FF Link/B

FOUNDATION™ Fieldbus H1 Master

User Manual

A-FFL/B

Document No. D122-007
09/2024
Revision 1.8

aparian

CONTENTS

1. Preface 8
1.1. Introduction to the FF Link 8
1.2. Features 10
1.3. Architecture 11
1.4. Additional Information 14
1.5. Support 15
2. Installation 16
2.1. Module Layout 16
2.2. Module Mounting 17
2.3. Power 19
2.4. H1 Bus 19
3. Setup 20
3.1. Install Configuration Software 20
3.2. Network Parameters 20
3.3. Device Description File Management 25
3.3.1. Adding DD Files 26
3.3.2. Exporting / Importing Catalog 29
3.4. Creating a New Project 31
3.5. FF Link – Basic Configuration 33
3.5.1. General 33
3.5.2. Physical Configuration 34
3.5.3. H1 36
3.5.4. Internal Map 39
3.5.4.1. Copy From 43
3.5.4.2. Copy To 48
3.5.5. Advanced 52
3.6. Add H1 Device 54
3.6.1. Manual H1 Device Add 54
3.6.2. Live List H1 Device Add 55
3.6.3. Configuration 56
3.6.3.1. Auto-Instantiate Control Strategy 56
3.6.3.2. Upload Control Strategy 56

3.6.3.3. General ... 57
3.6.3.4. Block Editor ... 59
3.6.3.5. Asynchronous Parameters ... 79
3.6.3.6. Advanced ... 81
3.6.4. H1 Device Import and Export Functions ... 83
3.6.4.1. Export H1 Device ... 83
3.6.4.1. Import H1 Device ... 83
3.7. EtherNet/IP Target ... 84
3.7.1. Logix Cyclic Class 1 ... 84
3.7.1.1. EDS AOP (Logix V21+) ... 84
3.7.1.2. Generic Module Profile (Logix Pre-V21) ... 87
3.7.1.3. ControlNet Router Profile ... 89
3.7.1.4. Multi-Connection ... 90
3.7.1.5. Logix Mapping ... 92
3.8. EtherNet/IP Originator ... 97
3.8.1. Explicit EtherNet/IP Messaging ... 97
3.8.1.1. EtherNet/IP Devices ... 97
3.8.1.2. EtherNet/IP Map ... 98
3.8.2. Cyclic Class 1 Connection ... 100
3.8.2.1. Manual Configuration ... 101
3.8.2.2. Import From Online Controller ... 103
3.8.2.3. Import From Controller L5X File ... 105
3.8.2.4. Import EDS File ... 107
3.8.2.5. Import Connection Library ... 108
3.9. Modbus TCP Slave ... 109
3.9.1. Modbus ... 109
3.10. Modbus TCP Master ... 111
3.10.1. Modbus ... 111
3.10.2. Modbus Auxiliary Map ... 112
3.11. Alert (Alarm) Notification ... 113
3.12. Cascade Control ... 117
3.12.1. Cascade Control Handshaking ... 118
3.12.2. Logix Cascade Control ... 118
3.13. System Details ... 121

3.13.1. Timing 121
3.13.2. Link Active Schedule 122
3.13.3. Master Publishing 122
3.13.4. Input Mapping Summary 122
3.13.5. Output Mapping Summary 123
3.13.6. DD Services 123
3.14. Module Download 124
3.15. Live List (Online) 125
3.15.1. Device Station Address / Tag Change 126
4. Operation 128
4.1. EtherNet/IP Target (Logix) 128
4.1.1. EtherNet/IP Class 1 Assembly Mapping 128
4.2. EtherNet/IP Originator 129
4.2.1. EtherNet/IP Class 1 Connections 129
4.2.1.1. Data received from H1 Device 130
4.2.1.2. Data sent to H1 Device 131
4.2.2. Explicit Messaging 133
4.2.2.1. Data received from H1 Device 133
4.2.2.1. Data sent to H1 Device 134
4.3. Modbus Master 135
4.3.1. Data sent to H1 Device 136
4.3.2. Data received from H1 Device 138
4.4. Modbus Slave 140
4.4.1. Data sent to H1 Device 141
4.4.2. Data received from H1 Device 142
4.5. H1 Alert (Alarm) Management 145
4.5.1. EtherNet/IP Target (Logix) 146
4.5.2. Modbus 147
4.5.3. EtherNet/IP Originator 147
4.6. Internal Map H1 Data Formats 147
4.6.1.1. FF Master - Master Status 148
4.6.1.2. FF Master - Advanced Status 149
4.6.1.3. FF Master - Alert 150
4.6.1.4. FF Master - Master Control 152

4.6.1.5. FF Master - Process Variable 152
4.6.1.6. FF Master - Alert Ack 153
4.6.1.7. FF Device - Device Status 153
4.6.1.8. FF Device - Device Control 154
4.6.1.9. FF Device - Process Variable 154
4.7. Internal Data Space (IDS) 155
4.7.1. Internal Status 155
5. Legacy Master Migration 157
6. FF Link Master Redundancy 161
6.1. Redundancy Strategy 161
6.2. Redundant Architecture 162
6.3. FF Link Configuration 163
6.3.1. General Configuration 163
6.3.2. Physical Configuration 163
6.3.2.1. Switch-Over TimeOut 164
6.4. Logix Configuration 165
6.5. Operation 166
7. Device Firmware Update 168
8. Device Type Manager (DTM) 171
8.1. Installation 171
8.2. Configuration 171
8.2.1. Configuration – FF Link (Direct) DTM 172
8.2.2. Configuration – FF Link-FT DTM 174
8.2.3. Adding Device DTMs 176
8.3. Operation 177
9. Diagnostics 181
9.1. LEDs 181
9.2. Module Status Monitoring 182
9.2.1. Project Explorer – Non-Redundant Mode 183
9.2.2. Project Explorer – Redundant Mode 183
9.2.3. FF Link Master 184
9.2.3.1. General 185
9.2.3.2. General Statistics 188
9.2.3.3. Live List 189

9.2.3.4. Master PVs ... 189
9.2.3.5. Alerts ... 190
9.2.3.6. EtherNet/IP Explicit ... 192
9.2.3.7. EtherNet/IP MAP ... 193
9.2.3.8. EtherNet/IP Originator ... 193
9.2.3.9. Logix ... 195
9.2.3.10. CIP Statistics ... 196
9.2.3.11. Modbus Statistics ... 196
9.2.3.12. Ethernet Clients ... 198
9.2.3.13. TCP/ARP ... 198
9.2.4. H1 Device Status ... 198
9.2.4.1. General ... 199
9.2.4.2. Statistics ... 200
9.2.4.3. PV Map ... 201
9.2.4.4. Asynchronous Parameters ... 202
9.2.4.5. Advanced Status ... 203
9.3. Explicit Messaging Utility ... 204
9.4. H1 Packet Capture ... 206
9.5. Modbus Packet Capture ... 208
9.6. Module Event Log ... 210
9.7. Web Server ... 212
9.8. Internal Register Viewer ... 213
9.9. Module Status Report ... 215
10. Technical Specifications ... 216
10.1. Dimensions ... 216
10.2. Electrical ... 217
10.3. Ethernet ... 217
10.4. EtherNet/IP Target ... 218
10.5. EtherNet/IP Originator ... 218
10.6. Modbus TCP Master ... 218
10.7. Modbus TCP Slave ... 219
10.8. H1 ... 219
10.9. Certifications ... 220
11. FOUNDATION™ Fieldbus H1 ... 221

11.1. Introduction 221
11.2. Topology of H1 221
11.3. H1 cable description 221
12. Appendix 223
12.1. Process Variable (PV) Status 223
12.1.1. Quality 223
12.1.2. Quality Sub Status 223
12.1.3. Limit 225
13. References 226
14. Index 227

Revision History

Revision	Date	Comment
1.0	5 August 2022	Initial document
1.1	6 September 2022	Added Cascade Control (3.12)
1.2	10 August 2023	Corrected error in PV Status Quality - Appendix
1.3	13 November 2023	Added UKCA Conformance Mark
1.4	27 November 2023	Updated EtherNet/IP Class 1 Connection Import options
1.5	12 April 2024	Added Logix Profile Options – Generic and ControlNet Router
1.6	6 May 2024	Added Upload Control Strategy from H1 device. Added H1 device download options. Updated H1 device block editor view. Updated H1 Asynchronous Parameter configuration.
1.7	11 June 2024	Added Explicit Message Utility
1.8	30 September 2024	Added Advanced Status mapping option

1. PREFACE

1.1. INTRODUCTION TO THE FF LINK

This manual describes the installation, operation, and diagnostics of the Aparian FF Link FOUNDATION™ Fieldbus H1 Master.

The FF Link allows the user to interface FOUNDATION™ Fieldbus (FF) H1 devices to EtherNet/IP (Target or Originator) or Modbus TCP (Master or Slave).

The FF Link operates as an FF H1 master allowing EtherNet/IP devices (e.g. Rockwell Automation Logix platform) or Modbus devices to exchange process, alarming, and diagnostic data with H1 devices as well as provide parameterization and asset management of H1 devices using either the Slate software or Device Type Managers (DTMs).

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

The **Primary Interface** can be configured to be any one of the following four interface modes:

EtherNet/IP Target
A Logix controller can own the FF Link over EtherNet/IP using up to four class 1 connections. This will allow the FF Link to exchange data with the Logix controller using the input and output assembly of the EtherNet/IP Class 1 connections. Data from H1 devices are mapped to the Logix controller over EtherNet/IP.

Modbus TCP Master
The data from H1 devices will be written to, or read from, the module’s internal Modbus Registers. The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Slave devices and the module’s internal Modbus registers. The Modbus communication utilizes Modbus TCP.

Modbus TCP Slave
The data from H1 devices will be written to, or read from, the module’s internal Modbus Registers using the internal mapping functions. These Modbus registers can be accessed by a remote Modbus Master using Modbus TCP.

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 EtherNet/IP network:

EtherNet/IP Class 1 Connection

The FF Link can be configured to own EtherNet/IP IO by using the Slate software to configure the Class 1 connections. Each FF Link can own up to 10 EtherNet/IP devices. Data from the EtherNet/IP IO (via the input and output assemblies) can be exchanged with the H1 devices.

EtherNet/IP Explicit Messaging

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

The FF Link has a built-in Isolated Power Conditioner which can supply up to 420mA. The power conditioner is protected against a configurable overcurrent limit. The module also has a configurable built-in fieldbus terminator.

The FF Link uses an internal mapping strategy allowing the user to map any FF H1 data to any supported interface and vice versa.

Up to 32 FOUNDATION™ Fieldbus H1 devices are supported by the FF Link. The data is formatted into the engineering units for use in either a Logix platform or Modbus Master/Slave device by using the automatically generated mapping imports for Logix User Defined Data Types (UDTs) or padding for Modbus Registers. The latter ensures alignment with the 16-bit data structure.

The FF Link supports module redundancy when using EtherNet/IP Target mode. This allows two identically configured FF Link modules to operate in an “Active – Standby” strategy. Using the provided Logix Add-On-Instruction (AOI), the H1 data from the Active FF Link is marshalled to the Logix user tags. Should the Active FF Link fail for any reason (e.g. loss of power, network failure, bus failure), then the previously Standby FF Link will automatically become the new Active module.

The FF Link will allow the user to monitor and extract FOUNDATION™ Fieldbus H1 device alarms from each H1 device on the connected H1 fieldbus from either a Logix controller or Modbus Master/Slave device.

A range of statistics and tools are available to provide a detailed diagnostic overview of each FF Link which simplifies commissioning. The Slate configuration utility allows the user to perform an H1 packet capture of the running fieldbus which can be used to analyse the bus behaviour and packets received. The FF Link also provides global and device specific statistics.

The FF Link module has two Ethernet ports allowing the user to configure either a Linear or Ring (Device Level Ring – DLR) Ethernet topology. The Ethernet ports can also be setup for port mirroring allowing for better fault analysis.

The FF Link can synchronize to an NTP Server allowing for automatic time synchronization. It also supports an onboard non-volatile event log for improved fault finding.

1.2. FEATURES

- Operates as a FOUNDATION™ Fieldbus H1 Master.
- Supports up to 32 FOUNDATION™ Fieldbus H1 devices.
- Module has various configurable primary interfaces:
    - EtherNet/IP Target
    - Modbus TCP Slave
    - Modbus TCP Master
    - EtherNet/IP Originator (Class 1 connection with up to 10 EtherNet/IP connections and Explicit Messaging with up to 10 EtherNet/IP devices).
- Provides a built-in Isolated Power Conditioner that can supply up to 420mA.
- Allows the use of external Power Conditioners for Intrinsically Safe (IS) applications.
- Provides a built-in software enabled Fieldbus terminator.
- Dual Ethernet ports which support Device-Level-Ring (DLR).
- Built-in H1 packet capture.
- Built-in Modbus packet capture.
- Supports detailed FOUNDATION™ Fieldbus H1 live list.
- Device Specific UDT, tag and logic generation for import into a Studio5000 project.
- Supports H1 Master Redundancy.
- Supports H1 Alert (Alarm) extraction.
- Network Time Protocol (NTP) supported for external time synchronization.
- Device Type Manager (DTM) supported.
- Online change configuration supported.
- Optional Advanced H1 scheduling.
- Small form factor – DIN rail mounted.

1.3. ARCHITECTURE

The figures below provide examples of typical network setup for an H1 Master architecture using either an EtherNet/IP or Modbus TCP Interface.

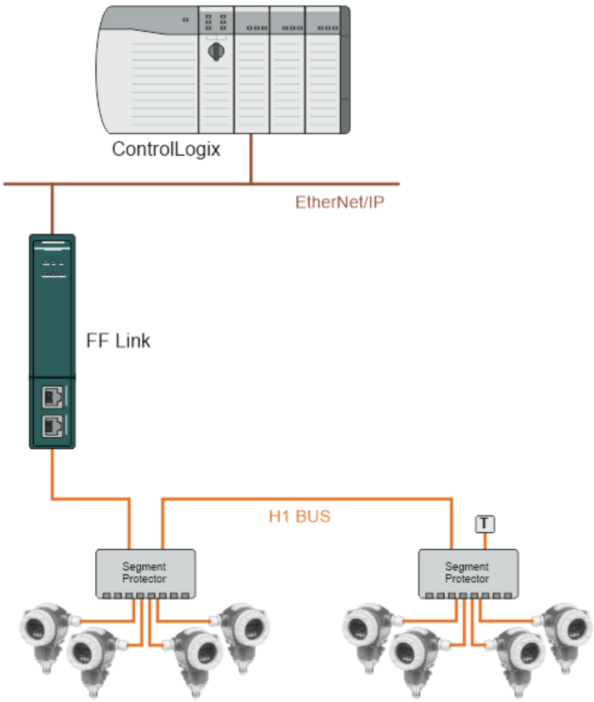


Figure 1.1 – FF Link - FF Link H1 Master to Logix (EtherNet/IP target) architecture

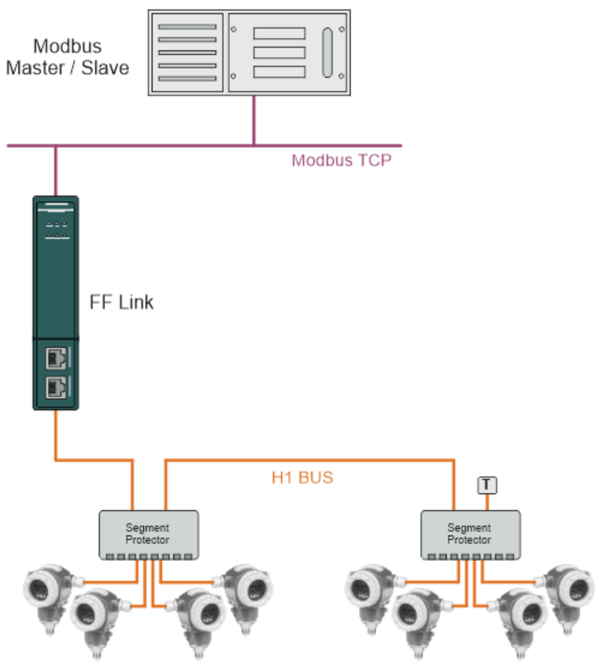


Figure 1.2 – FF Link H1 Master to Modbus TCP architecture

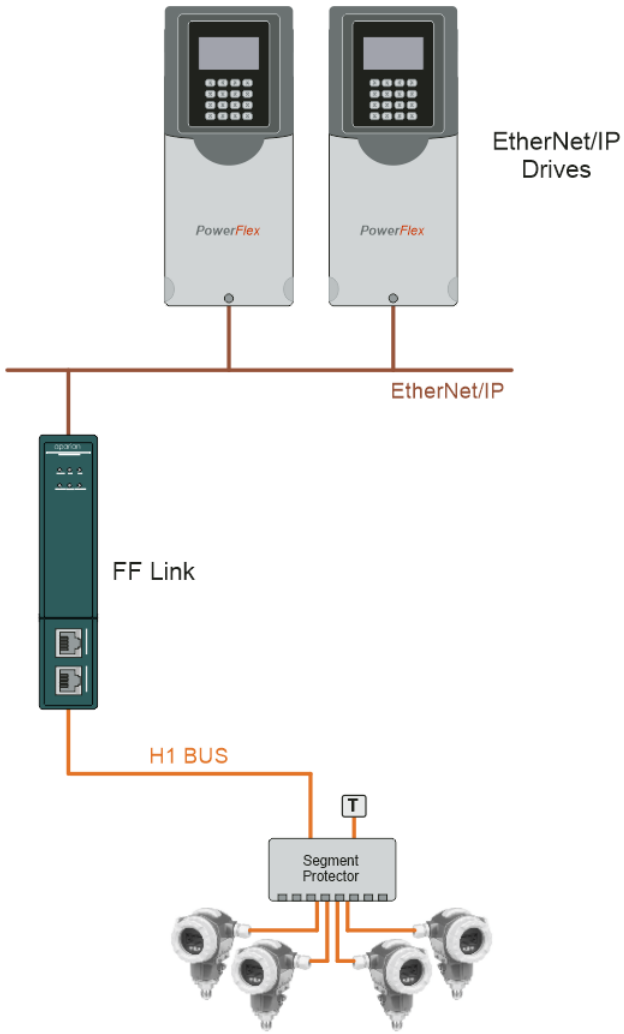


Figure 1.3 – FF Link H1 Master to EtherNet/IP Originator architecture

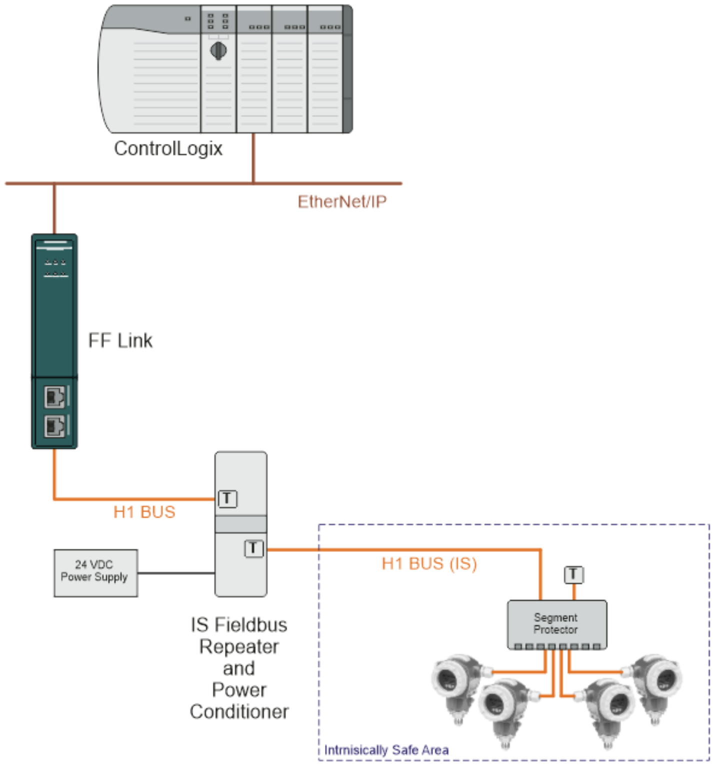


Figure 1.4 – FF Link H1 Master with External Intrinsically Safe Repeater

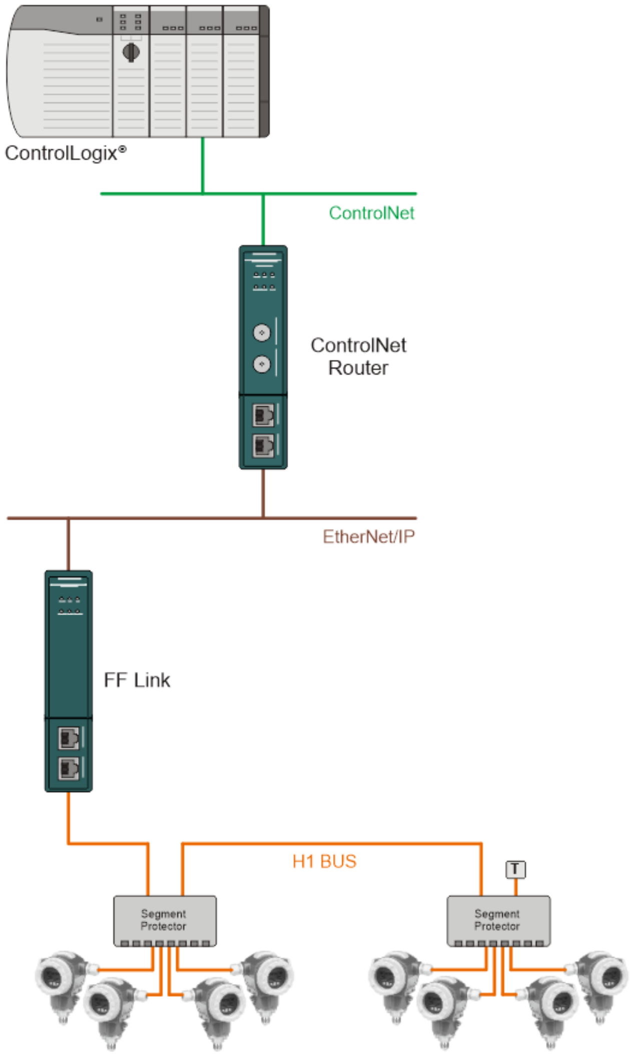


Figure 1.5 – FF Link H1 Master via ControlNet Router

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
FF Link User Manual FF Link Datasheet Example Code & UDTs	https://www.aparian.com/products/ff-link-b

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
FOUNDATION™ Fieldbus	FOUNDATION™ Fieldbus System Engineering Guidelines (AG-181) FOUNDATION™ Fieldbus Technical Overview (FD-043) www.fieldcommgroup.org/technologies/foundation-fieldbus

Table 1.1 - Additional Information

1.5. SUPPORT

Technical support is provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Resource	Link
Contact us web link	www.aparian.com/contact-us
Support email	support@aparian.com

Table 1.2 – Support Details

2. INSTALLATION

2.1. MODULE LAYOUT

The module has two Ethernet ports at the front of the module. The Ethernet cable must be wired according to industry standards which can be found in the additional information section of this document.

At the bottom of the FF Link modules, there is one 3-way power connector (back) and one 3-way H1 connector (front).

The modules provide six diagnostic LEDs as shown in the front view figure below. These LEDs are used to provide information regarding the module system operation, the Ethernet interface, the auxiliary communication (e.g. Modbus TCP), and the H1 network status.

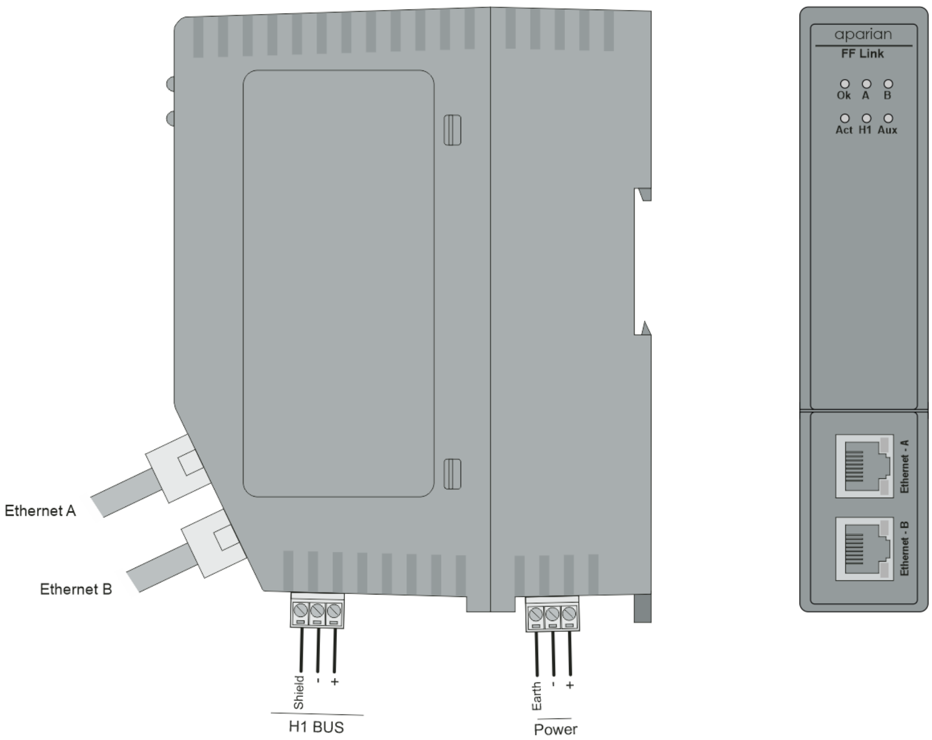


Figure 2.1 – FF Link Front and Side view

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

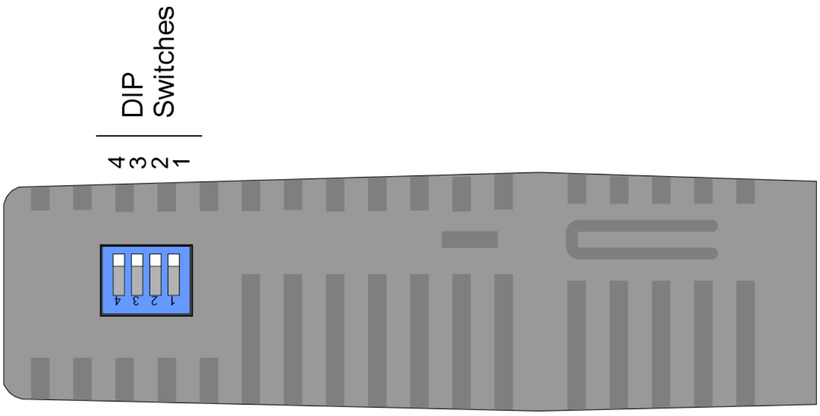


Figure 2.2 – FF Link Top view

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

Table 2.1. - DIP Switch Settings

2.2. MODULE MOUNTING

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

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

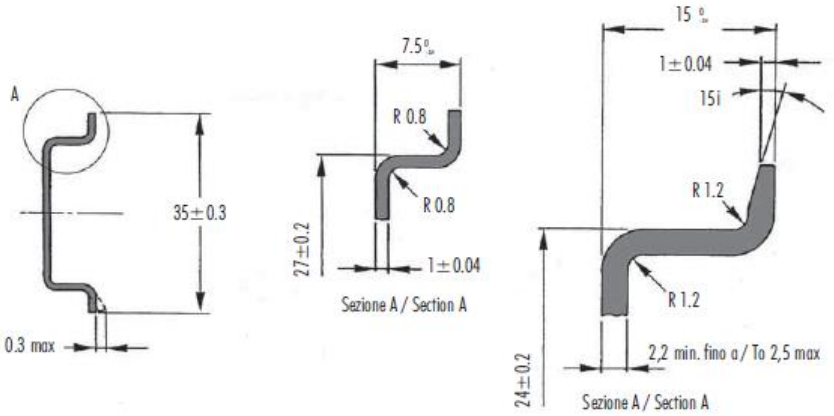


Figure 2.3 - DIN rail specification

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

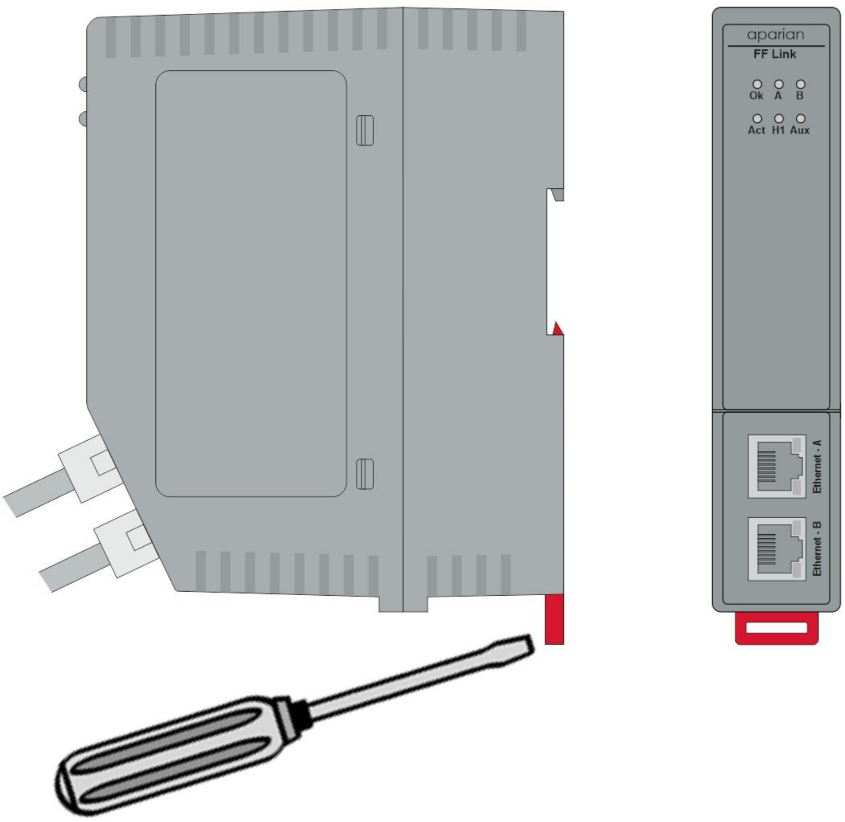


Figure 2.4 - DIN rail mouting

2.3. POWER

The FF Link has an input voltage range of 22-26 Vdc that needs to be applied to the module via the power connector. The power connector also provides an Earth connection for the FF Link.

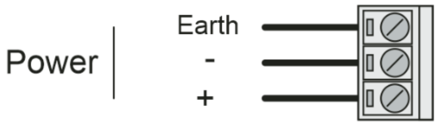


Figure 2.5 – FF Link Power connector

NOTE: It is recommended to always have a good clean earth connected to the module via the Earth connector on the power connector.

2.4. H1 BUS

The H1 connector has two conductors (positive and negative) for the H1 network. The connector also has a shield connection for the H1 bus which is internally connected to the Power Connector Earth.

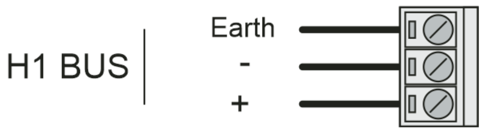


Figure 2.6 – FF Link H1 communications connector

3. SETUP

3.1. INSTALL CONFIGURATION SOFTWARE

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

Figure 3.1 - Aparian Slate Environment

3.2. NETWORK PARAMETERS

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

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

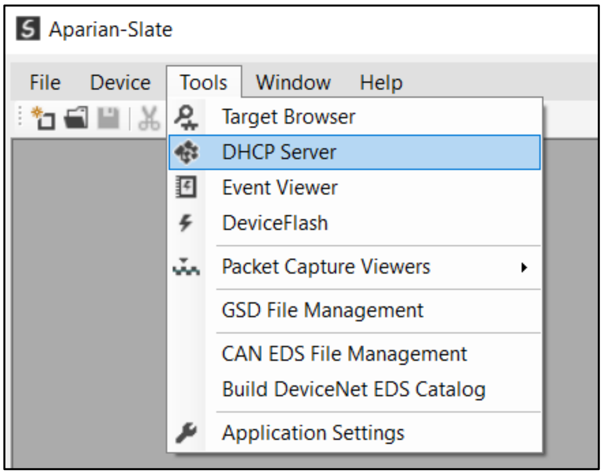


Figure 3.2 - Selecting DHCP Server

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

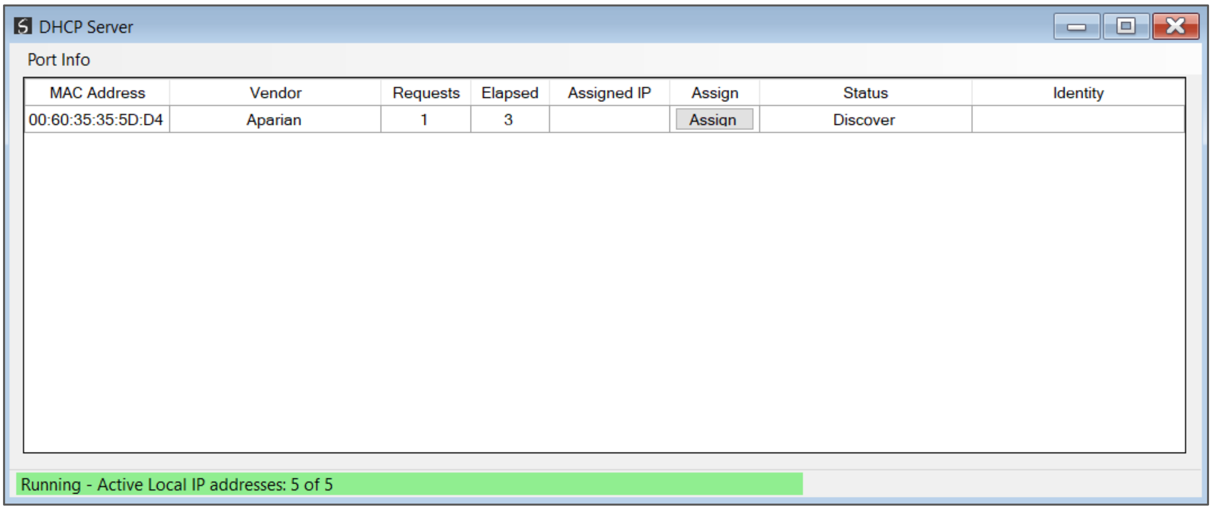


Figure 3.3 - DHCP Server

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

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

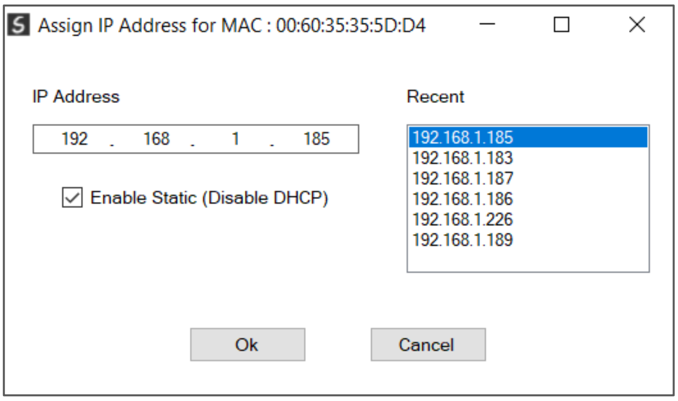


Figure 3.4 - Assigning IP Address

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

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

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

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

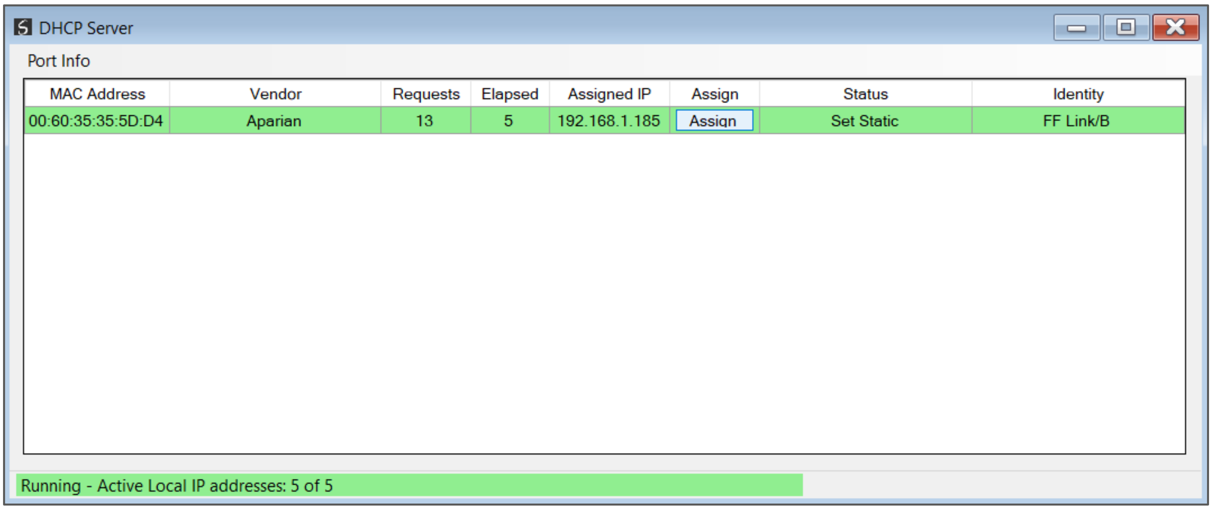


Figure 3.5 - Successful IP address assignment

It is possible to force the module back into DHCP mode by powering up the device with DIP switch 2 set to the ***On*** position.

A new IP address can then be assigned by repeating the previous steps.

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

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

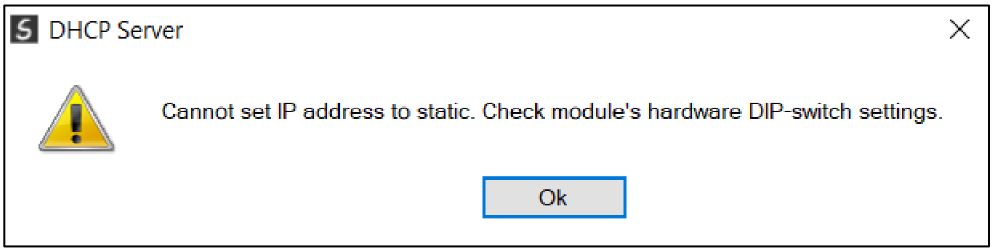


Figure 3.6 - Force DHCP warning

In addition to the setting the IP address, a number of other network parameters can be set during the DHCP process. These settings can be viewed and edited in Slate's ***Application Settings***, in the ***DHCP Server*** tab.

Once the DHCP process has been completed, the network settings can be set using the ***Ethernet Port Configuration*** via the ***Target Browser***.

The ***Target Browser*** can be accessed under the ***Tools*** menu.

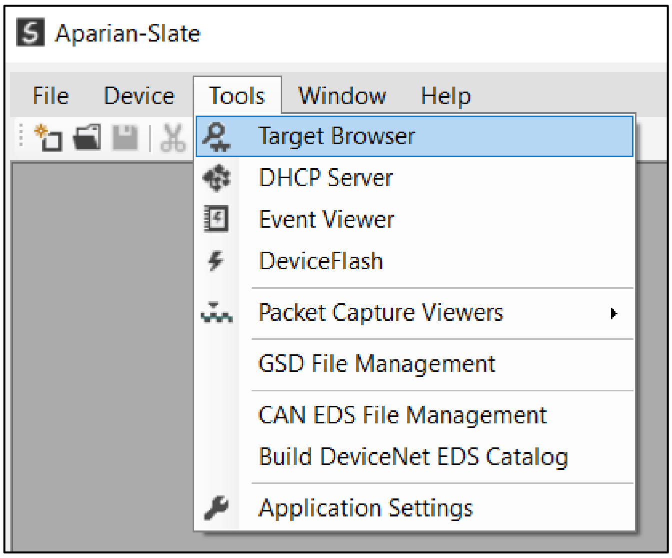


Figure 3.7 - Selecting the Target Browser

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

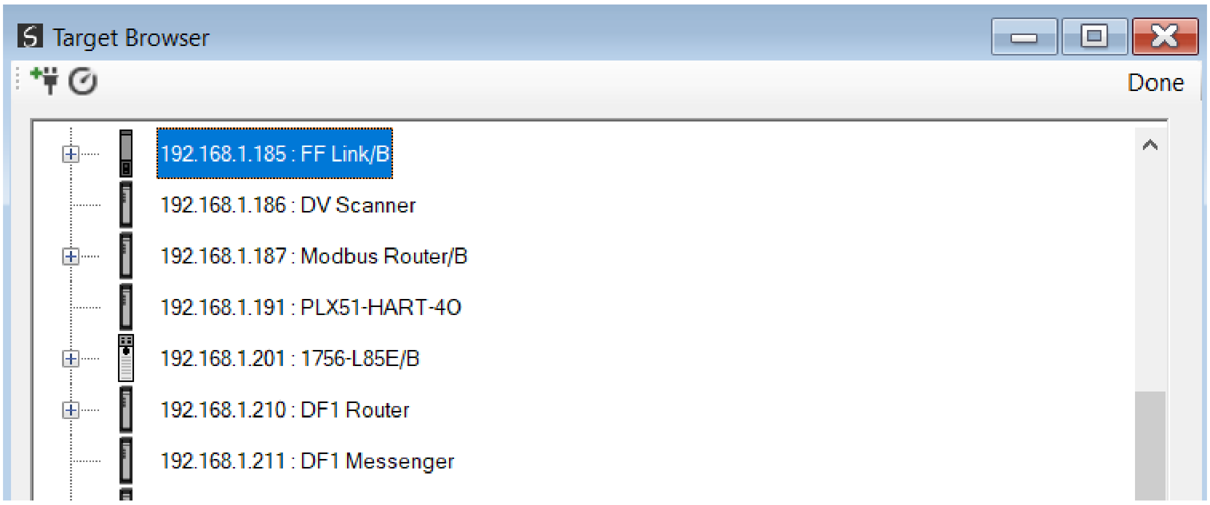


Figure 3.8 - Target Browser

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

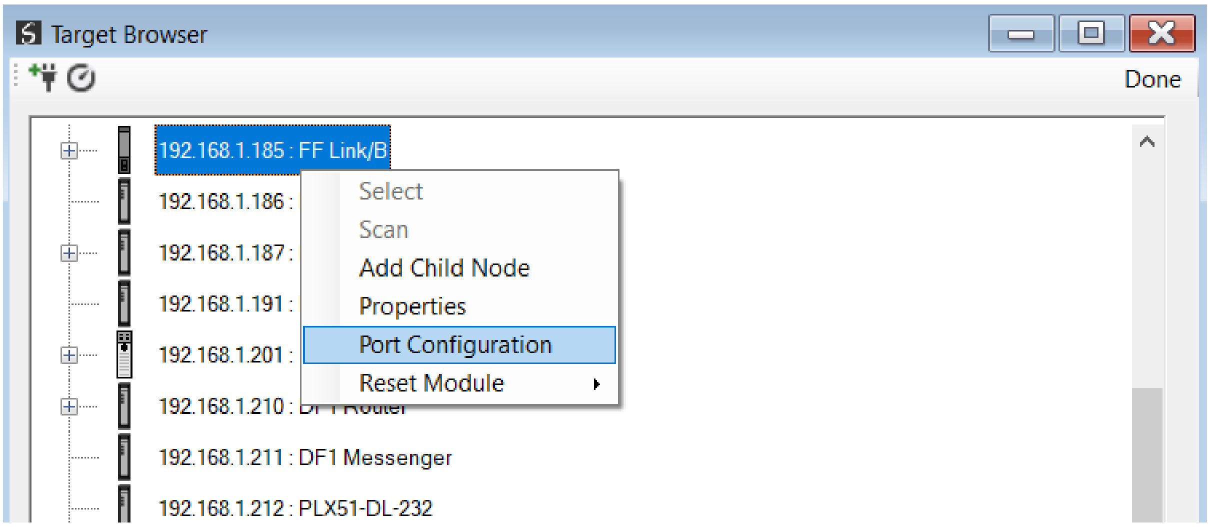


Figure 3.9 - Selecting Port Configuration

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

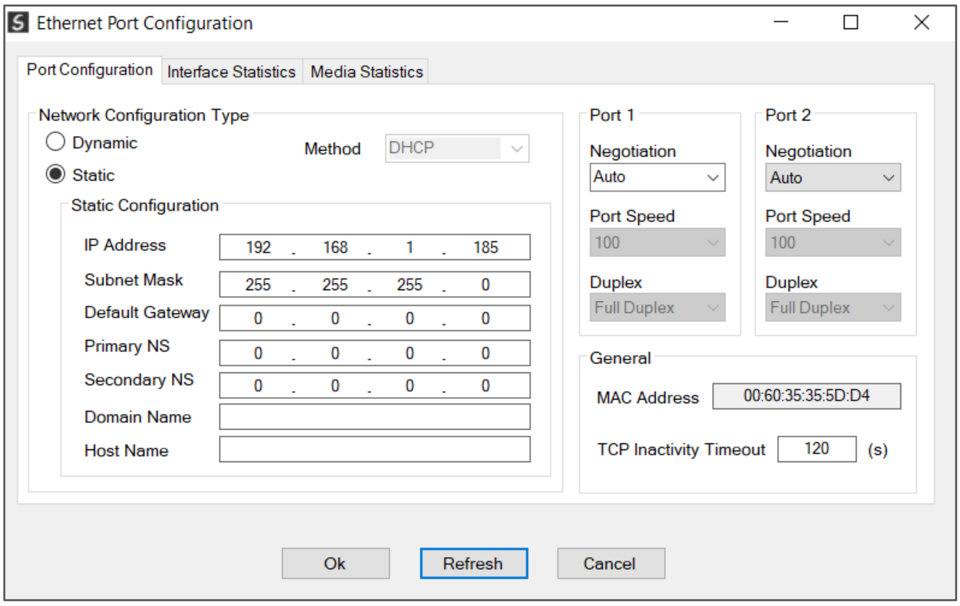


Figure 3.10 - Port Configuration

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

3.3. DEVICE DESCRIPTION FILE MANAGEMENT

Each FOUNDATION™ Fieldbus H1 device has a Device Description (DD) file that is required to provide information needed to configure the device for data exchange.

Slate manages the FF DD library which is used for adding devices to the FF Link.

The FF DD Management Tool is opened by selecting ***FF DD Management*** under the ***Tool*** menu in Slate.

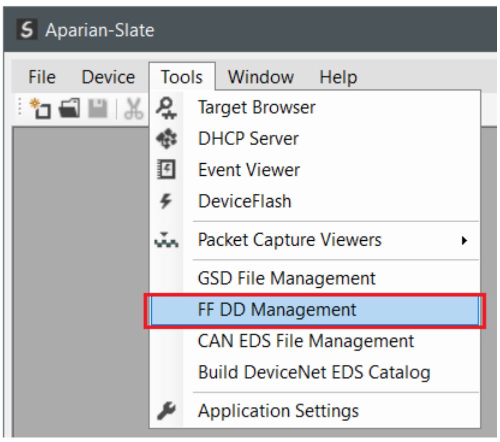


Figure 3.11 – Launching the FF DD Management Tool

Once the tool has been opened a list of H1 devices already registered using their FF DD files.

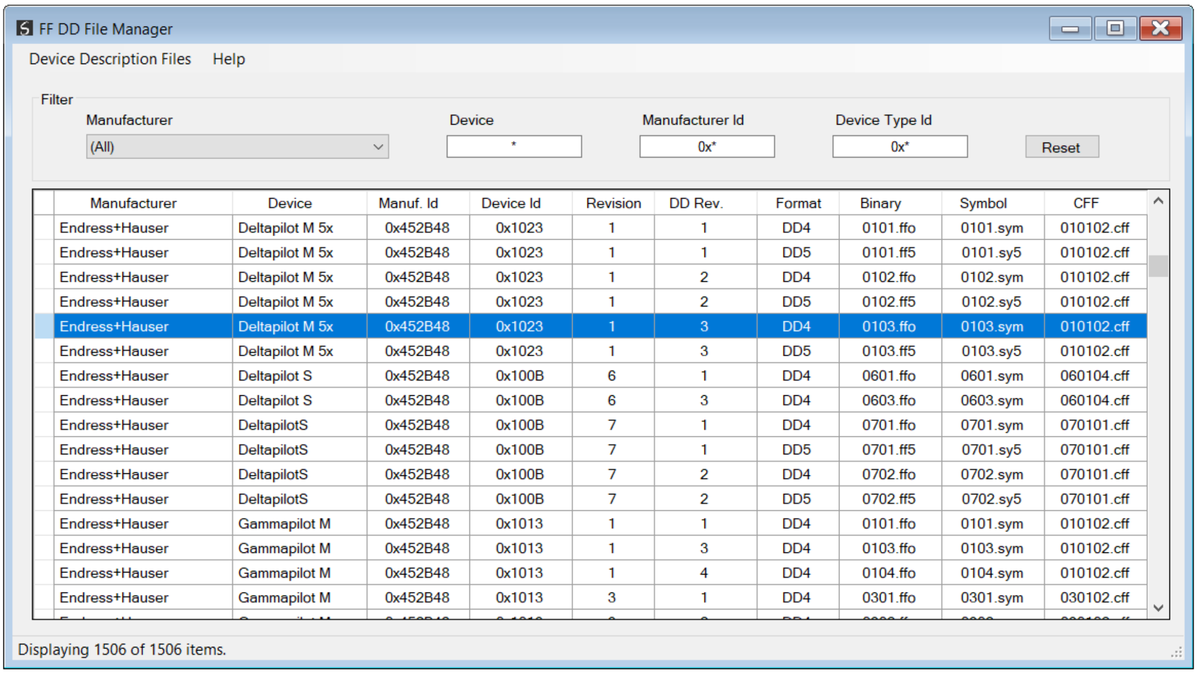


Figure 3.12 – FF DD Management Tool

3.3.1. Adding DD Files

To add an FF DD file the user will need to select the ***Add DD File*** option under the ***Device Descriptor Files*** menu.

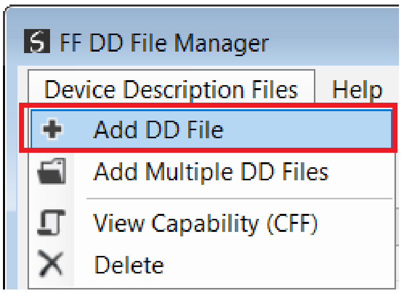


Figure 3.13 – FF DD File adding

The required FF Binary file will need to be selected as shown below.

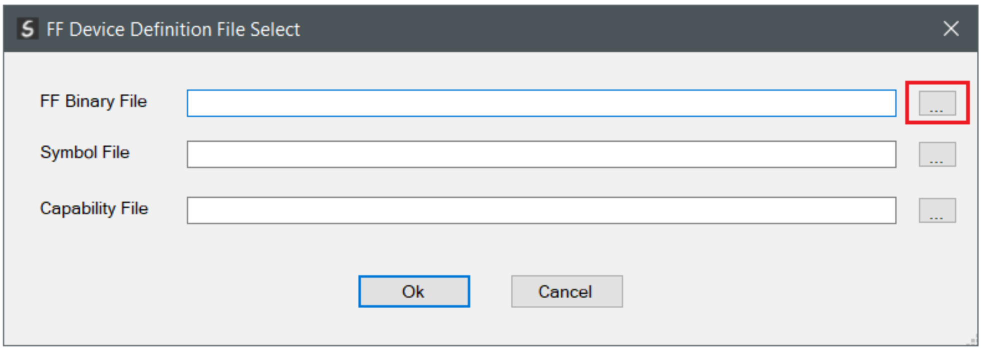


Figure 3.14 – FF DD File explorer launch

The user must select either a *.ffo or *.ff5 file as shown below.

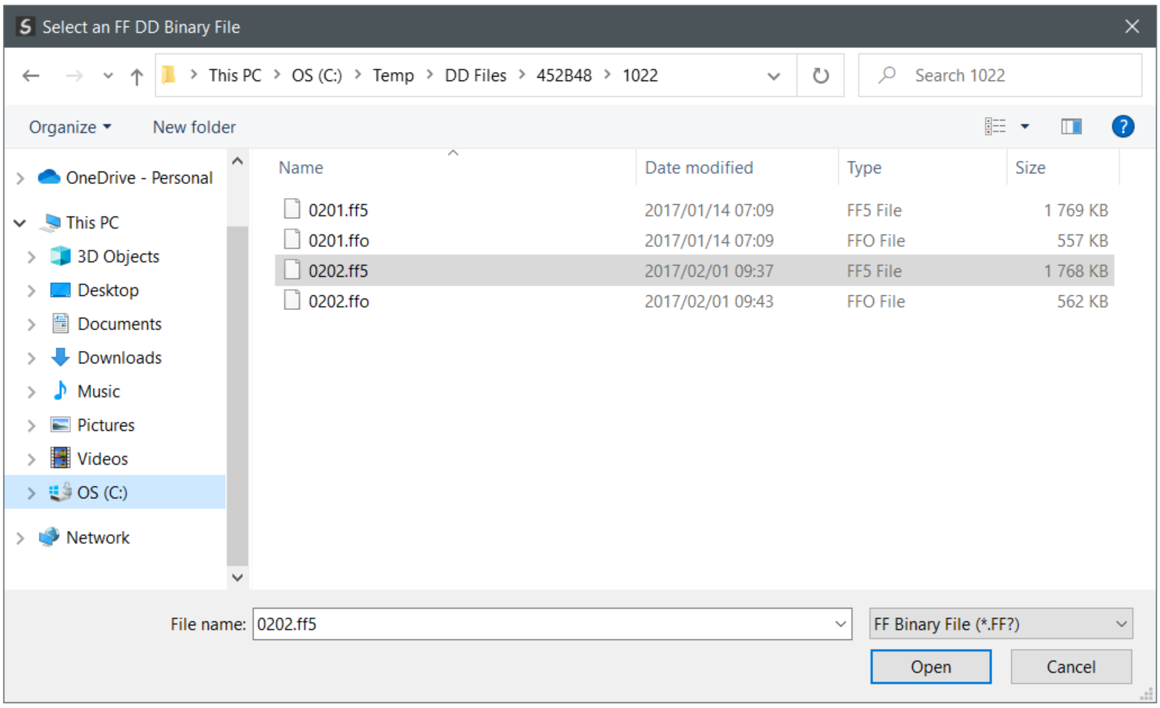


Figure 3.15 – FF DD File selection

Once selected, the DD Manager will attempt to find the matching symbol and capability file d in the same folder and if found, will automatically populate these in the respective textboxes.

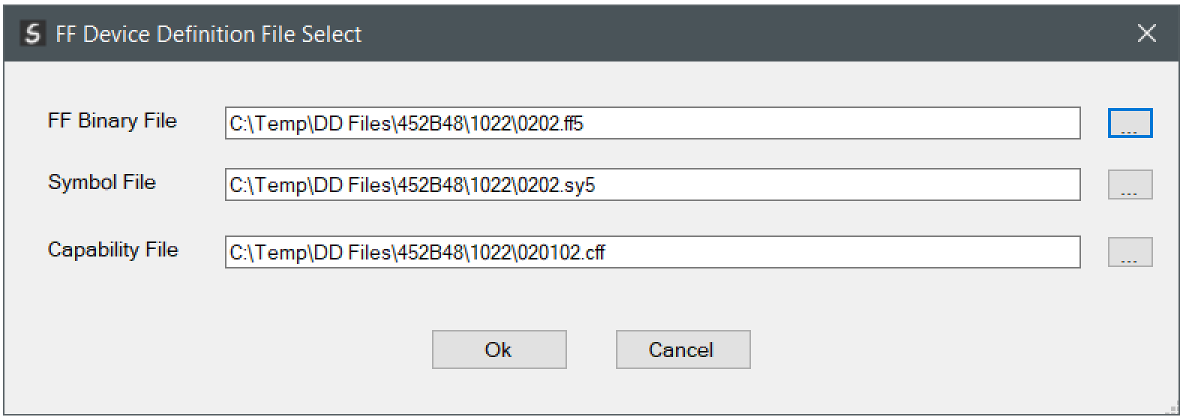


Figure 3.16 – FF DD File selected

Multiple DD Files can be added at the same time by selecting the ***Add Multiple DD Files*** option under the ***Device Descriptor Files*** menu.

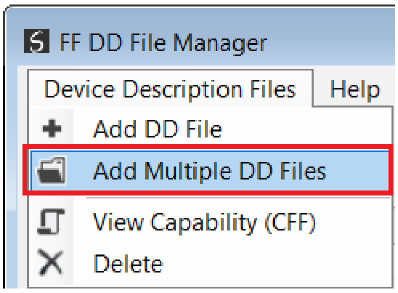


Figure 3.17 – FF DD Folder adding

Select the folder with the required DD files to be added.

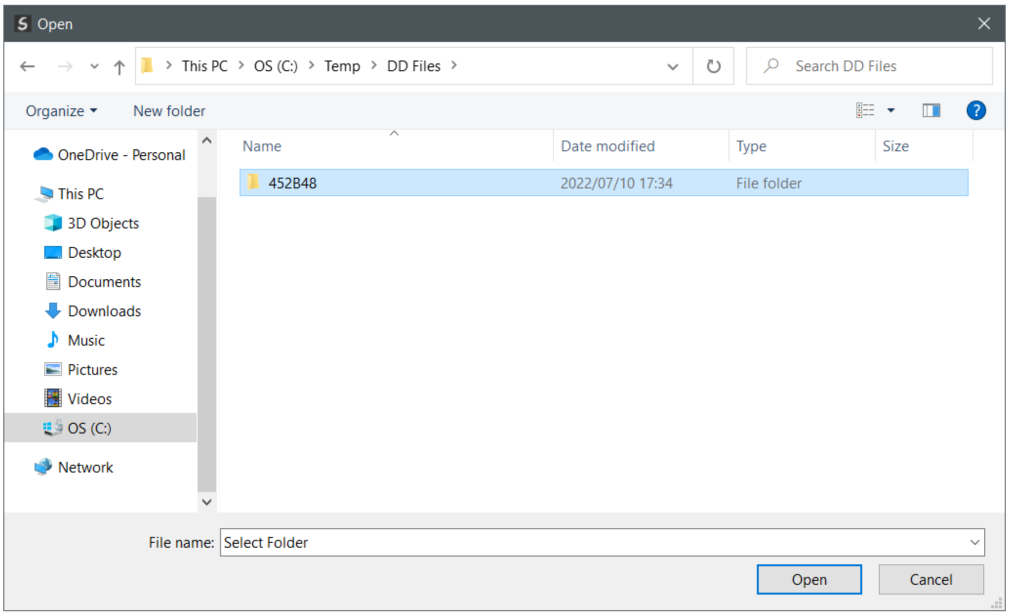


Figure 3.18 – FF DD Folder selection

Once the file has been selected the FF DD File Management tool will add the H1 device to the device list.

3.3.2. Exporting / Importing Catalog

An existing DD Catalog can be exported to a single file to simply the transfer to another system.

To export the complete DD Catalog, select the ***Export DD File Pack*** option under the ***Device Description Files*** menu item.

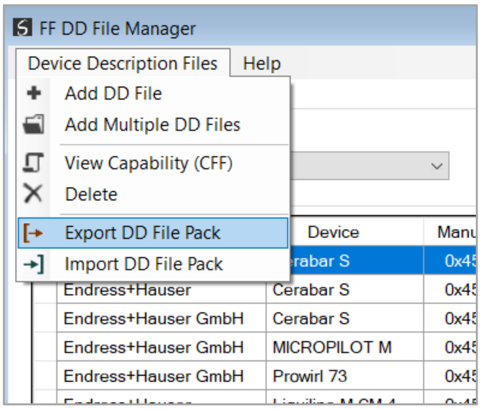


Figure 3.19 – Export DD File Pack

Select the filename of the DD Pack to be saved.

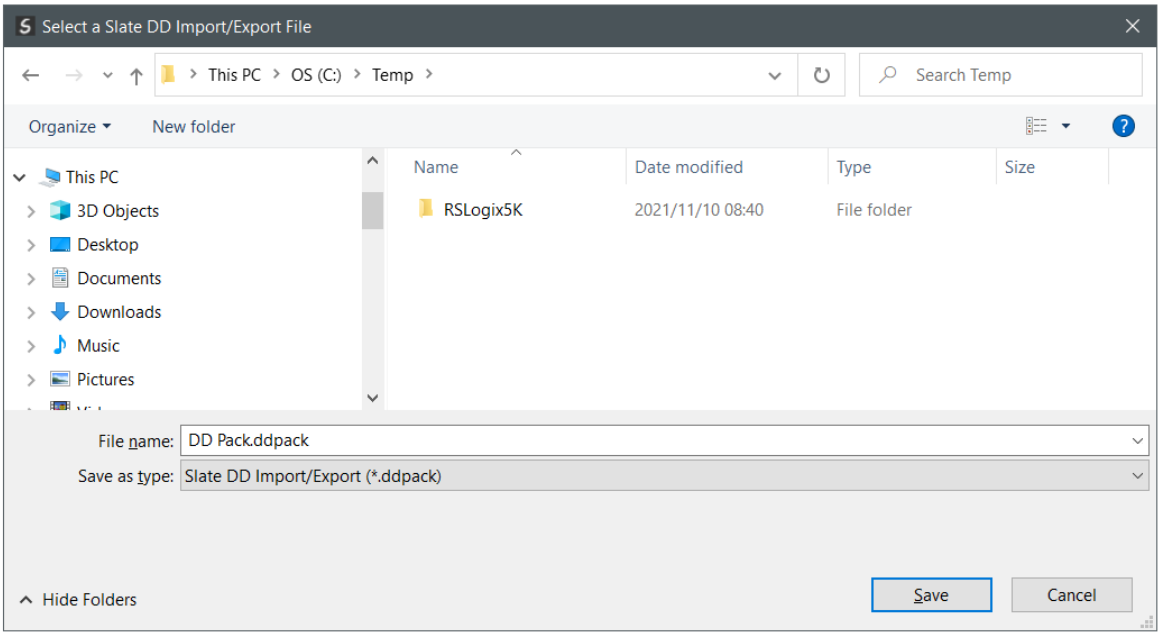


Figure 3.20 – Export DD File Pack – File Selection

The DD File pack can then be imported on another PC using the DD Catalog Import function. To import the pack, select the ***Import DD File Pack*** option under the ***Device Description Files*** menu item.

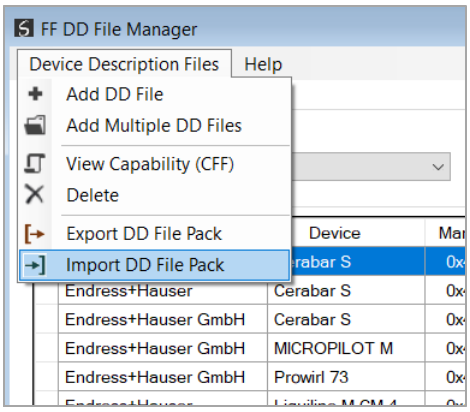


Figure 3.21 – Import DD File Pack

Select the **.ddpack* file to be imported.

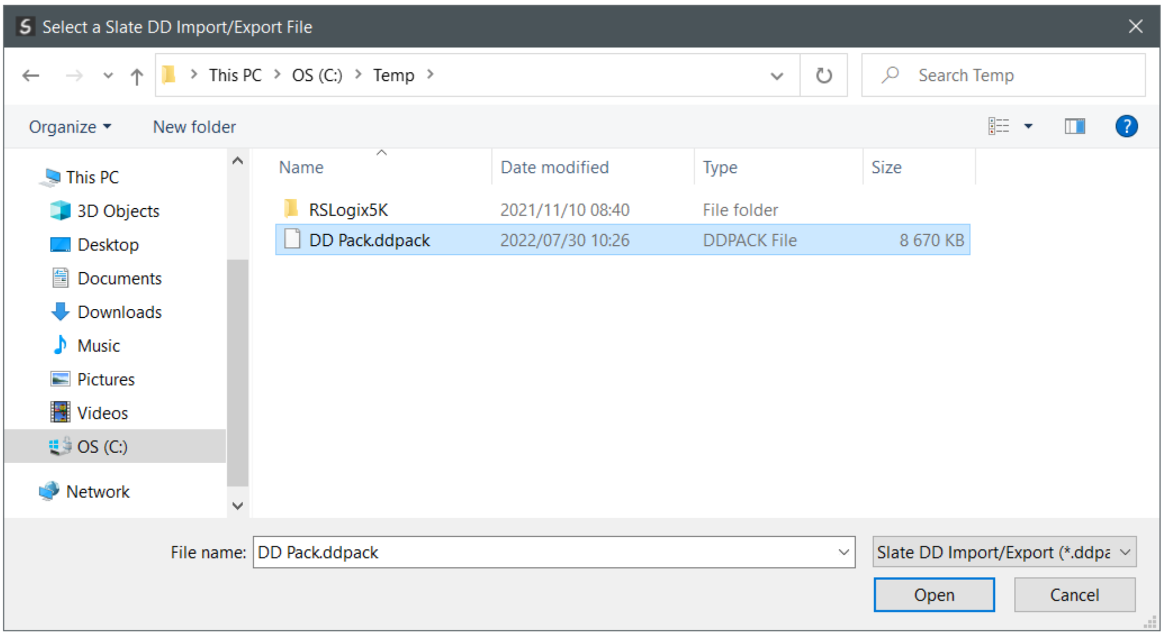


Figure 3.22 – Import DD File Pack – File selection

3.4. CREATING A NEW PROJECT

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

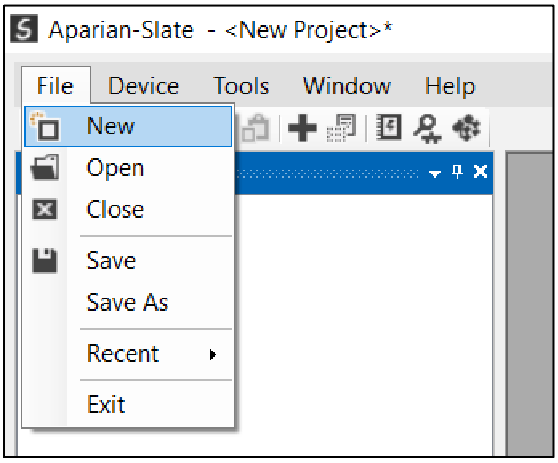


Figure 3.23 - Creating a new project

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.

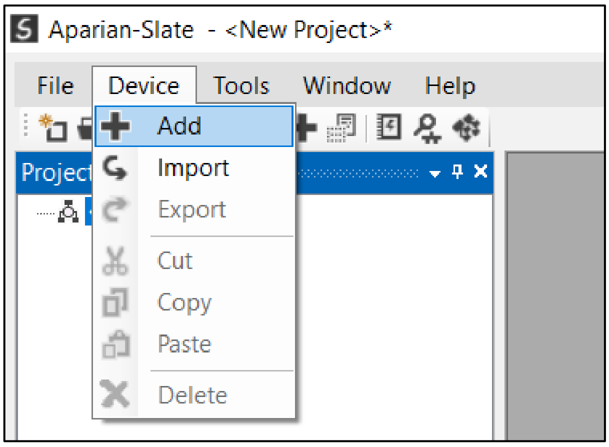


Figure 3.24 - Adding a new device

In the ***Add New Device*** window select the ***FF Link***, and click the ***Ok*** button.

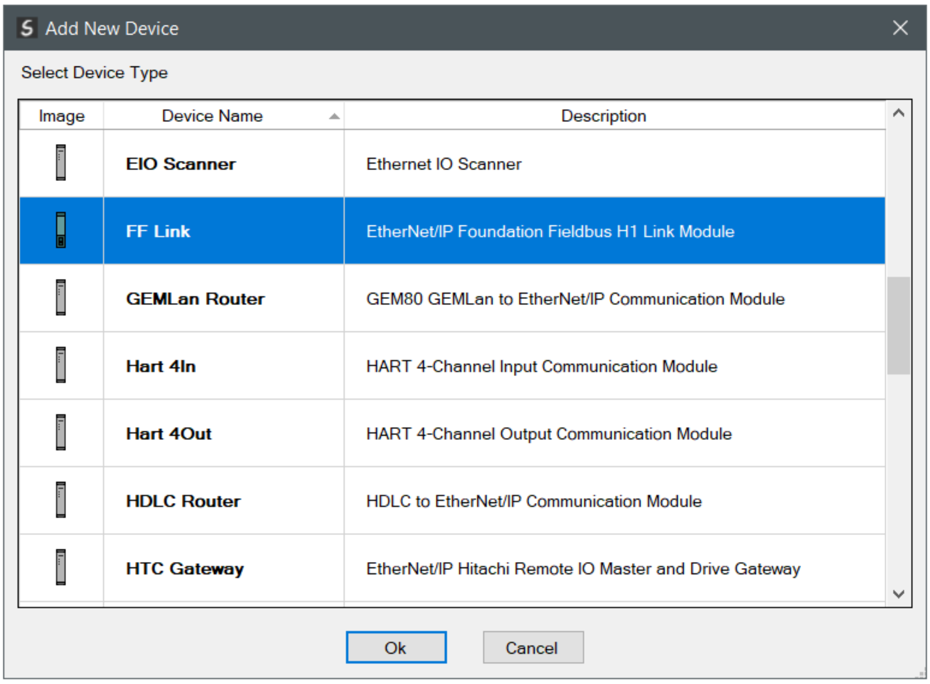


Figure 3.25 – Selecting a new FF Link

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***.

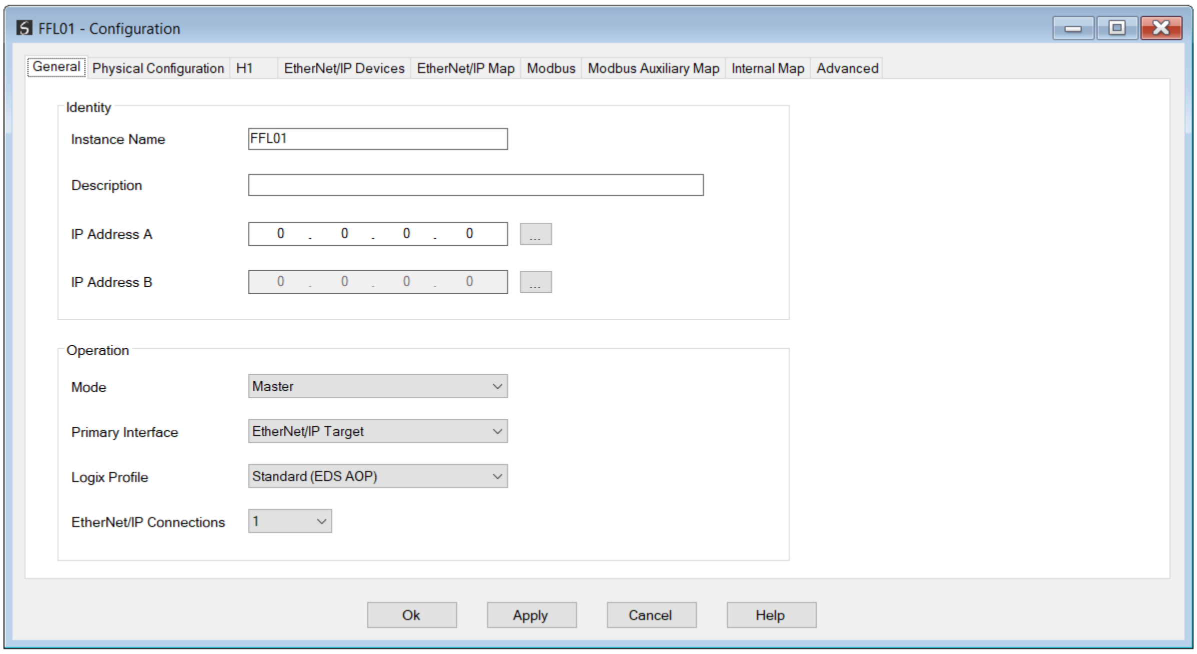


Figure 3.26 – FF Link Configuration

3.5. FF LINK – BASIC CONFIGURATION

The FF Link parameters are configured using the Slate configuration environment.

3.5.1. GENERAL

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

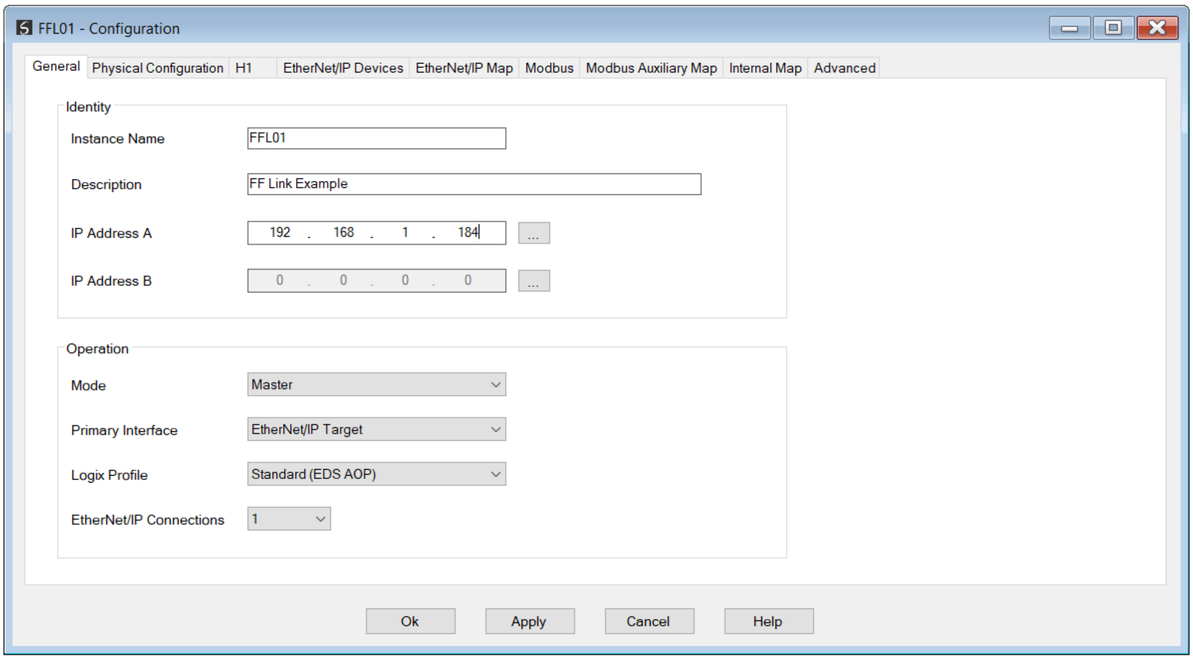


Figure 3.27 – FF Link General configuration

The General configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various FF Link modules.
Description	This parameter is used to provide a more detail description of the application for the module.
IP Address A	The IP address of the module when in (Standalone) Master mode. When in Redundant Master mode this will be the IP address of the "A" FF Link module
IP Address B	The IP address of the "B" FF Link module when in Redundant Master mode.
Mode	The FF Link can operate in one of three modes: **Quiet**

	This mode allows the user to connect the FF Link to an active bus and run an H1 packet capture. In this mode the FF Link will not communicate on the H1 Bus, but rather only listen. **Master** In this mode the FF Link is the H1 Master on the H1 network. **Redundant Master** In this mode the FF Link operates as one of two H1 Masters on the H1 network. See the section on Redundant Masters for more information.
Primary Interface	This is the network the FF Link will interface the H1 network. • EtherNet/IP Target (Logix) • Modbus TCP Slave • Modbus TCP Master • EtherNet/IP Originator
Logix Profile	The Logix Profile to be used for when generating Logix L5X mapping code. • Standard (EDS AOP) • Generic Profile • ControlNet Router (A-CNR)
EtherNet/IP Connections	The number of target EtherNet/IP (CIP) Connections to be used in the exchange with Logix (1 to 4). **NOTE:** This value must match that configured in the Logix IO tree.

Table 3.1 - General configuration parameters

3.5.2. Physical Configuration

The Physical Configuration is shown in the figure below. This allows the user to configure the physical media operation (e.g. enabling a terminator).

The FF Link ***Physical Configuration*** window is opened by either double clicking on the module in the tree or right clicking the module and selecting ***Configuration***.

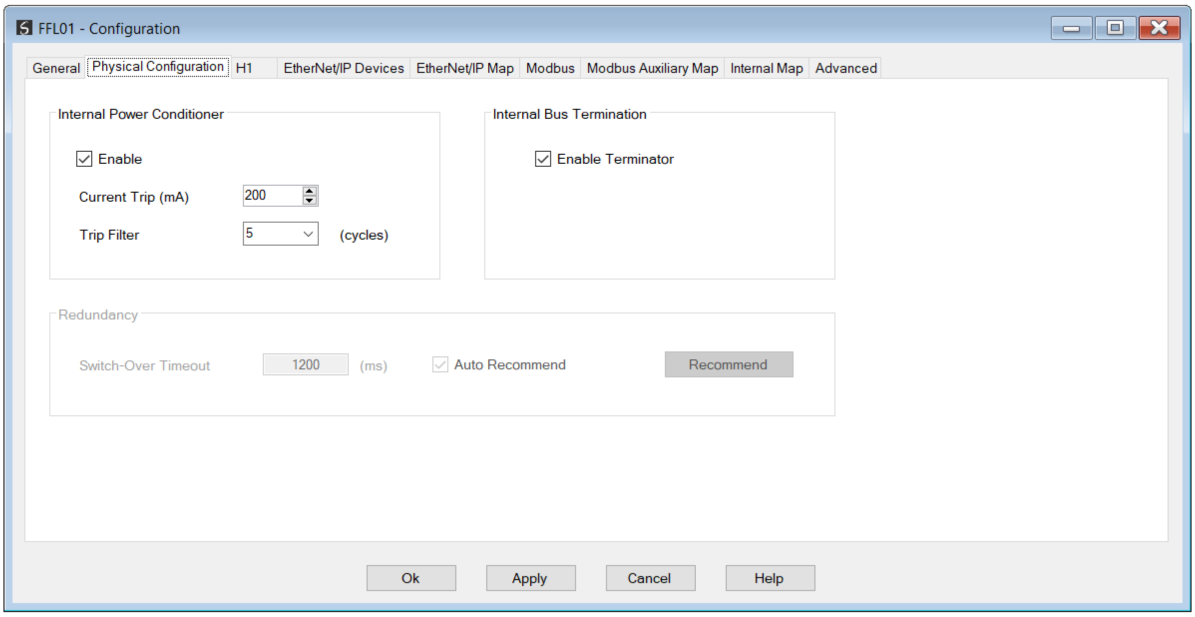


Figure 3.28 – FF Link Physcial Configuration configuration

The Physical Configuration consists of the following parameters:

Parameter	Description
Internal Power Conditioner	
Enable	This will enable/disable the internal power conditioner which will supply power to the H1 network. **NOTE:** When the internal power conditioner has been disabled the user will need to use and external power conditioner for the fieldbus to operate correctly.
Current Trip	The current limit (in mA) at which the FF Link will trip the fieldbus to protect against over current.
Trip Filter	The number of consecutive over current measurements cycles required before the fieldbus will be tripped and the power supply will be stopped. The cycle period is approximately 100us. Valid Range 0-50.
Internal Bus Terminator	
Enable Terminator	Enables or Disables the fieldbus terminator required for H1. Note that two terminators are required on the H1 bus, one at each bus extremity. See the H1 section in the appendix for more detail.
Redundancy	
Switch-Over Timeout	**This is only relevant in Redundant Master mode** This parameter is the amount of time (in milliseconds) the Standby Master will be the active Master without the Logix output assembly directing it to be the active master. After this time the module will revert to a Standby mode.

	See Redundancy section
Auto Recommend	**This is only relevant in Redundant Master mode** When this option is selected, the ***Switch-Over Timeout*** will be automatically calculated each time the configuration is applied.

Table 3.2 - Physcial Configuration parameters

3.5.3. H1

The H1 configuration is shown in the figure below. The FF Link ***H1*** configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

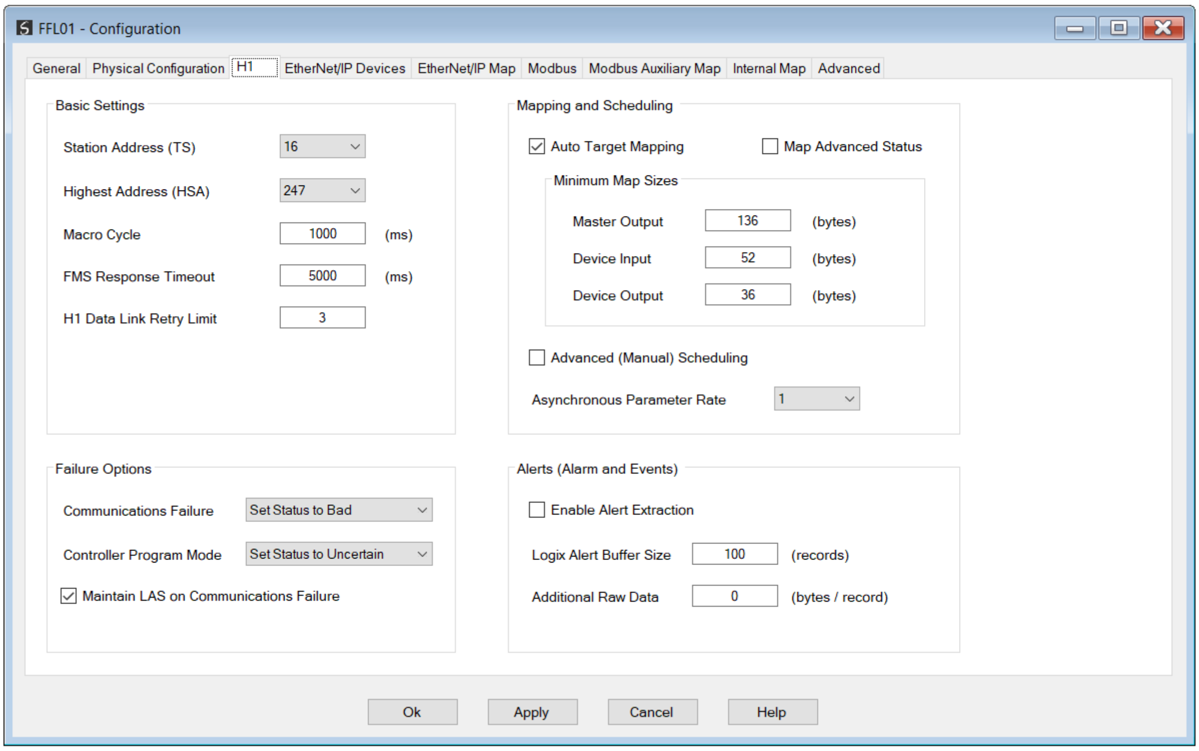


Figure 3.29 – FF Link H1 configuration

The H1 configuration consists of the following parameters:

Parameter	Description
	Basic Settings
Station Address (TS)	**H1 Station Address** for the FF Link module. TS should be different than any other station address on the H1 network, it should also be less-than or equal to the HSA below: Min: *16* Max: *247* Default: *16*

Highest Address (HSA)	**Highest Station Address**. This is the highest station address of the active stations (masters) will poll. A low HSA is better for H1 performance. Min: *19* Max: *247* Default: *247*
Macro Cycle	The macrocycle is a single iteration of a schedule within an H1 field device, which means it is the slowest update rate of each H1 field device. Devices can have their blocks and connectors scheduled to execute multiple times in a single macrocycle to increase their update rate. Default: *1000ms*
FMS Response Timeout	The amount of time the FF Link module will wait for an H1 device to respond to an FMS request. Default: *3000ms*
H1 Data Link Retry Limit	The number of retries the FF Link will attempt to pass the token on the H1 bus to a specific H1 device before flagging the device as offline. Default: *3*
Failure Options	
Communication Failure	The FF Link will send one of the following status options to all H1 devices receiving process variable data (from the FF Link) when the communication to the primary interface (e.g., EtherNet/IP or Modbus TCP) has failed. This will allow the device to go into the configure failsafe state. **Continue** – when communication is lost, the PVs sent to H1 devices will continue as per normal. **Set Status to Bad** – when communication is lost, the PV status sent to H1 devices will be set to *BAD – Comms Lost – Constant Value*. **Set Status to Uncertain** – when communication is lost, the PV status sent to H1 devices will be set to *Uncertain – Comms Lost – Constant Value*.
Controller Program Mode	The FF Link will send one of the status options to all H1 devices receiving process variable data (from the FF Link) when the Logix controller operating state (when the primary interface is configured for EtherNet/IP Target) is in Program Mode. This will allow the device to go into the configure failsafe state. **Continue** – when communication is lost, the PVs sent to H1 devices will continue as per normal. **Set Status to Bad** – when communication is lost, the PV status sent to H1 devices will be set to *BAD – Comms Lost – Constant Value*. **Set Status to Uncertain** – when communication is lost, the PV status sent to H1 devices will be set to *Uncertain – Comms Lost – Constant Value*.
Maintain LAS on Communications Failure	When the primary interface communication interface has failed, this options determines whether the FF Link continue as the Link Active Scheduler (LAS) on the H1 bus, or if it should become a backup LAS.
Mapping and Scheduling	
Auto Target Mapping	Slate will auto configure all the required mapping in the *Internal Map* tab for all H1 devices and interfaces, when this option is set. When this option is unchecked the user will manually have to configure the internal mapping to map the data between the H1 FF network and the primary interface. **NOTE:** It is strongly recommended that the user keep this option enabled for EtherNet/IP Target (Logix), Modbus TCP Master, and Modbus TCP Slave).
Map Advanced Status	Selecting this option will cause the Advanced Status to be added to the *Internal Map*.

Minimum Map Sizes	The configuration of suitable Minimum Map Sizes allows future changes to the configuration of the H1 Master and H1 field devices without affecting the mapping of other H1 devices. Since the mapping data for the Master and H1 devices is packed sequentially, the adding of a new PV in one H1 device (for example) would shift all the other H1 device's status and PV data down. The configuration of Minimum Map Sizes assigns spare data to each device to prevent this. ***Master Output (Minimum Map Size):*** The minimum number of bytes assigned for the H1 Master Output. This is typically consumed by PVs being sent from the Master to H1 Devices. ***Device Input (Minimum Map Size):*** The minimum number of bytes assigned for each H1 Device Input. This is typically consumed by PVs being compelled from the H1 Device, and configured Asynchronous Parameters being read from the H1 Device. ***Device Output (Minimum Map Size):*** The minimum number of bytes assigned for each H1 Device Output. This is typically consumed by the H1 Device Control, and configured Asynchronous Parameters being written to the H1 Device.
Advanced (Manual) Scheduling	This option allows the user to manually configure the scheduling of H1 device function blocks and connectors. ***Unchecked (Automatic)*** When the option is unchecked, each function block within each device is scheduled sequentially based on the execution time of each function block. Each function block is executed only once per macrocycle. The order of the function blocks (Schedule Order) can be modified by the user. The Connectors (compel) are scheduled immediately after the execution of the block to which they are connected. This option is suitable for most applications. ***Checked (Manual – Advanced)*** Here the user is required to configure the schedule for each function block and connector, by specifying the Initial Time, the number of occurrences (per macrocycle) and time between executions. This option is required for time critical systems and when different H1 devices require different PV update rates. It does however require the user to have a more advanced knowledge of FOUNDATION™ Fieldbus.
Asynchronous Parameter Rate	This parameter provides the number of asynchronous requests that can be sent every Macro Cycle. See the *Asynchronous Parameters* section of the H1 device for more information regarding the asynchronous parameters. Default: *1*
Alerts (Alarm and Events)	
Enable Alert Extraction	This parameter is the global enable for alert extraction from H1 devices. When this this is disabled, no alerts from H1 field devices will be processed (regardless of the field device *Alert Extraction* setting).
Logix Alert Buffer Size	When generating the Logix L5X file for the FF Link configuration (see the EtherNet/IP Target section), this parameter specifies the size of the Logix Alert UDT array.

Additional Raw Data	This parameter specifies the number of additional bytes that can be saved for each alert in the Logix buffer. Default: 0

Table 3.3 - H1 configuration parameters

NOTE: For details on the **EtherNet/IP Device** and **EtherNet/IP Map** configuration, please refer to the EtherNet/IP Originator Specific section 3.8.

NOTE: For details on the **Modbus** and **Modbus Auxiliary Map** configuration, refer to the Modbus TCP Slave section 3.9 or Modbus TCP Master section 3.10.

3.5.4. Internal Map

The internal data map is used to exchange data from the Ethernet primary interface to the H1 network and vice versa. Up to 500 items can be mapped. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

NOTE: When using EtherNet/IP Target, Modbus TCP Slave, or Modbus TCP Master, it is recommended to use the ***Auto Target Mapping*** feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

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 3.4 – Internal Map Copy functions

The data can also be reformatted when copying from the source to the destination. The reformat option provides nine different reformat options.

NOTE: The reformat option is only available for ***Byte to Byte*** Copy Functions.

<table>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>None</td><td>No reformatting applied (AA BB CC DD)</td></tr>
<tr><td>BB AA</td><td>16bit Byte swap</td></tr>
<tr><td>BB AA DD CC</td><td>32bit Byte Pair Swap</td></tr>
<tr><td>CC DD AA BB</td><td>Word Swap</td></tr>
<tr><td>DD CC BB AA</td><td>Word and Byte Pair Swap</td></tr>
<tr><td>Logix PV Real</td><td>This reformat is used for Logix L5X generation when the primary interface is EtherNet/IP target. It will format the data specifically for the Logix UDT created for Process Variables with an analog value (i.e., float).

Source Type
<table>
<tr><td>Byte 0</td><td>H1 Device PV Status (Enum)
Bit 0 – PV Status – Bad
Bit 1 – PV Status – Uncertain
Bit 2 – PV Status – Good
Bit 3 – PV Status – Good Cascade
Bit 4 – PV Limit – No Limit
Bit 5 – PV Limit – Low Limit
Bit 6 – PV Limit – High Limit
Bit 7 – PV Limit – Constant</td></tr>
<tr><td>Byte 1</td><td>H1 Device PV SubStatus (Raw)</td></tr>
<tr><td>Byte 2 - 3</td><td>Pad Byte</td></tr>
<tr><td>Byte 4 - 7</td><td>H1 Device PV Float Value</td></tr>
</table>

Destination Type
<table>
<tr><td>Byte 0</td><td>H1 Device PV Status (Raw)</td></tr>
<tr><td>Byte 1 - 2</td><td>Pad Byte</td></tr>
<tr><td>Byte 3</td><td>H1 Device PV Status (Enum)
Bit 0 – PV Status – Bad
Bit 1 – PV Status – Uncertain
Bit 2 – PV Status – Good
Bit 3 – PV Status – Good Cascade</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><td></td><td>Bit 4 – PV Limit – No Limit
Bit 5 – PV Limit – Low Limit
Bit 6 – PV Limit – High Limit
Bit 7 – PV Limit – Constant</td></tr>
<tr><td>Byte 4 - 7</td><td>H1 Device PV SubStatus (Enum)
Bit 0 – Bad – Configuration Error
Bit 1 – Bad – Not Connected
Bit 2 – D Bad – Device Failure
Bit 3 – Bad – Sensor Failure
Bit 4 – Bad – No Comms with last usable value
Bit 5 – Bad – No Comms with no usable value
Bit 6 – Bad – Out of service
Bit 7 – Bad – Transducer in MAN
Bit 8 – Bad – SIS
Bit 9 – Uncertain – Last usable value
Bit 10 - Uncertain – Substitute
Bit 11 - Uncertain – Initial Value
Bit 12 - Uncertain – Sensor Conversion Not Accurate
Bit 13 - Uncertain – Engineering Unit Range Violation
Bit 14 - Uncertain – Sub Normal
Bit 15 - Uncertain – Transducer in MAN
Bit 16 – Good – Active Block Alarm
Bit 17 - Good – Active Advisory Alarm
Bit 18 - Good – Active Critical Alarm
Bit 19 - Good – Unacknowledged Block Alarm
Bit 20 - Good – Unacknowledged Advisory Alarm
Bit 21 - Good – Unacknowledged Critical Alarm
Bit 22 - Good – Initial Fault State
Bit 23 – Good CAS – Initialization Acknowledge
Bit 24 – Good CAS – Initialization Request
Bit 25 – Good CAS – Not Invited
Bit 26 – Good CAS – Not Selected
Bit 27 – Good CAS – Local Override
Bit 28 – Good CAS – Fault Statue Active
Bit 29 – Good CAS – Initial Fault Active</td></tr>
<tr><td>Byte 8 - 11</td><td>H1 Device PV Float Value</td></tr>
</table>
</td>
</tr>
<tr>
<td>Logix PV Int.</td>
<td>This reformat is used for Logix L5X generation when the primary interface is EtherNet/IP target. It will format the data specifically for the Logix UDT created for Process Variables with a discrete value (i.e., integer).
Source Type
<table>
<tr><td>Byte 0</td><td>H1 Device PV Status (Enum)
Bit 0 – PV Status – Bad</td></tr>
</table>
</td>
</tr>
</table>

	Bit 1 – PV Status – Uncertain Bit 2 – PV Status – Good Bit 3 – PV Status – Good Cascade Bit 4 – PV Limit – No Limit Bit 5 – PV Limit – Low Limit Bit 6 – PV Limit – High Limit Bit 7 – PV Limit – Constant
Byte 1	H1 Device PV SubStatus (Raw)
Byte 2 - 3	Pad Byte
Byte 4	H1 Device PV Int Value

Destination Type

Byte 0	H1 Device PV Status (Raw)
Byte 1 - 2	Pad Byte
Byte 3	**H1 Device PV Status (Enum)** Bit 0 – PV Status – Bad Bit 1 – PV Status – Uncertain Bit 2 – PV Status – Good Bit 3 – PV Status – Good Cascade Bit 4 – PV Limit – No Limit Bit 5 – PV Limit – Low Limit Bit 6 – PV Limit – High Limit Bit 7 – PV Limit – Constant
Byte 4 - 7	**H1 Device PV SubStatus (Enum)** Bit 0 – Bad – Configuration Error Bit 1 – Bad – Not Connected Bit 2 – D Bad – Device Failure Bit 3 – Bad – Sensor Failure Bit 4 – Bad – No Comms with last usable value Bit 5 – Bad – No Comms with no usable value Bit 6 – Bad – Out of service Bit 7 – Bad – Transducer in MAN Bit 8 – Bad – SIS Bit 9 – Uncertain – Last usable value Bit 10 - Uncertain – Substitute Bit 11 - Uncertain – Initial Value Bit 12 - Uncertain – Sensor Conversion Not Accurate Bit 13 - Uncertain – Engineering Unit Range Violation Bit 14 - Uncertain – Sub Normal Bit 15 - Uncertain – Transducer in MAN

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>Bit 16 – Good – Active Block Alarm
Bit 17 - Good – Active Advisory Alarm
Bit 18 - Good – Active Critical Alarm
Bit 19 - Good – Unacknowledged Block Alarm
Bit 20 - Good – Unacknowledged Advisory Alarm
Bit 21 - Good – Unacknowledged Critical Alarm
Bit 22 - Good – Initial Fault State
Bit 23 – Good CAS – Initialization Acknowledge
Bit 24 – Good CAS – Initialization Request
Bit 25 – Good CAS – Not Invited
Bit 26 – Good CAS – Not Selected
Bit 27 – Good CAS – Local Override
Bit 28 – Good CAS – Fault Statue Active
Bit 29 – Good CAS – Initial Fault Active</td></tr>
<tr><td>Byte 8</td><td>H1 Device PV Int Value</td></tr>
</table>
</td></tr>
<tr><td>Modbus PV Real</td><td>This reformat is used when the primary interface is Modbus TCP Master/Slave. It will format the data specifically for Modbus for Process Variables with an analog value (i.e., float). The data will be formatted as below.

Source or Destination Type
<table>
<tr><td>MB Register 0</td><td>Pad byte + H1 Device PV Status</td></tr>
<tr><td>MB Register 1</td><td>H1 Device PV Float Value (lower 16 bits)</td></tr>
<tr><td>MB Register 2</td><td>H1 Device PV Float Value (upper 16 bits)</td></tr>
</table>
</td></tr>
<tr><td>Modbus PV Int.</td><td>This reformat is used when the primary interface is Modbus TCP Master/Slave. It will format the data specifically for Modbus for Process Variables with a discrete value (i.e., integer). The data will be formatted as below.

Source or Destination Type
<table>
<tr><td>MB Register 0</td><td>Pad byte + H1 Device PV Status</td></tr>
<tr><td>MB Register 1</td><td>H1 Device Int PV Value</td></tr>
</table>
</td></tr>
</table>

Table 3.5 – Internal Map Reformat Options

3.5.4.1. COPY FROM

One of six sources can be selected to copy from:
Internal, EIP Target, EIP Originator, MB Register, FF Master, or ***FF Device***.

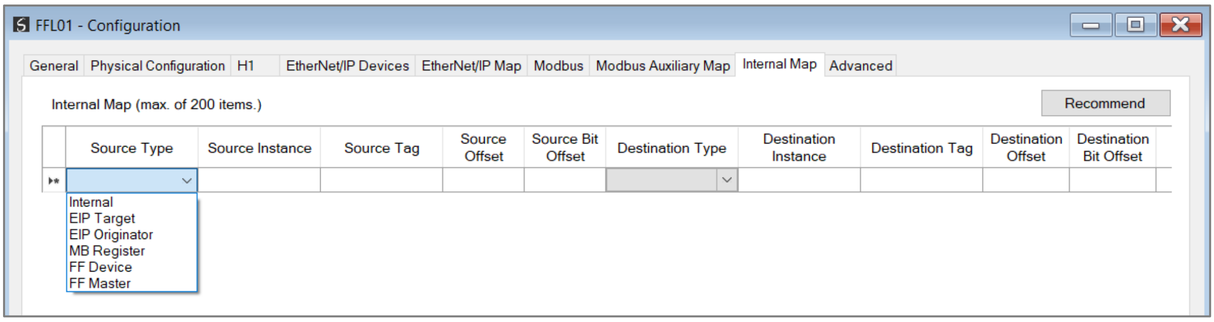


Figure 3.30 – Internal Map – Source Type

3.5.4.1.1. *INTERNAL*

Set the ***Source Type*** to ***Internal***, when copying data from the internal data space (IDS).

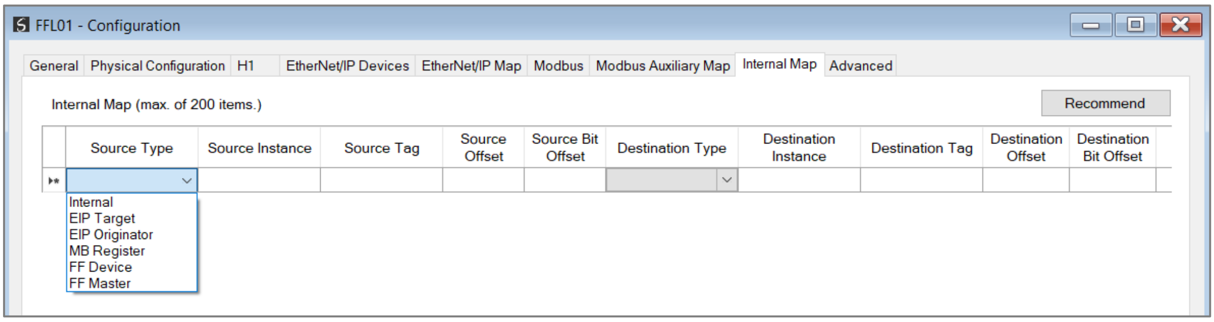


Figure 3.31 – Internal Map – Internal Source Type

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 first 2,000 bytes are however reserved for internal system use.
The ***Count*** is the number of **bytes** that will be copied.

3.5.4.1.2. *EIP TARGET*

Set the ***Source Type*** to ***EIP Target*** when copying data from a connection originator (e.g. the output assembly from the Logix Controller).

NOTE: When using EtherNet/IP Target it is recommended to use the ***Auto Target Mapping*** feature (in the H1 tab configuration). This will automatically map all the required data in the Internal Map.

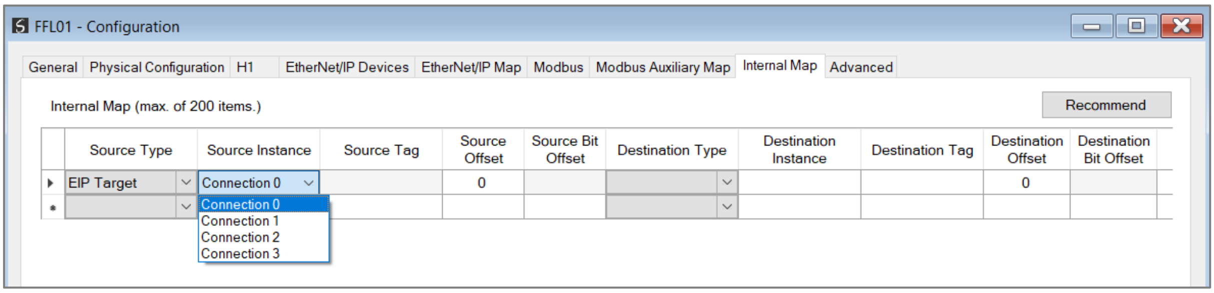


Figure 3.32 – Internal Map – EtherNet/IP Target Source Type

The ***Source Instance*** will be the connection number (0 to 3).
The ***Source Offset*** is the offset in the connection data of the EtherNet/IP output assembly from where the data must be copied.
The ***Count*** is the number of bytes that will be copied.

3.5.4.1.3. EIP ORIGINATOR

Set the ***Source Type*** to ***EIP Originator*** when copying data from an EtherNet/IP Device.

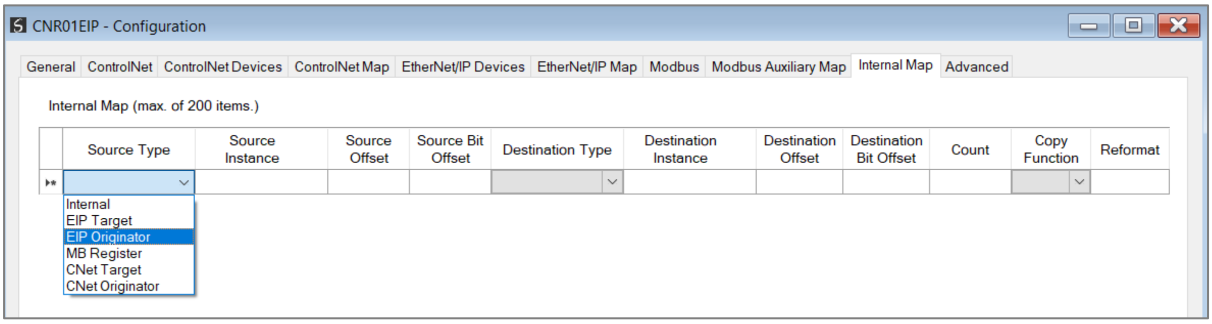


Figure 3.33 – Internal Map – EtherNet/IP Originator Source Type

The ***Source Instance*** will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

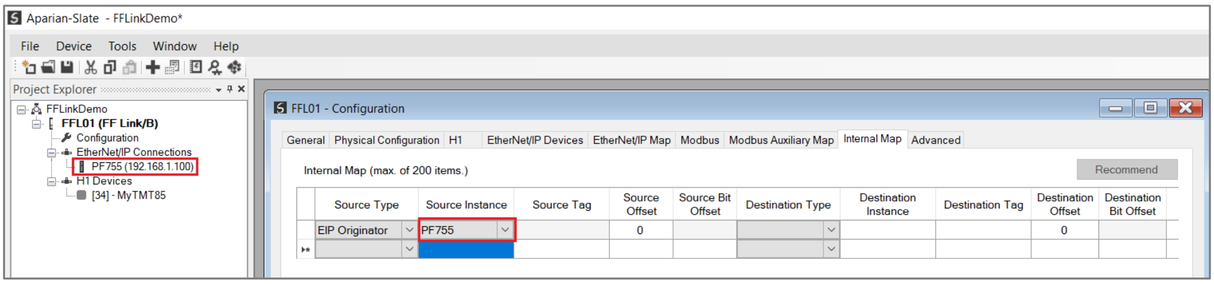


Figure 3.34 – Internal Map – EtherNet/IP Originator Source Instance

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

3.5.4.1.4. MODBUS REGISTER

Set the ***Source Type*** to ***MB Register*** (Modbus Register) when copying from Modbus data.

NOTE: When using Modbus TCP Master / Slave, it is recommended to use the ***Auto Target Mapping*** feature (in the H1 tab configuration). This will automatically map all the required data in the Internal Map.

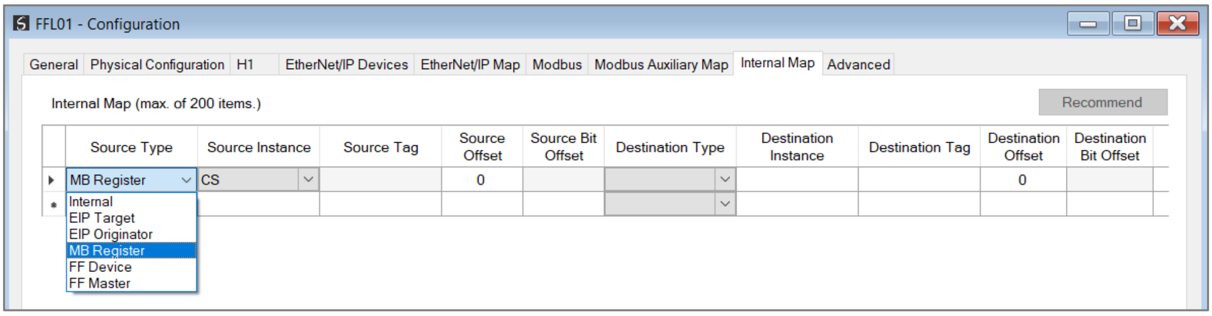


Figure 3.35 – Internal Map – Modbus Source Type

The ***Source Instance*** will be the Modbus register type required.

- HR – Holding Register
- IR – Internal Register
- CS – Coil
- IS – Discrete Input

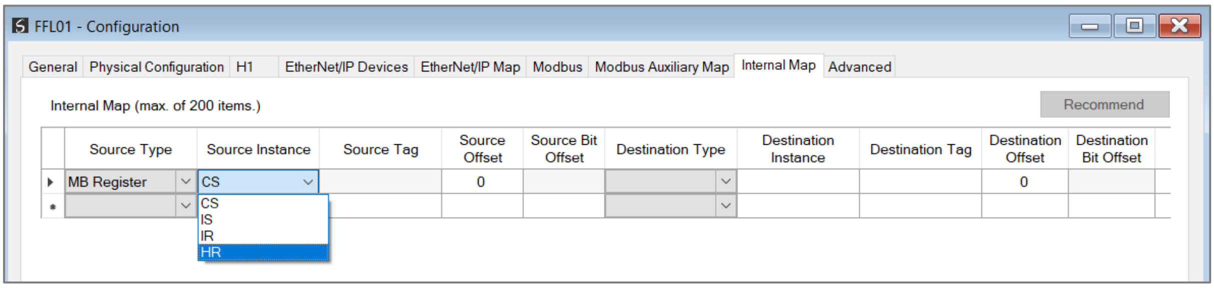


Figure 3.36 – Internal Map - Modbus Source Instance

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

3.5.4.1.5. FF DEVICE

Set the ***Source Type*** to ***FF Device***, when copying data from an H1 Device.

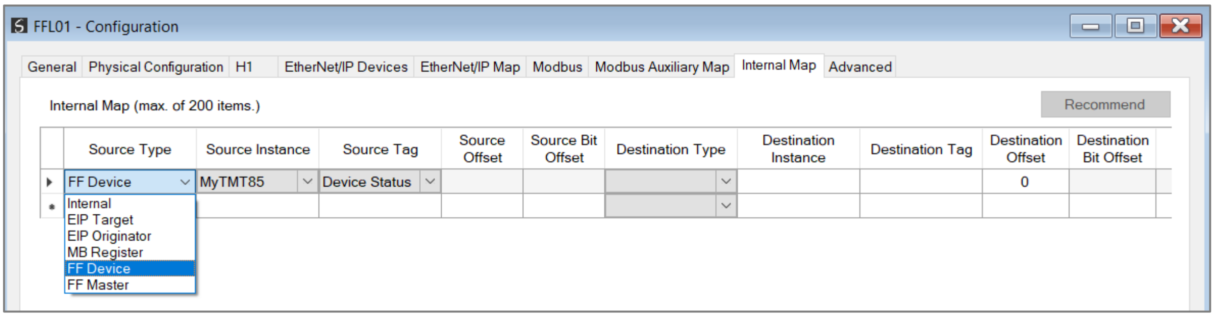


Figure 3.37 – Internal Map – FF Device Source Type

The ***Source Instance*** will be the H1 Device configured in the Slate H1 Device IO tree.
The ***Source Tag*** is the data that must be copied (*DeviceStatus, PV, Async Parameter*).
The ***Count*** is the number of bytes that will be copied (which will automatically be populated).
See section 4.6 with regards to the data formats for the various FF Device Source.

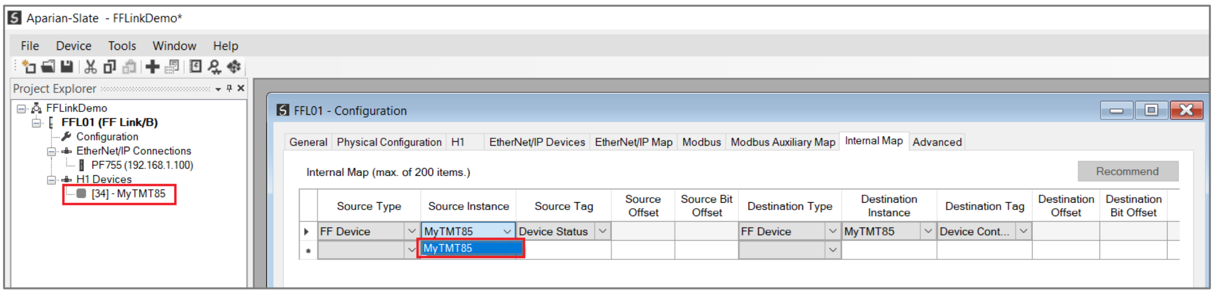


Figure 3.38 – Internal Map – FF Device Source Instance

3.5.4.1.6. FF MASTER

Set the ***Source Type*** to ***FF Master***, when copying data from the FF Link (i.e., the FF Master).

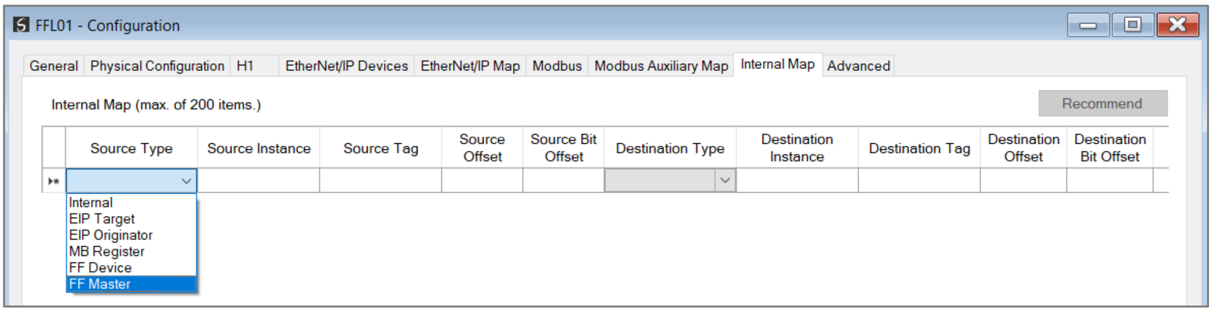


Figure 3.39 – Internal Map – FF Master Source Type

The ***Source Tag*** is the data that must be copied from the FF Master (*Master Status, Advanced Status* or *Alert*).
The ***Count*** is the number of bytes that will be copied (which will automatically be populated).
See section 4.6 with regards to the data formats for the various FF Master Source.

3.5.4.2. COPY TO

One of six destinations can be selected to copy to:
Internal, EIP Target, EIP Originator, Modbus Register, FF Device, or ***FF Master.***

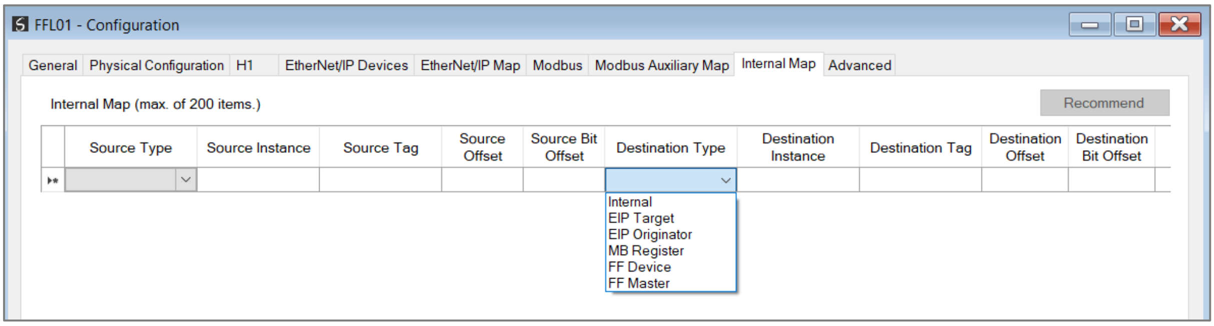


Figure 3.40 – Internal Map – Destination Type

3.5.4.2.1. INTERNAL

Set the ***Destination Type*** to ***Internal***, when copying data to the internal data space (IDS).

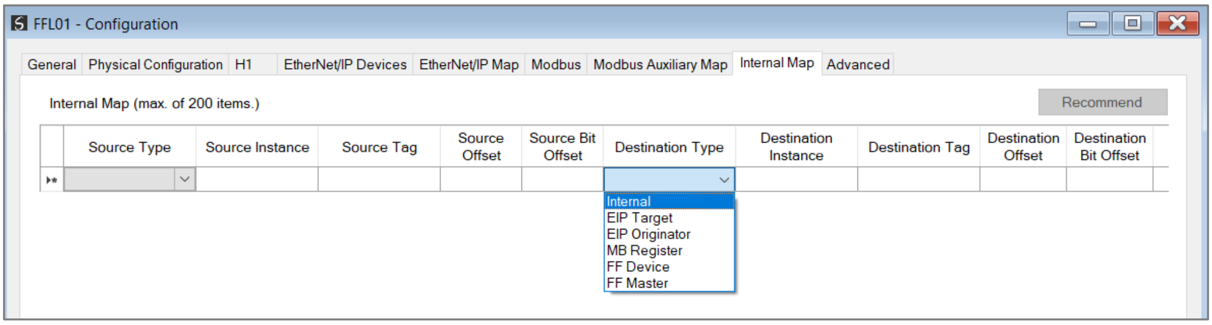


Figure 3.41 – Internal Map – Internal Destination

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 first 2,000 bytes are however reserved for internal system use.
The ***Count*** is the number of <u>bytes</u> that will be copied.

3.5.4.2.2. EIP TARGET

Set the ***Destination Type*** to ***EIP Target***, when copying data to the EtherNet/IP Target input assembly.

NOTE: When using EtherNet/IP Target it is recommended to use the ***Auto Target Mapping*** feature (in the H1 tab configuration). This will automatically map all the required data in the Internal Map.

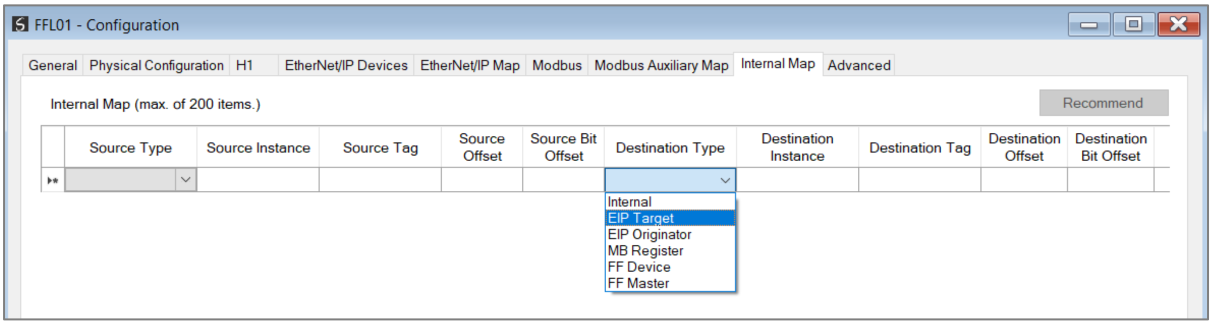


Figure 3.42 – Internal Map – EtherNet/IP Target Destination

The ***Destination Instance*** will be the connection number (0 – 3).
The ***Destination Offset*** is the offset in the Connection Data of the EtherNet/IP input assembly from where the data must be copied.
The ***Count*** is the number of bytes that will be copied.

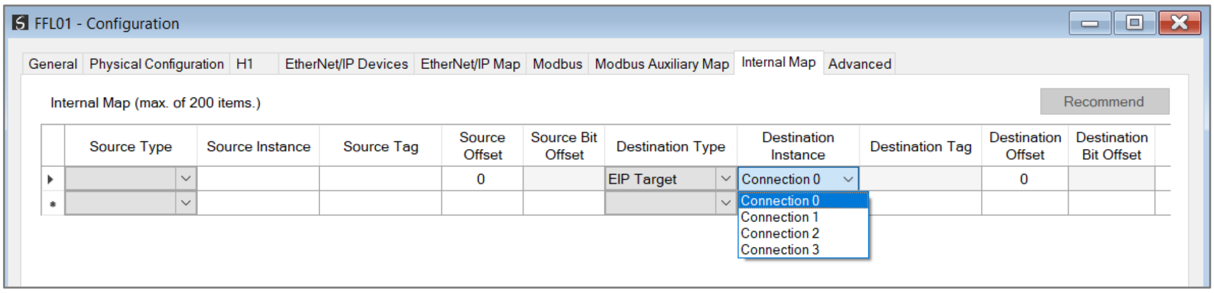


Figure 3.43 – Internal Map – EtherNet/IP Target Connection Number

3.5.4.2.3. EIP ORIGINATOR

Set the ***Destination Type*** to ***EIP Originator***, when copying data to an EtherNet/IP IO device **Output** Assembly.

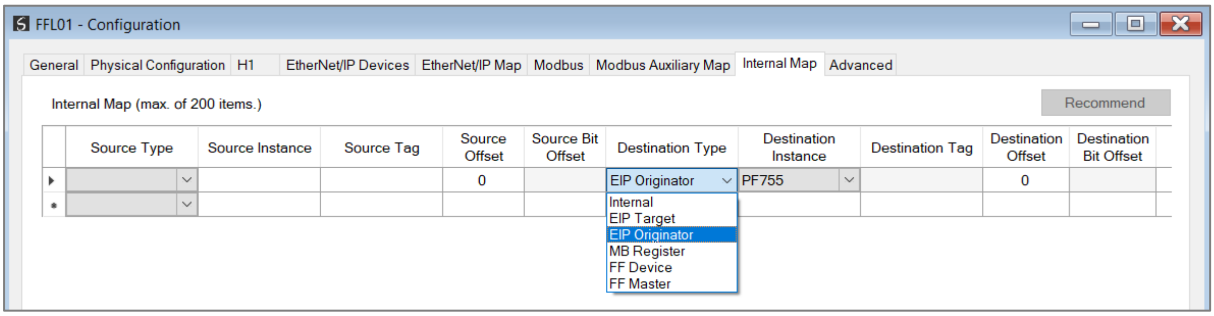


Figure 3.44 – Internal Map – EtherNet/IP Originator Destination

The ***Destination Instance*** will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

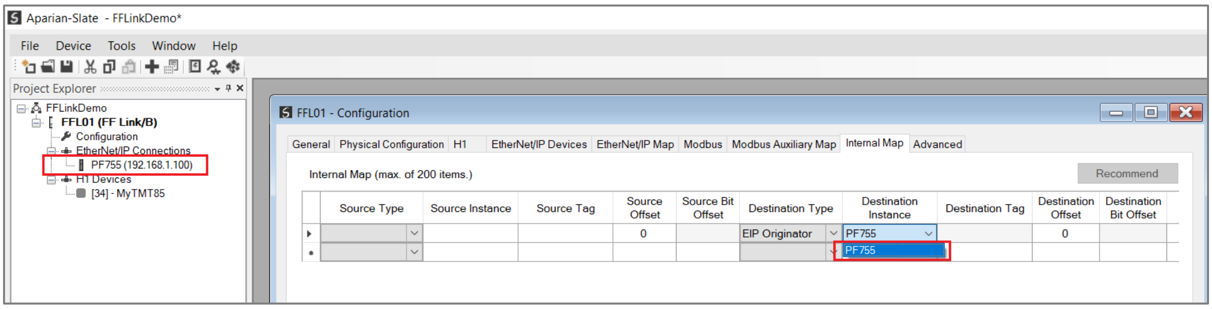


Figure 3.45 – Internal Map – EtherNet/IP Originator Destination Instance

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

3.5.4.2.4. MODBUS REGISTER

Set the ***Destination Type*** to ***MB Register*** (Modbus Register), when copying data to a Modbus Register.

NOTE: When using Modbus TCP Master / Slave, it is recommended to use the ***Auto Target Mapping*** feature (in the H1 tab configuration). This will automatically map all the required data in the Internal Map.

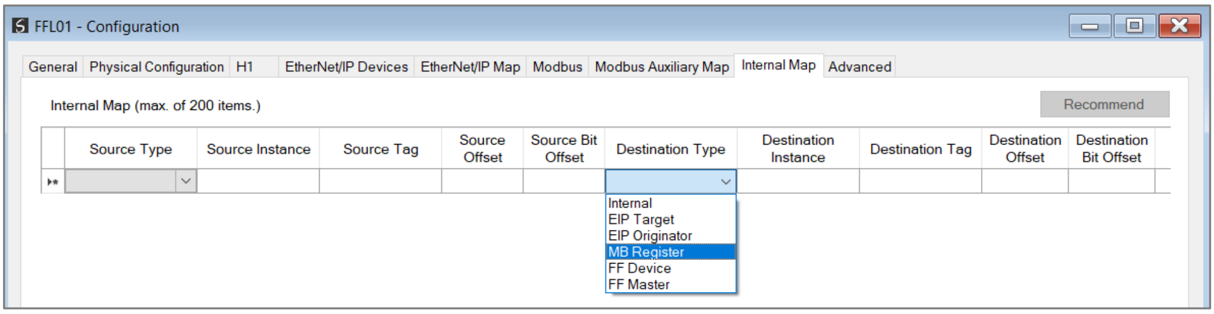


Figure 3.46 – Internal Map - Modbus Destination Type

The ***Destination Instance*** will be the Modbus register type required.

- HR – Holding Register
- IR – Internal Register
- CS – Coil
- IS – Discrete Input

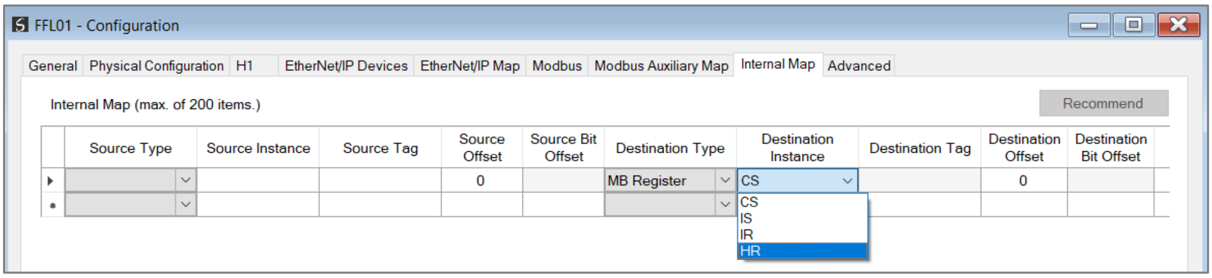


Figure 3.47 – Internal Map – Modbus Destination Instance

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

3.5.4.2.5. FF DEVICE

Set the ***Destination Type*** to ***FF Device***, when copying data to an H1 Device.

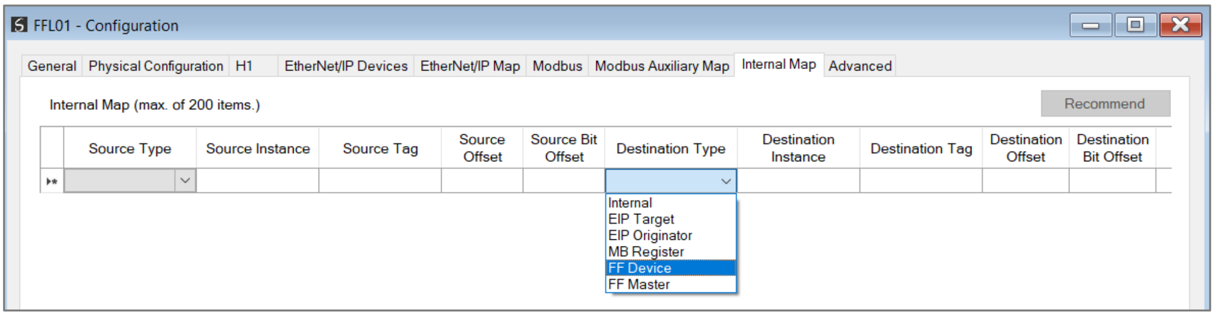


Figure 3.48 – Internal Map – FF Device Target Destination Type

The ***Destination Instance*** will be the H1 Device configured in the Slate H1 Device IO tree.
The ***Destination Tag*** is the H1 Device data to which the source data must be copied (*Device Control, PV, Async Parameter*).
The ***Count*** is the number of bytes that will be copied (which will automatically be populated).
See section 4.6 with regards to the data formats for the various FF Device Destination.

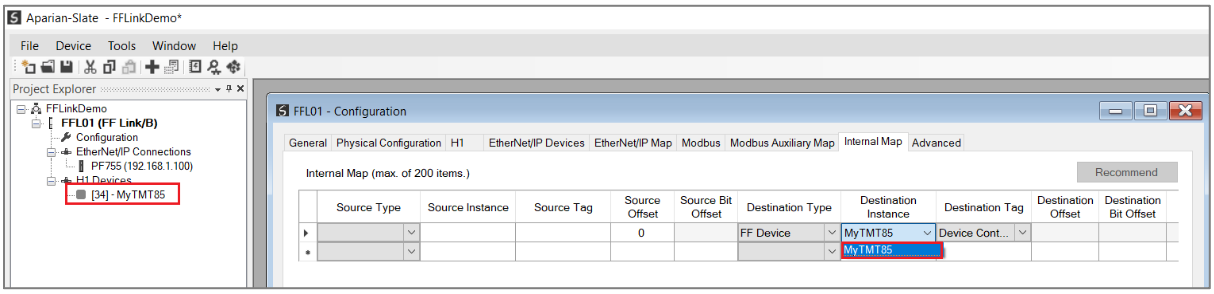


Figure 3.49 – Internal Map – FF Device Target Destination Instance

3.5.4.2.6. FF MASTER

Set the ***Destination Type*** to ***FF Master***, when copying data to the FF Link (i.e., FF Master).

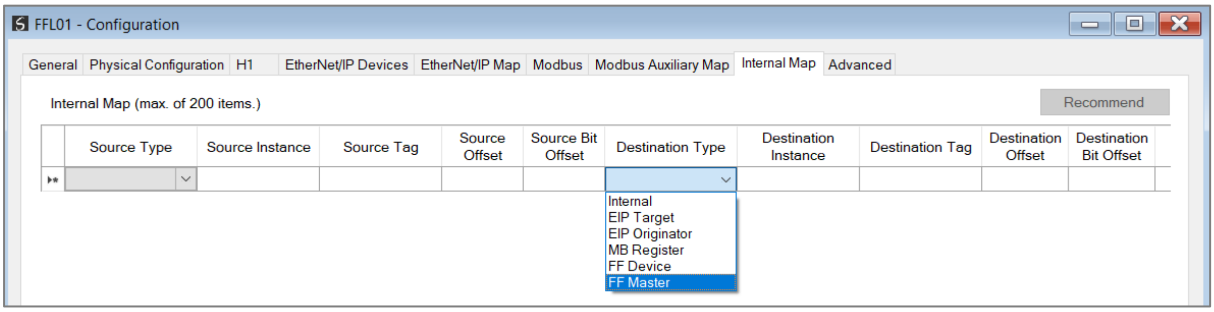


Figure 3.50 – Internal Map – FF Master Originator Destination Type

The ***Destination Tag*** will be destination (*Master Control* or *Alert Ack*) to where the source data is copied.

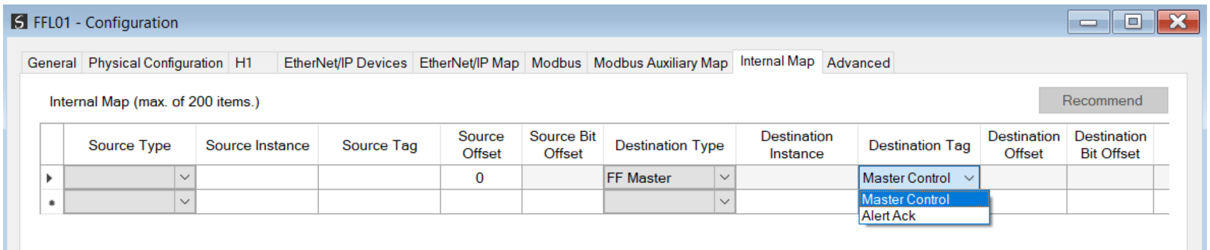


Figure 3.51 – Internal Map – FF Mater Destination Tag

The ***Count*** is the number of bytes that will be copied (which will automatically be populated). See section 4.6 with regards to the data formats for the various FF Master Destination.

3.5.5. ADVANCED

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

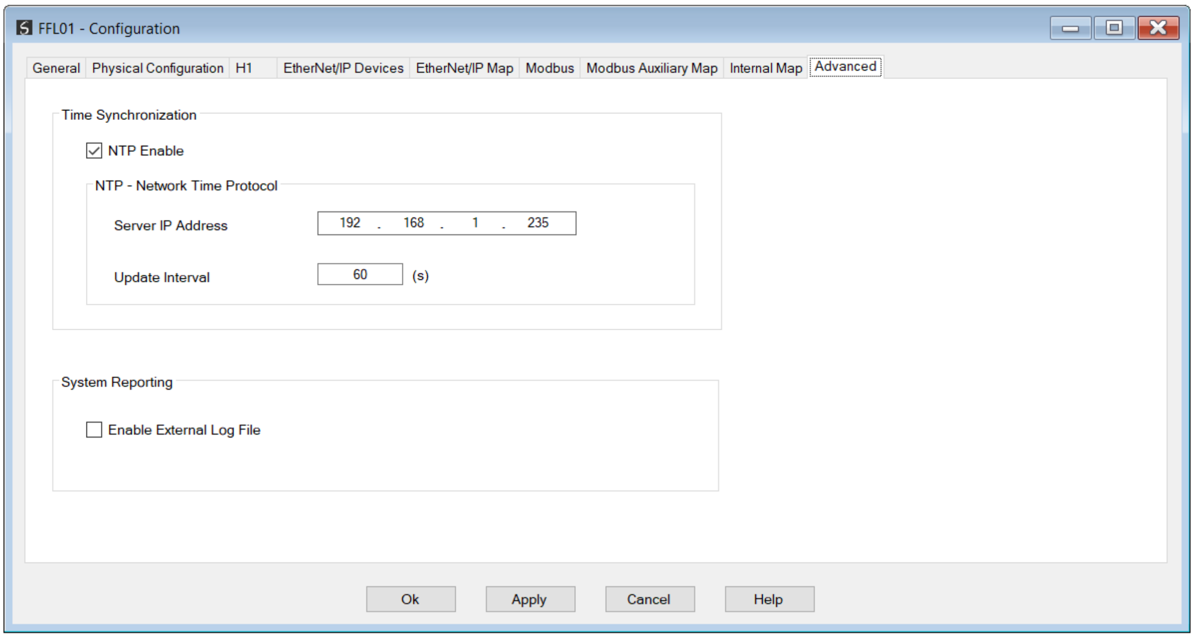


Figure 3.52 – FF Link Advanced configuration

The Advanced configuration consists of the following parameters:

Parameter	Description
	NTP
NTP Enable	The FF Link 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 FF Link will request time from the NTP Server.
	System Reporting
Enable External Log File	This option enables the generation of external log files which can be used by software engineers to determine the source of user system related issues. This option should remain unselected for performance. *Default: Unchecked*

Table 3.6 - Advanced configuration parameters

3.6. ADD H1 DEVICE

The user will need to add each H1 device to the FF Link which can then be configured. This can either be done by manually selecting an H1 device or by using the Live List to add the H1 device.

3.6.1. Manual H1 Device Add

To manually add an H1 device, right-click on the ***H1 Devices*** item in the tree and selecting ***Add H1 Device***.

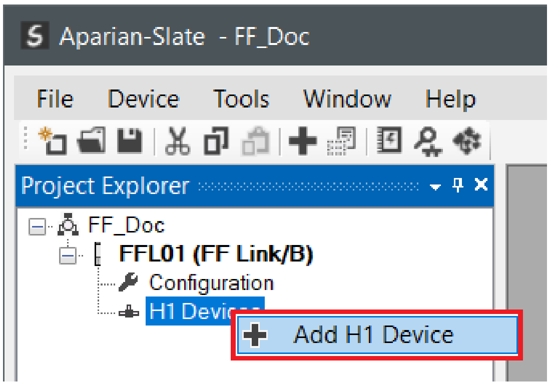


Figure 3.53 – Adding an H1 Field Device

Next the user will need to select the device to be added to the FF Link. This is done by selecting the device from the ***Device Description File Selector*** and pressing ***Ok***.

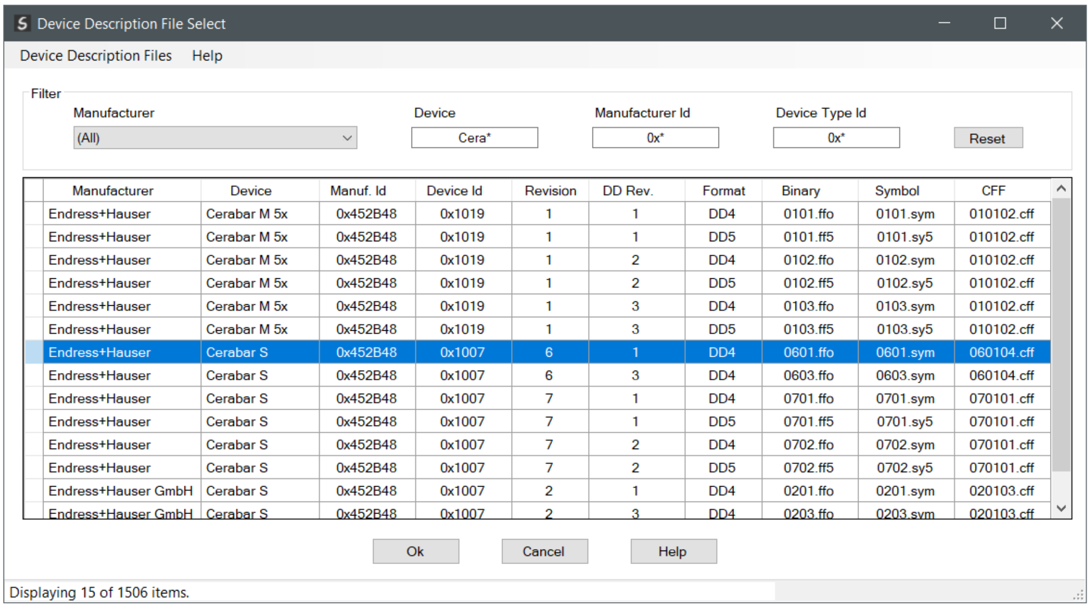


Figure 3.54 – Selecting an H1 Field Device

Once the device has been selected, the General Configuration page will be opened and the device will be assigned the first available H1 Station Address. Note that the device will only be added once the ***Ok*** or ***Apply*** button has been pressed.

3.6.2. Live List H1 Device Add

When adding a device using the Live List, the user will need to be online with the FF Link module in Slate. Once online, right-click on the module and select *Status.*

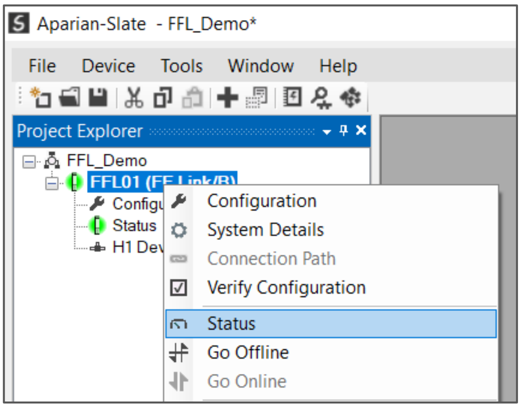


Figure 3.55 – Opening Status Window

Next select the ***Live List*** tab in the status window to see all the detected devices.

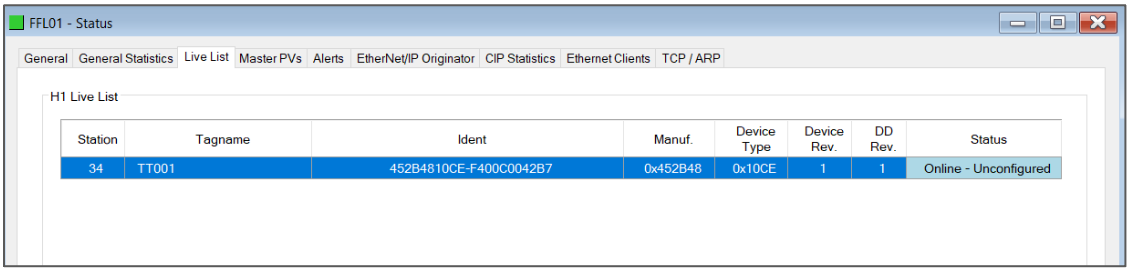


Figure 3.56 – Live List tab

Right-click on the device that must be added and select ***Add Device***.

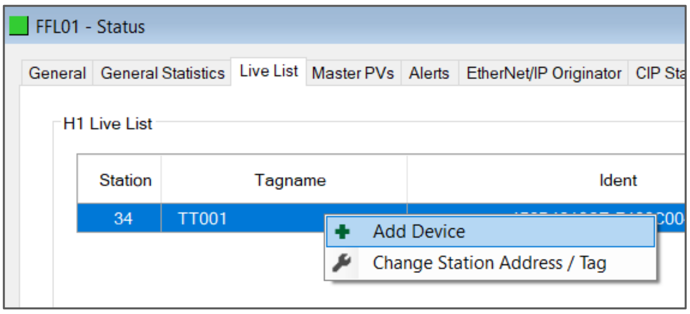


Figure 3.57 – Live List add device

The software will then attempt to match the online device in the live list with a device in the FF DD Catalog. If a matching DD file is found, the device will be added at the station number in the live list. If no exact match is found in the DD catalog, then the best possible alternative will be provided for the user to add.

3.6.3. CONFIGURATION

3.6.3.1. AUTO-INSTANTIATE CONTROL STRATEGY

Once a device is added, the user will be prompted to optionally ***Auto-Instantiate*** a ***Control Strategy***. The process will instantiate the Resource block, all Transducer blocks and the selected Function blocks with recommended connectors.

NOTE: The Auto-instantiated Control Strategy provides a basic configuration that should be sufficient in most applications. Depending on the application required, additional Function Blocks and/or Connectors may need to be instantiated by the user.

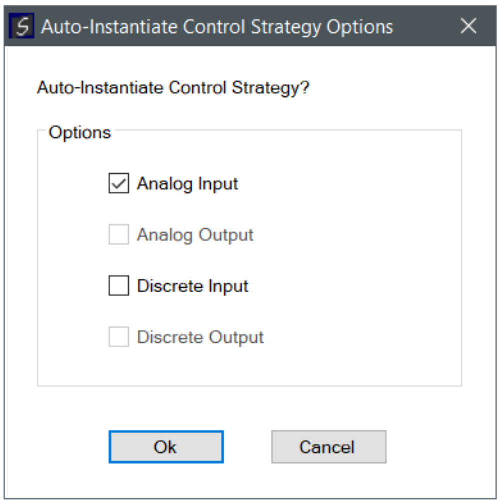


Figure 3.58 – Auto-Instantiate Prompt

3.6.3.2. UPLOAD CONTROL STRATEGY

As an alternative to the Auto-Instantiate process, the current control strategy in the device can be uploaded.

NOTE: In order to Upload from the H1 device, the FF Link must be online, and the current configuration must be blank.

To upload from an H1 device, right-click on a blank space in the ***Block Editor*** tab and select the ***Upload Control Strategy*** option.

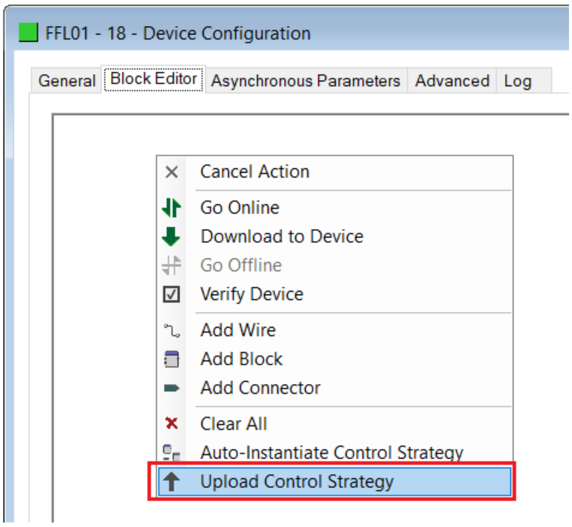


Figure 3.59 – Upload Control Strategy

NOTE: The Upload Control Strategy provides a best match for the device configuration within the FF Link. Depending on the application required, and the previous configuration environment, certain connectors may need to be modified.

NOTE: Once the Upload Control Strategy has been completed, the configuration will need to be re-downloaded to the H1 device in order to synchronise the VCRs and other FF Link parameters.

3.6.3.3. GENERAL

The General configuration is shown in the figure below. The Device configuration window is opened by either double clicking on the H1 device in the tree or right-clicking the H1 device and selecting ***Configuration***.

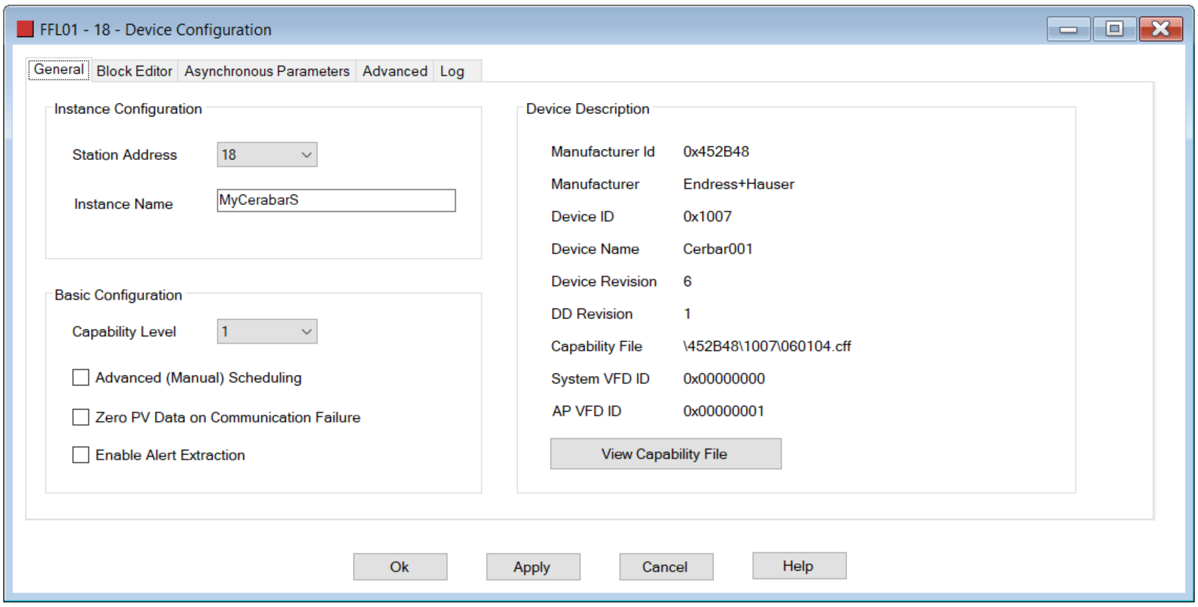


Figure 3.60 – Field Device General configuration parameters

The General configuration consists of the following parameters:

Parameter	Description
Instance Configuration	
Station Address	This is the station address configured for the H1 device. This is the address the FF Link will use to look for and configure the device for Data Exchange.
Instance Name	The device instance name which will be used to create the Tag names and UDTs in Logix.
Basic Configuration	
Capability Level	For devices that support multiple Capability Levels, the selected Capability is selected here.
Advanced (Manual) Scheduling	This option allows the user to manually configure the scheduling of this H1 device's function blocks and connectors. ***Unchecked (Automatic)*** When the option is unchecked, each function block is scheduled sequentially based on the execution time of the previous function block. Each function block is executed only once per macrocycle. The order of the function blocks (Schedule Order) can be modified by the user. The Connectors (compel commends) are scheduled immediately after the execution of the block to which they are connected. This option is suitable for most applications. ***Checked (Manual – Advanced)*** Here the user is required to configure the schedule for each function block and connector, by specifying the Initial Time, the number of occurrences (per macrocycle) and time between executions.

	This option is required for time critical systems and when different H1 devices require different PV update rates. It does however require the user to have a more advanced knowledge of FOUNDATION™ Fieldbus.
Zero PV Data on Communication Failure	When this option is set, the PV data returned to the Primary Interface for a specific H1 device will be forced to zero if the H1 device goes offline.
Enable Alert Extraction	This parameter enables the alert extraction functionality for the H1 device. When this this is disabled, no alerts from this specific H1 field device will be processed. **NOTE**: This setting is only valid if the global Alert Extraction has been enabled (see General module H1 settings).

Table 3.7 –Device General configuration parameters

The Device Description frame shows the information about the DD files (binary and capability) used to configure the H1 device.

The raw Capability file can be viewed by clicking on the ***View Capability File*** button.

3.6.3.4. BLOCK EDITOR

The H1 configuration is shown in the figure below. The Device ***H1 Configuration*** window is opened by either double clicking on the H1 device in the tree or right-clicking the H1 device and selecting ***Configuration***.

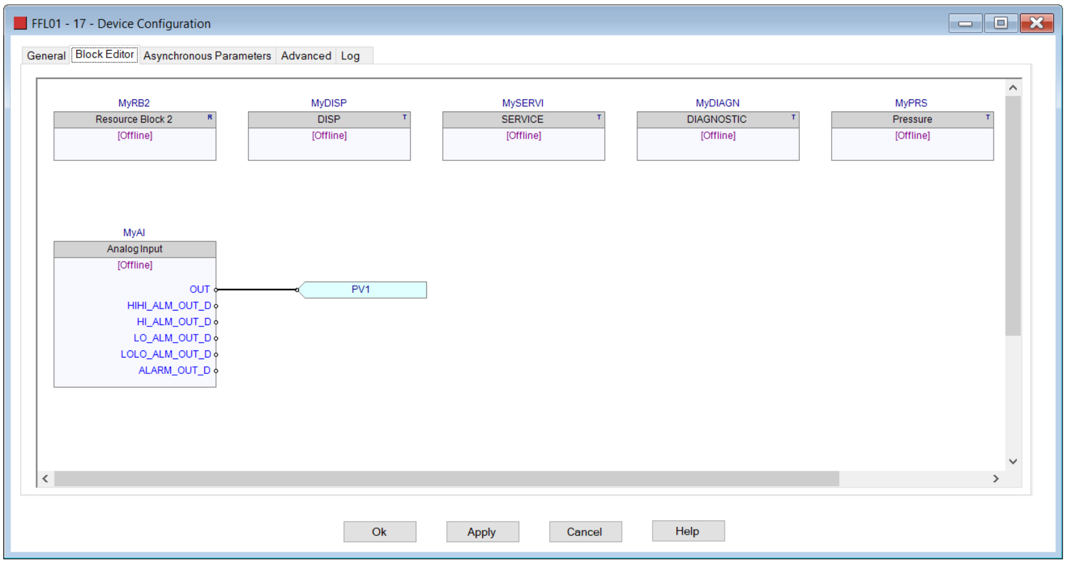


Figure 3.61 – Field Device H1 configuration parameters

The block editor is used to configure the device's control strategy by:

- Adding blocks (Resource, Transducer and Function Blocks),
- Adding connectors to publish to, or subscribe from, other H1 devices,
- Wiring between blocks and connectors,

- Configuring block and connector parameters.

3.6.3.4.1. ADD / DELETE BLOCK

If a device was auto instantiated when added, the Resource, Transducer(s), and at least one Function Block will be added. Certain device can have more than one Function block added (e.g., a multi-variable device).

If a device was **not** auto instantiated when added, then the user will need to manually add all the required blocks.

⚠ **NOTE:** Each H1 device will require at least one Resource block, one Transducer block, and one Function block to operate correctly.

To add a block, the user will need to right-click on an open space in the Block Editor window and select ***Add Block***.

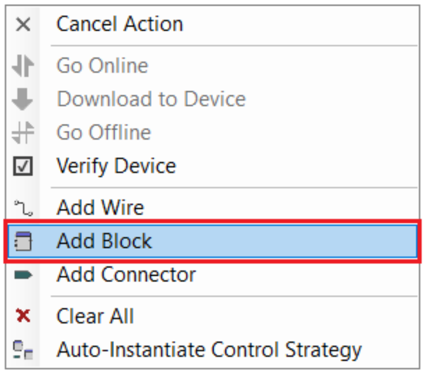


Figure 3.62 – Block Editor – Add Block selection

Next the user will need to select the block to be added (as shown below).

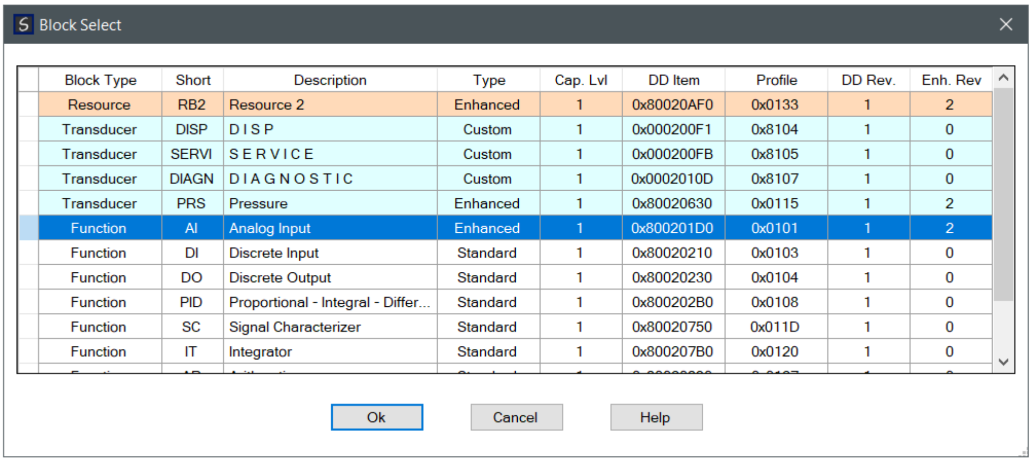


Figure 3.63 – Block Editor – Add Function Block

Once the block has been added it will appear in the Block Editor window. Blocks can be moved in the block editor window by left-clicking (and holding) on the block and dragging it to the new preferred position.

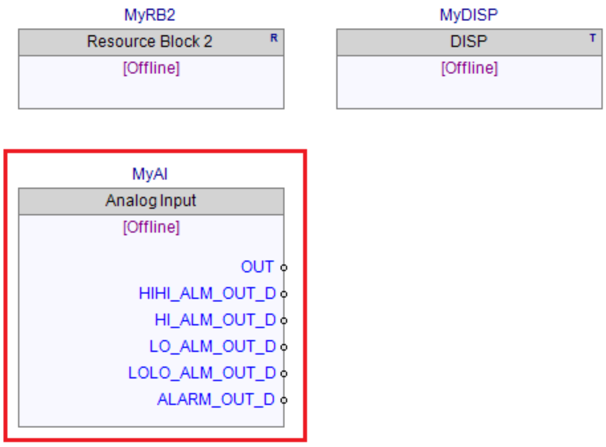


Figure 3.64 – Block Editor – Added function block

To delete a block, either right-click on the block and select the ***Delete*** *option* or, select the block and press the ***Delete*** key*.*

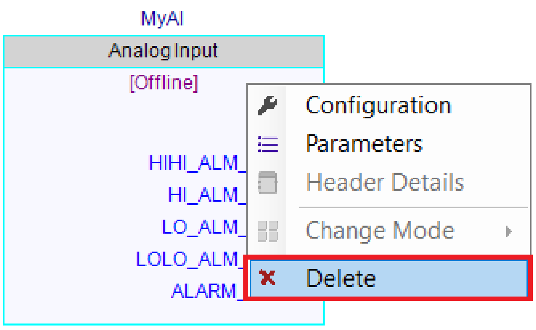


Figure 3.65 – Block Editor – Delete block

3.6.3.4.2. BLOCK CONFIGURATION

The block configuration will allow the user to change the Block Tag (in the block editor), view detailed information about the block, and allow the user to manually schedule the block (if required).

The block configuration can be opened by either, double-clicking on the block or, right-clicking on the block and selecting the ***Configuration*** *option*

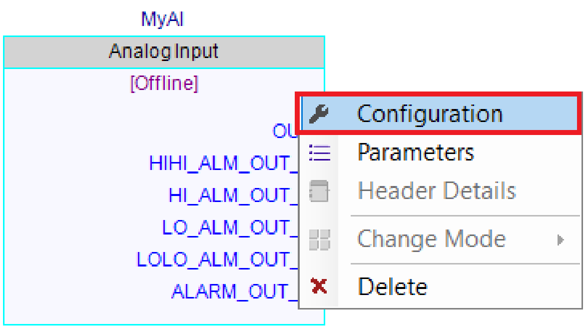


Figure 3.66 – Block Editor – Block Configuration

The General tab of the block configuration window has the Block Identity and Block Definition (as shown below). The Block Tag can be changed to provide the Block a specific name in the Block Editor window.

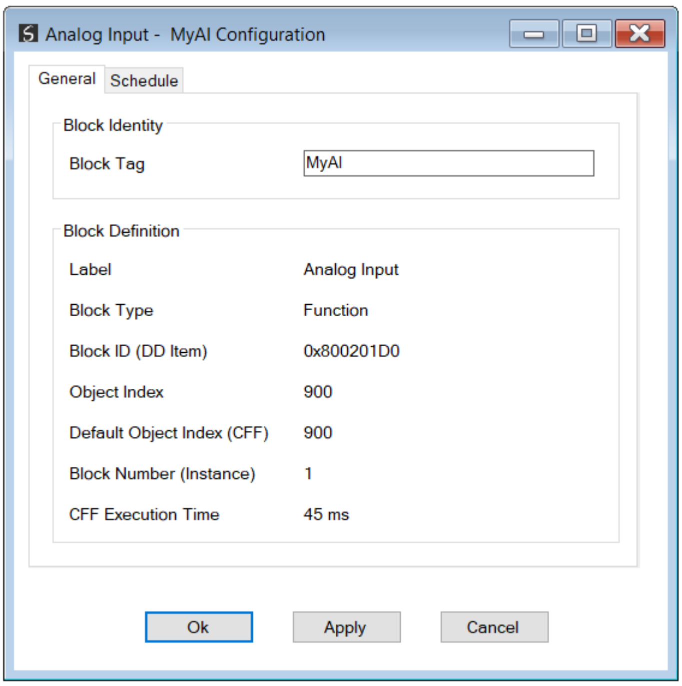


Figure 3.67 – Block Configuration – General

The Schedule tab of the block configuration window will allow the user to specify how the (Function) block is scheduled.

NOTE: Only Function blocks can be scheduled, and thus this tab is disabled for Resource and Transducer blocks.

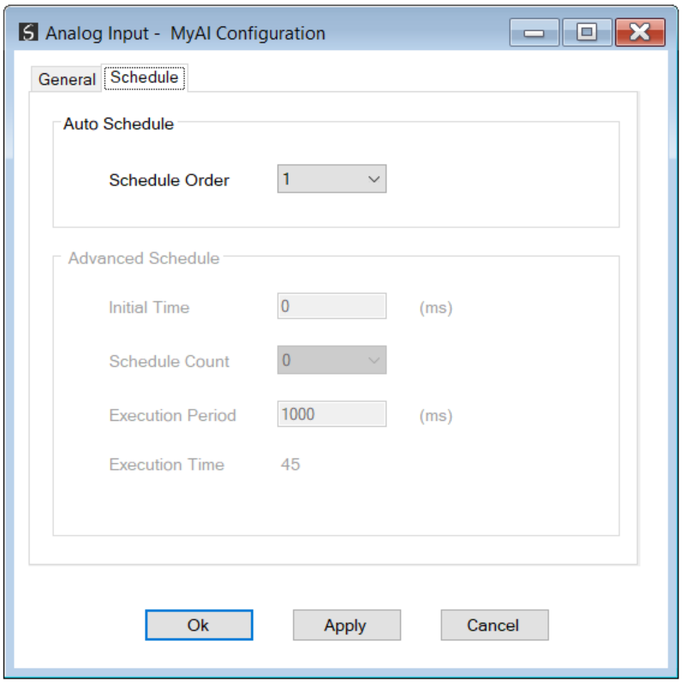


Figure 3.68 – Block Configuration – Schedule (Auto)

The configuration options of the Schedule tab depend on the configuration of the H1 device's ***Advanced (Manual) Scheduling*** option.

When ***Advanced (Manual) Scheduling*** is disabled, then the user can modify only the function block's ***Schedule Order***.

Parameter	Description
Schedule Order	The automatic schedule orders the function blocks in increasing Schedule Order (that is the lowest value is scheduled first). Where two or more blocks have the same Schedule Order, then the blocks will be scheduled in the order in which they were instantiated. *Default: 1*

Table 3.8 – Block Schedule - Advanced Scheduling Disabled

NOTE: When ***Advanced (Manual) Scheduling*** is disabled, each function block is scheduled to execute only once per Macro Cycle.

When ***Advanced (Manual) Scheduling*** is enabled, then the user is required to configure a suitable Schedule Strategy using the following:

Parameter	Description
Initial Time	The time (in milliseconds) from the start of the Macro Cycle when the function block will first execute.
Repeat Count	The number of times (in addition to the initial execution) that the function block will repeat within the Macro Cycle. For example, if the Repeat Count is set to 2, then the block will execute 3 times per Macro Cycle.
Execution Period	The time (in milliseconds) between the start of the block execution and the start of the subsequent execution.
Execution Time	The time that the function block will take to execute. This parameter is read only and is provided for guideline purposes only.

Table 3.9 – Block Schedule - Advanced Scheduling Enabled

NOTE: It is important that the entered parameters do not cause the function block to execute beyond the Macro Cycle.

NOTE: It is important that the entered parameters do not cause the function block to execute at the same time as another function block in the same device.

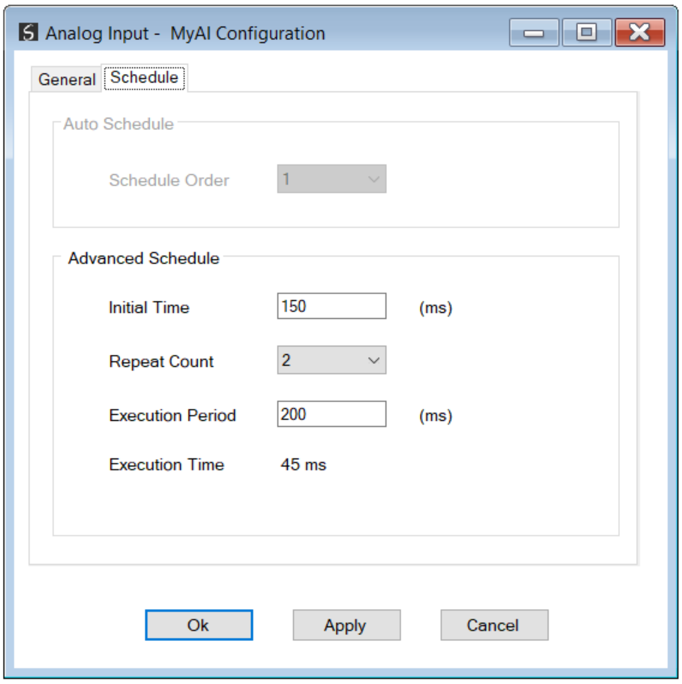


Figure 3.69 – Block Configuration – Schedule (Manual - Advanced)

The timing diagram below shows the block execution for the example configuration above.

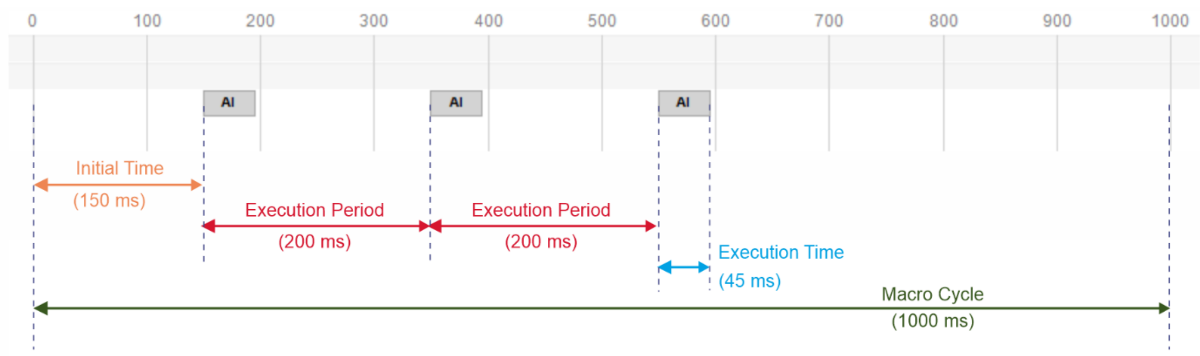


Figure 3.70 – Block Schedule Timing (Manual - Advanced)

3.6.3.4.3. BLOCK PARAMETERIZATION

The parameter view of the block will allow the user to change various settings specific to the block and the H1 device.

The block parameters can be opened by either right-clicking on the block and selecting ***Parameters***, or by double-clicking on the block whilst holding down the Shift key.

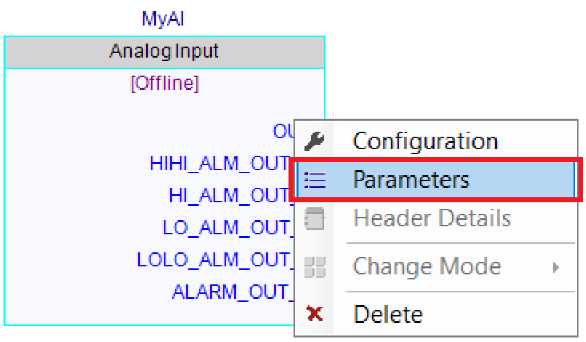


Figure 3.71 – Block Editor – Select Block Parameters

The Block Parameter window is shown below.

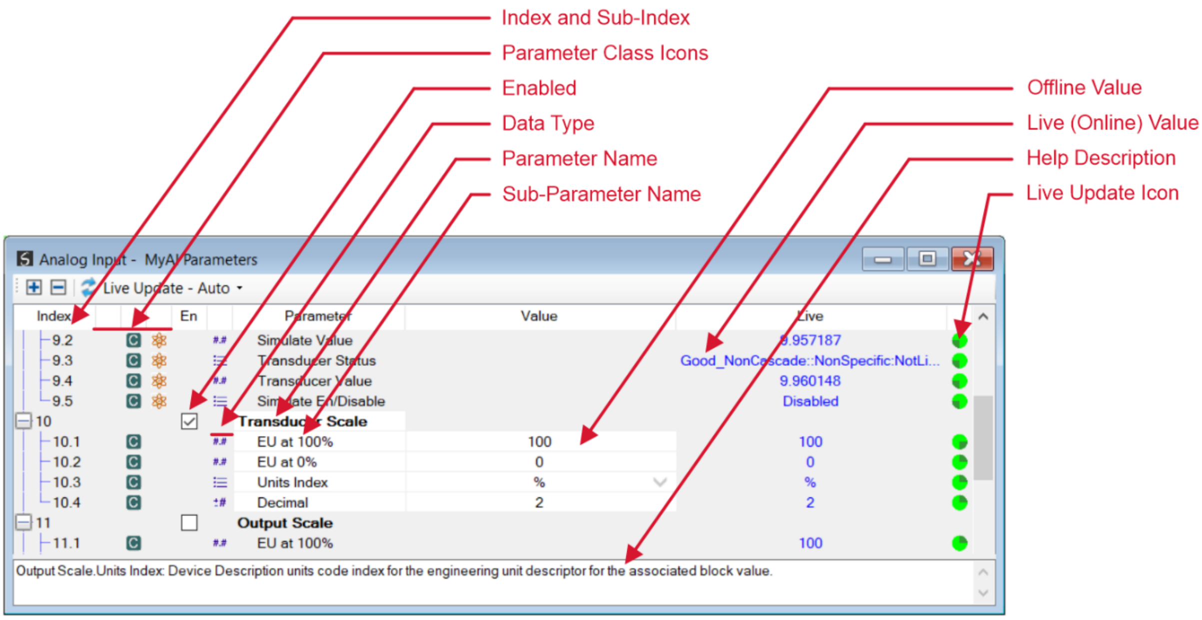


Figure 3.72 – Block Editor – Block Parameters

The Parameter Class is indicated by the combination of up to 3 icons as follows:

Class Icon	Description
	Alarm
	Operator
	Tuning
	Contained
	Local
	Input
	Output
	Dynamic

Table 3.10 – Block Parameters – Class Icons

The Parameter Data Type is indicated by one of the following icons:

Data Type Icon	Description
	Bool
±#	Integer
#	Unsigned Integer
#.#	Float
$\#.\#_d$	Double
	Enumerated
	Bit Enumerated
	Index
asc	ASCII
asc	Packed ASCII
	Password
010 110	Bitstring
	Date
	Time / Time Value
	Date Time
	Duration
euc	EUC
str^{o}	Octet String
str^{v}	Visible String

Table 3.11 – Block Parameters – Data Type Icons

The block parameters can be change either offline or online.

OFFLINE CHANGE

When changing the parameter offline, the user will need to enable the block parameter by selecting the (Enable) option in the ***En*** column (as shown below).

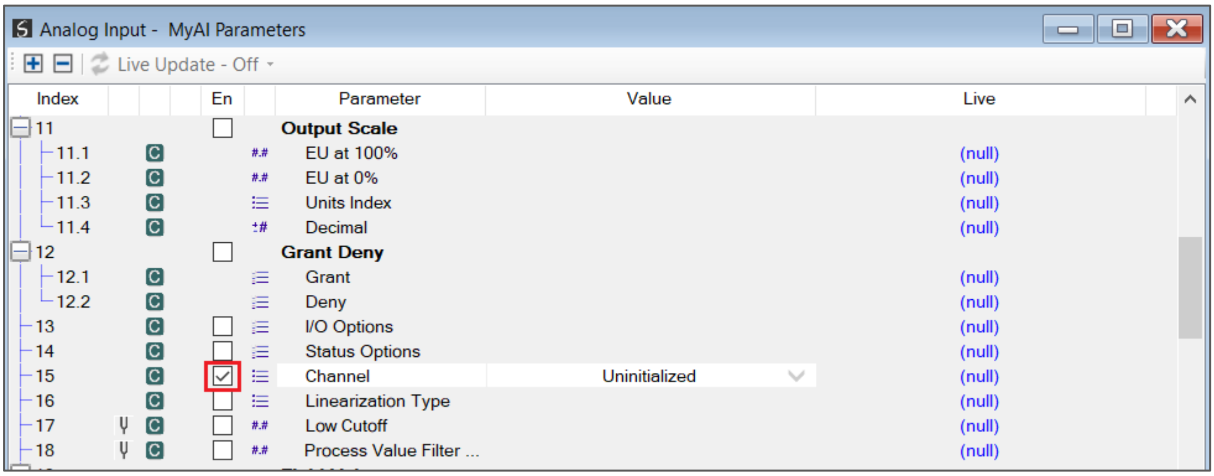


Figure 3.73 – Block Editor – Enable Offline Parameter

Once the parameter has been enabled, its offline value can be modified by the user. The values of enabled parameters are stored in the offline Slate project and also written to the H1 device during a device download.

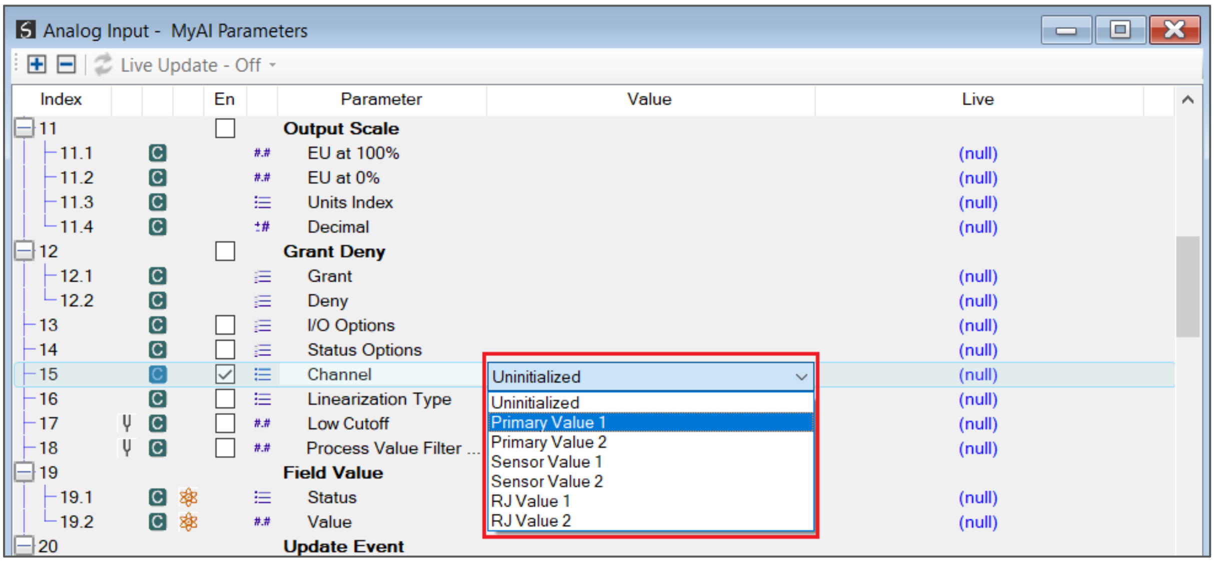


Figure 3.74 – Block Editor – Offline Parameter Selection

When a parameter containing sub-parameters is enabled, then all of the sub-parameters are automatically enabled.

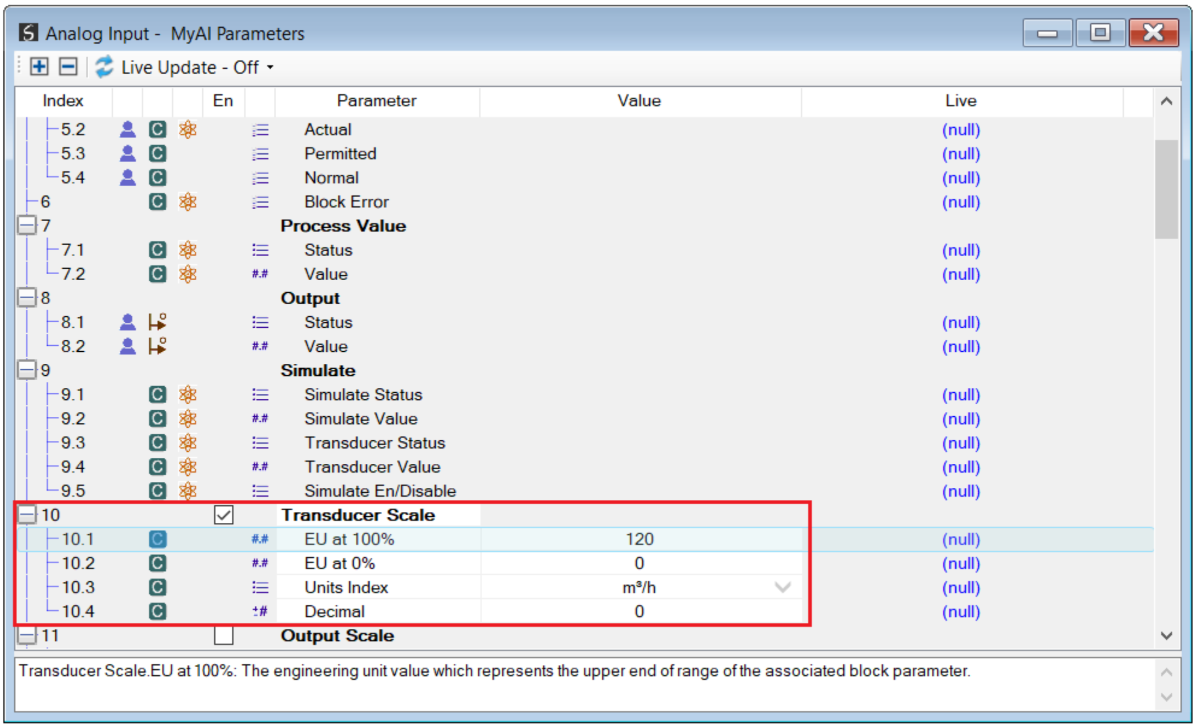


Figure 3.75 – Block Editor – Offline Parameter Group Selection

When downloading to a device, the enabled block parameters will be updated in the specific device during the download.

NOTE: See the *Download to H1 Device* section for details regarding the H1 device download.

ONLINE CHANGE

When changing the block parameters online, the user can either double click on the ***Live*** value, or right-click on the parameter and select the ***Write Live Value*** option.

The Block Parameter Write dialog will open. The user can then select the desired value, and then press the ***Write*** button to write the parameter to the H1 device.

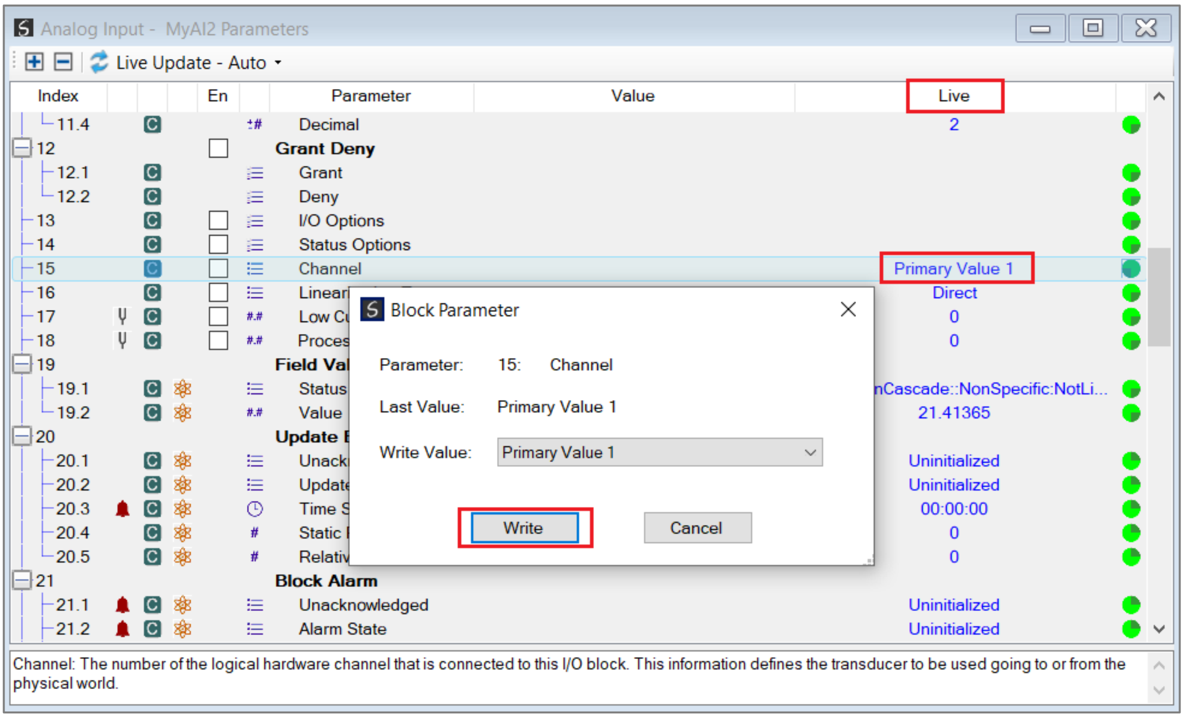


Figure 3.76 – Block Editor – Online Parameter change

NOTE: See the *Go Online with H1 Device* section for details regarding going online the H1 device.

NOTE: When changing certain parameters online, the user will need to ensure that the offline parameter has also been enabled with the correct value. Failing to do this could result in a replacement H1 device to not have the same parameter values as the existing device when downloading.

NOTE: When changing certain parameters online, certain blocks may need to be placed into the Out-of-Service (OoS) mode to avoid the parameter write fail. This can be done by either right-clicking on the block (when online with the Block Editor) and selecting *Change Mode -> Out of Service (OoS)* or changing the Target Parameter in the Parameter Editor to OOS (as shown below):

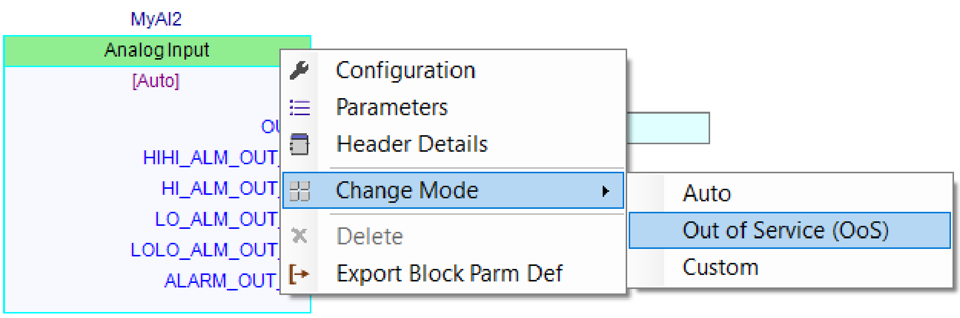


Figure 3.77 – Block Editor – Block Mode set to OOS from Block Editor

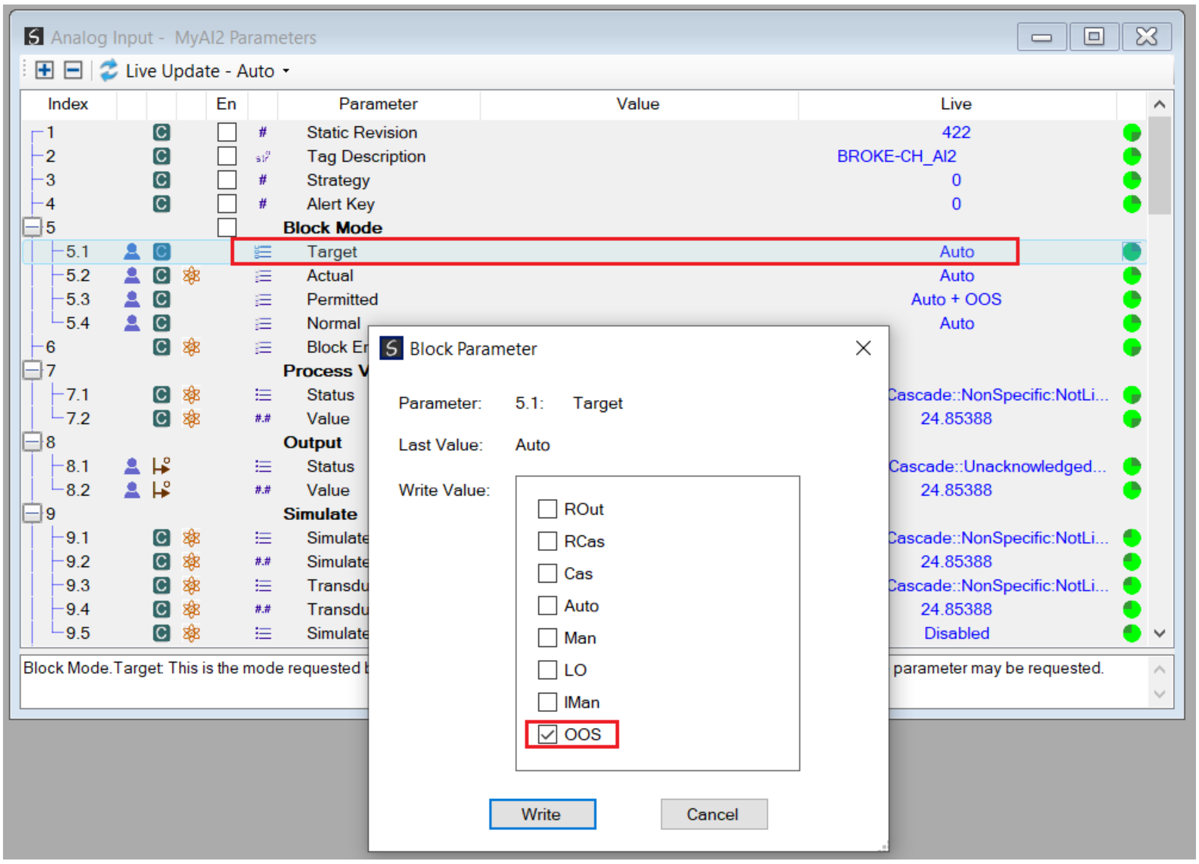


Figure 3.78 – Block Editor – Block Mode set to OOS from Parameter Editor

NOTE: When after the online parameter update is done, the block needs to be placed in the previously running Block Mode (e.g., Auto).

3.6.3.4.4. ADD CONNECTOR

Connectors are used to link inputs and outputs from blocks in an H1 device to the primary interface (e.g., EtherNet/IP) by providing H1 device sources and destinations in the internal mapping. To add a connector in the block editor, right-click on an open space of the block editor window and select ***Add Connector*** option.

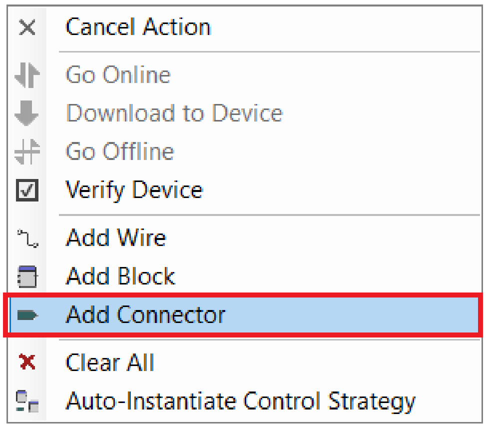


Figure 3.79 – Block Editor – Add connector

The Connector Configuration window will open.

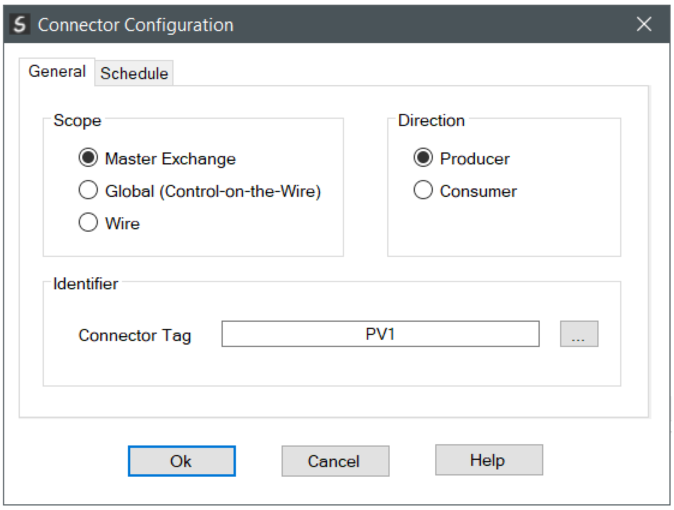


Figure 3.80 – Block Editor – Connector configuration - General

Below are the types of connectors that can be used.

Parameter	Description
	Scope
Master Exchange	The data from the H1 device function block will only be exchanged with the FF Link which is the H1 Master on the network.
Global (Control-on-the-Wire)	The data from the H1 device function block will be exchange with any H1 device on the network (including the FF Link).
Wire	A wire connector is used to connect two wires in the Block Editor window (if the user does not want to draw the wire across the Block Editor page). This is useful in certain applications where complicated block designs need to be used. **NOTE:** A wire connector does not interface to any variable in a function block, but rather to virtually link wires.
	Direction

Producer	When a connector is a producer, then the process variable data will be sent **from** the H1 device function block. Depending on the Scope selected, the data will either be sent directly to the FF Link (*Master Exchange)* or to other H1 devices (*Global*) on the network.
Consumer	When a connector is a consumer, then the process variable will be sent **to** the H1 device function block. Depending on the Scope selected, the data will either be sent from the FF Link (*Master Exchange*) or from another H1 device on the network (*Global*).
Identifier	
Connector Tag	The name given to the connector instance. In the case of Master Exchange and Global Producer, this name will be used for the Logix tag name when creating the L5X export. **Note**: Connector Tags must be unique within an H1 device. In addition to this, ***Master Exchange - Consumer*** and ***Global - Producer*** Connectors Tags must be unique within the entire FF Link configuration.

Table 3.12 – Block Editor – Connector configuration

The Schedule tab is only enabled when the configuration of the H1 device's ***Advanced (Manual) Scheduling*** option is set.

When ***Advanced (Manual) Scheduling*** is disabled, then the Producer Connector is scheduled (compelled) immediately after the execution of the block to which it is connected. Consumer Connectors are scheduled (compelled) as soon as possible at the start of the Macro Cycle.

NOTE: When ***Advanced (Manual) Scheduling*** is disabled, each connector is scheduled (compelled) only once per Macro Cycle.

When ***Advanced (Manual) Scheduling*** is enabled, then the user is required to configure a suitable Schedule Strategy using the following:

Parameter	**Description**
Initial Time	The time (in milliseconds) from the start of the Macro Cycle when the connector will first be compelled.
Repeat Count	The number of times (in addition to the initial compel) that the connector will repeat within the Macro Cycle. For example, if the Repeat Count is set to 2, then the connector will be compelled 3 times per Macro Cycle.
Repeat Time	The time (in milliseconds) between subsequent scheduling of the same connector.

Table 3.13 – Connector Timing - Advanced Scheduling Enabled

NOTE: It is important that the entered parameters do not cause the connector scheduling to execute beyond the Macro Cycle.

NOTE: The FF Link Master in its role as Link Active Schedular (LAS) will attempt to compel all the connectors based on the user timing configuration provided. However, where this timing conflicts with other H1 devices, the connector will be compelled as soon as the H1 bus is available immediately following the configured time.

NOTE: To ensure maximum communication efficiency (and reduce latency) Connectors should be scheduled in relation to the function blocks to which they are connected. A Producer Connector (connected to a function block output port) should be executed after that function block has completed its execution. (i.e., Start Time + Execution Time).
A Consumer Connector (connected to a function block input port) should be scheduled so as, to complete before the function block executes (typically 16ms before the function block executes.)

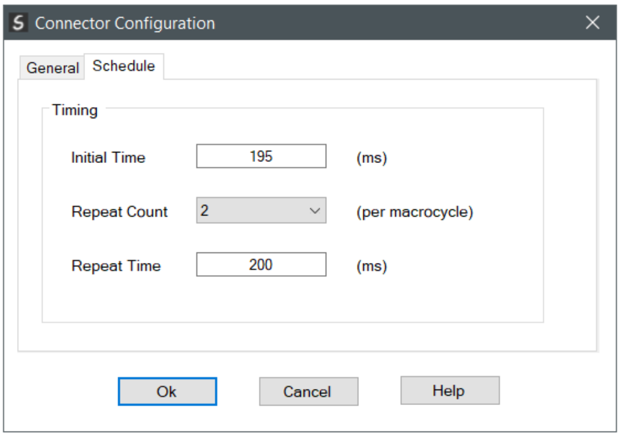


Figure 3.81 – Connector Configuration – Schedule (Manual - Advanced)

The timing diagram below shows the connector timing in relation to the block execution for the example configuration above.

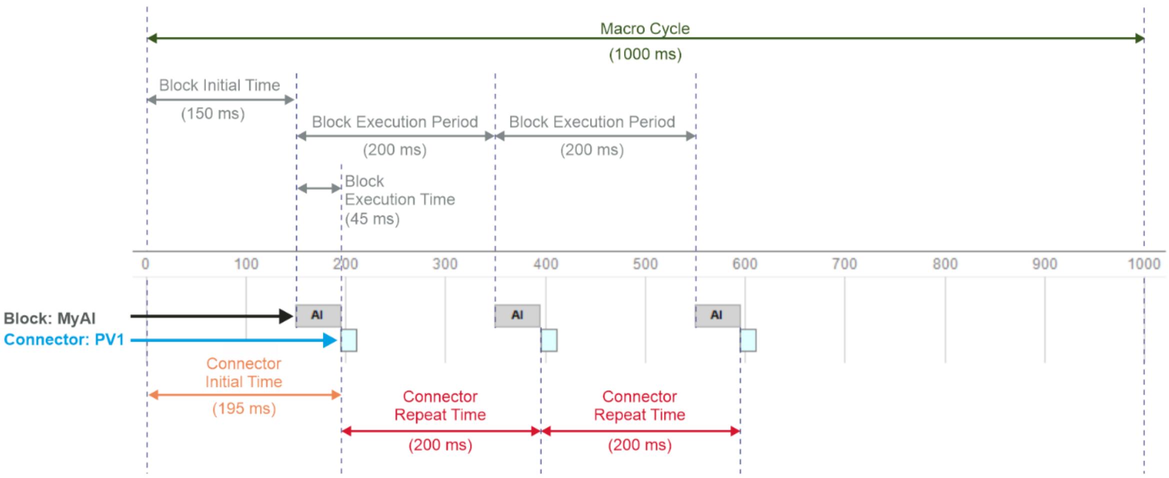


Figure 3.82 – Connector Schedule Timing (Manual - Advanced)

NOTE: In the example above, the connector's ***Initial Time*** is 45 ms greater than that of the function block's ***Initial Time*** to allow for the block's ***Execution Time***.

Once the configuration has been applied, the connector will be added to the Block Editor window.

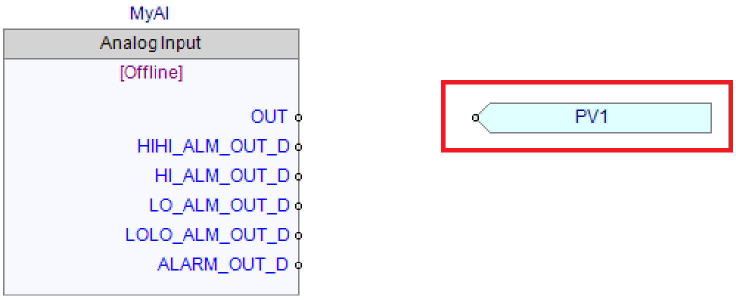


Figure 3.83 – Block Editor – Connector added

After the connector has been added to the Block Editor window, the configuration for the connector can be accessed by either double-clicking or, right-clicking on the connector in the Block Editor and selecting ***Configuration***.

Figure 3.84 – Block Editor – Connector configuration selected

The Connector symbol displayed depends on the type of connector configured.

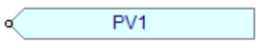
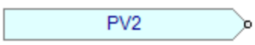
Symbol	Direction	Scope
	Producer	Master Exchange
	Consumer	
	Producer	Global (Control-on-the-Wire)
	Consumer	
	Producer	Wire
	Consumer	

Table 3.14 – Connector Symbols

3.6.3.4.5. ADD WIRE

In the Block Editor **wires** are used to connect between block ports and connectors. To add a wire, right-click on an open space in the Block Editor and select the ***Add Wire*** option.

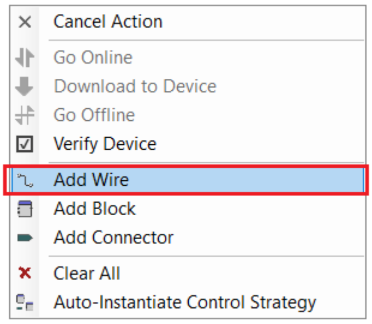


Figure 3.85 – Block Editor – Adding a wire

A wire will then be drawn once the user left-clicks on the Block Editor page and moves the mouse cursor. Finishing a segment of the wire is done by left-clicking on the Block Editor page. The wire drawing will still be active and if the mouse cursor is moved another segment of the wire will be drawn. To stop drawing a wire, the user can either left double-click on the last point of the wire, right-click and select the *Cancel Action option*. The wire drawing mode will automatically terminate if the wire terminates on the port of a block or connector.

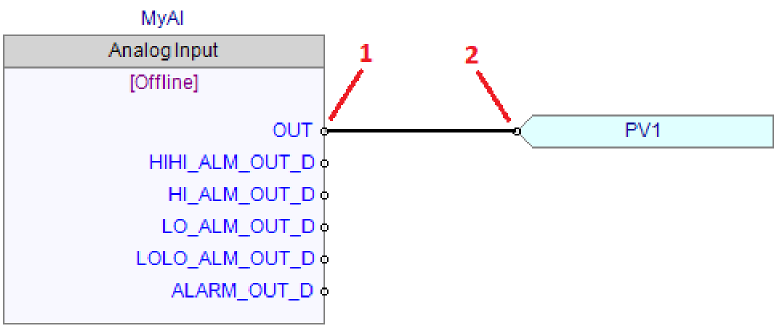


Figure 3.86 – Block Editor – Drawing a wire between a block and connector

A wire segment can be deleted by right-clicking on the wire and selecting the ***Delete Segment*** option, or by pressing the ***Delete*** key.

An entire Wire can be deleted by right-clicking on any wire segment and selecting the ***Delete Wire (All Segments)*** option.

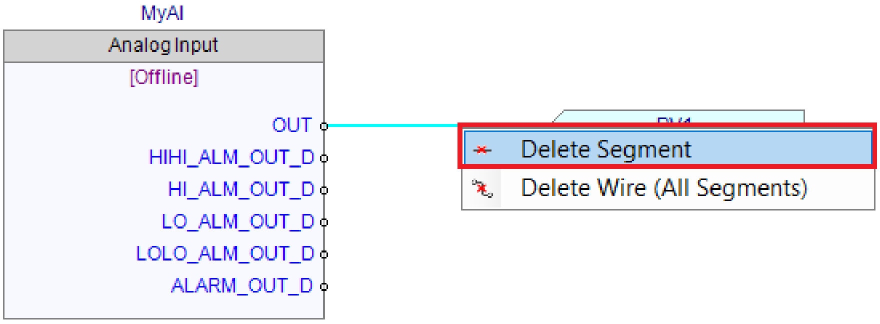


Figure 3.87 – Block Editor – Delete Segment

3.6.3.4.6. *DOWNLOAD TO H1 DEVICE*

Once the H1 device has been correctly setup, the control strategy (configuration) will need to be downloaded to the H1 device. The H1 device download can take up to a minute to complete depending on the block configuration complexity and number of H1 devices on the H1 network. Once complete, the H1 device will store the block configuration in Non-volatile (NV) memory. Thus, when the H1 device is power cycled it will still have the updated block configuration.

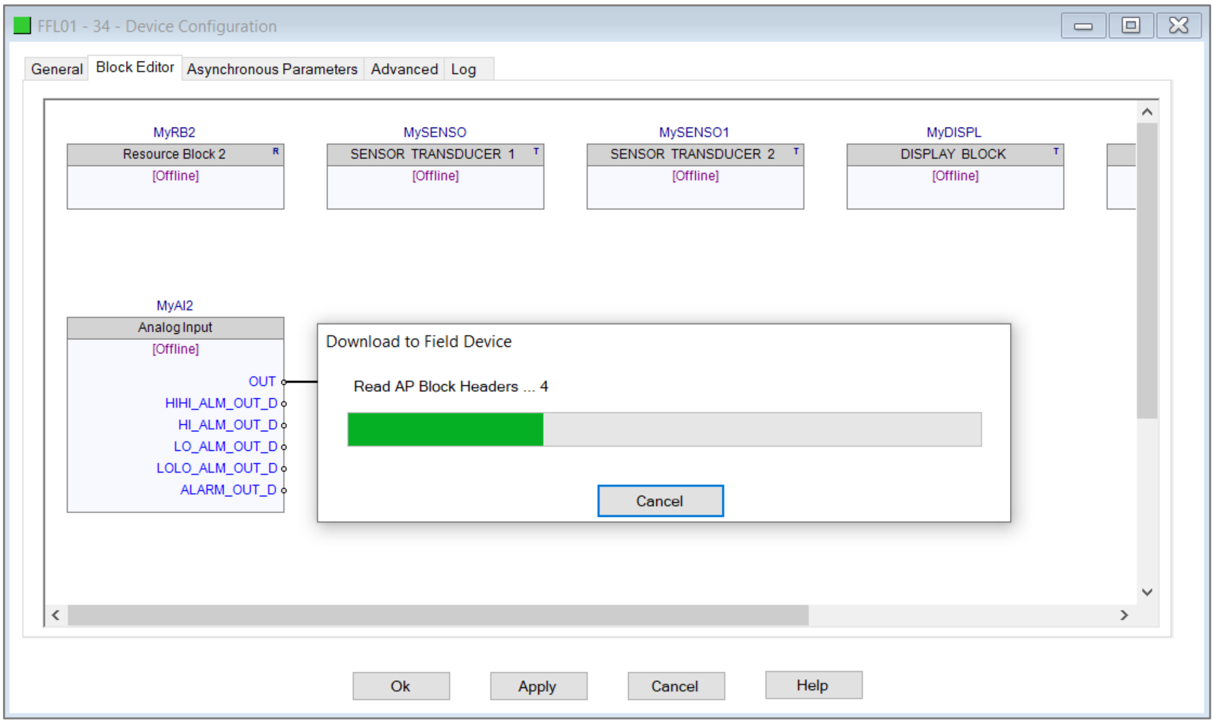


Figure 3.88 – Block Editor – Download

NOTE: The user will be requested to Apply the FF Link configuration once the block configuration has been downloaded to the H1 device and the FF Link configuration has changed. This will ensure that the updated H1 device block configuration is applied to the FF Link device configuration.

3.6.3.4.7. GO ONLINE WITH H1 DEVICE

The user can also go online with the H1 device after the download is complete. This will allow the user to view or edit live block parameters. To go online with an H1 device, right-click on a empty space in the Block Editor and select ***Go Online***.

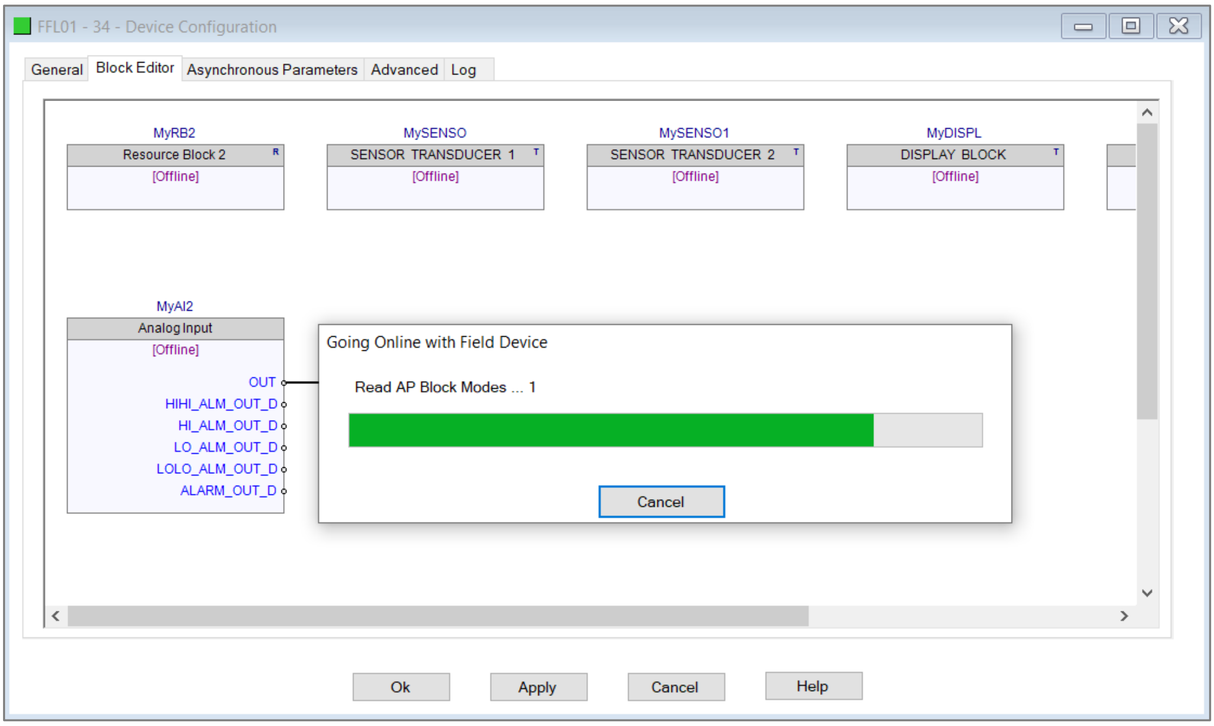


Figure 3.89 – Block Editor – Go Online

Once online the user can see the output values being produced by the blocks.

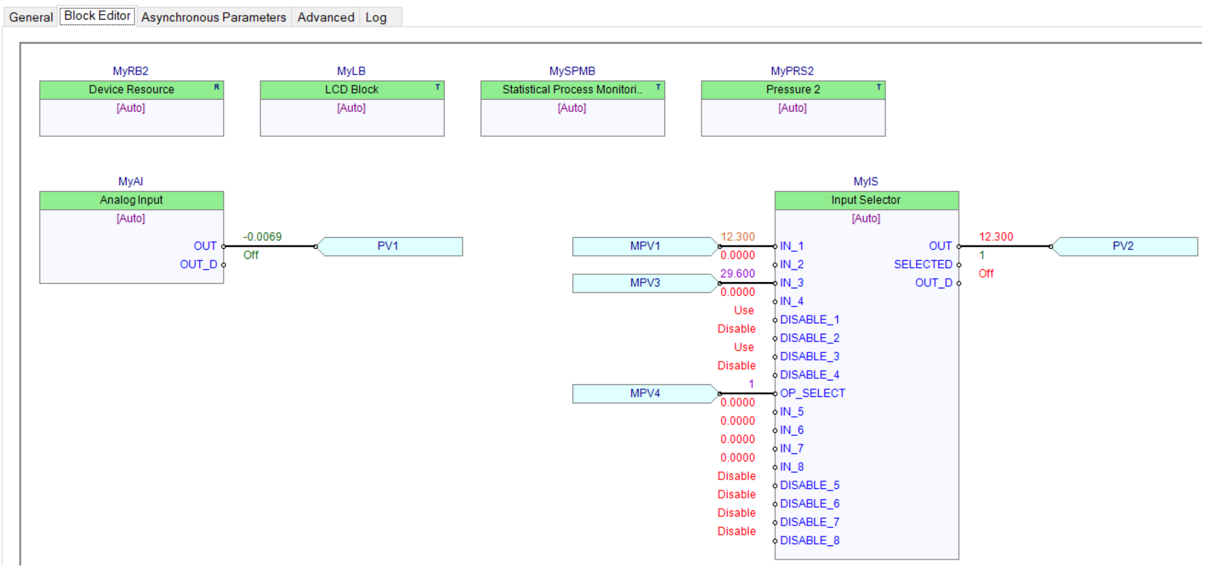


Figure 3.90 – Block Editor – Online Block View

The status of each process variable is indicated by its color as follows:

PV Color	PV Status
Green	Good
Purple	Good Cascade
Orange	Uncertain
Red	Bad

Table 3.15 – PV Colors

The user can also view the online values of the parameters in each block by opening the parameter view for the block.

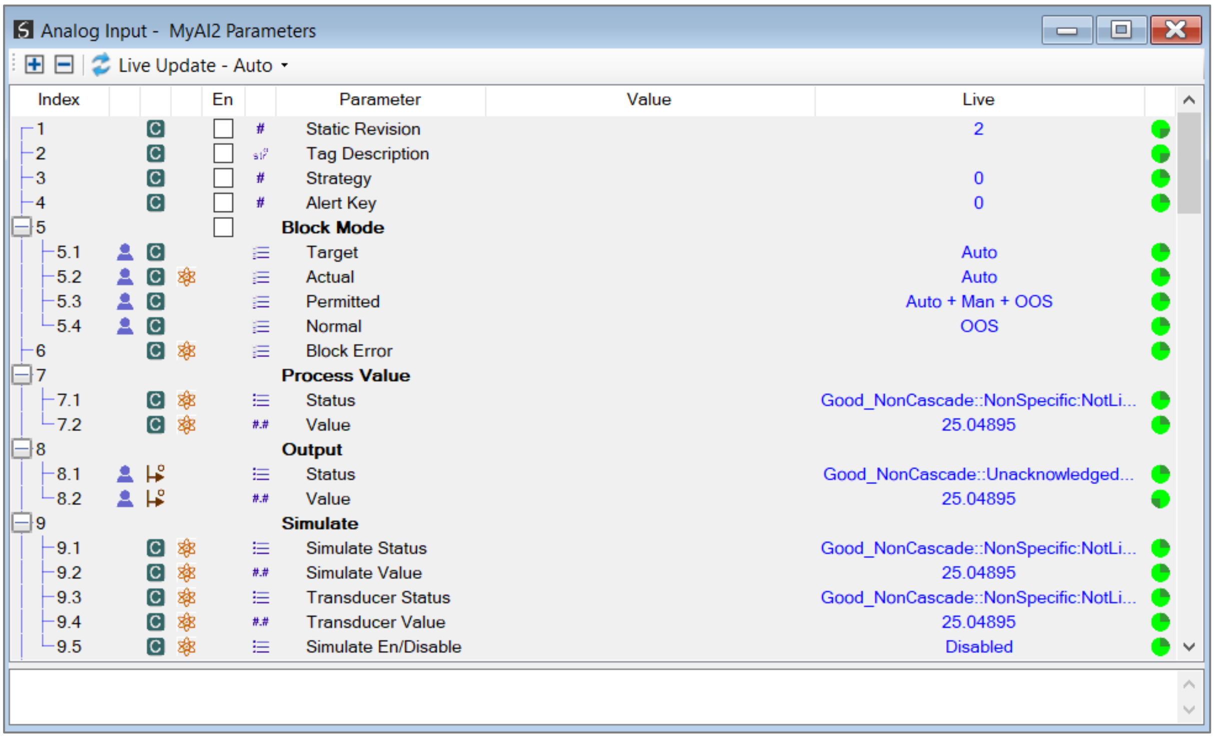


Figure 3.91 – Block Editor – Online Parameter View

3.6.3.5. ASYNCHRONOUS PARAMETERS

Asynchronous Parameter can be used to read or write data from the Primary Interface to specific block parameters which are not accessible through the normal connector and wire method. Within each Macro Cycle a configurable number of asynchronous parameters will be executed (see the *Asynchronous Parameter Rate* in the general H1 configuration).

The Asynchronous Parameters configuration is shown in the figure below.

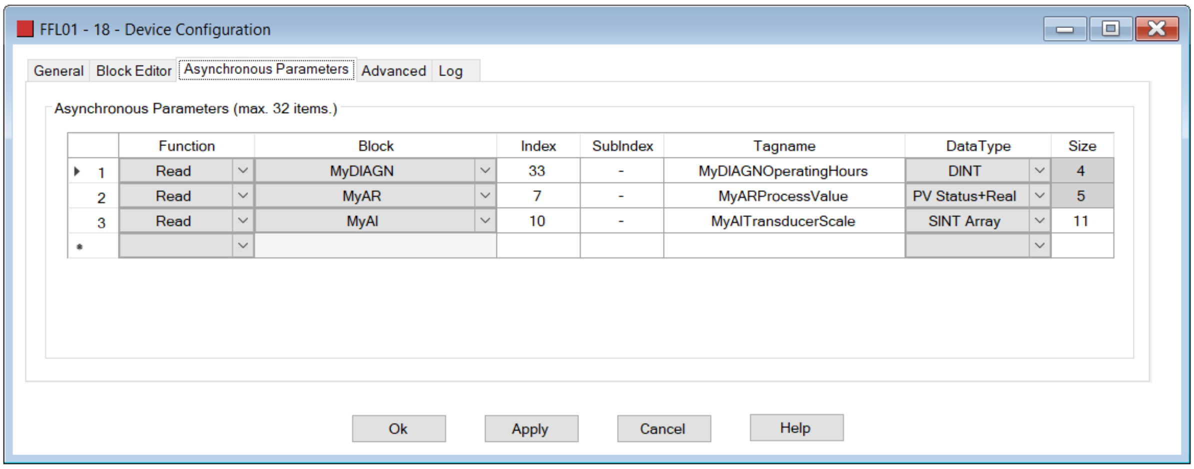


Figure 3.92 – Device Asynchronous Parameters configuration

The Device Asynchronous Parameter configuration consists of the following attributes:

Parameter	Description
Function	The FF Link can either **read** or **write** an asynchronous parameter to, or from, a device.
Block	The instance name of the block which has the parameter to be used in the asynchronous parameter read or write.
Index	The parameter index of the selected block to be used in the asynchronous parameter read or write. **NOTE:** See the section below on how to browse for a specific parameter.
SubIndex	The parameter subindex of the selected block to be used in the asynchronous parameter read or write. Note: Select 0, for a parameter with no sub-parameters.
Tagname	The name given to the asynchronous parameter which will be used in the internal mapping as well as the Logix tagname in the L5X creation.
DataType	The data type of the parameter.
Size	The size of the parameter in bytes. This field is read only unless the selected ***Data Type*** is ***SINT Array***.

Table 3.16 – Device Asynchronous Parameters configuration attributes

3.6.3.5.1. PARAMETER BROWSER

The asynchronous parameter selection index, subindex, and datatype can be either manually entered or the user can browse to the required parameter. To open the block parameter browse view, right-click on the row, in the Asynchronous Parameter mapping grid and select the ***Browse*** option.

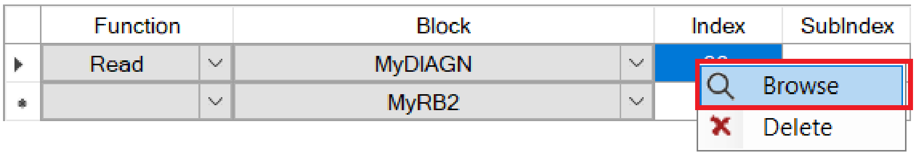


Figure 3.93 – Device Asynchronous Parameter Browse selection

The Block Parameter Select dialog will open, displaying all the parameters for the selected block.

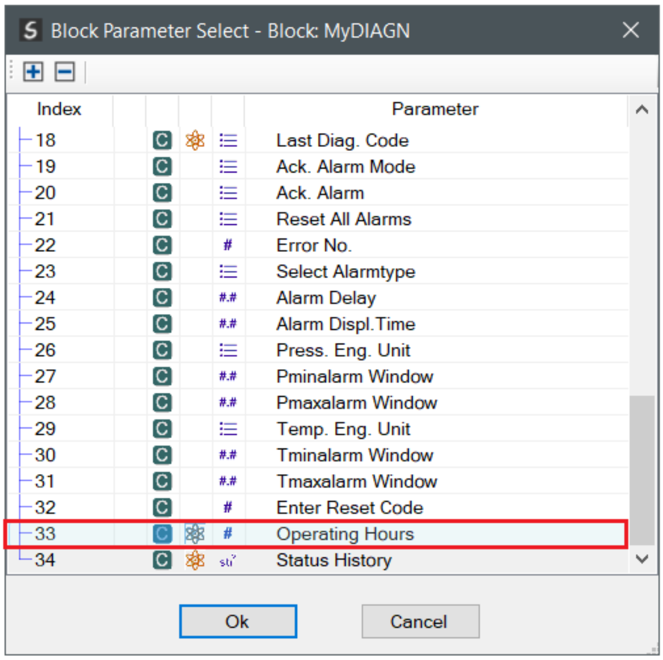


Figure 3.94 – Device Asynchronous Parameters browsing

3.6.3.6. ADVANCED

The Advanced configuration is shown in the figure below.

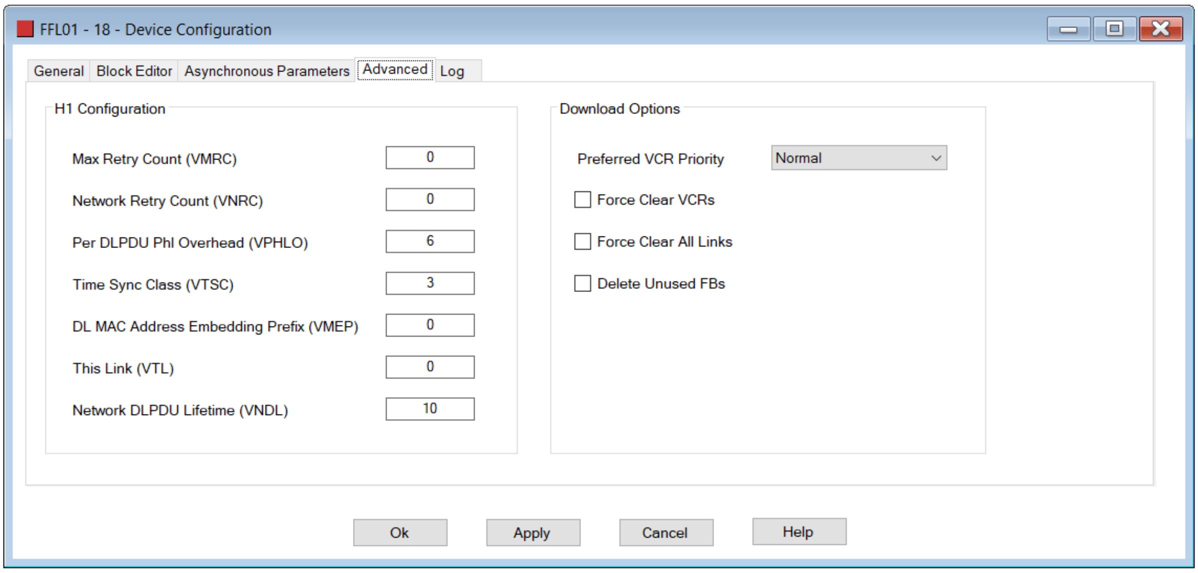


Figure 3.95 – H1 device Advanced Configuration

3.6.3.6.1. ADVANCED H1 CONFIGURATION

NOTE: It is advised that the user do not alter these setting as it could result in unpredictable behaviour.

3.6.3.6.2. H1 DOWNLOAD OPTIONS

The following H1 Download Options consist of the following attributes:

Parameter	**Description**
Preferred VCR Priority	The Priority to be configured for all H1 device publish VCRs. The Options are: • Normal (default) • Urgent • Time Available
Force Clear VCRs	When selected, this option forces the software to attempt to clear all the configured non-system VCRs, before the download configuration is attempted.
Force Clear All Links	When selected, this option forces the software to attempt to clear all the configured function block links, before the download configuration is attempted. When unchecked, the software clears links until it finds 3 consecutive unconfigured links.
Delete Unused FBs	When selected, this option forces the software to attempt to delete all unused function blocks. This is typically not required, because unused function blocks are not scheduled and thus should have no impact on the control strategy.

Table 3.17 – H1 Download Options

3.6.4. H1 Device Import and Export Functions

A specific H1 device configuration can be replicated in other FF Link projects through the use of the H1 device Import and Export functions.

3.6.4.1. EXPORT H1 DEVICE

To export an H1 device, right-click on the specific H1 device and select the ***Export H1 Device*** option.

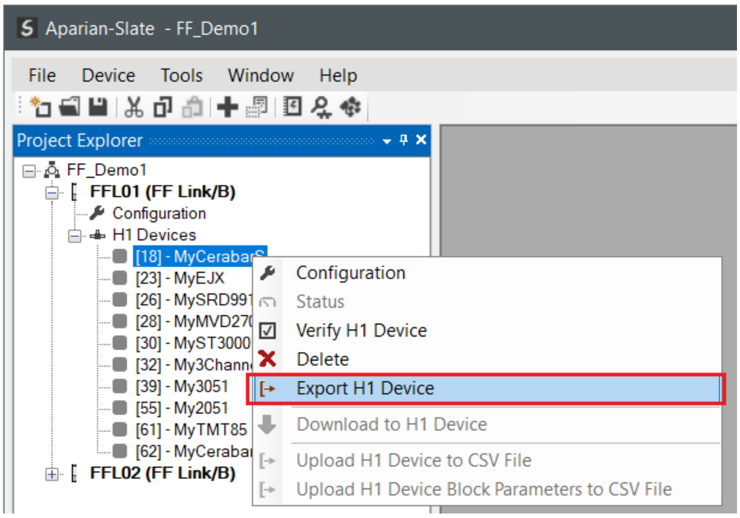


Figure 3.96 – H1 device Export

The user will then be prompted to select a suitable folder and filename for the exported (.s1hd) file.

3.6.4.1. IMPORT H1 DEVICE

To import a previously exported H1 device instance, right-click on the ***H1 Devices*** tree item and select the ***Import H1 Device*** option.

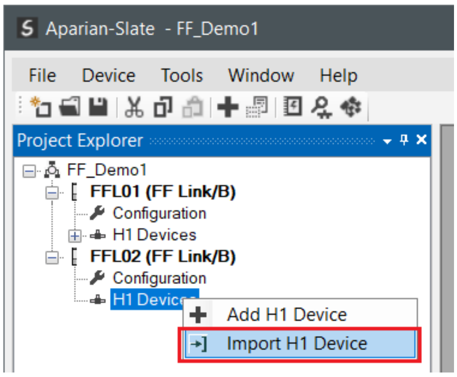


Figure 3.97 – H1 device Import

3.7. ETHERNET/IP TARGET

3.7.1. LOGIX CYCLIC CLASS 1

The FF Link module can be easily integrated with the Allen-Bradley Logix family of controllers. Integration with the Logix family in Studio5000 makes use of either the EDS Add-On-Profile (AOP), or Generic Module Profile, or in the case of ControlNet via a ControlNet Router, the ControlNet Generic Profile.

3.7.1.1. EDS AOP (LOGIX V21+)

NOTE: When using the EDS AOP, ensure that the FF Link's ***Logix Profile*** has been set to the ***Standard (EDS AOP)*** option, otherwise the generated L5X Logix code will import with errors.

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.

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.

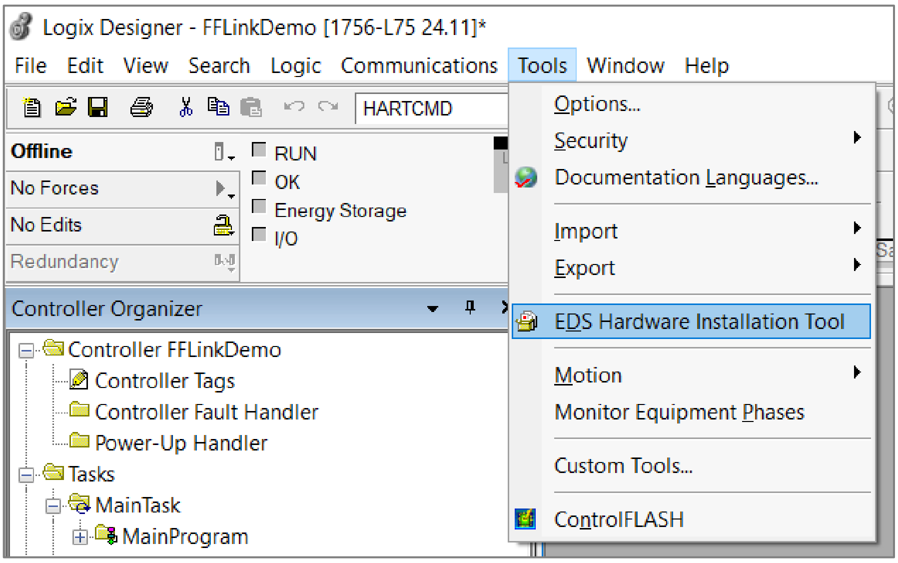


Figure 3.98 - EDS Hardware Installation Utility

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.

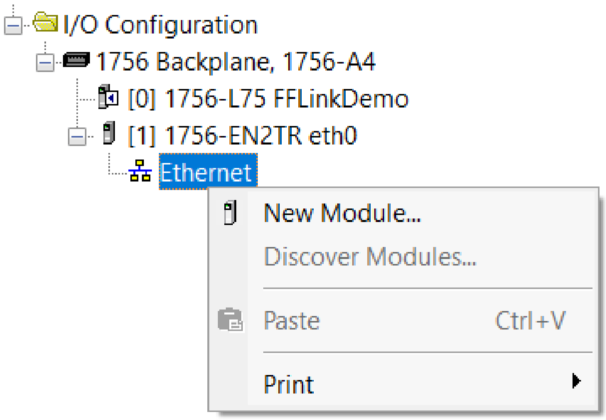


Figure 3.99 – Adding a module

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.

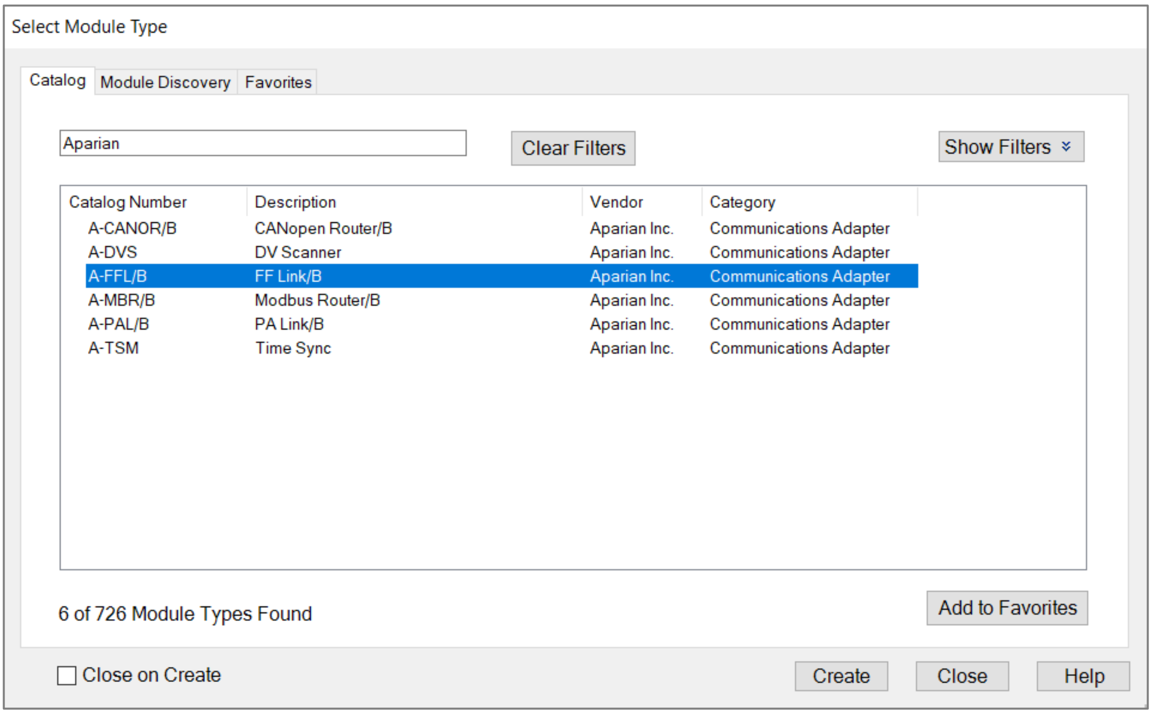


Figure 3.100 – Selecting the module

Locate and select the FF Link 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.

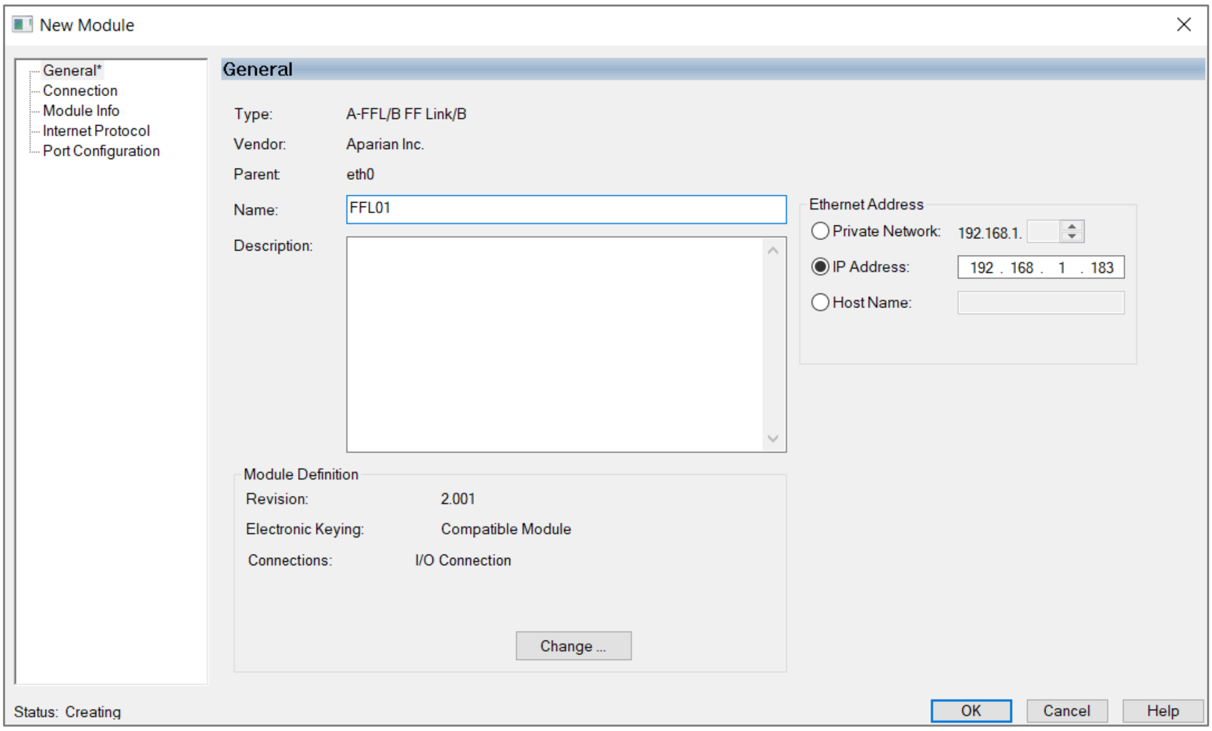


Figure 3.101 – Module instantiation

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

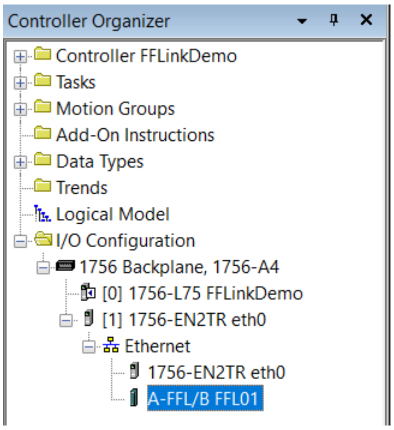


Figure 3.102 – Logix IO tree

Name	Value	For	Style	Data Type
⊟ FFL01:I1	{...}	{...		_055A:A_FFLB_7E6CF713:I:0
FFL01:I1.ConnectionFaulted	0		Decimal	BOOL
⊞ FFL01:I1.Data	{...}	{...	Decimal	SINT[500]
⊞ FFL01:I2	{...}	{...		_055A:A_FFLB_7E6CF713:I:0
⊞ FFL01:I3	{...}	{...		_055A:A_FFLB_7E6CF713:I:0
⊞ FFL01:I4	{...}	{...		_055A:A_FFLB_7E6CF713:I:0
⊞ FFL01:O1	{...}	{...		_055A:A_FFLB_78F5E13D:O:0
⊞ FFL01:O2	{...}	{...		_055A:A_FFLB_78F5E13D:O:0
⊞ FFL01:O3	{...}	{...		_055A:A_FFLB_78F5E13D:O:0
⊞ FFL01:O4	{...}	{...		_055A:A_FFLB_78F5E13D:O:0

Figure 3.103 – Module Defined Data Type

3.7.1.2. GENERIC MODULE PROFILE (LOGIX PRE-V21)

NOTE: When using a Generic Module Profile only a **single** connection is supported limiting the total H1 data that can be exchanged.

NOTE: When using the Generic Profile, ensure that the FF Link's ***Logix Profile*** has been set to the ***Generic Profile*** option, otherwise the generated L5X Logix code will import with errors.

When using Logix versions prior to version 21, then the FF Link module must be added to the Logix I/O tree as a generic Ethernet module. This is achieved by right clicking on the Ethernet Bridge in the Logix and selecting *New Module* after which the *ETHERNET-MODULE* is selected to be added as shown in the figure below.

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

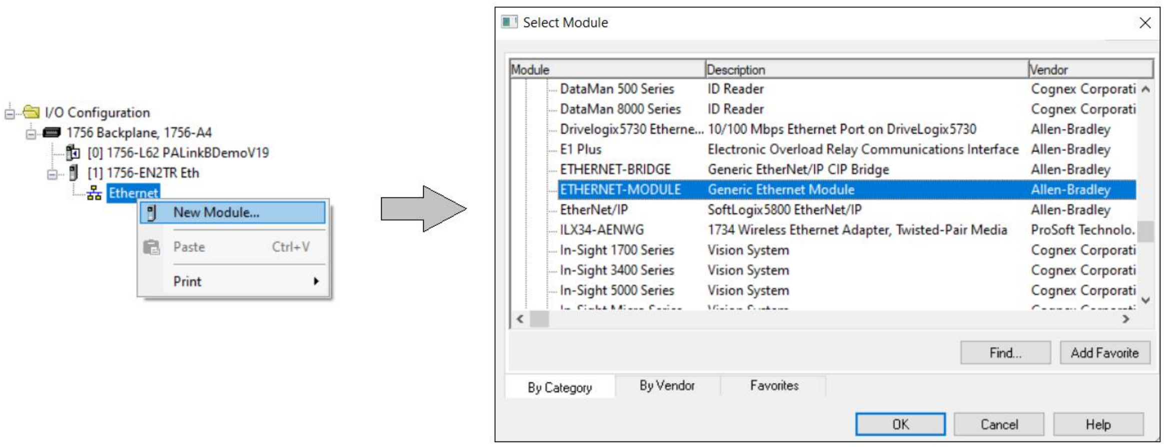


Figure 3.104 - Add a Generic Ethernet Module in Logix

The user must enter the ***IP Address*** of the FF Link module that will be used. The assembly instance and size must also be added for the input, output, and configuration in the connection parameters section.

The required connection parameters for the FF Link module are shown below:

Connection Parameter	Assembly Instance	Size
Input	140	500 (8-bit)
Output	141	496 (8-bit)
Configuration	102	0 (8-bit)

Table 3.18 - Logix class 1 connection parameters for the FF Link module

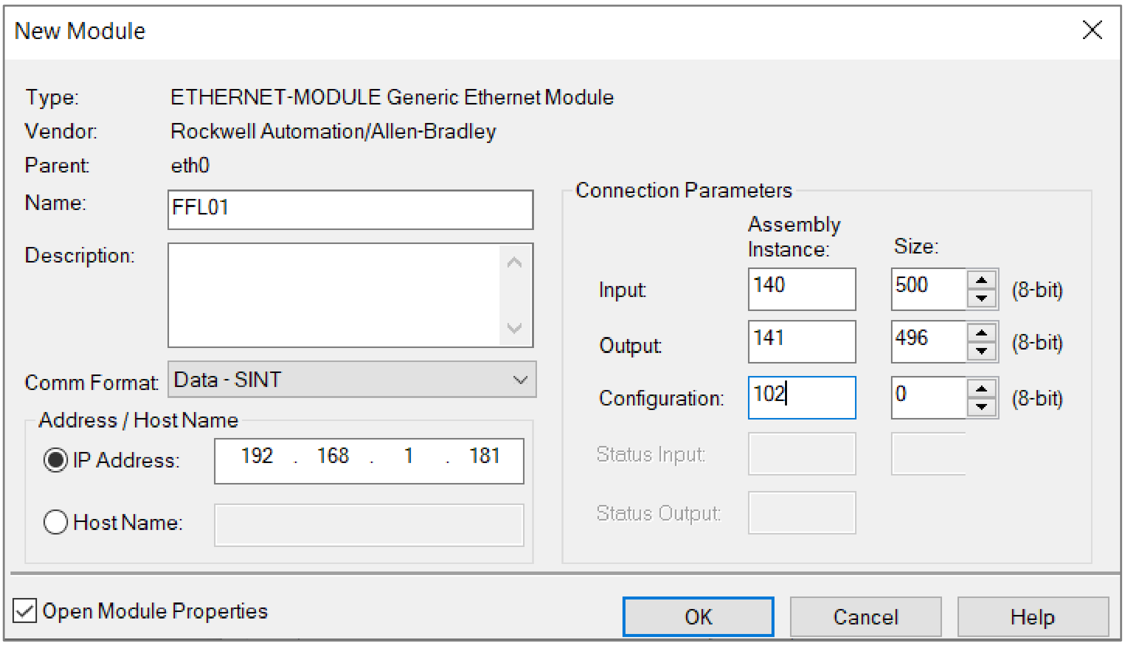


Figure 3.105 - Logix General module properties for FF Link module

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. Refer to the technical specification section in this document for further details on the limits of the RPI.

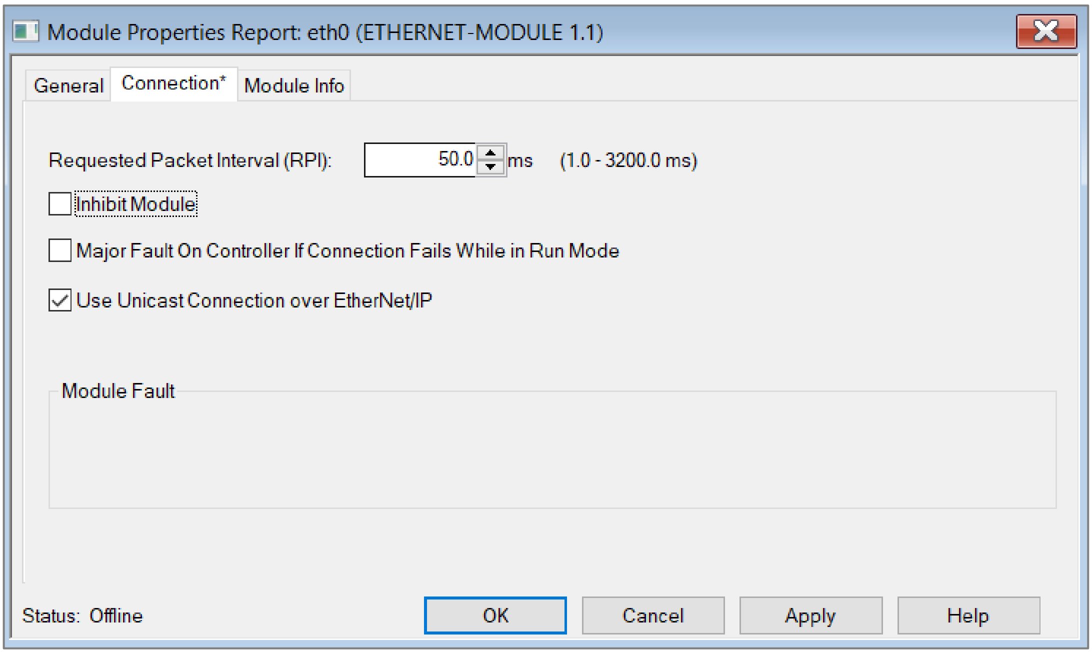


Figure 3.106 - Connection module properties in Logix

Once the module has been added to the Logix I/O tree the Logix controller will be ready to connect to the FF Link with a Class 1 connection.

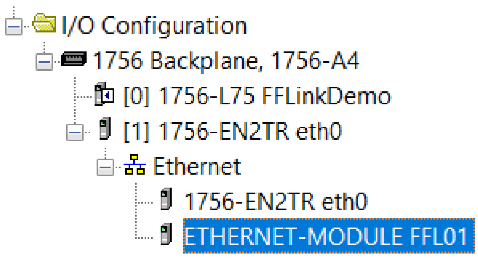


Figure 3.107 – Logix I/O module tree

3.7.1.3. CONTROLNET ROUTER PROFILE

The FF Link can be added to an existing ControlNet network by making use of the Aparian ControlNet Router.

NOTE: When using the ControlNet Router option only a **<u>single</u>** connection is supported limiting the total H1 data that can be exchanged.

NOTE: When using the ControlNet Router, ensure that the FF Link's ***Logix Profile*** has been set to the ***ControlNet Router (A-CNR)*** option, otherwise the generated L5X Logix code will import with errors.

The ControlNet Router (A-CNR) must be added to the Logix IO tree under a suitable ControlNet bridge (e.g. 1756-ACN / 1756-ACNR). The details of adding the module are described in the ***ControlNet Router Enhanced User Manual***.

NOTE: It is important that the name of the ControlNet Router in the Logix IO tree matches the name given to the FF Link in the Slate configuration. Failure to do so will result in errors when importing the generated Logix L5x file.

The FF Link requires the ControlNet Router to be configured in a specific way, including the EtherNet/IP Originator connection to the FF Link.

It is strongly recommended to make use of the ControlNet Router configuration:
ControlNetRouter - FFLink Bridge Example.spx
This file can be can be downloaded from the Aparian website and imported into an existing (FF Link) Slate project.

For more details, please download the following document:
D122-018 FF Link - Technical Application Note - Connection via ControlNet Router.pdf

3.7.1.4. MULTI-CONNECTION

The FF Link supports up to four Class 1 (cyclic data exchange) connections. This will allow the user to have more field devices per FF Link because more data can be exchanged between the Logix controller and the FF Link.

NOTE: This only applies when the user has implemented the FF Link into Logix using an EDS AOP. When using a Generic Module Profile in Logix (pre-Logix v21) the user will only be able to use 1 Logix Connection.

When the user verifies the Slate project (achieved by right-clicking on the device and selecting ***Verify Configuration***), the software will indicate if the current configuration will fit into the selected ***EtherNet/IP Connection*** count. If not, the user will need to increase the connection count.

In Slate the user can set the number of ***EtherNet/IP Connections*** in the Logix tab of the configuration window (as shown below):

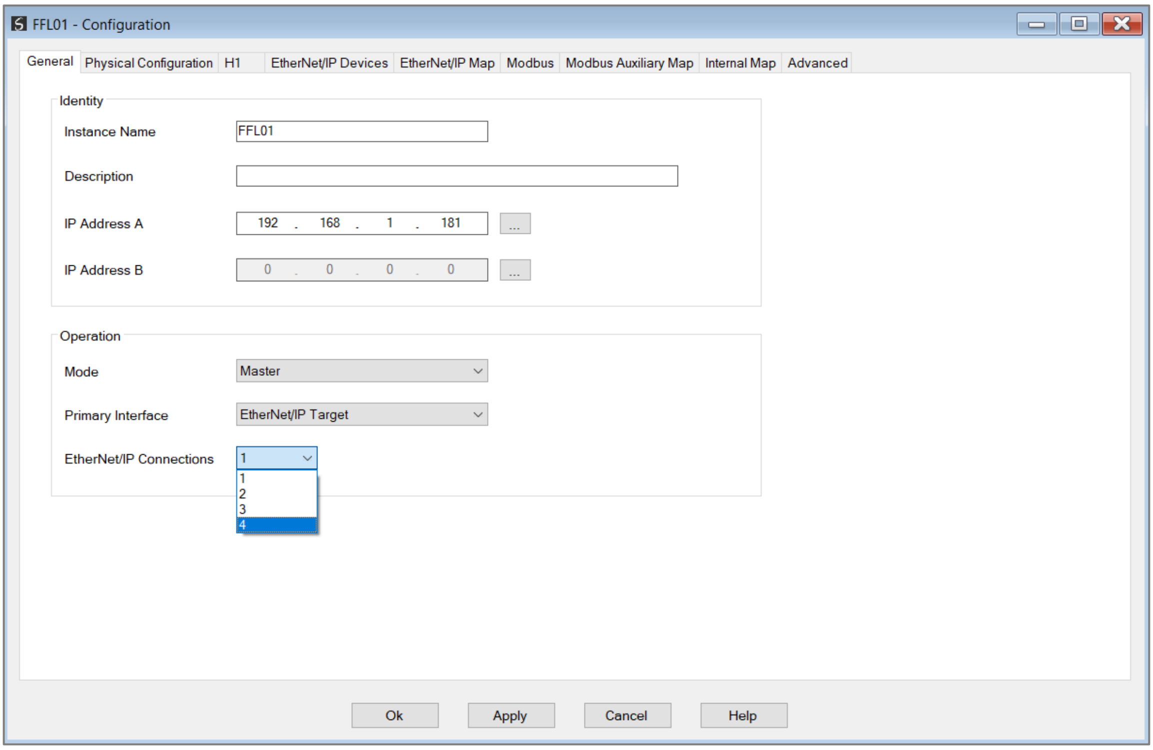


Figure 3.108 – Slate EtherNet/IP Connection Count

In Logix the user can increase/decrease the connection count using the EDS AOP (as shown below):

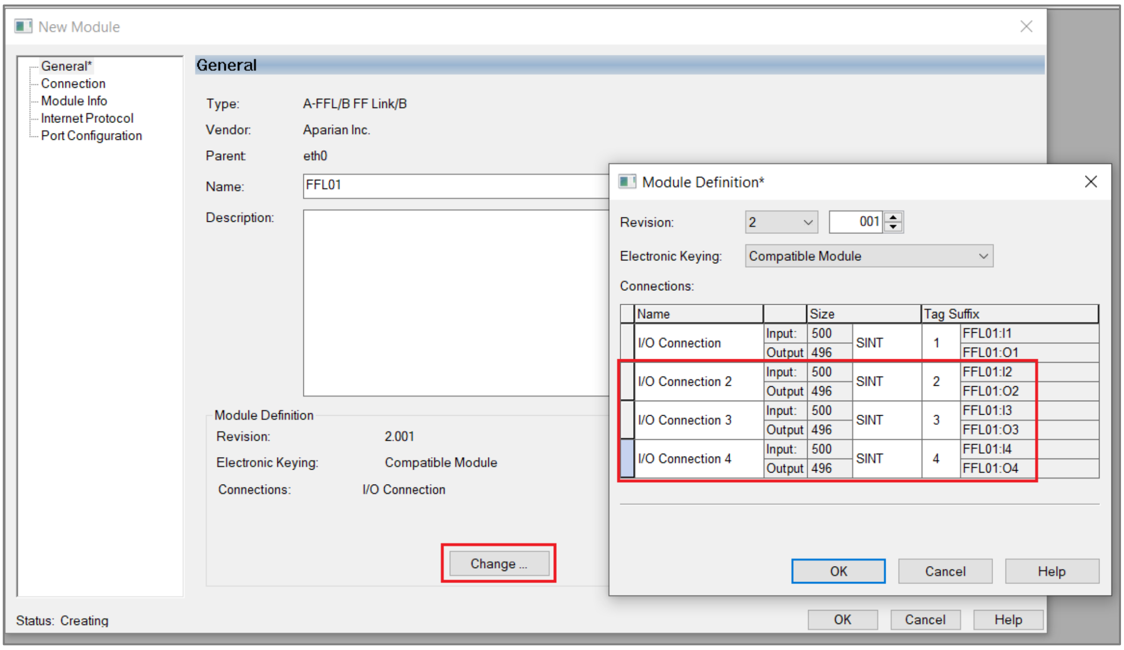


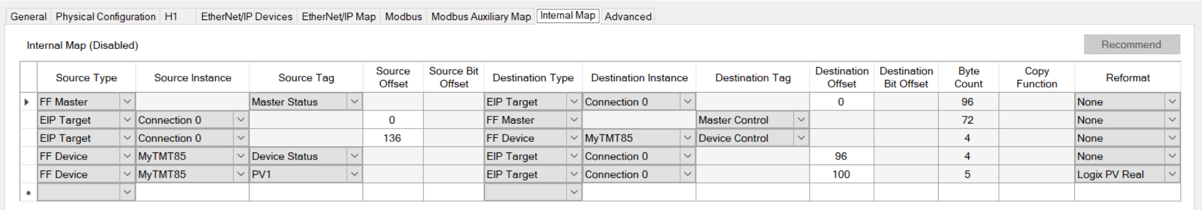
Figure 3.109 – Logix EtherNet/IP Connection Count

NOTE: The connection count selected in Logix must match the configured *EtherNet/IP Connections* count in Slate.

3.7.1.5. LOGIX MAPPING

Slate will generate the required UDTs and Routines (based on the Internal Map) to map the required H1 Device input and output data.

NOTE: When *Auto Target Mapping* is selected in the H1 configuration, then the Internal Map will automatically be assembled as the configuration is updated.



Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			EIP Target	Connection 0		0		96		None
EIP Target	Connection 0		0		FF Master		Master Control			72		None
EIP Target	Connection 0		136		FF Device	MyTMT85	Device Control			4		None
FF Device	MyTMT85	Device Status			EIP Target	Connection 0		96		4		None
FF Device	MyTMT85	PV1			EIP Target	Connection 0		100		5		Logix PV Real

Figure 3.110 – Internal Mapping

The user will need to generate the required Logix and UDTs by right-clicking on the module in Slate and selecting the ***Generate Logix L5X*** option.

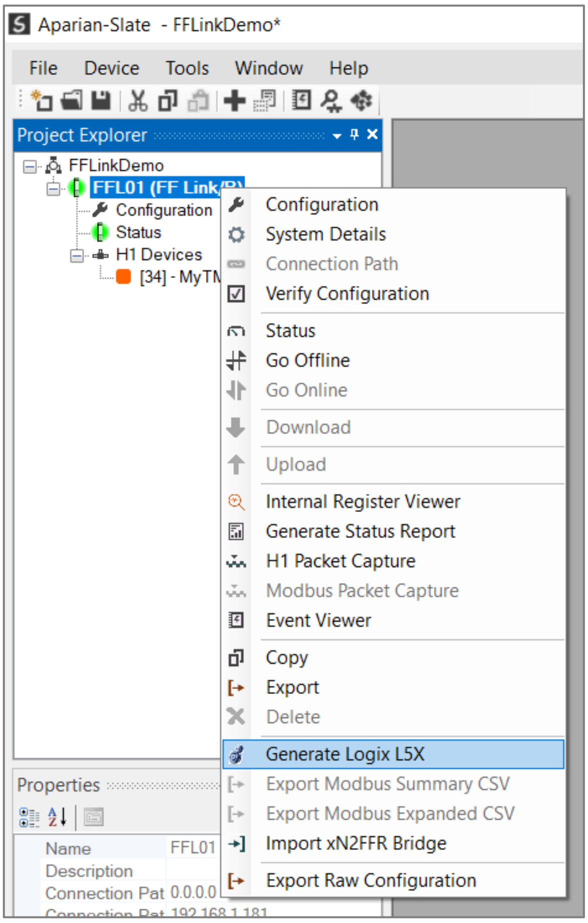


Figure 3.111 – Selecting Generate Logix L5X

The user will then be prompted to select a suitable file name and path for the L5X file.

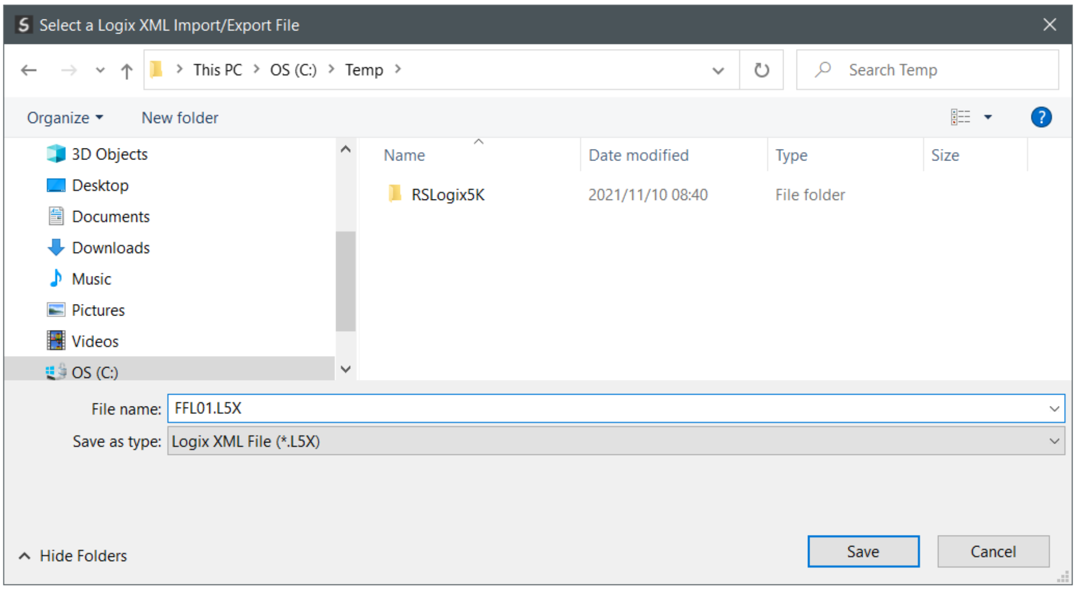


Figure 3.112 – Selecting the Logix L5X file name

This L5X file can now be imported into the Studio 5000 project by right-clicking on a suitable ***Program*** and selecting ***Add***, and then ***Import Routine***.

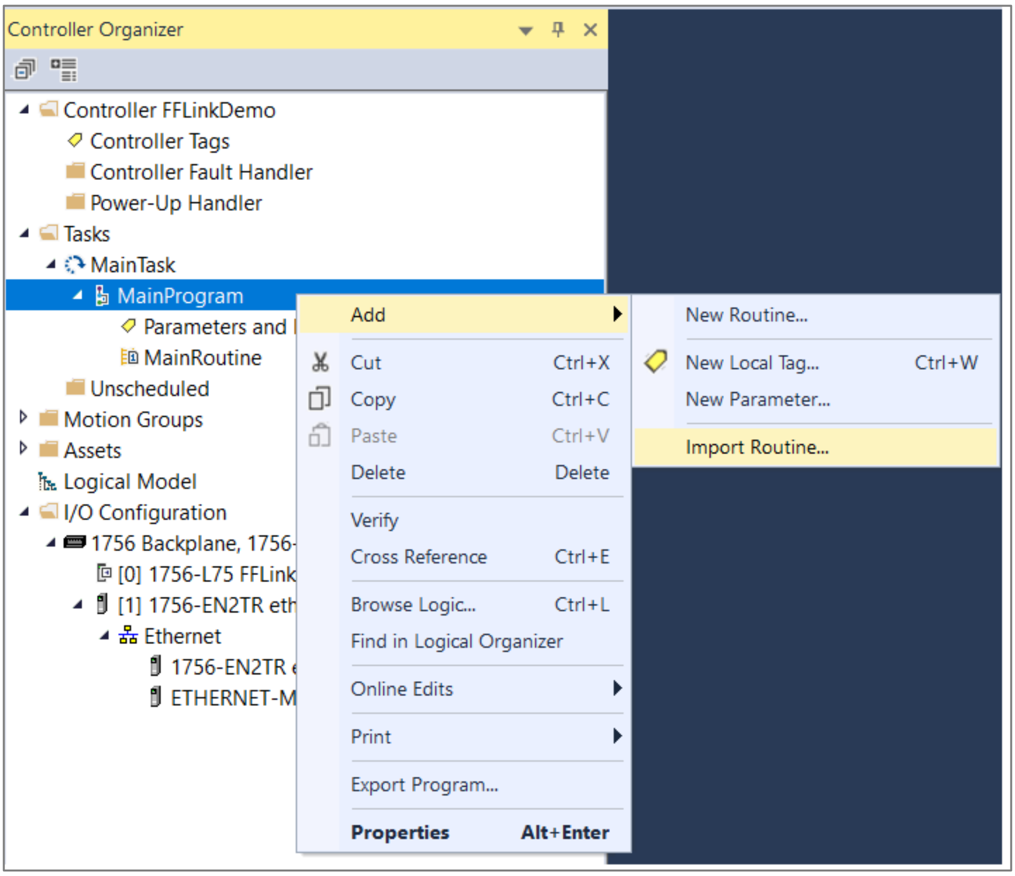


Figure 3.113 – Importing the L5X file into Studio 5000

In the file open dialog select the previously created L5X file and accept the import by pressing ***Ok***.

The import will create the following:

- Mapping Routine
- Multiple UDT (User-Defined Data Types)
- Multiple Controller Tags

Since the imported mapping routine is not a Main Routine, it will need to be called from the current Main Routine.

Figure 3.114 – Calling the mapping routine

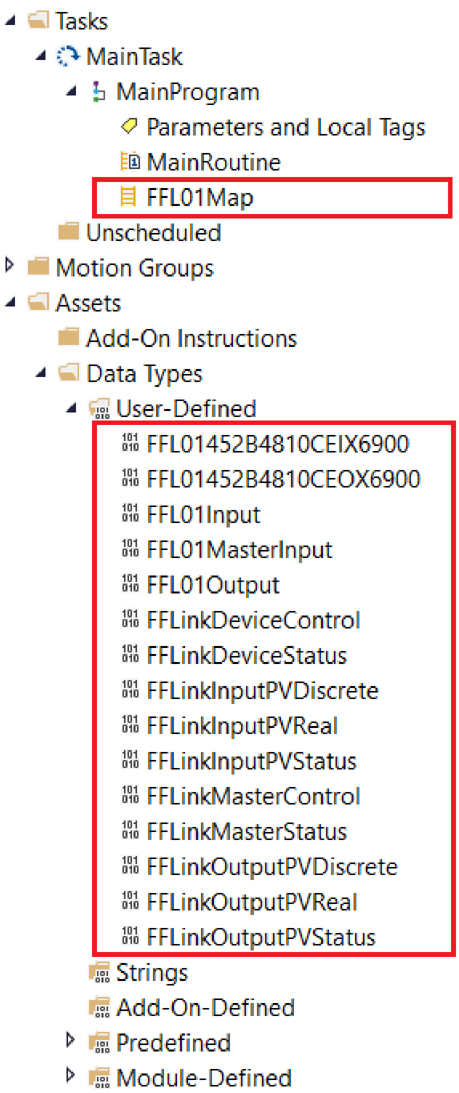


Figure 3.115 – Imported Logix Objects

A number of FF Link specific (UDT) tags are created. The Master Control tag is used H1 Master Redundancy Control as well as FF Link PV mapping verification. These parameters do not need to be modified as the L5X code generated will change these as required.

FFL01Out	{...}	{...}		FFL01Output	
FFL01Out.MasterControl	{...}	{...}		FFLinkMaster...	
FFL01Out.MasterControl.RedundancyControl	0		Decimal	SINT	Master Redundancy Command
FFL01Out.MasterControl.OtherMasterConfigCRC	0		Decimal	INT	Redundant Master Configuration ...
FFL01Out.MasterControl.PVVCR	{...}	{...}	Hex	SINT[64]	Master PV VCR

Figure 3.116 – Master Control tag

The Master Status tag displays the status of the H1 Master, including arrays to show the LiveList and Data Exchange Active status of each H1 device.

Name	Value	Force	Style	Data Type	Description
FFL01In	{...}	{...}		FFL01Input	
FFL01In.ConnectionOk	1		Decimal	BOOL	Connection Ok (0=Fault, 1=Ok)
FFL01In.Master	{...}	{...}		FFL01MasterI...	
FFL01In.Master.Status	{...}	{...}		FFLinkMaster...	
FFL01In.Master.Status.ConfigValid	1		Decimal	BOOL	Configuration Valid
FFL01In.Master.Status.DuplicateNode	0		Decimal	BOOL	Duplicate Node Detected
FFL01In.Master.Status.ActiveLAS	1		Decimal	BOOL	Active LAS
FFL01In.Master.Status.EIPOwned	1		Decimal	BOOL	Class 1 Ownership: 0=Not Owned,...
FFL01In.Master.Status.EIPOriginatorCommsOk	0		Decimal	BOOL	
FFL01In.Master.Status.MBCommsOK	0		Decimal	BOOL	
FFL01In.Master.Status.PLCModeRun	1		Decimal	BOOL	Bit 0-Prog 1-RUN
FFL01In.Master.Status.RedundancyEnabled	0		Decimal	BOOL	Redundancy Config: 0=Standalon...
FFL01In.Master.Status.RedundancyStatus	0		Decimal	BOOL	Redundancy Status: 0=Standby, 1...
FFL01In.Master.Status.PowerConditionerActive	1		Decimal	BOOL	Internal Power Conditioner Active
FFL01In.Master.Status.TerminatorActive	1		Decimal	BOOL	Internal Terminator Active
FFL01In.Master.Status.PortTripped	0		Decimal	BOOL	FF Bus Port Tripped
FFL01In.Master.Status.PortProtection	0		Decimal	BOOL	FF Bus Port Protection
FFL01In.Master.Status.BusVoltage	21.620079		Float	REAL	PA Bus Voltage (V)
FFL01In.Master.Status.BusCurrent	25.093683		Float	REAL	PA Bus Current (mA)
FFL01In.Master.Status.CPUTemperature	42.87983		Float	REAL	CPU Temperature
FFL01In.Master.Status.ConfigCRC	16#b4de		Hex	INT	Configuration Checksum
FFL01In.Master.Status.ActiveNodeCount	1		Decimal	SINT	Number of Active Slave Devices
FFL01In.Master.Status.SwitchOverTimeOut	1200		Decimal	INT	Redundancy Switch-Over Timeout

Figure 3.117 – Master Status tag

The Advanced Status tag displays additional information about the FF Link, including firmware revision and the status of the Ethernet ports. The ***Map Advanced Status*** option (located in the H1 configuration tab) must be selected for this tag to appear.

Name	Value	Style	Data Type	Description
FFL01In.Master.AdvancedStatus	{...}		FFLinkMasterAdvancedStatus	
FFL01In.Master.AdvancedStatus.EthPort1LinkUp	1	Decimal	BOOL	Port 1 Link: 0=Down, 1=Up
FFL01In.Master.AdvancedStatus.EthPort1Mirror	0	Decimal	BOOL	Port 1 Mirror: 0=Normal, 1=Mirror
FFL01In.Master.AdvancedStatus.EthPort2LinkUp	0	Decimal	BOOL	Port 2 Link: 0=Down, 1=Up
FFL01In.Master.AdvancedStatus.EthPort2Mirror	0	Decimal	BOOL	Port 2 Mirror: 0=Normal, 1=Mirror
FFL01In.Master.AdvancedStatus.DLREnabled	1	Decimal	BOOL	Device Level Ring Config: 0=Disabled, 1=Enabled
FFL01In.Master.AdvancedStatus.DLRRingMode	0	Decimal	BOOL	Device Level Ring Topology: 0=Linear, 1=Ring
FFL01In.Master.AdvancedStatus.DLRRingFault	0	Decimal	BOOL	Device Level Ring State: 0=Normal, 1=Fault
FFL01In.Master.AdvancedStatus.NTPEnabled	0	Decimal	BOOL	NTP Config: 0=Disabled, 1=Enabled
FFL01In.Master.AdvancedStatus.NTPLocked	0	Decimal	BOOL	NTP Status: 0=Not Locked, 1=Locked
FFL01In.Master.AdvancedStatus.FWRevMajor	2	Decimal	SINT	Firmware Major Revision
FFL01In.Master.AdvancedStatus.FWRevMinor	1	Decimal	SINT	Firmware Minor Revision
FFL01In.Master.AdvancedStatus.FWRevMicro	10	Decimal	SINT	Firmware Micro Revision
FFL01In.Master.AdvancedStatus.DIPSwitch	0	Decimal	SINT	DIP Switch Status
FFL01In.Master.AdvancedStatus.ConfigCRC	16#b348	Hex	INT	Configuration Checksum
FFL01In.Master.AdvancedStatus.MACAddress	{...}	Hex	SINT[6]	MAC Address
FFL01In.Master.AdvancedStatus.PortTripCounter	0	Decimal	DINT	Port Trip Counter
FFL01In.Master.AdvancedStatus.UpTime	2145	Decimal	DINT	Module Up Time
FFL01In.Master.AdvancedStatus.DTYear	2024	Decimal	INT	Module Year
FFL01In.Master.AdvancedStatus.DTMonth	9	Decimal	SINT	Module Month
FFL01In.Master.AdvancedStatus.DTDay	30	Decimal	SINT	Module Day
FFL01In.Master.AdvancedStatus.DTHour	8	Decimal	SINT	Module Hour
FFL01In.Master.AdvancedStatus.DTMinute	43	Decimal	SINT	Module Minute
FFL01In.Master.AdvancedStatus.DTSecond	21	Decimal	SINT	Module Second

Figure 3.118 – Advanced Status tag

There is also a tag created for each configured H1 device. The structure of which comprises the following:

- Input Status - Status related to H1 device
- Input Data – As specified in the block configuration and internal mapping
- Output Control – Used to mapping and station verification
- Output Data – As specified in the block configuration and internal mapping

FFL01MyTMT85In	{...}	{...}		FFL01452B48...	
FFL01MyTMT85In.Status	{...}	{...}		FFLinkDevice...	
FFL01MyTMT85In.Status.Online	1		Decimal	BOOL	Device Online (0=Offline, 1=Online)
FFL01MyTMT85In.Status.DataExchangeActive	1		Decimal	BOOL	Data Exchange Active (0=Inactive, 1=...
FFL01MyTMT85In.Status.IdentMismatch	0		Decimal	BOOL	Device Identity Mismatch (0=Ok, 1=...
FFL01MyTMT85In.Status.DeviceError	0		Decimal	BOOL	FF Device Error (0=Ok, 1=Error)
FFL01MyTMT85In.Status.SytemUpdateBusy	0		Decimal	BOOL	FFLink Busy Updating System Informa...
FFL01MyTMT85In.Status.APUpdateBusy	0		Decimal	BOOL	FFLink Busy Updating Application Pro...
FFL01MyTMT85In.Status.MapChecksumMismatch	0		Decimal	BOOL	Device Mapping Checksum Mismatch...
FFL01MyTMT85In.Status.Station	34		Decimal	INT	Station Address
FFL01MyTMT85In.PV1	{...}	{...}		FFLinkInputP...	
FFL01MyTMT85In.PV1.Status	{...}	{...}		FFLinkInputP...	
FFL01MyTMT85In.PV1.Value	25.449615		Float	REAL	PV Value

Figure 3.119 – Slave Device-Specific tag

3.8. ETHERNET/IP ORIGINATOR

The FF Link module 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 Master configuration while the cyclic class 1 connections are added to the *EtherNet/IP Connections* node under the FF Link module in the Slate project tree.

3.8.1. EXPLICIT ETHERNET/IP MESSAGING

Up to ten EtherNet/IP devices can be added for explicit messaging. The user will need to add each device as explained in section 3.8.1.1 below. Once the EtherNet/IP devices have been added the user can then configure the require mapping for the EtherNet/IP Explicit messaging as shown in section 3.8.1.2 below.

3.8.1.1. ETHERNET/IP DEVICES

This tab is enabled only 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 table with a size of 100Kbytes.

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

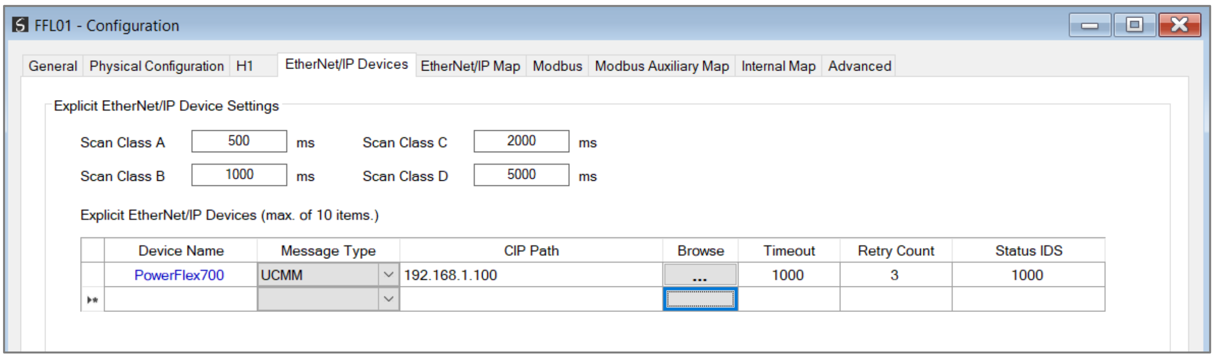


Figure 3.120 – EtherNet/IP Devices configuration

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 mapped item 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 of the following message types when communicating with the target EtherNet/IP device: **UCMM** – Unconnected Explicit Message **Class 3** – Connected Explicit Message **Logix Tag** – Connected Explicit Message (when reading or writing to Logix tags)
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 data table (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 3.19 – EtherNet/IP Devices configuration parameters

3.8.1.2. ETHERNET/IP MAP

This tab is enabled only 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 a data table with a size of 100Kbytes.

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

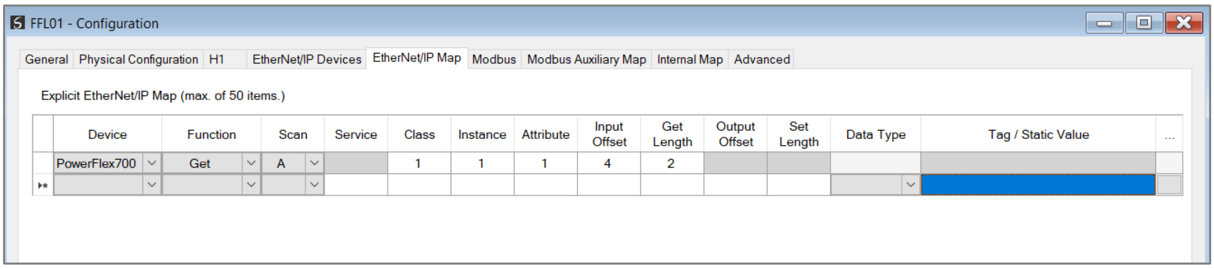


Figure 3.121 – EtherNet/IP Map configuration

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 Data Table 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 Data Table 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 FF Link module can be setup to write the fixed value only once when the target device communication has been established. **Custom** This function allows the user to use a custom Service to write and read data in the same transaction. The user will need to see which custom services that target device supports in that device's user manual. **Read Tag** When using a Logix controller as a EtherNet/IP Device, the module can read a Logix tag from the target Logix controller using Class 3 or UCMM messaging. The value from the tag will be saved at the configured Input Offset. **Write Tag**

	When using a Logix controller as a EtherNet/IP Device, the 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 Table 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 Table. This will only be available for ***Get*** and ***Custom*** functions.
Output Offset	The location in the internal Data Table from where the data to be written to the target device will be read. This will only be available for ***Set*** and ***Custom*** functions.
Set Length	The length of the data to be written. This will only be available for ***Set*** and ***Custom*** functions.
Data Type	The data type of the Static Value. This will only be available for ***Set Static*** function.
Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.20 – EtherNet/IP Map configuration parameters

3.8.2. Cyclic Class 1 Connection

The FF Link 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

3.8.2.1. 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***.

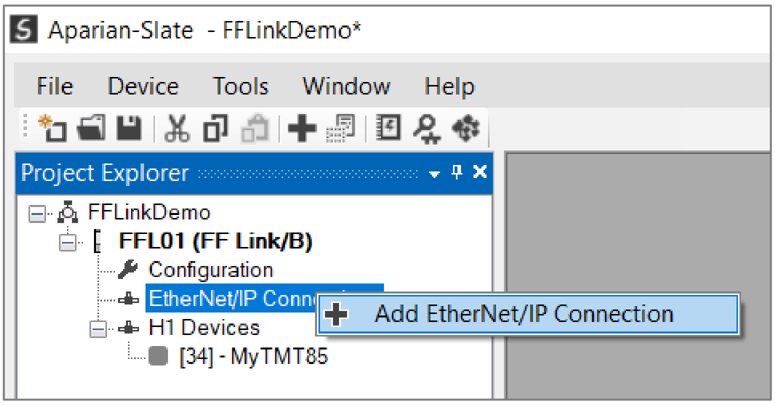


Figure 3.122 – Adding EtherNet/IP Class 1 Connection

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

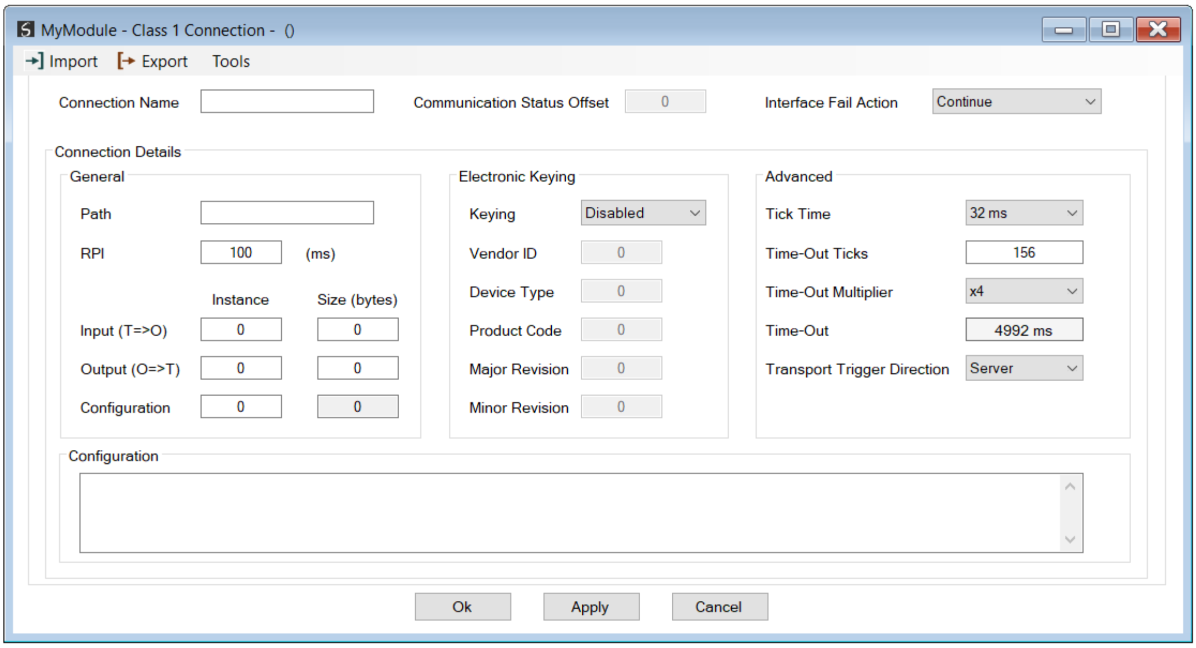


Figure 3.123 – EtherNet/IP Class 1 Connection Parameters

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

Parameter	Description
Connection Name	The instance name given to the Class 1 Connection.
Comm Status Offset	This is the offset in the data table (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.
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*** textbox below.
Electronic Keying	
Keying	Electronic Keying can be used to ensure that the target device is the correct device type. The available options are: **Disabled** Keying is disabled 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	
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 3.21 – EtherNet/IP Class 1 Connection Parameters

3.8.2.2. IMPORT FROM ONLINE CONTROLLER

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

3.8.2.2.1. 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.

3.8.2.2.2. 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.

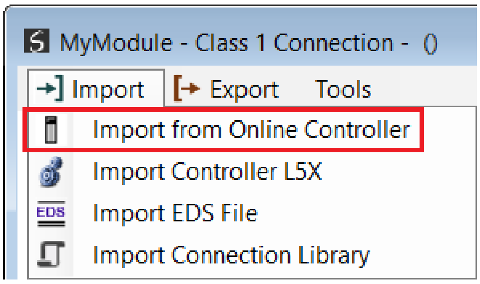


Figure 3.124 – Import from Online Controller

The Import Connection Parameters form will open.

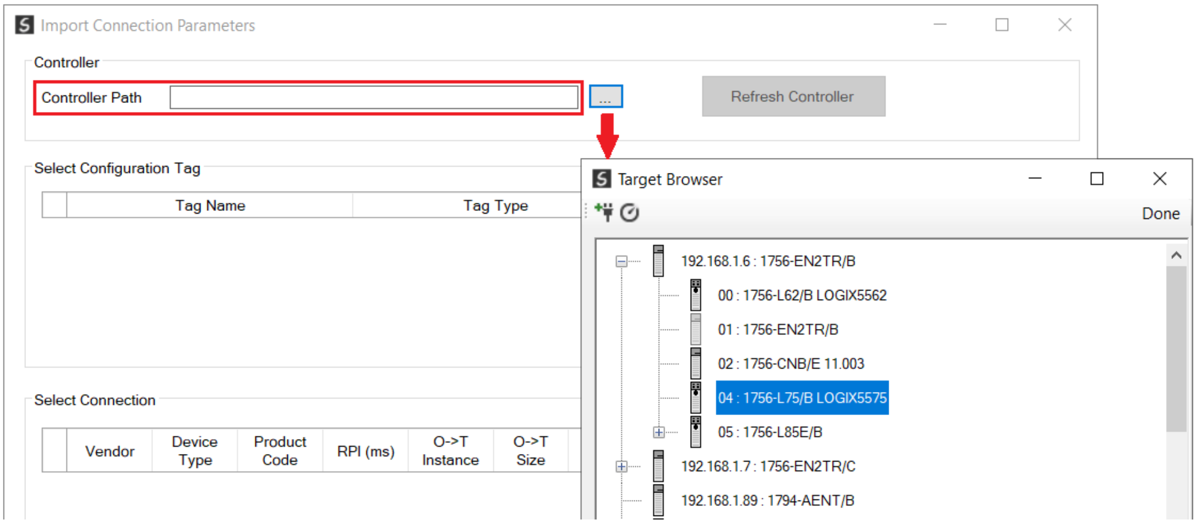


Figure 3.125 – Import Connection Parameters – Controller Path

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.

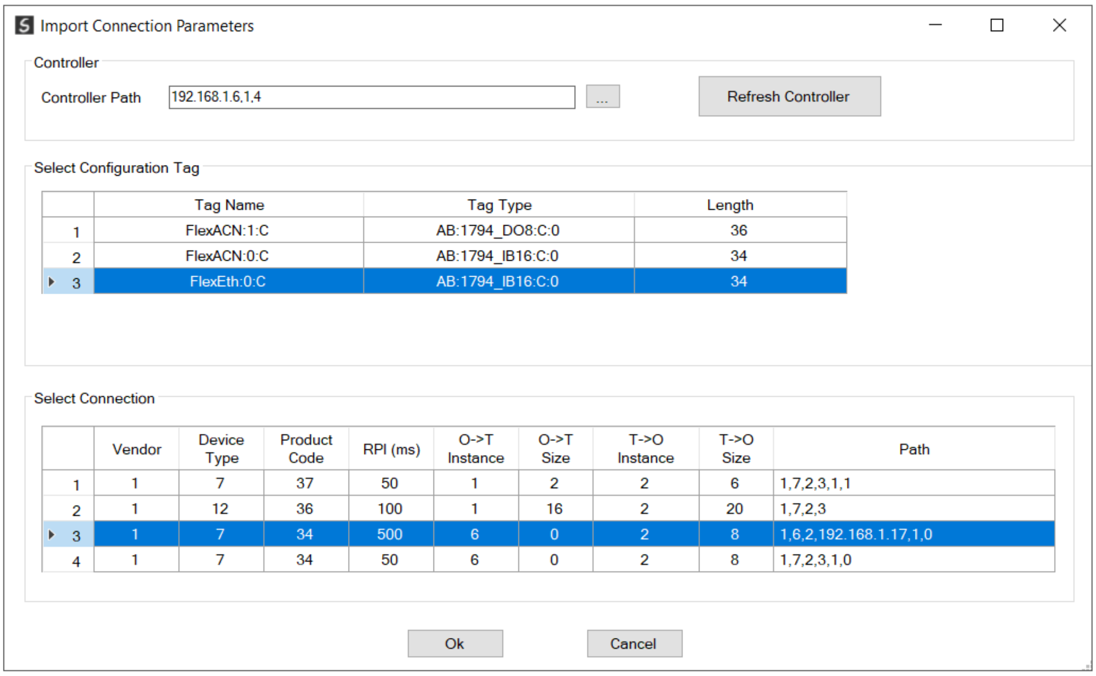


Figure 3.126 – Import Connection Parameters – Select Connection

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.

3.8.2.3. IMPORT FROM CONTROLLER L5X FILE

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

3.8.2.3.1. 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.

3.8.2.3.2. 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.

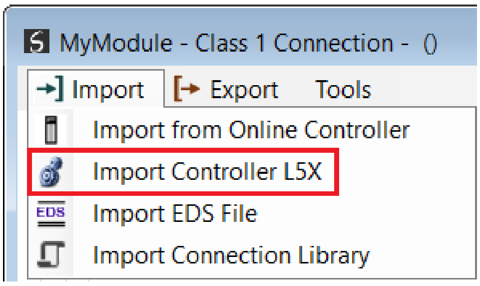


Figure 3.127 – Import from Controller L5X

The Import Connection Parameters form will open.

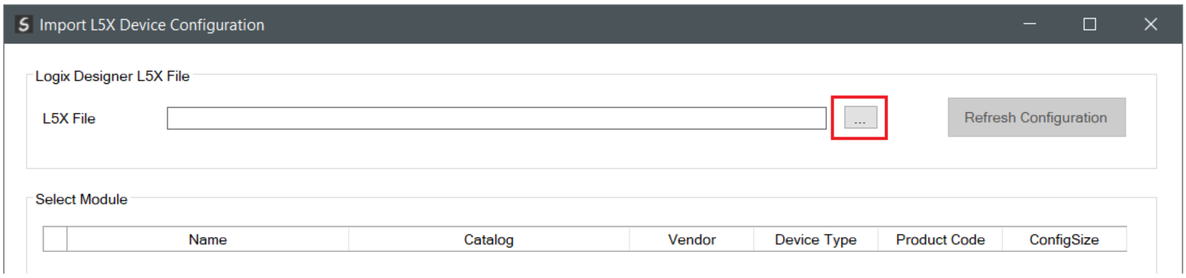


Figure 3.128 – Import L5X Device Configuration – Select L5X

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

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

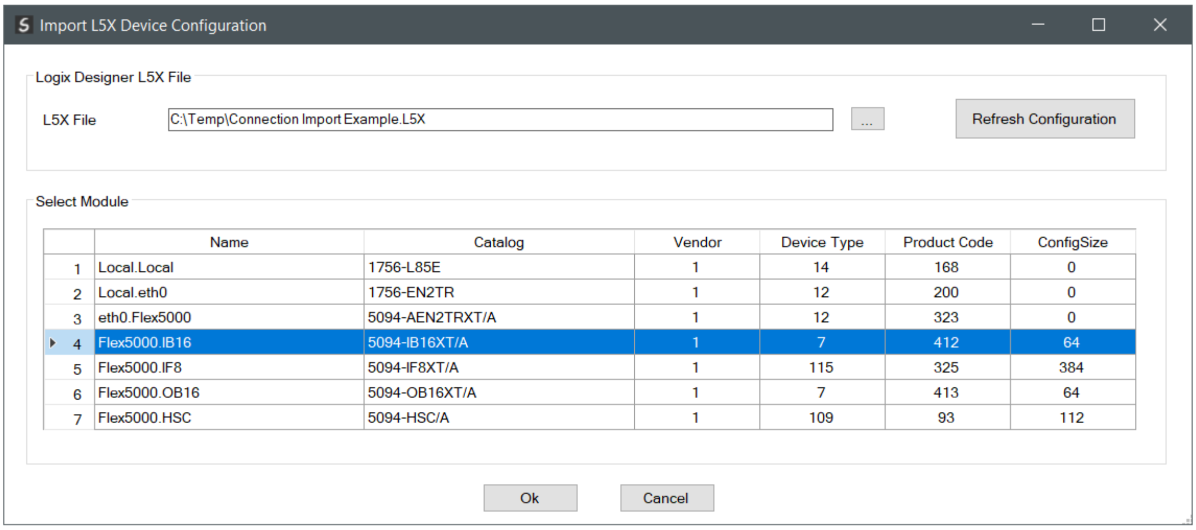


Figure 3.129 – Import L5X Device Configuration

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.

3.8.2.4. 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.

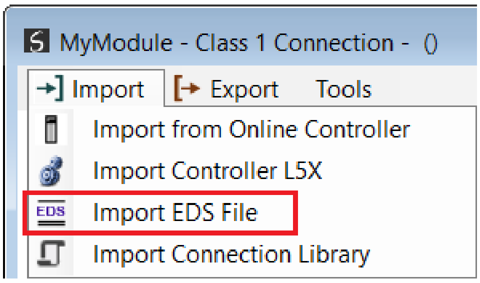


Figure 3.130 – Import EDS File

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

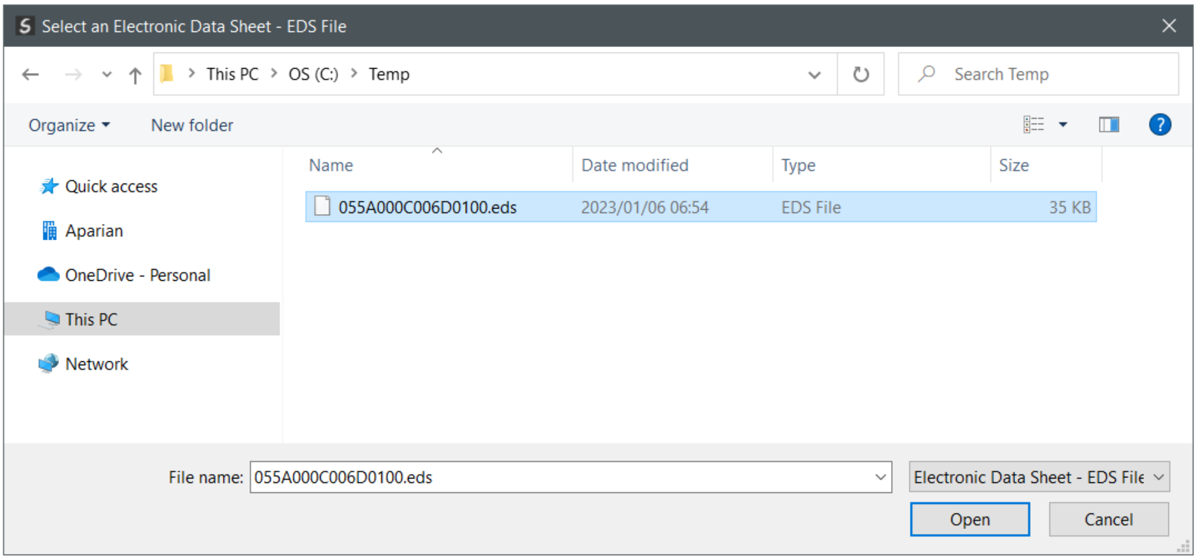


Figure 3.131 – Browse to EDS File

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.

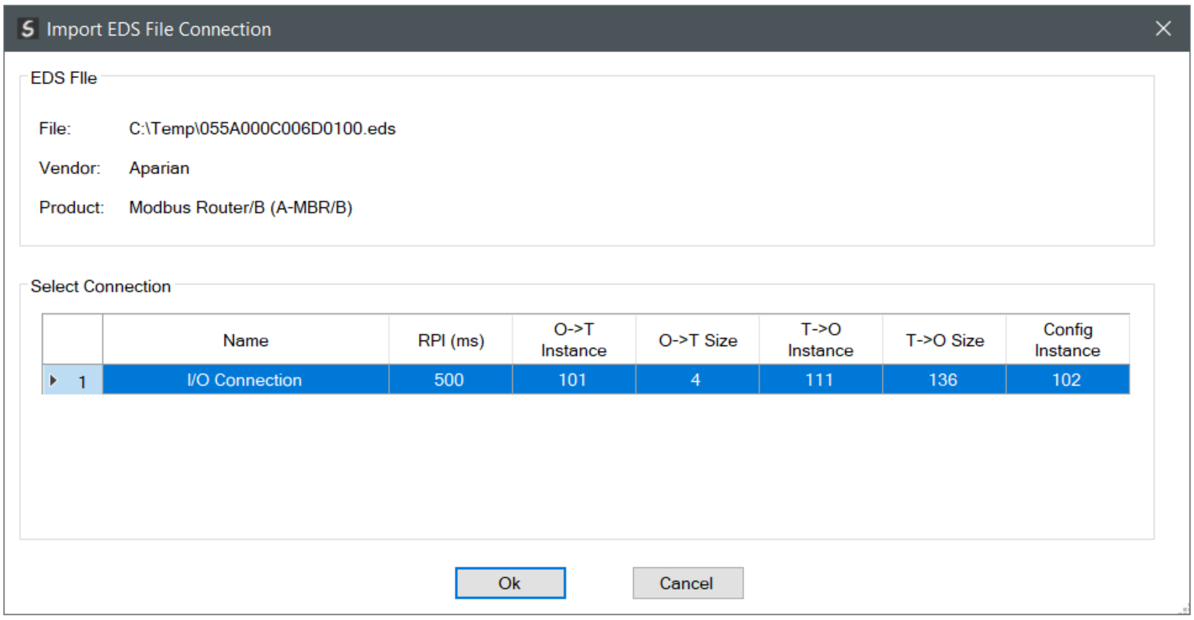


Figure 3.132 – Select Connection

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.

3.8.2.5. 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.

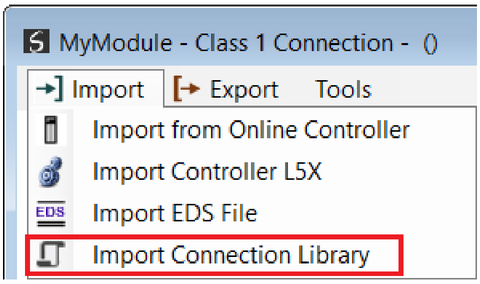


Figure 3.133 – Import Connection Library File

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.

3.8.2.5.1. EXPORT LIBRARY FILE

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

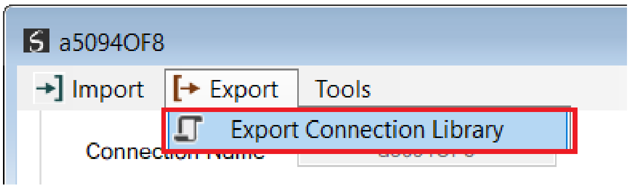


Figure 3.134 – Export Connection Library File

3.9. MODBUS TCP SLAVE

3.9.1. Modbus

The Modbus configuration is shown in the figure below. The Modbus configuration is used when the ***Primary Interface*** has been set to ***Modbus TCP Master*** or ***Modbus TCP Slave***.

NOTE: When *Auto Target Mapping* is selected in the H1 configuration, then the Internal Map with the respective Modbus Registers will automatically be built as the configuration is updated.

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

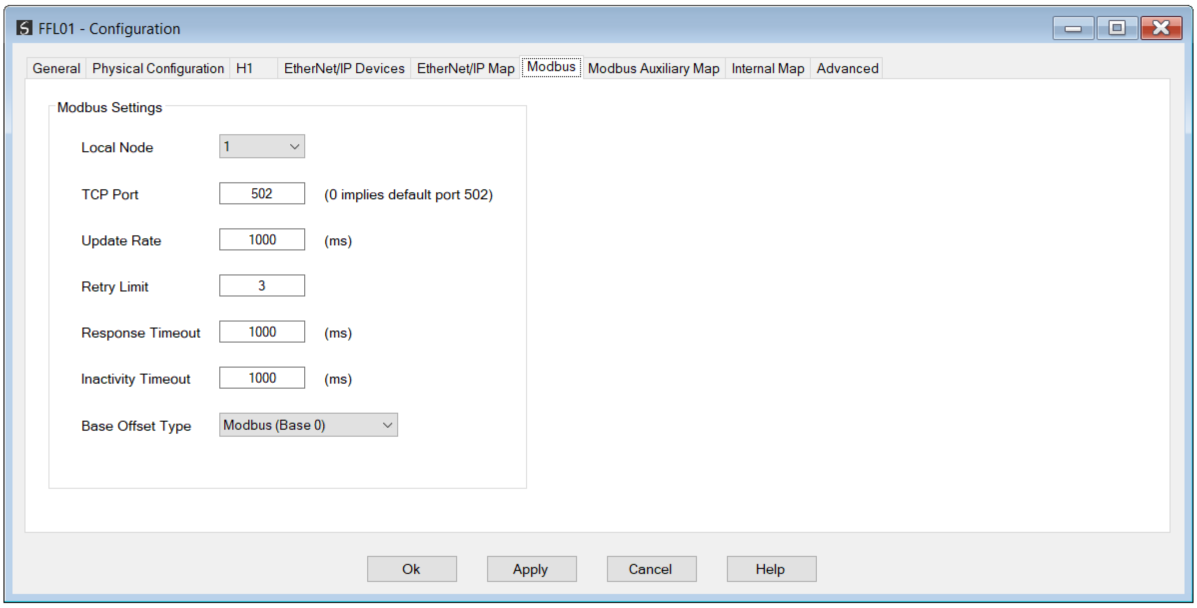


Figure 3.135 – FF Link Modbus Addressing configuration

The Modbus configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node Number assumed by the FF Link.
TCP Port	The TCP port to be used for the Modbus communication can be configured. If a zero is entered, the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between master requests to the target Modbus device. **(Modbus TCP Master mode only)**
Retry Limit	The number of successive Modbus request retries. **(Modbus TCP Master mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a Modbus response. **(Modbus TCP Master mode only)**
Inactivity Timeout	The slave timeout time in milliseconds. If no communication from the remote Master is received within this time communication is deemed to have failed. **(Modbus TCP Slave mode only)**

Base Offset Type	**Modbus (Base 0)** Conventional Modbus addressing where the first address is 0. **PLC (Base 1)** PLC addressing, where the first address is 1.

Table 3.22 - Modbus configuration parameters

3.10. MODBUS TCP MASTER

3.10.1. MODBUS

The Modbus configuration is shown in the figure below. The Modbus configuration is used when the ***Primary Interface*** has been set to ***Modbus TCP Master*** or ***Modbus TCP Slave***.

NOTE: When *Auto Target Mapping* is selected in the H1 configuration, then the Internal Map with the respective Modbus Registers will automatically be built as the configuration is updated.

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

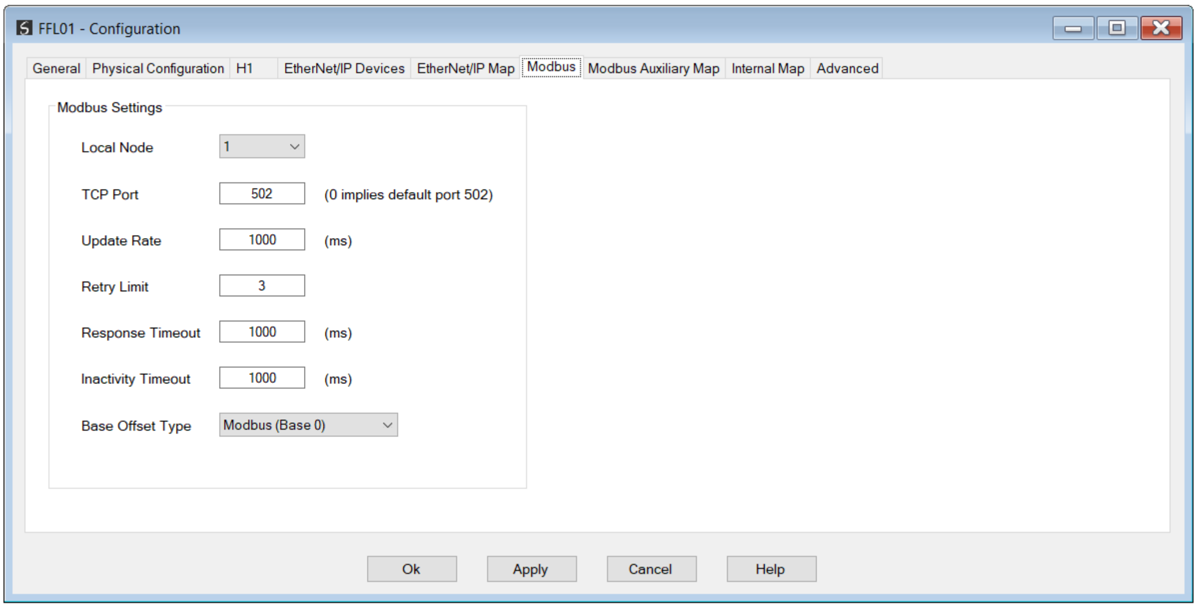


Figure 3.136 – FF Link Modbus Addressing configuration

The Modbus configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node Number assumed by the FF Link.

TCP Port	The TCP port to be used for the Modbus communication can be configured. If a zero is entered, the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between master requests to the target Modbus device. **(Modbus TCP Master mode only)**
Retry Limit	The number of successive Modbus request retries. **(Modbus TCP Master mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a Modbus response. **(Modbus TCP Master mode only)**
Inactivity Timeout	The slave timeout time in milliseconds. If no communication from the remote Master is received within this time communication is deemed to have failed. **(Modbus TCP Slave mode only)**
Base Offset Type	**Modbus (Base 0)** Conventional Modbus addressing where the first address is 0. **PLC (Base 1)** PLC addressing, where the first address is 1.

Table 3.23 - Modbus configuration parameters

3.10.2. Modbus Auxiliary Map

The Modbus Auxiliary Map configuration is shown in the figure below. This table will be enabled when the FF Link has the primary interface set to ***Modbus TCP Master***. This will allow the user to read and/or write any internal FF Link Modbus Register to any Modbus Slave. Up to 20 Modbus Slaves can be connected and up to 200 mapped items can be configured.

The FF Link ***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***.

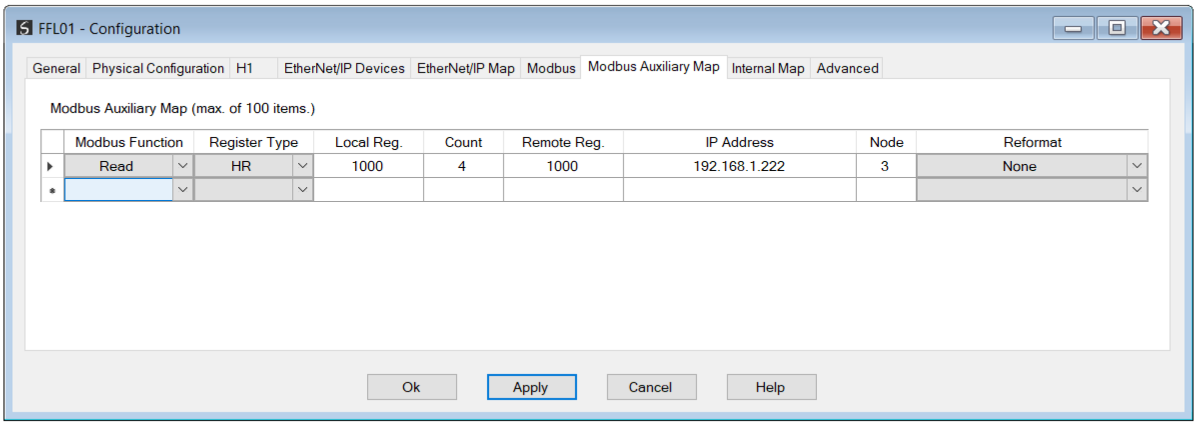


Figure 3.137 – FF Link Modbus Auxiliary Map configuration

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Modbus Function	This is the Modbus function is the used with the Modbus Slave. **Read** – Read a Modbus Register (e.g.: HR, IR, CS, or IR) from a Modbus Slave. **Write** – Write a Modbus Register (e.g.: HR, IR, CS, or IR) to a Modbus Slave.
Register Type	Modbus Register Type: **CS** – Coil Status **IS** – Input Status **IR** – Input Register **HR** – Holding Register
Local Reg.	The local (internal) FF Link Modbus address.
Count	The number of Modbus elements to read or write.
Remote Reg.	The remote slave Modbus address.
IP Address	The IP address of the remote Modbus TCP slave.
Node	The Modbus Node address of the remote Modbus slave.
Reformat	How the data is formatted before reading or writing from/to the Modbus slave. **None** – No reformatting will be done. **BB AA** – 16-bit Byte swap **BB AA DD CC** – 32-bit Byte Swap **CC DD AA BB** – Word Swap **DD CC BB AA** – Word and Byte Swap

Table 3.24 - Modbus Auxiliary Map configuration parameters

3.11. ALERT (ALARM) NOTIFICATION

The FF Link supports H1 Alert management by buffering report notifications from H1 devices and allowing the Primary Interface to unload these events. Events must be enabled and configured in the FF Link as well as the H1 device blocks.

In the General H1 configuration, the ***Enable Alert Extraction*** must be set.

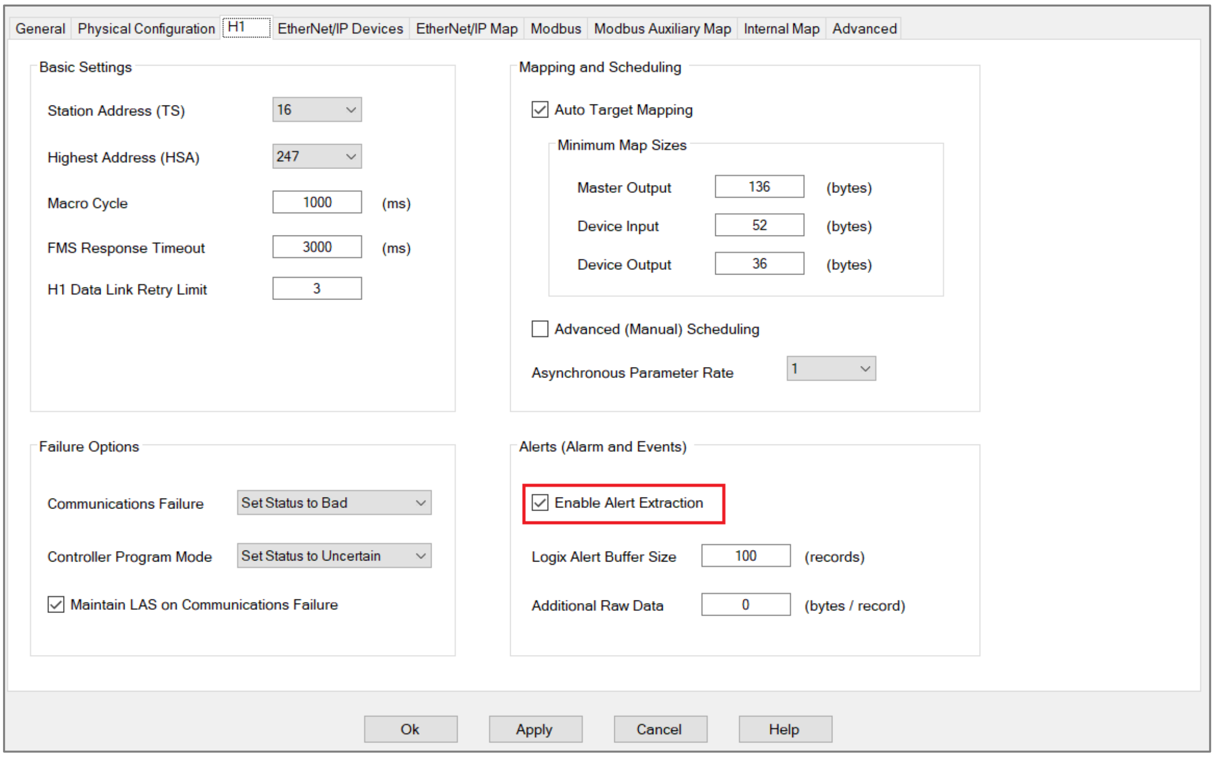


Figure 3.138 – General H1 configuration *Enable Alarm Extraction*

In the Device General configuration, the ***Enable Alert Extraction*** must be set.

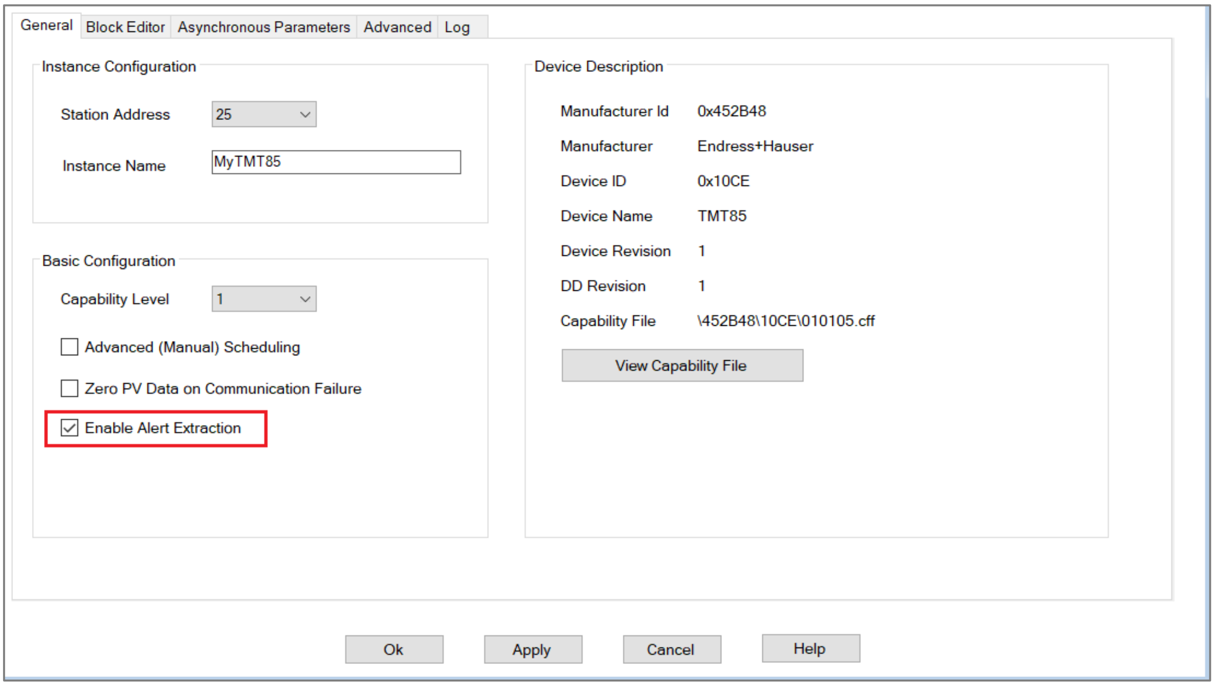


Figure 3.139 – Device General configuration *Enable Alarm Extraction*

Next, the H1 device will need to have Reports enabled to allow Alerts to be sent automatically. This is achieved by right-clicking on the H1 device Resource block (in the Block Editor) and selecting ***Parameters***.

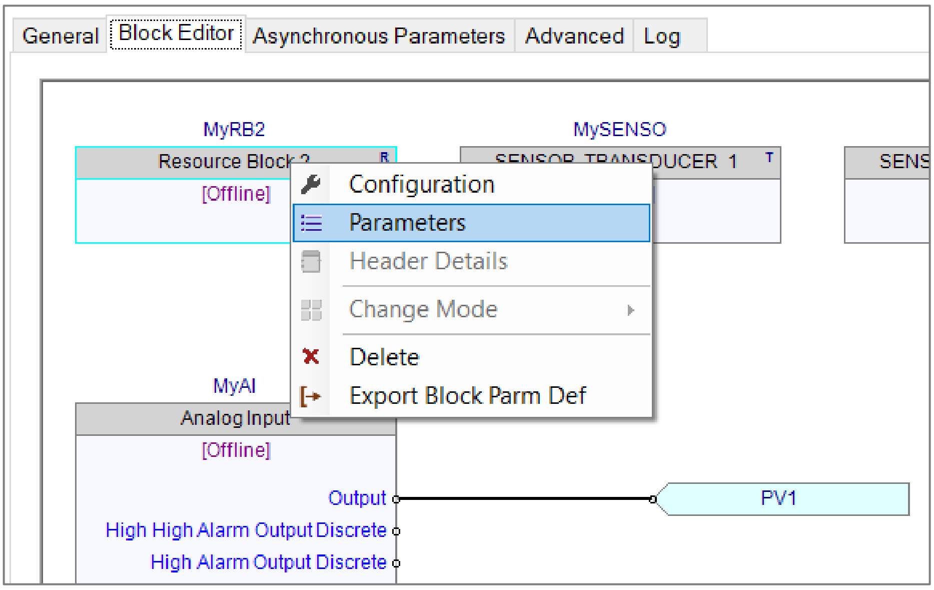


Figure 3.140 – Device General configuration *Enable Alarm Extraction*

Under the ***Features*** parameter, select the ***Reports*** option.

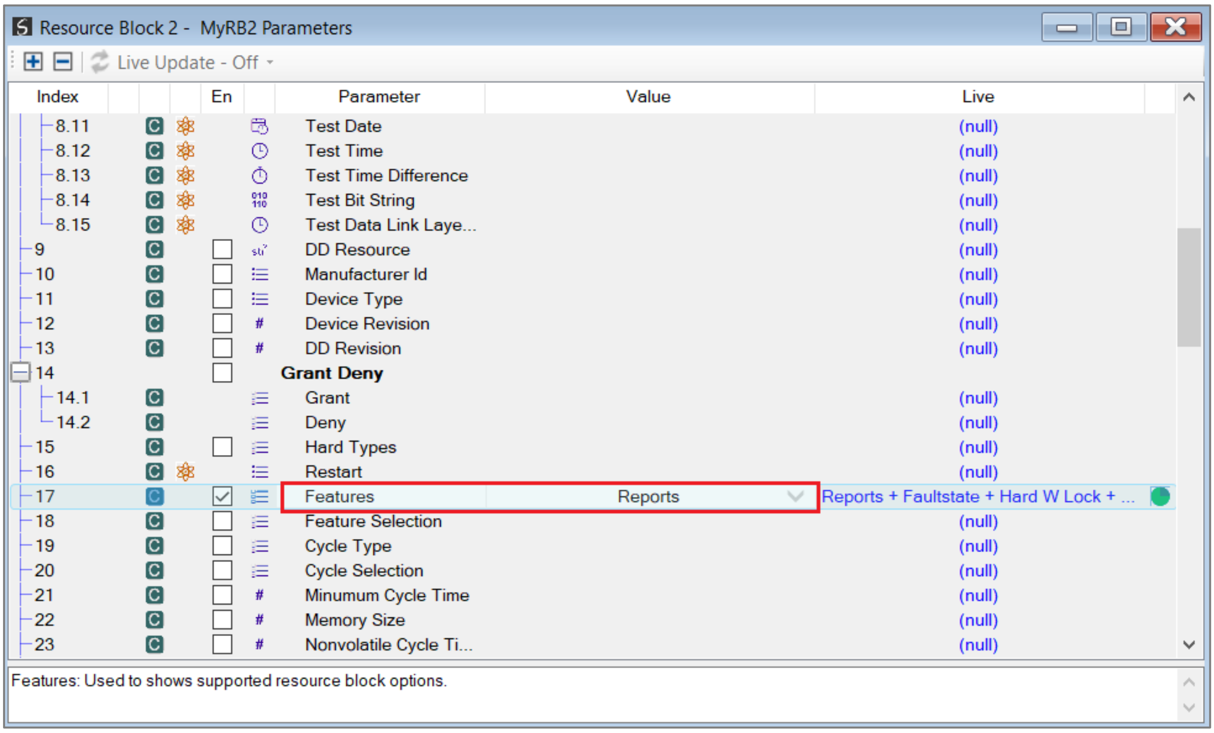


Figure 3.141 – H1 Device Features - Reports

If analog alerts need to be used in the function block, then certain parameters in the function block will need to be set. Right-click on the function block (in the Block Editor) and selecting ***Parameters***.

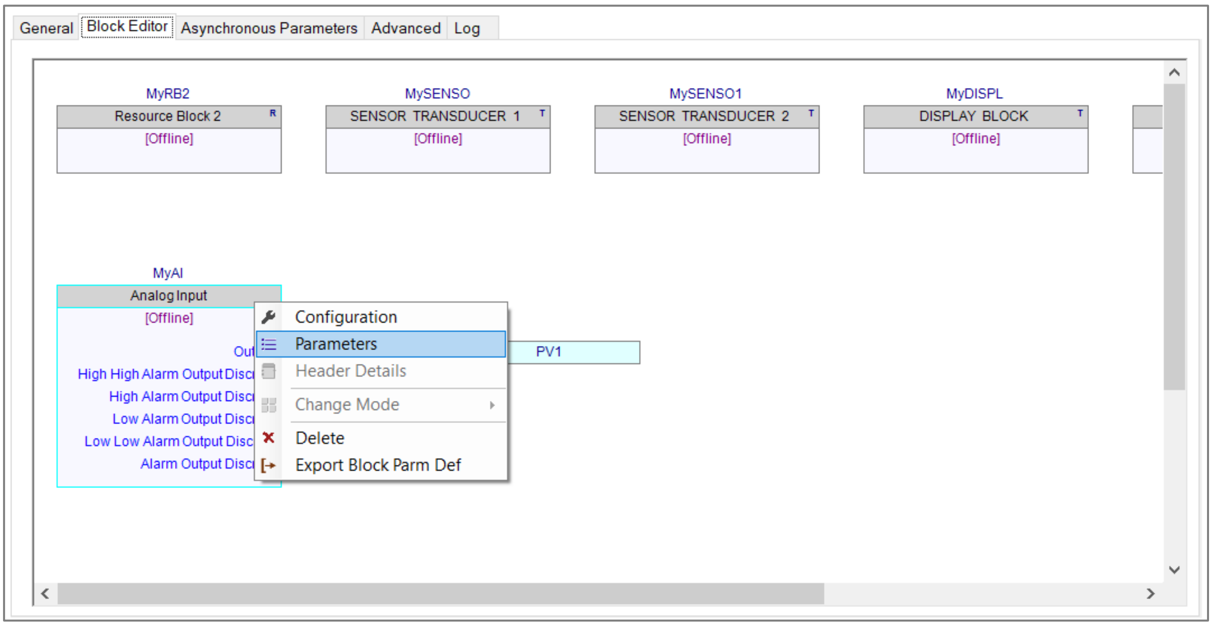


Figure 3.142 – H1 Device – Function Block Parameter Selection

For each of the analog alarms, the user will need to configure the ***Limit*** as well as the ***Priority*** parameters.

NOTE: Certain H1 devices require the ***Priority*** to be set to a value between 3 and 7.

The user will also need to set the ***Acknowledge Option*** to ***Auto Acknowledge*** the analog alarms.

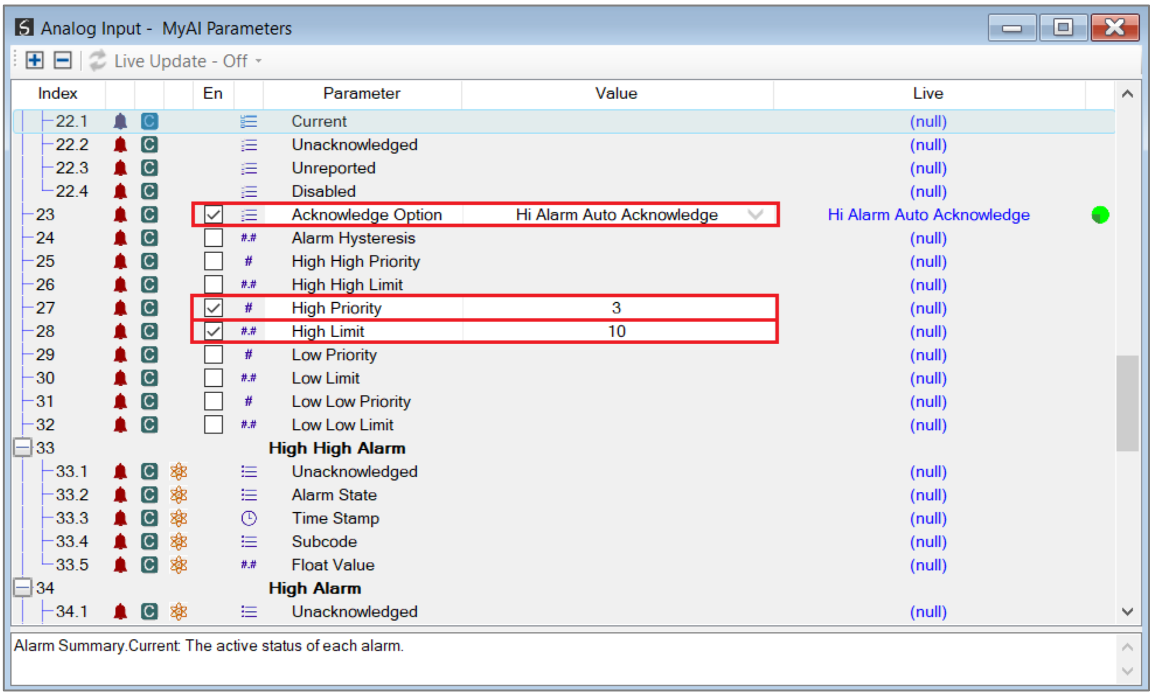


Figure 3.143 – H1 Device – Function Block Parameters

NOTE: If the *Enable Alert Extraction* configuration option has been enabled on an H1 device (*General Configuration* page in Slate), and it has been downloaded, then the user will also need to re-download to the H1 device in the Block Editor window. This is required to configure the Alarm VCRs.

3.12. CASCADE CONTROL

Cascade control involves the output of one function block being used to control another. These function blocks may be in the same field device, or different field devices.

When a function block (e.g. Analog Output) needs to be controlled by an external controller (e.g. Logix) via the FF Link, then the correct handshaking between this block and the controller must be provided. The handshaking process makes use of the ***Status*** byte of both the ***CAS_IN*** (forward path) and the ***BKCAL_OUT*** (backward path).

In the example below the forward control path is provided by the Master PV “MPV2” tag connected to the “CAS_IN” input, and the backward path by the “BKCAL_OUT” port connected to the device PV “PV2”.

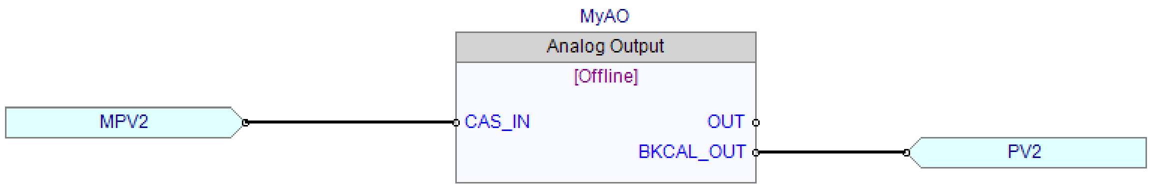


Figure 3.144 – Typical Cascade Control Strategy

3.12.1. Cascade Control Handshaking

When the AO block is in ***Auto*** mode, the BKCAL_OUT will indicate the cascade ***NotInvited*** status.

If the block is then commanded to go to ***Auto*** + ***Cascade*** mode, then the BKCAL will raise the ***InitializeRequest*** flag. The upstream function block (or "virtual block" logic in the case of Logix control), will then Acknowledge the request, by setting the ***InitAck*** flag (sub-status bit 0) of the status byte connected to the CAS_IN port.

The AO function block, after receiving the Acknowledgment, will then enter ***Cascade*** mode, and clear the ***InitializeRequest*** flag.

This handshaking procedure is summarized in the diagram below.

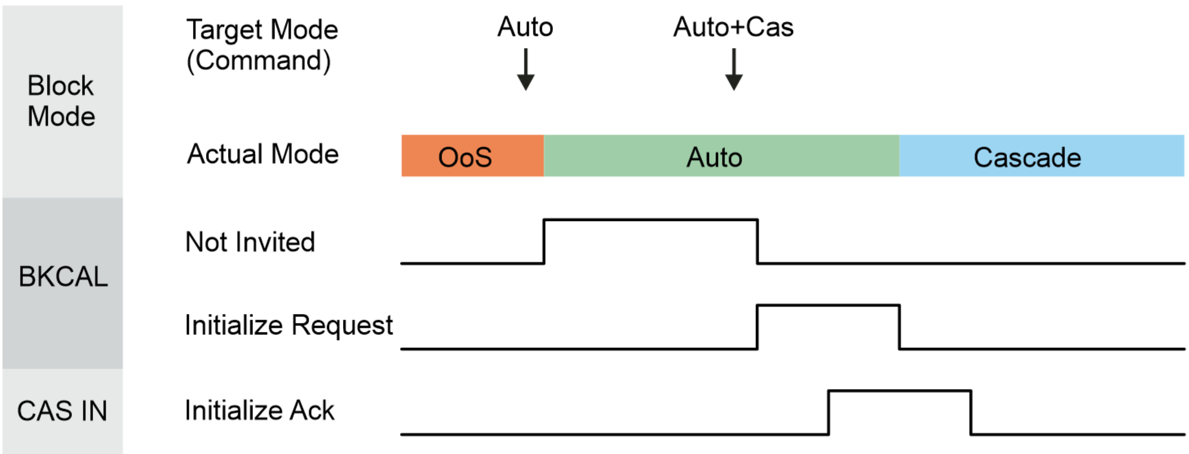


Figure 3.145 – Cascade Control Handshaking

NOTE: When using the FFLink with a Logix controller, the generated Logix code will automatically include the Cascade Add-On-Instruction (AOI) to manage the Cascade handshaking process

3.12.2. Logix Cascade Control

Cascade control with Logix is simplified through the use of the Cascade control AOI (FFLinkCASControl) which is automatically generated in the Logix L5X export file.

The following steps describe how to place a block in Casacde mode.

Continuing with the previous example, once the control strategy has been downloaded to the field device, the Analog Output (AO) block should be in ***Auto*** mode.

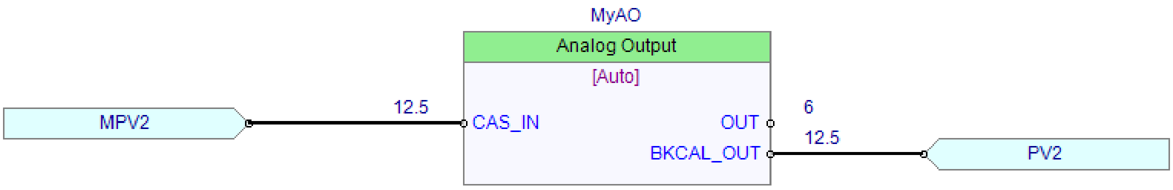


Figure 3.146 – AO block in Auto

In Logix, the FFLinkCASControl AOI would show that the Cascade control is ***NotInvited***.

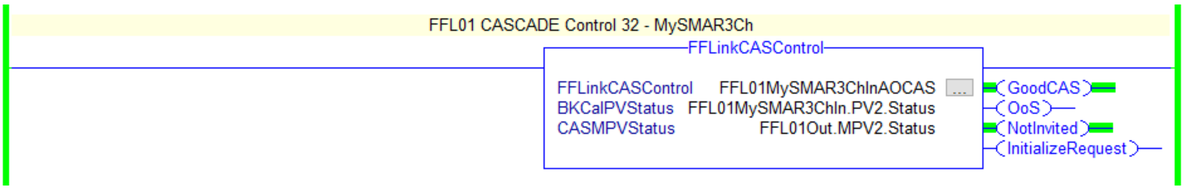
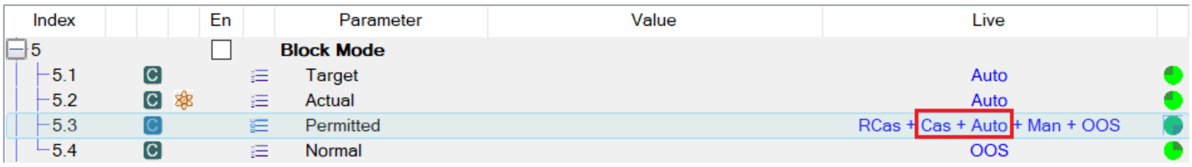


Figure 3.147 – CASControl AOI – Not Invited

Before the AO block can be set to ***Auto+CAS*** mode, the user should ensure that ***CAS*** mode is available in the ***Block Mode - Permitted*** parameter. If not, then it should be enabled by writing to this parameter.



Index	En	Parameter	Value	Live
5		Block Mode		
5.1		Target		Auto
5.2		Actual		Auto
5.3		Permitted		RCas + Cas + Auto + Man + OOS
5.4		Normal		OOS

Figure 3.148 – Block Mode – Permitted

The user can now initiate the Cascade control process, by right-clicking on the ***AO*** block and selecting the ***Change Mode***, and then ***Auto + CAS*** option.

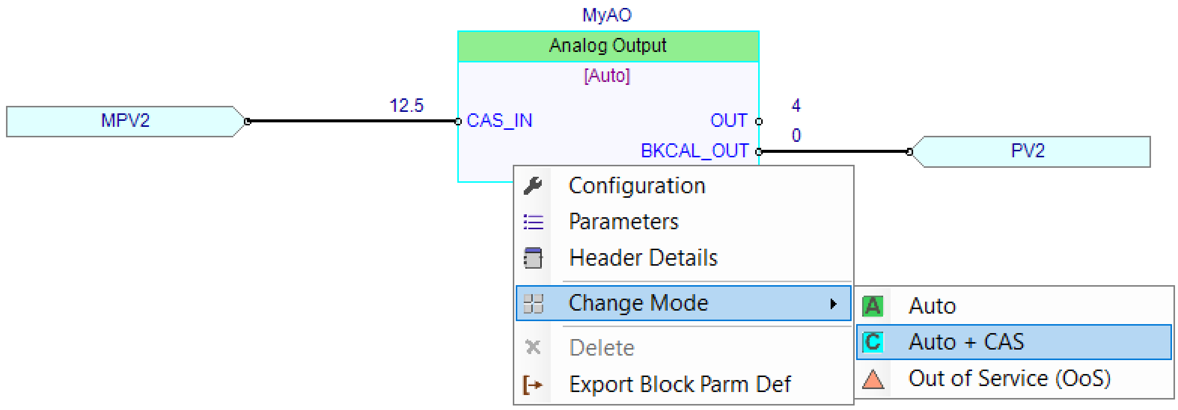


Figure 3.149 – Change Mode – Auto + CAS

In Logix, the CASControl AOI, will indicate the ***InitializeRequest*** flag being raised for a short interval.

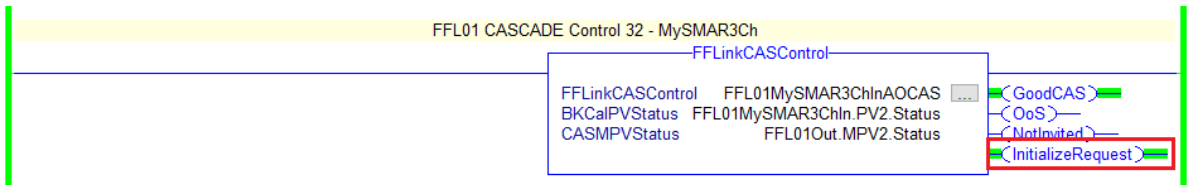


Figure 3.150 – CASControl AOI – Initialize Request

The AO block will then enter ***Cascade*** (CAS) mode.

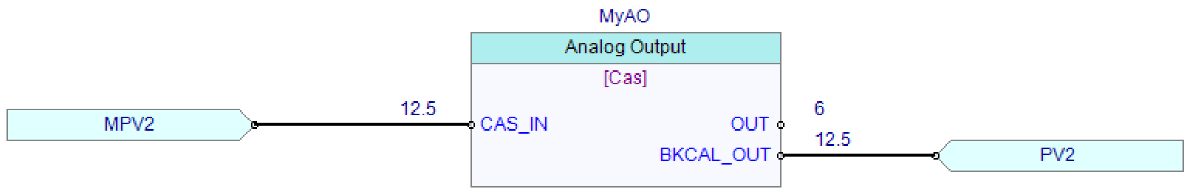


Figure 3.151 – AO Block – Cascade mode

The user will now be able to command the AO block through the referenced PV Master tag's ***Value*** member.

Name	Value	Data Type
FFL01Out.MPV2	{...}	FFLinkOutputPVReal
FFL01Out.MPV2.Status	{...}	FFLinkOutputPVStatus
FFL01Out.MPV2.Value	12.5	REAL

Figure 3.152 – Master PV

3.13. SYSTEM DETAILS

The System Details window provides a summary of the FF link configuration, allowing the user to verify the configuration before downloading to the H1 devices and FF Link.

To open the System Details window right-click on the FF Link in the project explorer and select the ***System Details*** option.

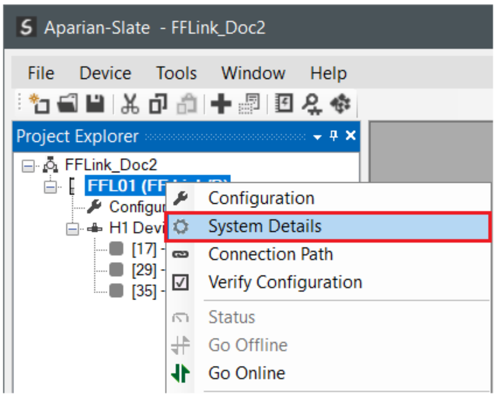


Figure 3.153 – Selecting System Details

3.13.1. TIMING

The ***Timing*** tab of the System Details window displays a summary of the H1 bus timing.
The x-axis is time (displayed in milliseconds) and spans one complete Macro Cycle.
The FF Link Master and H1 devices are listed on the left (y-axis) and include their instantiated function blocks and connector schedules.

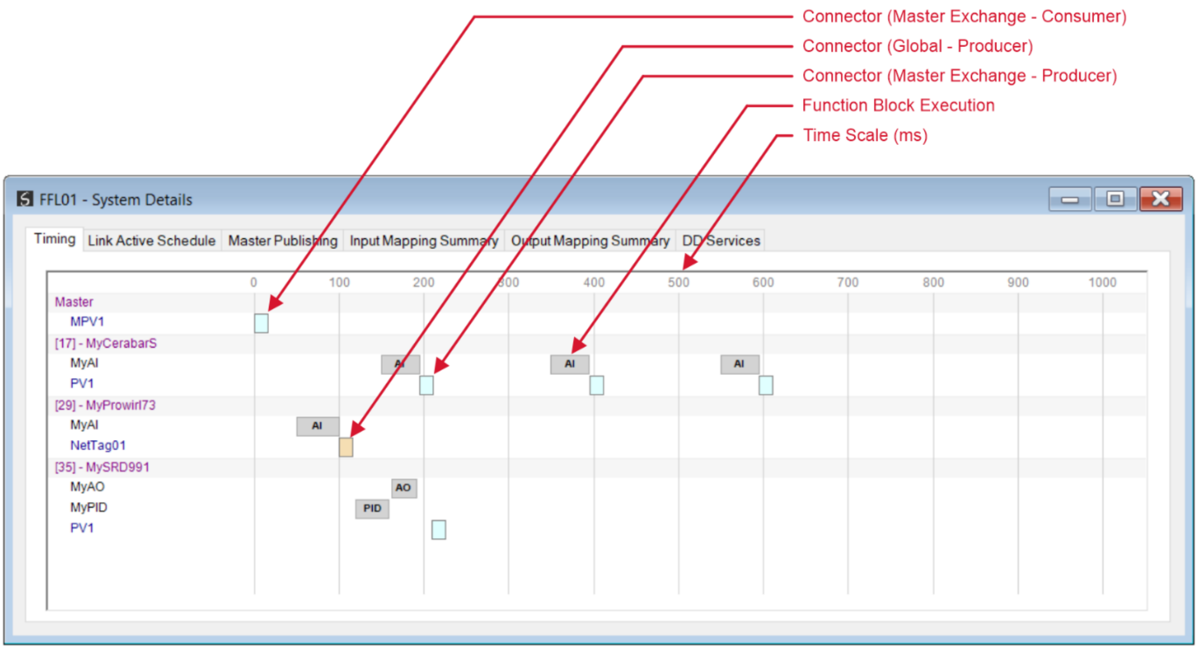
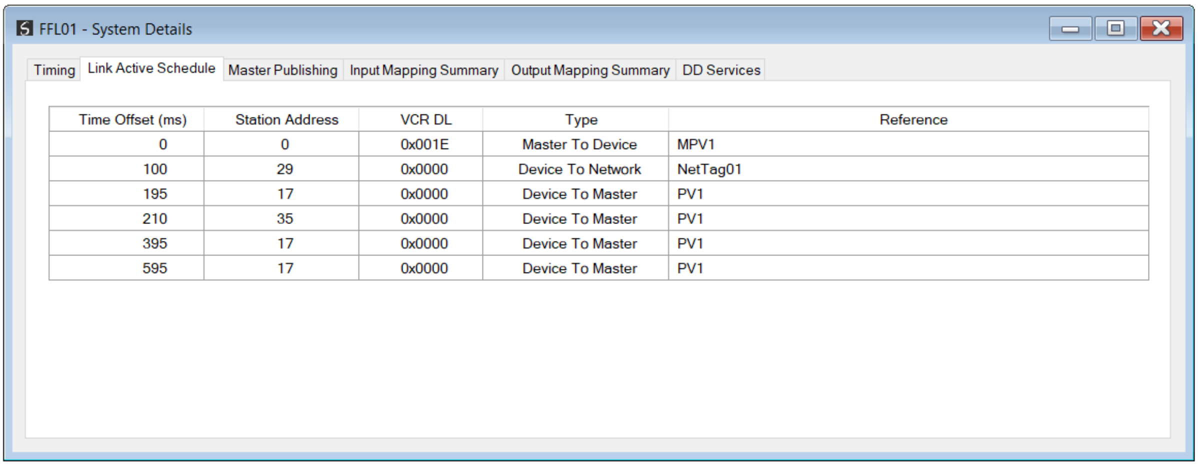


Figure 3.154 – System Details – Timing

3.13.2. Link Active Schedule

The ***Link Active Schedule*** tab displays a summary of the data points to be compelled within each Macro Cycle.

Note that the ***VCR DL*** may be zero for an H1 device that has not yet been downloaded.

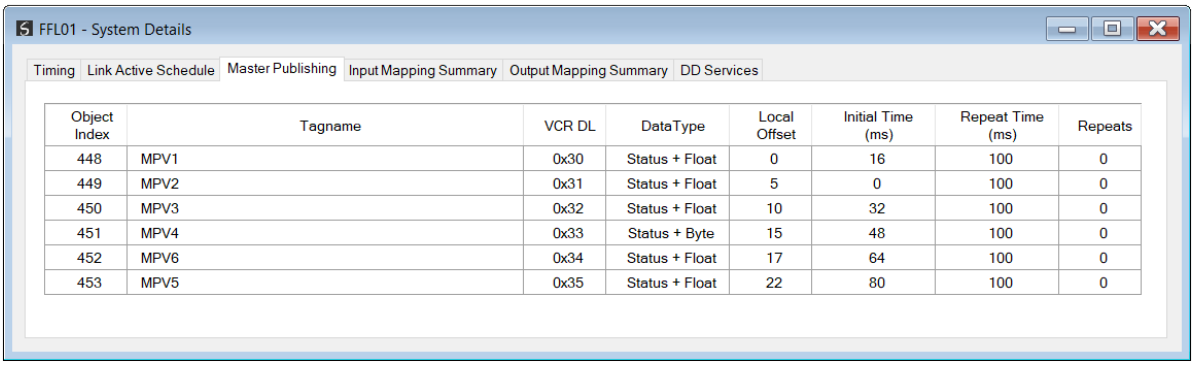


Time Offset (ms)	Station Address	VCR DL	Type	Reference
0	0	0x001E	Master To Device	MPV1
100	29	0x0000	Device To Network	NetTag01
195	17	0x0000	Device To Master	PV1
210	35	0x0000	Device To Master	PV1
395	17	0x0000	Device To Master	PV1
595	17	0x0000	Device To Master	PV1

Figure 3.155 – System Details – Link Active Schedule

3.13.3. Master Publishing

The ***Master Publishing*** tab displays a summary of all the data points that are sent from the FF Link Master to the H1 devices.



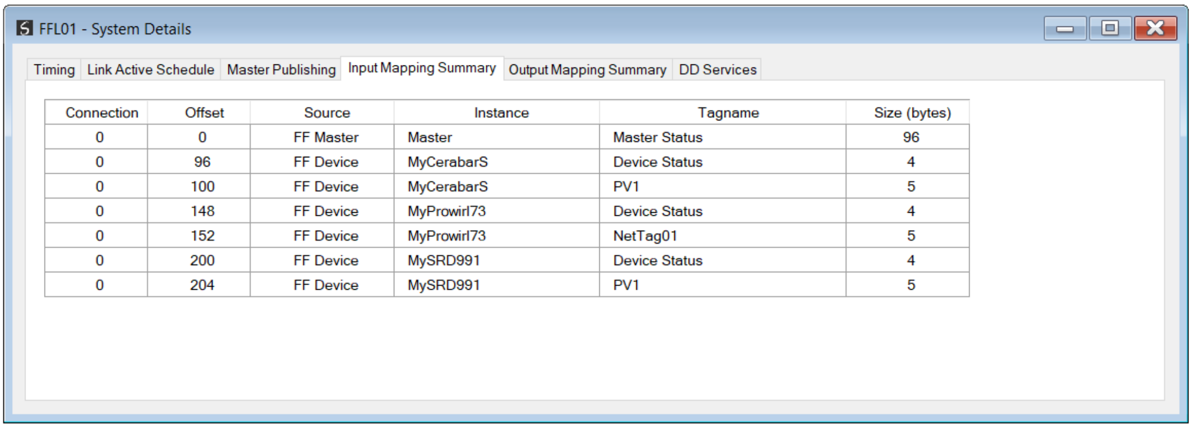
Object Index	Tagname	VCR DL	DataType	Local Offset	Initial Time (ms)	Repeat Time (ms)	Repeats
448	MPV1	0x30	Status + Float	0	16	100	0
449	MPV2	0x31	Status + Float	5	0	100	0
450	MPV3	0x32	Status + Float	10	32	100	0
451	MPV4	0x33	Status + Byte	15	48	100	0
452	MPV6	0x34	Status + Float	17	64	100	0
453	MPV5	0x35	Status + Float	22	80	100	0

Figure 3.156 – System Details – Master Publishing

3.13.4. Input Mapping Summary

The ***Input Mapping Summary*** tab displays all the data points that are being transferred from the FF Link Master to the Primary Interface. (Input Data).

These points would typically include the Master Status, Device Status, Master Exchange-Producer connectors and Read Asynchronous Parameters.



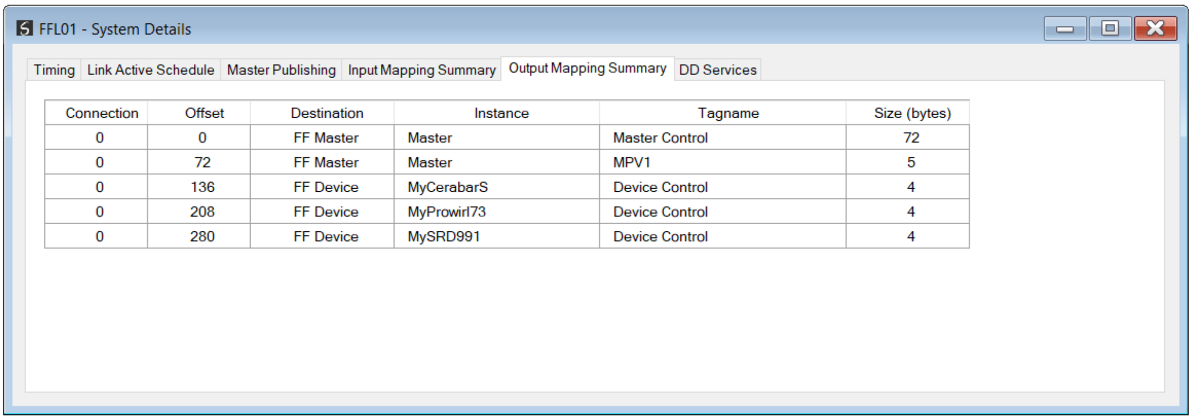
Connection	Offset	Source	Instance	Tagname	Size (bytes)
0	0	FF Master	Master	Master Status	96
0	96	FF Device	MyCerabarS	Device Status	4
0	100	FF Device	MyCerabarS	PV1	5
0	148	FF Device	MyProwirl73	Device Status	4
0	152	FF Device	MyProwirl73	NetTag01	5
0	200	FF Device	MySRD991	Device Status	4
0	204	FF Device	MySRD991	PV1	5

Figure 3.157 – System Details – Input Mapping Summary

3.13.5. OUTPUT MAPPING SUMMARY

The ***Output Mapping Summary*** tab displays all the data points that are being transferred to the FF Link Master from the Primary Interface. (Output Data).

These points would typically include the Master Control, Device Control, Master Exchange-Consumer connectors and Write Asynchronous Parameter.



Connection	Offset	Destination	Instance	Tagname	Size (bytes)
0	0	FF Master	Master	Master Control	72
0	72	FF Master	Master	MPV1	5
0	136	FF Device	MyCerabarS	Device Control	4
0	208	FF Device	MyProwirl73	Device Control	4
0	280	FF Device	MySRD991	Device Control	4

Figure 3.158 – System Details – Output Mapping Summary

3.13.6. DD SERVICES

The DD Services tab displays internal diagnostics information pertaining to the embedded DD Services. This information can be used by software engineers to determine the source of user system related issues.

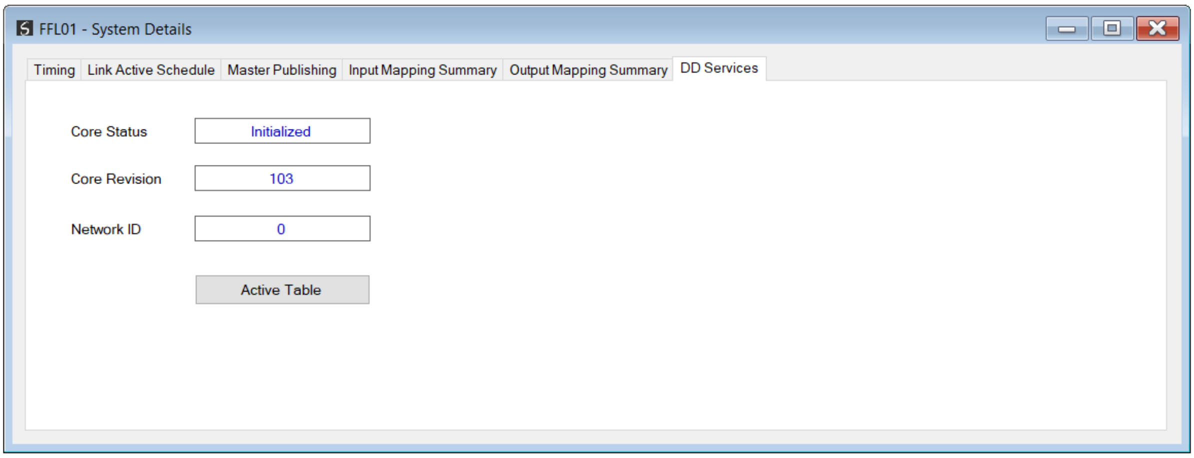


Figure 3.159 – System Details – DD Services

3.14. MODULE DOWNLOAD

Once the FF Link configuration has been completed, it must be downloaded to the module. The configured IP address of the module will be used to connect to the module.

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

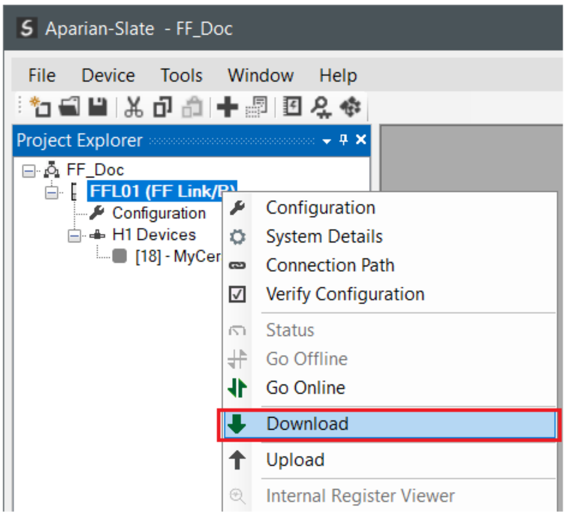


Figure 3.160 - Selecting Download

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

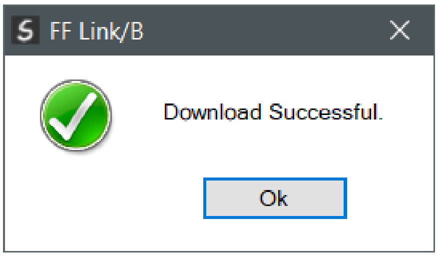


Figure 3.161 - Successful download

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

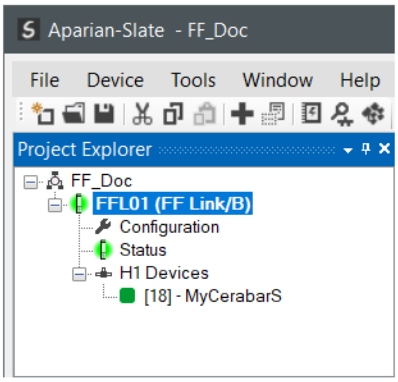


Figure 3.162 - Module online

3.15. LIVE LIST (ONLINE)

Once online with the FF Link in Slate, the user will be able to see all the devices on the H1 network.

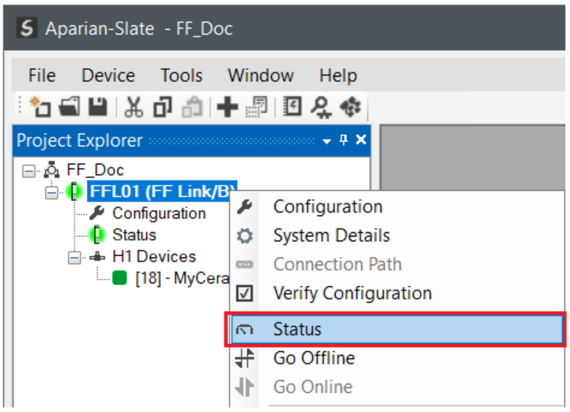


Figure 3.163 – Open Master Status

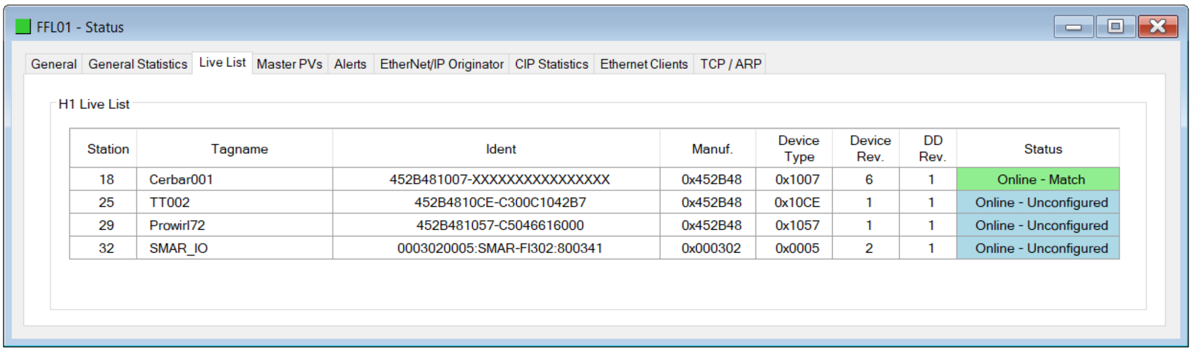


Figure 3.164 – Live List

If a device has been found that is not currently in the FF Link configured device list the user will be able to add the device from this window by right-clicking on the device and selecting ***Add Device***.

NOTE: The FF DD files will need to be already registered before a device can be added to the FF Link configuration.

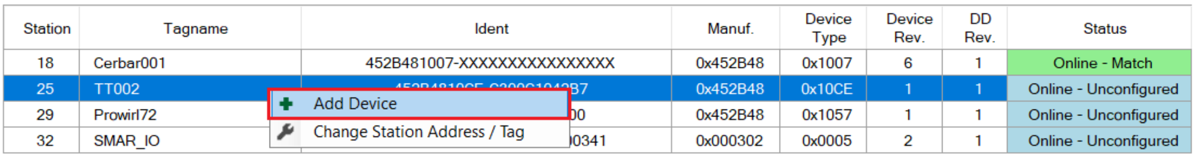


Figure 3.165 – Adding the Field Devices Found

3.15.1. Device Station Address / Tag Change

The user can change the station address and the Tag of a device by right-clicking on the device in the ***List List*** and selecting ***Change Station Address / Tag***.

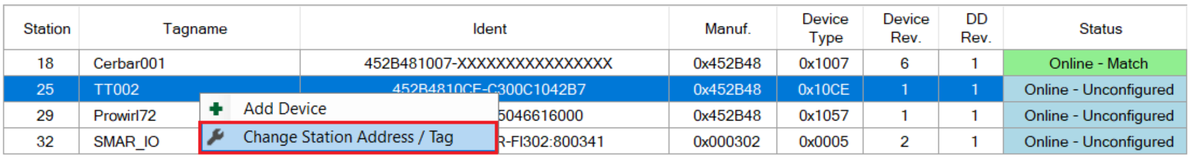


Figure 3.166 – Changing Station Address and/or Tag

Next, the user will need to select the ***New Station Address*** and/or ***PD Tagname*** for the device. Once selected press the ***Set*** button.

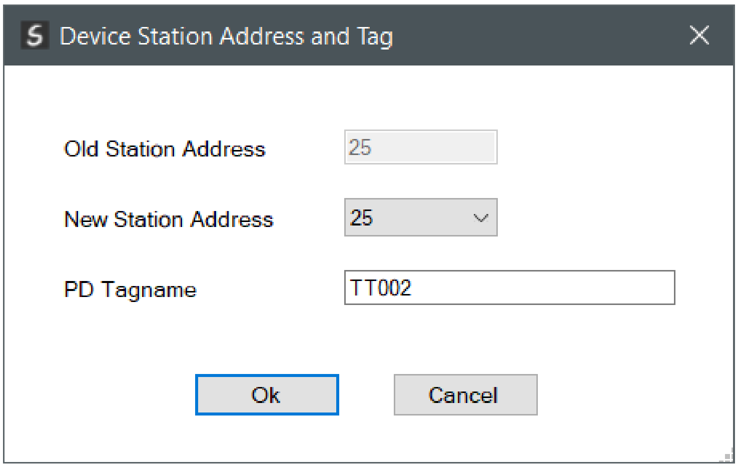


Figure 3.167 – Selecting new Station Address.

Once the request has been sent the user must monitor the ***LiveList*** to confirm that the H1 device Tag and/or Station Address has changed.

NOTE: The amount of time for the device to appear at the new station address is device type dependant. In the ***LiveList*** there will be a period where both node addresses may show up while the original station address is timing out.

NOTE: If the user sets the station address to an address that is already present on the H1 network it could result in communication failure of both devices.

NOTE: Generally, the device will need to be in the correct state before it will accept a command to change its station address (i.e. the device must not be in data exchange state).

4. OPERATION

4.1. ETHERNET/IP TARGET (LOGIX)

When the FF Link has been configured for Logix communication (by setting the ***Primary Interface*** to ***EtherNet/IP***), it must be added in Studio 5000 under an Ethernet bridge in the IO tree. The controller (e.g.: ControlLogix or CompactLogix) will then establish one or more Class 1 connections to the FF Link module and will start exchanging data.

NOTE: When using EtherNet/IP Target, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

4.1.1. ETHERNET/IP CLASS 1 ASSEMBLY MAPPING

When the module operates in a Logix “owned” mode the Logix controller will establish a class 1 cyclic communication connection to the FF Link. Up to four input and output assemblies are exchanged at a fix interval (RPI).

NOTE: The module input and output assembly of each connection will be an undecorated array of data. The imported Logix routine (generated by Slate) will copy this data to the input and output assemblies.

Once the generate L5X file has been imported (which will match the Internal Mapping in the configuration), the user will be able to use the tags generated for the specific FF Link. The data of the various tags (Master Status, Device PVs, etc.) will be in the format as shown in section 4.6.

Name	Value	Style	Data Type	Description
FFL01In	{...}		FFL01Input	
FFL01In.ConnectionOk	1	Decimal	BOOL	Connection Ok (0=Fault, 1=Ok)
FFL01In.Master	{...}		FFL01MasterInput	
FFL01In.Master.Status	{...}		FFLinkMasterStatus	
FFL01In.Master.Status.ConfigValid	1	Decimal	BOOL	Configuration Valid
FFL01In.Master.Status.DuplicateNode	0	Decimal	BOOL	Duplicate Node Detected
FFL01In.Master.Status.ActiveLAS	1	Decimal	BOOL	Active LAS
FFL01In.Master.Status.EIPOwned	1	Decimal	BOOL	Class 1 Ownership: 0=Not Owned, 1=Owned
FFL01In.Master.Status.PLCModeRun	1	Decimal	BOOL	Mode: 0=Prog 1=RUN
FFL01In.Master.Status.RedundancyEnabled	1	Decimal	BOOL	Redundancy Config: 0=Standalone, 1=Redundant
FFL01In.Master.Status.RedundancyStatus	1	Decimal	BOOL	Redundancy Status: 0=Standby, 1=Active
FFL01In.Master.Status.PowerConditionerActive	1	Decimal	BOOL	Internal Power Conditioner Active
FFL01In.Master.Status.TerminatorActive	0	Decimal	BOOL	Internal Terminator Active
FFL01In.Master.Status.PortTripped	0	Decimal	BOOL	H1 Bus Port Tripped
FFL01In.Master.Status.PortProtection	0	Decimal	BOOL	H1 Bus Port Protection
FFL01In.Master.Status.RedundancyConfigMismatch	0	Decimal	BOOL	Redundancy Config: 0=Ok, 1=Mismatch
FFL01In.Master.Status.BusVoltage	21.072735	Float	REAL	H1 Bus Voltage (V)
FFL01In.Master.Status.BusCurrent	87.82789	Float	REAL	H1 Bus Current (mA)
FFL01In.Master.Status.CPUTemperature	44.596565	Float	REAL	CPU Temperature
FFL01In.Master.Status.ConfigCRC	16#dab7	Hex	INT	Configuration Checksum
FFL01In.Master.Status.ActiveNodeCount	10	Decimal	SINT	Number of Active Slave Devices
FFL01In.Master.Status.SwitchOverTimeOut	1200	Decimal	INT	Redundancy Switch-Over Timeout
FFL01In.Master.Status.AlertPendingTotalCount	0	Decimal	INT	Number of Pending Alerts
FFL01In.Master.Status.DeviceLiveList	{...}	Decimal	BOOL[256]	Live List (0=Not Live, 1=Live)
FFL01In.Master.Status.DeviceDataExchangeActive	{...}	Decimal	BOOL[256]	Data Exchange Active (0=Inactive, 1=Active)
FFL01In.Master.Status.MasterPVVCRMismatch	{...}	Decimal	BOOL[64]	Master PV VCR Mismatch (0=Ok, 1=Mismatch)

Figure 4.1 – Logix Master Status Tag

Name	Value	Style	Data Type	Description
FFL01MyTMT85In	{...}		FFL01452B4810CEIX47E9	
FFL01MyTMT85In.Status	{...}		FFLinkDeviceStatus	
FFL01MyTMT85In.Status.Online	1	Decimal	BOOL	Device Online (0=Offline, 1=Online)
FFL01MyTMT85In.Status.DataExchangeActive	1	Decimal	BOOL	Data Exchange Active (0=Inactive, 1=Active)
FFL01MyTMT85In.Status.IdentMismatch	0	Decimal	BOOL	Device Identity Mismatch (0=Ok, 1=Mismatch)
FFL01MyTMT85In.Status.DeviceError	0	Decimal	BOOL	FF Device Error (0=Ok, 1=Error)
FFL01MyTMT85In.Status.SytemUpdateBusy	0	Decimal	BOOL	FFLink Busy Updating System Information (0=Done, 1...
FFL01MyTMT85In.Status.APUpdateBusy	0	Decimal	BOOL	FFLink Busy Updating Application Process Informatio...
FFL01MyTMT85In.Status.MapChecksumMismatch	0	Decimal	BOOL	Device Mapping Checksum Mismatch(0 = Ok, 1 = Mis...
FFL01MyTMT85In.Status.Station	25	Decimal	INT	Station Address
FFL01MyTMT85In.PV1	{...}		FFLinkInputPVReal	
FFL01MyTMT85In.PV1.Status	{...}		FFLinkInputPVStatus	
FFL01MyTMT85In.PV1.Value	25.532288	Float	REAL	PV Value
FFL01MyTMT85Out	{...}		FFL01452B4810CEOX47E9	
FFL01MyTMT85Out.Control	{...}		FFLinkDeviceControl	
FFL01MyTMT85Out.Control.MapCRC	16#47e9	Hex	INT	Mapping Checksum
FFL01MyTMT85Out.Control.Station	25	Decimal	INT	Station Address

Figure 4.2 – Logix Device Status Tag

4.2. ETHERNET/IP ORIGINATOR

The FF Link module can operate as an EtherNet/IP originator. In this mode the module can exchange H1 Master and H1 Device data with EtherNet/IP devices using either the input and output assemblies of the Class 1 EtherNet/IP connection to the device, or using an explicit (Class 3 or UCMM) EtherNet/IP message to read or write data.

4.2.1. ETHERNET/IP CLASS 1 CONNECTIONS

Once the EtherNet/IP Class 1 connections are configured and established, the received H1 device data can be written to the output assembly of selected EtherNet/IP device (Originator to Target) and the input assembly of the selected EtherNet/IP device can be written to H1 devices.

4.2.1.1. DATA RECEIVED FROM H1 DEVICE

In the configuration the user will specify the offset where the data must be in the assembly.

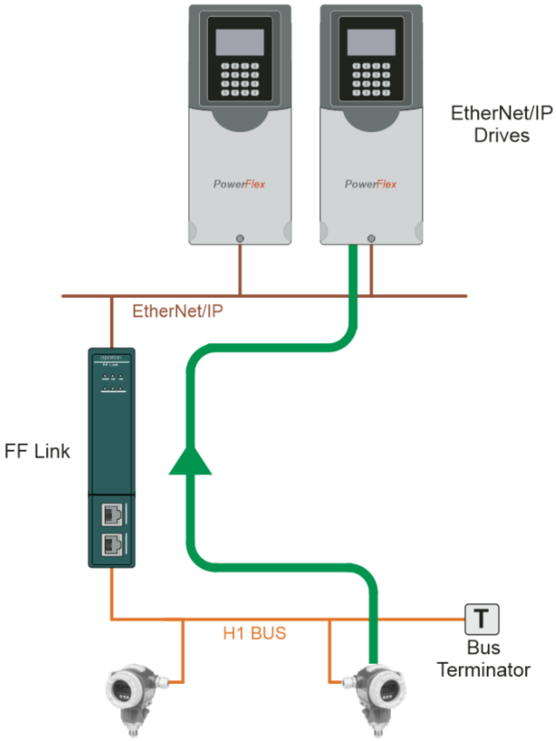


Figure 4.3 – H1 Device data to EtherNet/IP Device

The FF Link H1 Master will be configured to receive data from the H1 device by adding a producer PV connector (PV1).

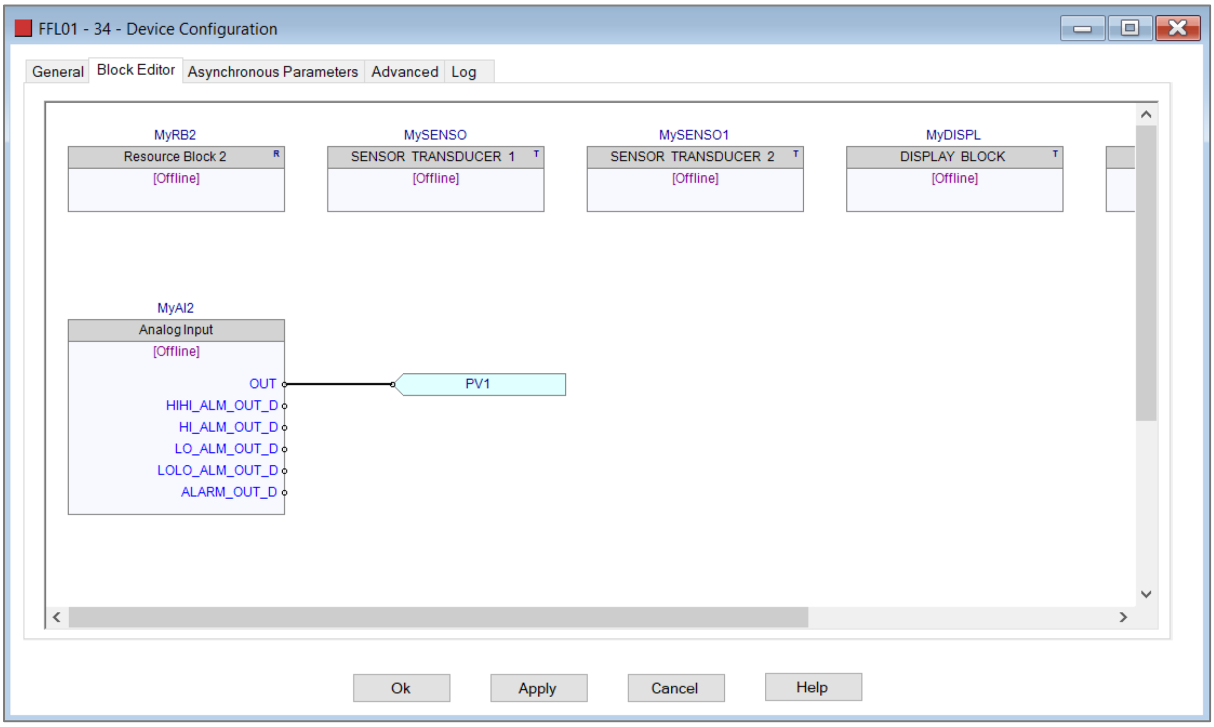


Figure 4.4 – H1 Device with producer connector

In the internal map the Process variable from the H1 Device (MyTMT85 – PV1) is mapped to the PF755 EtherNet/IP device output assembly at offset 4 (as shown below).

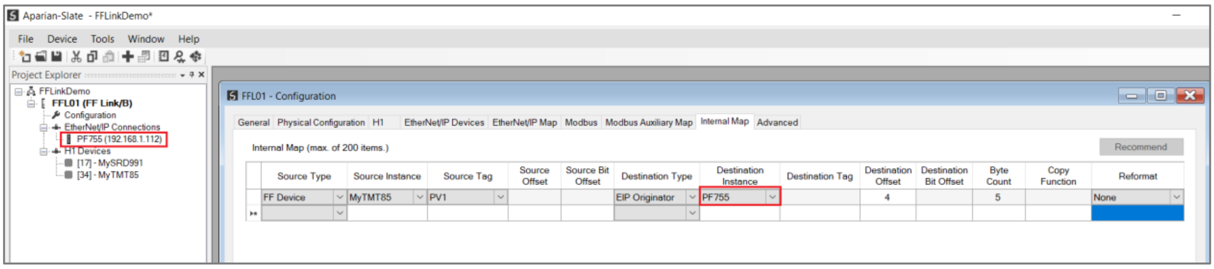


Figure 4.5 – Process variable (PV1) from H1 Device to EtherNet/IP Device

4.2.1.2. DATA SENT TO H1 DEVICE

In the configuration the user will specify the offset where the data resides in the (input) assembly.

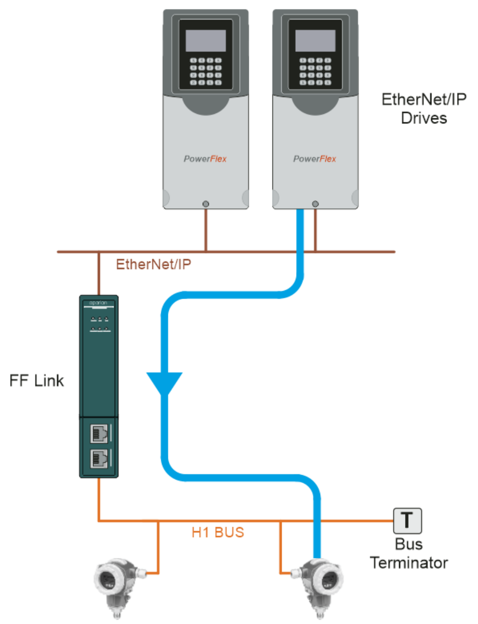


Figure 4.6 – H1 Device data from EtherNet/IP Device

The FF Link will be configured to read data from the EtherNet/IP Device Input Assembly (at a configured offset) and write the specific data to the FF Link H1 Master process variable (MPV1) that will send the data to the H1 Device.

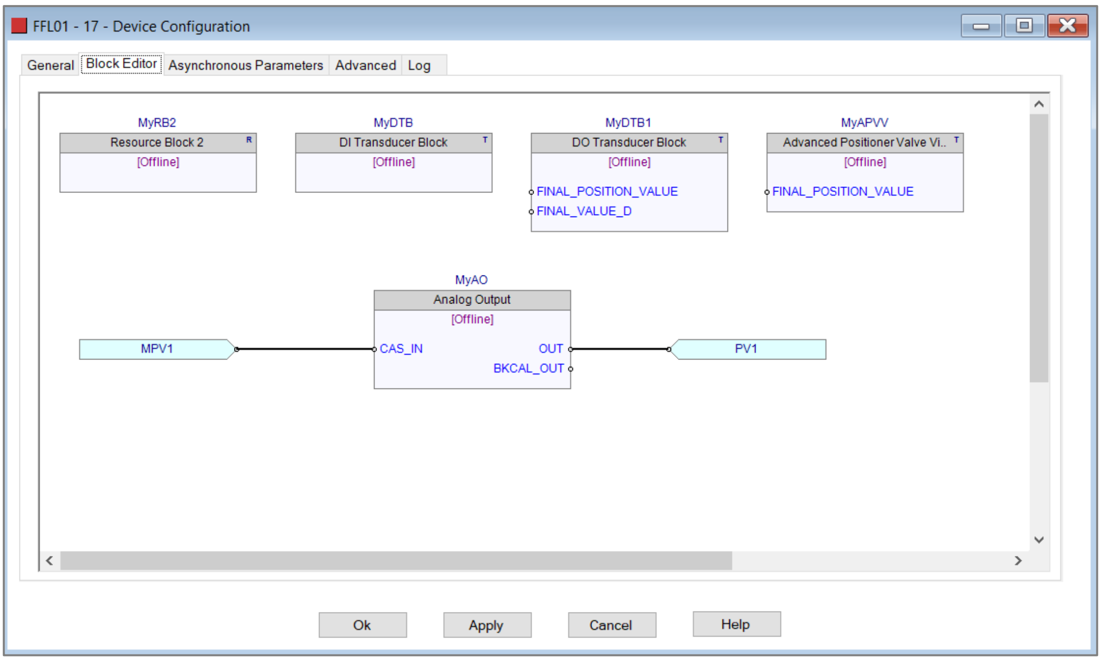


Figure 4.7 – H1 Device with consumer connector

In the internal map the input assembly of PF755 (offset 10) is mapped to the process variable of the H1 Device (MyTMT85 – PV1).

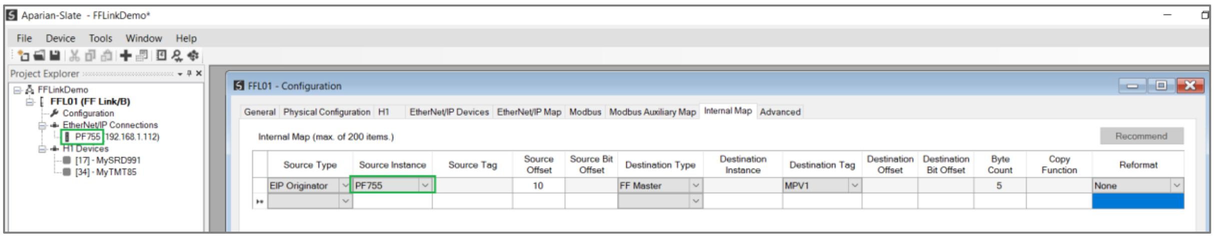


Figure 4.8 – Data from EIP Input Assembly to process variable (PV1) of H1 Device

4.2.2. Explicit Messaging

When using the EtherNet/IP Explicit Messaging, the user can configure up to ten EtherNet/IP devices which will be used for the Explicit Messaging. This is configuration is located in the *EtherNet/IP Devices* tab. Following this, the EtherNet/IP Map of explicit messages needs to be configured. The Explicit Messaging has 100KB of Internal Data Space (IDS) where data can be stored for exchanges between the explicit EtherNet/IP devices and the H1 Master and H1 Device data.

4.2.2.1. DATA RECEIVED FROM H1 DEVICE

When receiving H1 Device data, it must be copied to a specific address in the IDS which can then be written to the EtherNet/IP device using explicit messaging with the data source the offset in the IDS where the H1 Device data was stored.

In the example below the H1 Device is producing data which is stored in the Internal Data Space (IDS) offset 2020 using the internal mapping.

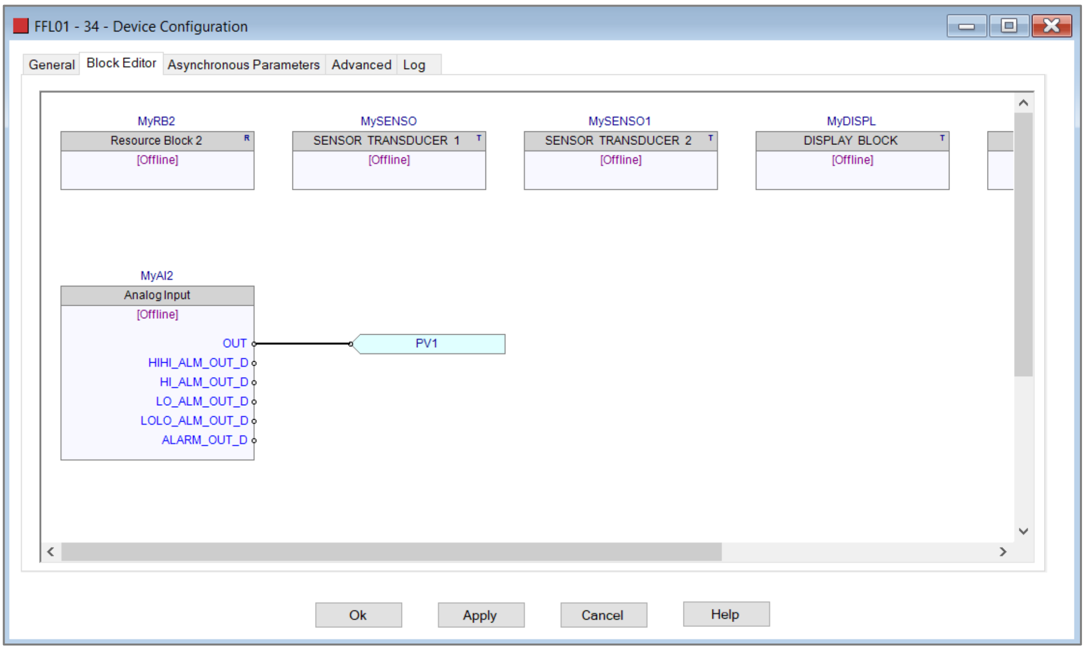


Figure 4.9 – H1 Device with producer connector

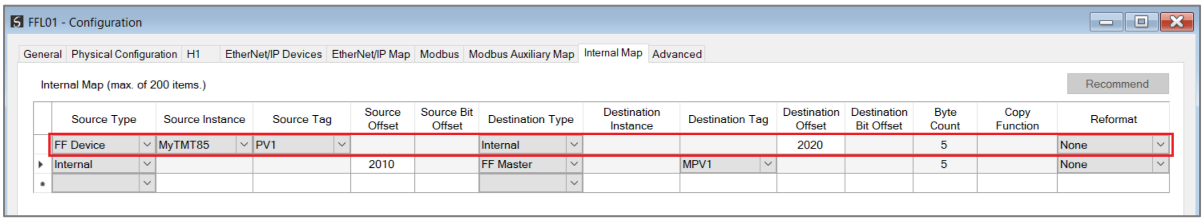


Figure 4.10 – IDS Offset for H1 Device Data

In the EtherNet/IP Map data is being written (using a Set function) with data from ***Output Offset*** 2020 where the H1 device data was stored in the internal mapping.

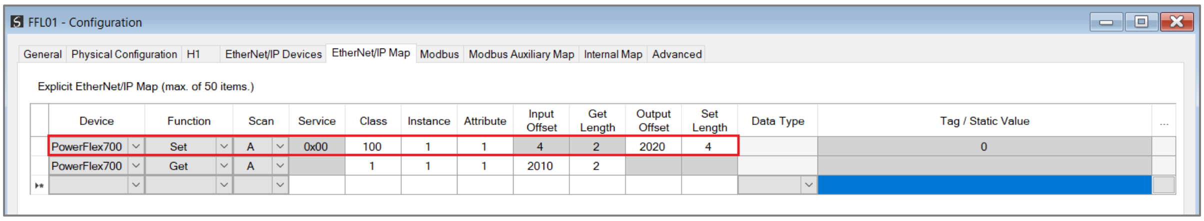


Figure 4.11 – EIP Table Interface Offset for H1 Device Data

4.2.2.1. DATA SENT TO H1 DEVICE

When reading data from the EtherNet/IP device (for example using a Get function in the EtherNet/IP Mapping) the data is stored at a configured offset in the IDS mapping table. The offset is configured by configuring the interface ***Input Offset*** of 2010 in the EtherNet/IP Map.

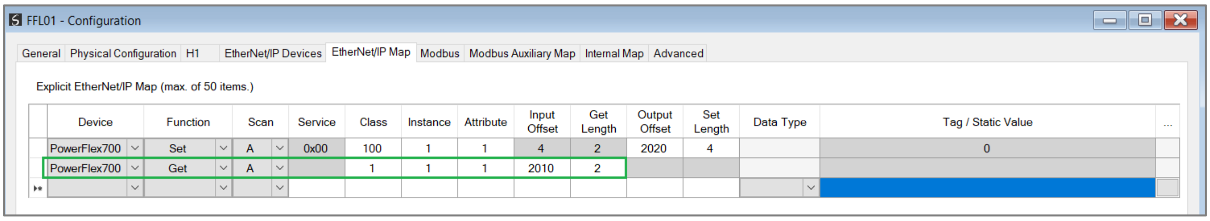


Figure 4.12 – EtherNet/IP Map Input Offset for IDS

The data stored at IDS offset 2010 from the EtherNet/IP device can then be sent to an H1 Device using the internal mapping.

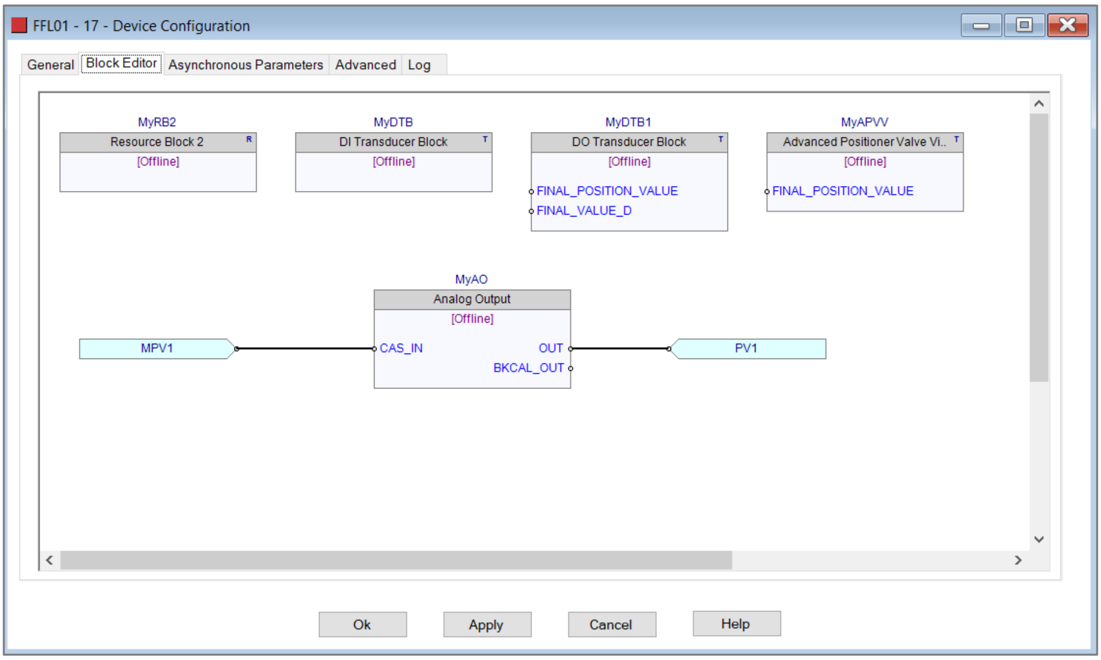


Figure 4.13 – H1 Device with consumer connector

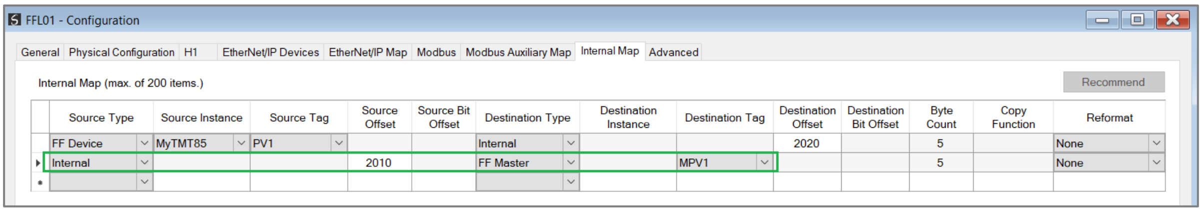


Figure 4.14 – Internal mapping from IDS to H1 Master data

4.3. MODBUS MASTER

When the FF Link module's ***Primary Interface*** is set to ***Modbus TCP Master***, then the FF Master and Device data can be mapped to and from configurable internal Modbus Registers and offsets.

NOTE: When using Modbus TCP Master, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

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

4.3.1. DATA SENT TO H1 DEVICE

In the example below, the FF Link with the ***Primary Interface*** set to ***Modbus TCP Master*** will read Modbus Holding Registers from a Modbus Slave device and then transmit that data to an H1 device.

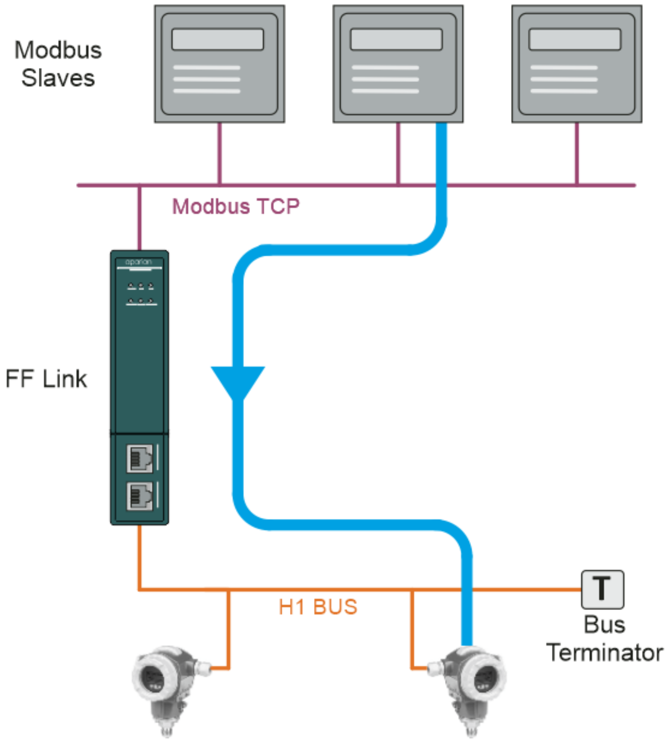


Figure 4.15 – Modbus Master to H1 Device operation

The H1 device will be configured to receive data from the FF Link H1 Master by adding a consumer PV connector (MPV1).

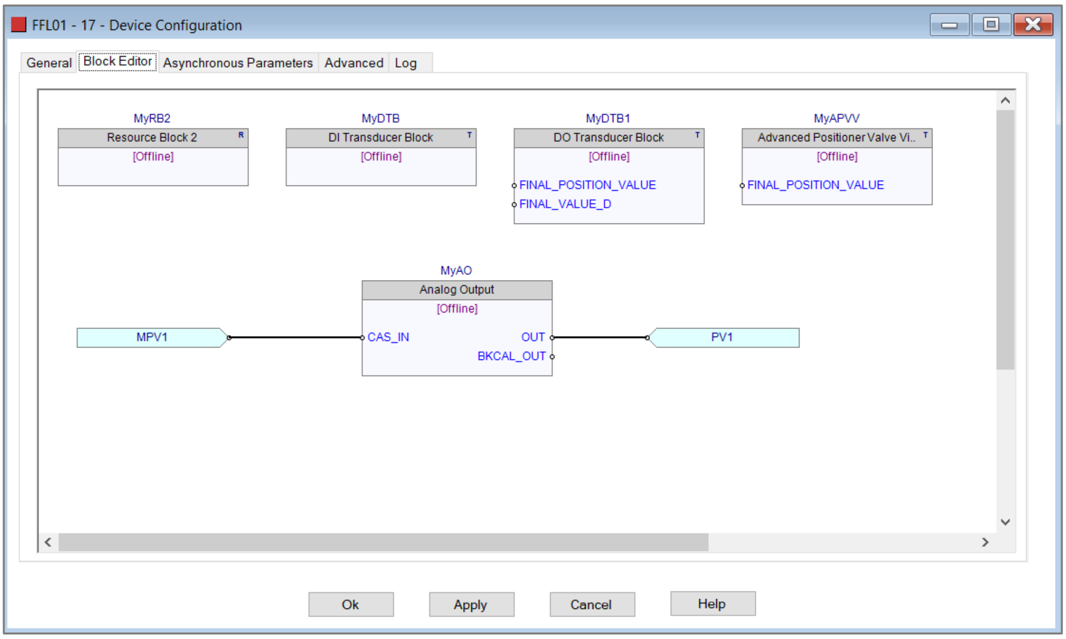
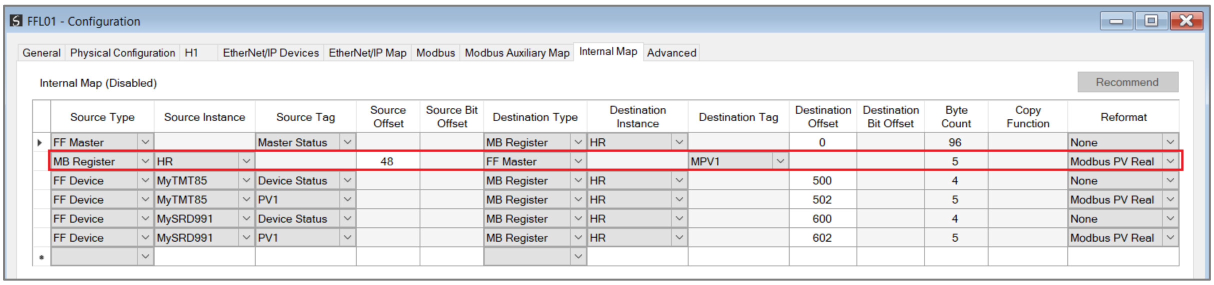


Figure 4.16 – H1 Device with consumer connector

In the internal map (which was automatically compiled with the *Auto Target Mapping* set in the configuration) Modbus Holding Register 48 is mapped to the FF Master MPV1 (as shown below). Note that the data was reformatted to *Modbus PV Real*. See section 3.5.4 for more information regarding the formatting.


FFL01 - Configuration

General | Physical Configuration | H1 | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Advanced

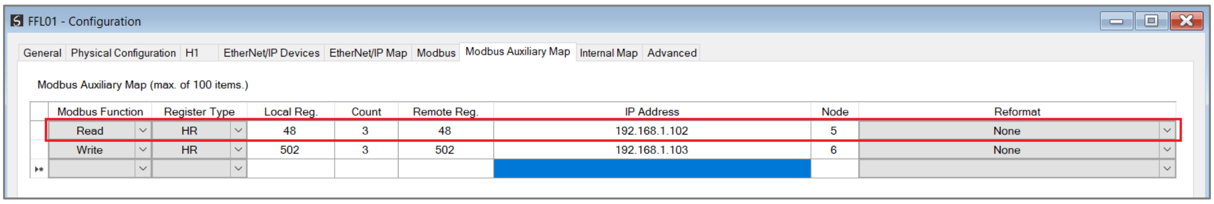
Internal Map (Disabled)

Recommend

Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			MB Register	HR		0		96		None
MB Register	HR		48		FF Master		MPV1			5		Modbus PV Real
FF Device	MyTMT85	Device Status			MB Register	HR		500		4		None
FF Device	MyTMT85	PV1			MB Register	HR		502		5		Modbus PV Real
FF Device	MySRD991	Device Status			MB Register	HR		600		4		None
FF Device	MySRD991	PV1			MB Register	HR		602		5		Modbus PV Real

Figure 4.17 – Internal Mapping for H1 Master

The user will then need to configure the ***Modbus Auxiliary Map*** to read the Modbus data from the Modbus Slave at the specific Modbus Holding Registers.


FFL01 - Configuration

General | Physical Configuration | H1 | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Advanced

Modbus Auxiliary Map (max. of 100 items.)

Modbus Function	Register Type	Local Reg.	Count	Remote Reg.	IP Address	Node	Reformat
Read	HR	48	3	48	192.168.1.102	5	None
Write	HR	502	3	502	192.168.1.103	6	None

Figure 4.18 – FF Link Modbus Master Auxiliary Reading

4.3.2. Data received from H1 Device

In the example below, the FF Link with the ***Primary Interface*** set to ***Modbus TCP Master*** will write H1 data received from an H1 device to Modbus Holding Registers in a Modbus Slave device.

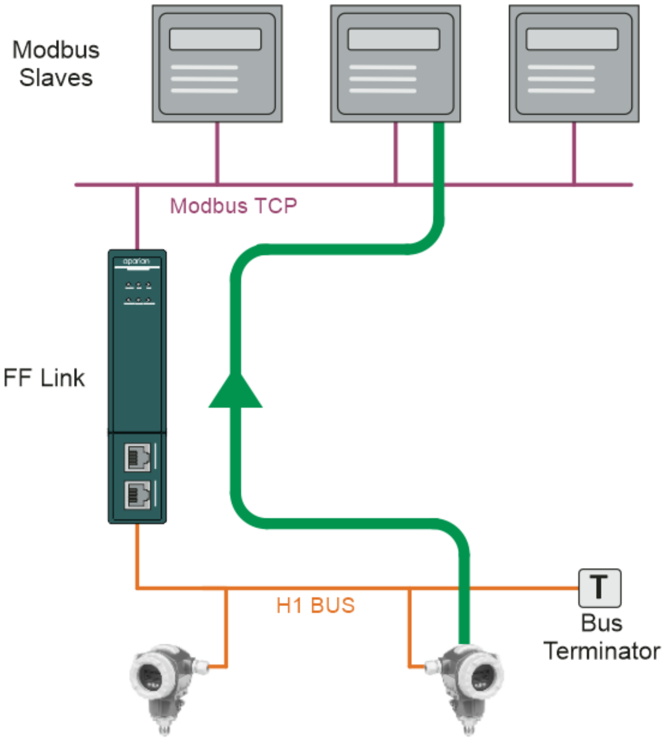


Figure 4.19 – H1 Device to Modbus Master operation

The FF Link H1 Master will be configured to receive data from the H1 device by adding a producer PV connector (PV1).

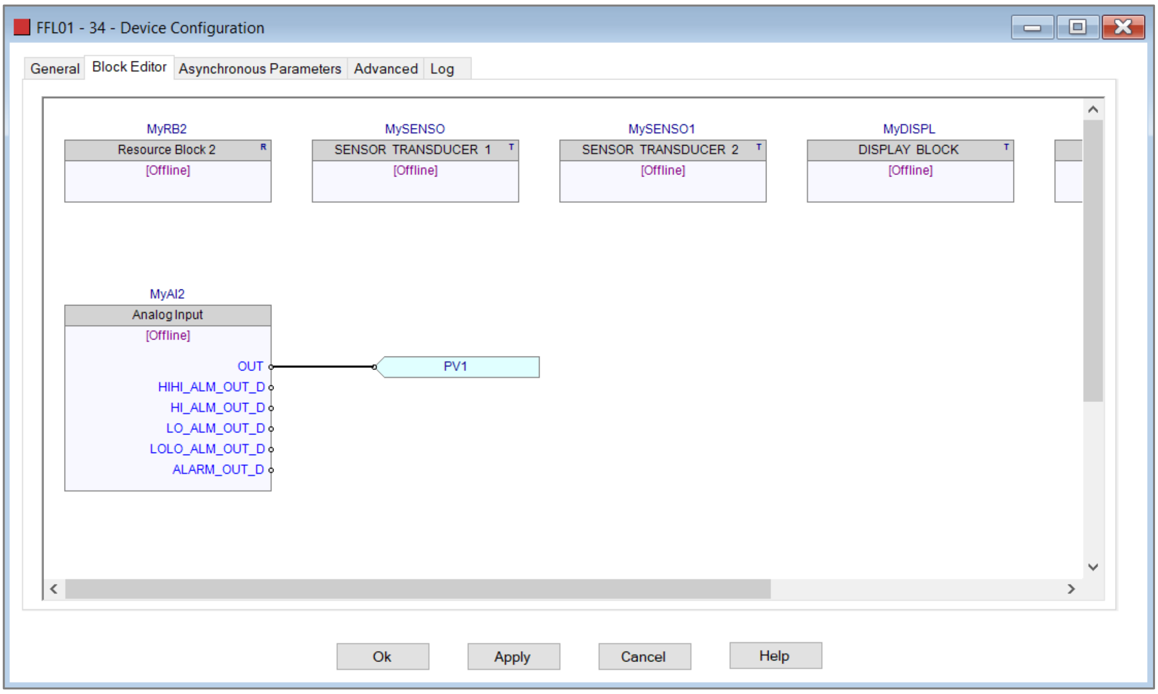


Figure 4.20 – H1 Device with producer connector

In the internal map (which was automatically compiled with the *Auto Target Mapping* set in the configuration) the Process variable from the H1 Device (MyTMT85 – PV1) is mapped to Modbus Holding Register 502 (as shown below). Note that the data was reformatted to *Modbus PV Real*. See section 3.5.4 for more information regarding the formatting.

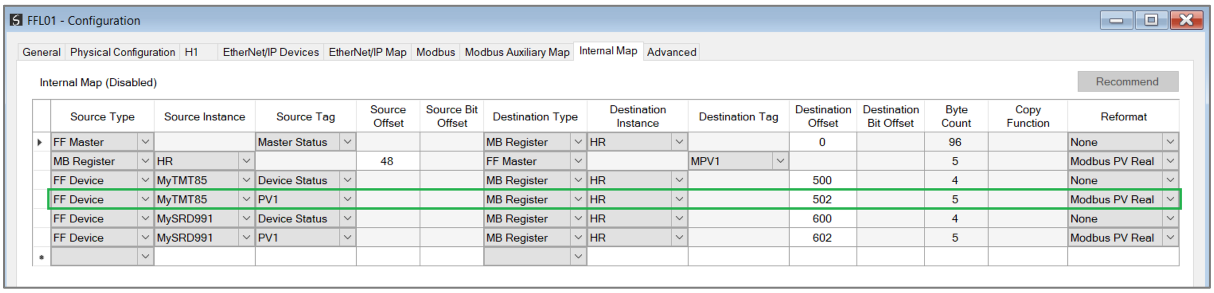


Figure 4.21 – Internal Mapping for H1 Device

The user will then need to configure the ***Modbus Auxiliary Map*** to write the Modbus data to the Modbus Slave at the specific Modbus Holding Registers.

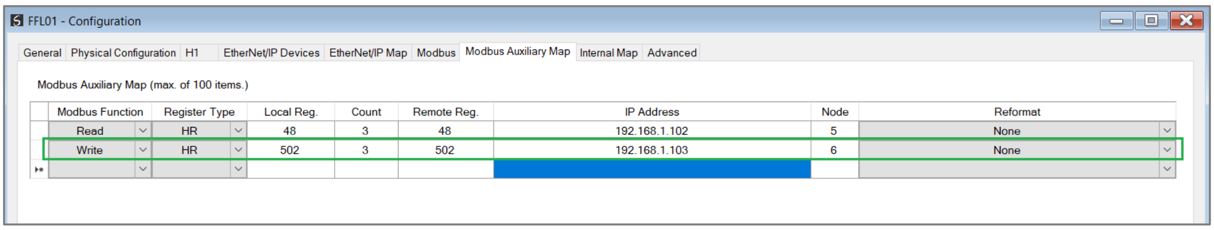


Figure 4.22 – FF Link Modbus Master Auxiliary Writing

A summary of the utilized Modbus Registers can be viewed by right-clicking on the module and selecting the ***Export Modbus Summary CSV*** option. This produces a CSV (comma-separated-variable) file showing the Modbus registers used.

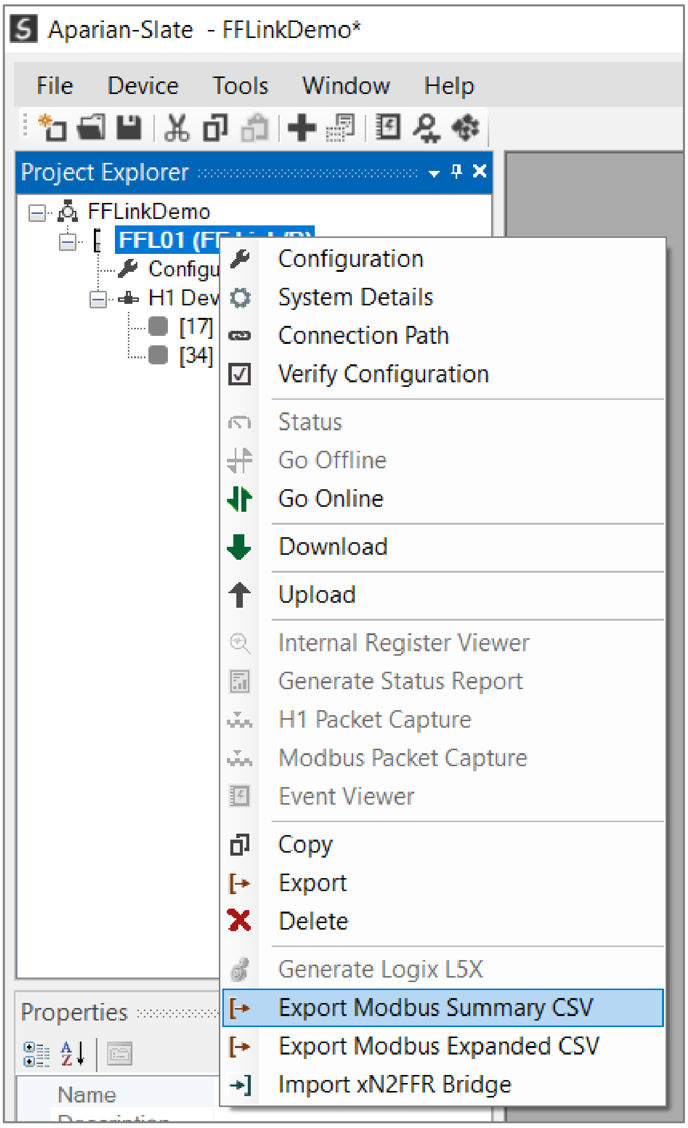


Figure 4.23 – Export Modbus Summary Report

4.4. MODBUS SLAVE

When the FF Link module's ***Primary Interface*** is set to ***Modbus TCP Slave***, then the FF Master and Device data can be mapped to and from configurable internal Modbus Registers and offsets.

NOTE: When using Modbus TCP Slave, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

The internal Modbus Registers can then be asynchronously exchanged with a remote Modbus TCP Master. The remote Modbus Master can read or write to the configured Modbus addresses to access the H1 Master and Device data.

4.4.1. Data sent to H1 Device

In the example below the FF Link with the ***Primary Interface*** set to ***Modbus TCP Slave*** will read Internal Modbus Holding Registers (which need to be populated by an external Modbus TCP Master) and then transmit that data to an H1 device.

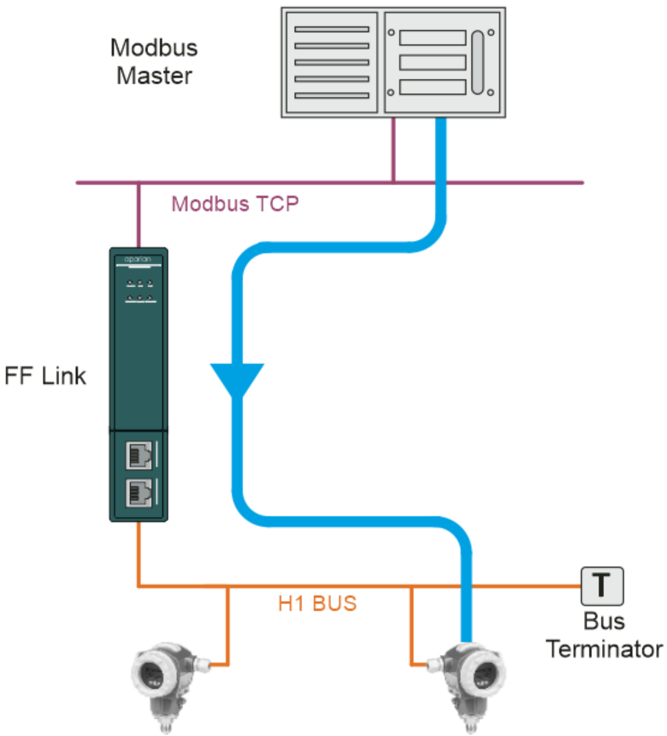


Figure 4.24 – Modbus Master to H1 Device operation

The H1 device will be configured to receive data from the FF Link H1 Master by adding a consumer PV connector (MPV1).

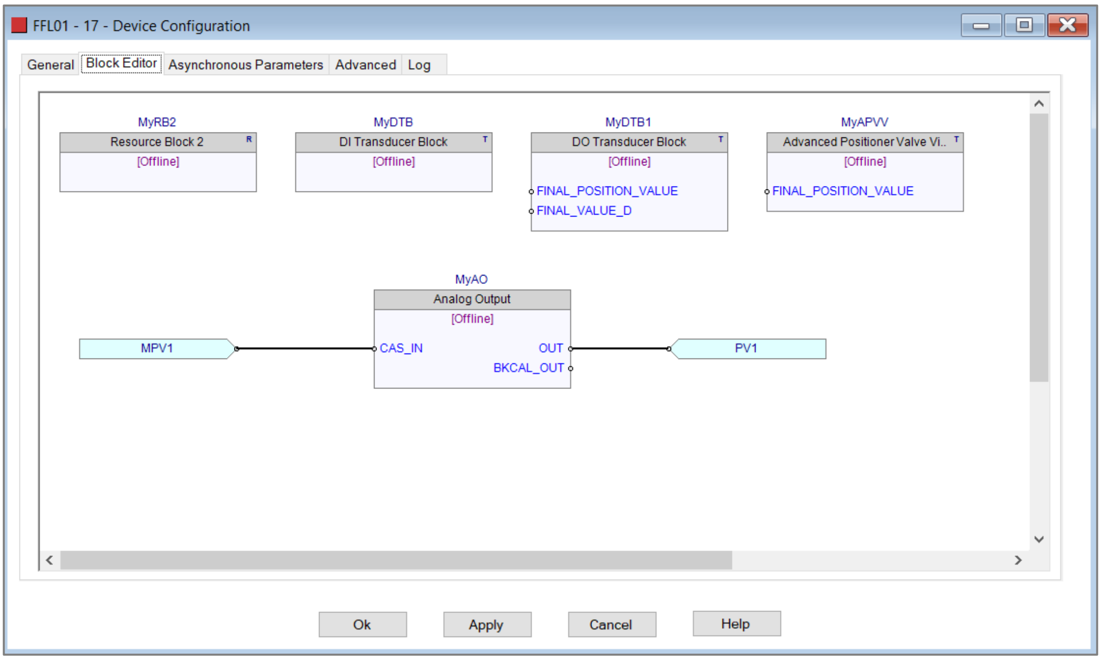
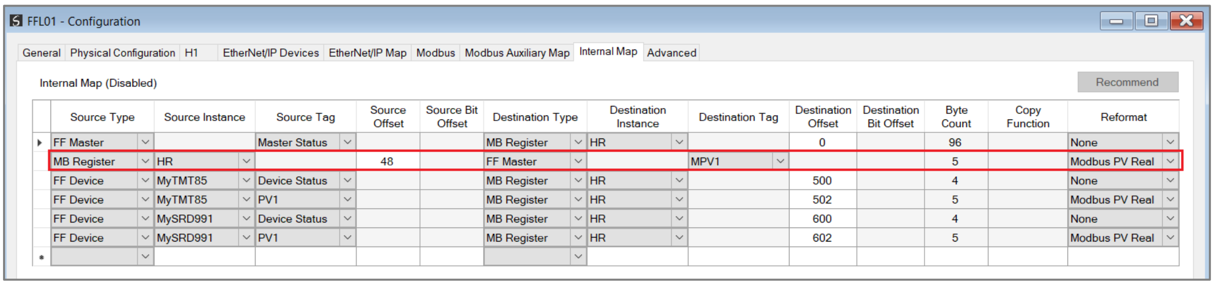


Figure 4.25 – H1 Device with consumer connector

In the internal map (which was automatically compiled with the *Auto Target Mapping* set in the configuration) Modbus Holding Register 48 is mapped to the FF Master MPV1 (as shown below). Note that the data was reformatted to *Modbus PV Real*. See section 3.5.4 for more information regarding the formatting.



FFL01 - Configuration

General | Physical Configuration | H1 | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Advanced

Internal Map (Disabled)

Recommend

Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			MB Register	HR		0		96		None
MB Register	HR		48		FF Master		MPV1			5		Modbus PV Real
FF Device	MyTMT85	Device Status			MB Register	HR		500		4		None
FF Device	MyTMT85	PV1			MB Register	HR		502		5		Modbus PV Real
FF Device	MySRD991	Device Status			MB Register	HR		600		4		None
FF Device	MySRD991	PV1			MB Register	HR		602		5		Modbus PV Real

Figure 4.26 – Internal Mapping for H1 Master

The external Modbus Master will need to write to Modbus Holding Register 48 of the FF Link to update the data that must be sent from the FF Link H1 Master to the H1 device.

4.4.2. DATA RECEIVED FROM H1 DEVICE

In the example below the FF Link with the ***Primary Interface*** set to ***Modbus TCP Slave*** will write H1 data received from an H1 device to Internal Modbus Holding Registers.

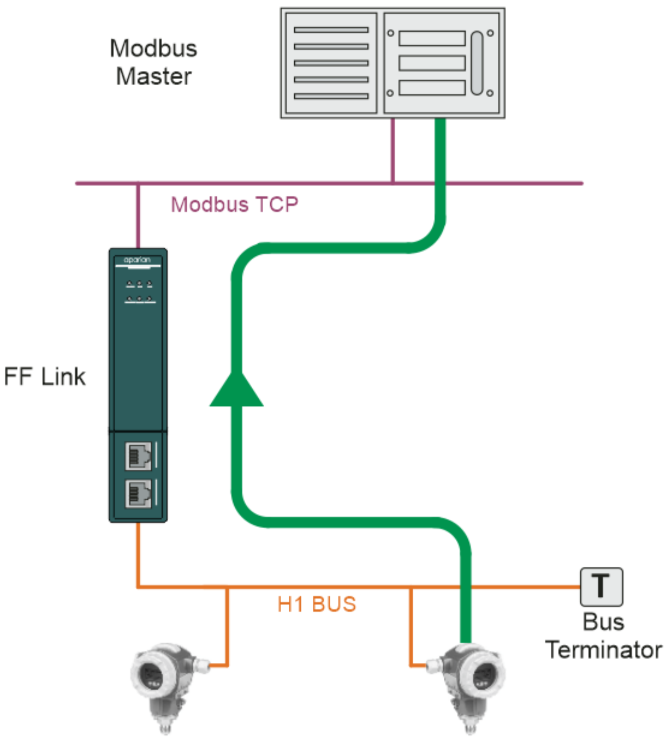


Figure 4.27 – H1 Device to Modbus Master operation

The FF Link H1 Master will be configured to receive data from the H1 device by adding a producer PV connector (PV1).

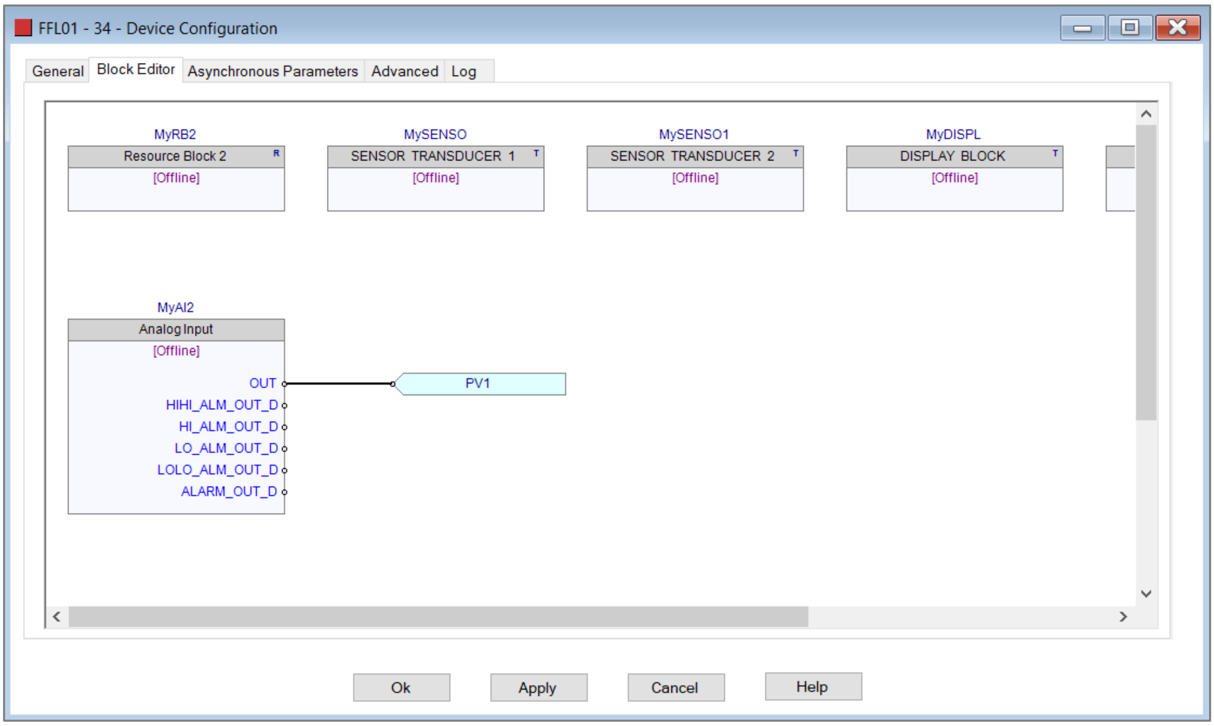
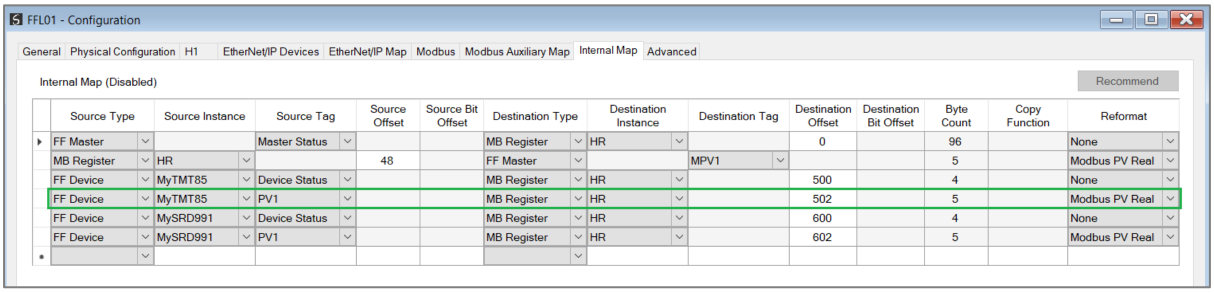


Figure 4.28 – H1 Device with producer connector

In the internal map (which was automatically compiled with the *Auto Target Mapping* set in the configuration) the Process variable from the H1 Device (MyTMT85 – PV1) is mapped to Modbus Holding Register 502 (as shown below). Note that the data was reformatted to *Modbus PV Real*. See section 3.5.4 for more information regarding the formatting.



Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			MB Register	HR		0		96		None
MB Register	HR		48		FF Master		MPV1			5		Modbus PV Real
FF Device	MyTMT85	Device Status			MB Register	HR		500		4		None
FF Device	MyTMT85	PV1			MB Register	HR		502		5		Modbus PV Real
FF Device	MySRD991	Device Status			MB Register	HR		600		4		None
FF Device	MySRD991	PV1			MB Register	HR		602		5		Modbus PV Real

Figure 4.29 – Internal Mapping for H1 Device

The external Modbus Master will need to read Modbus Holding Register 502 of the FF Link to receive the updated data that was sent from the H1 device to the FF Link H1 Master.

A summary of the utilized Modbus Registers can be viewed by right-clicking on the module and selecting the ***Export Modbus Summary CSV*** option. This produces a CSV (comma-separated-variable) file showing the Modbus registers used.

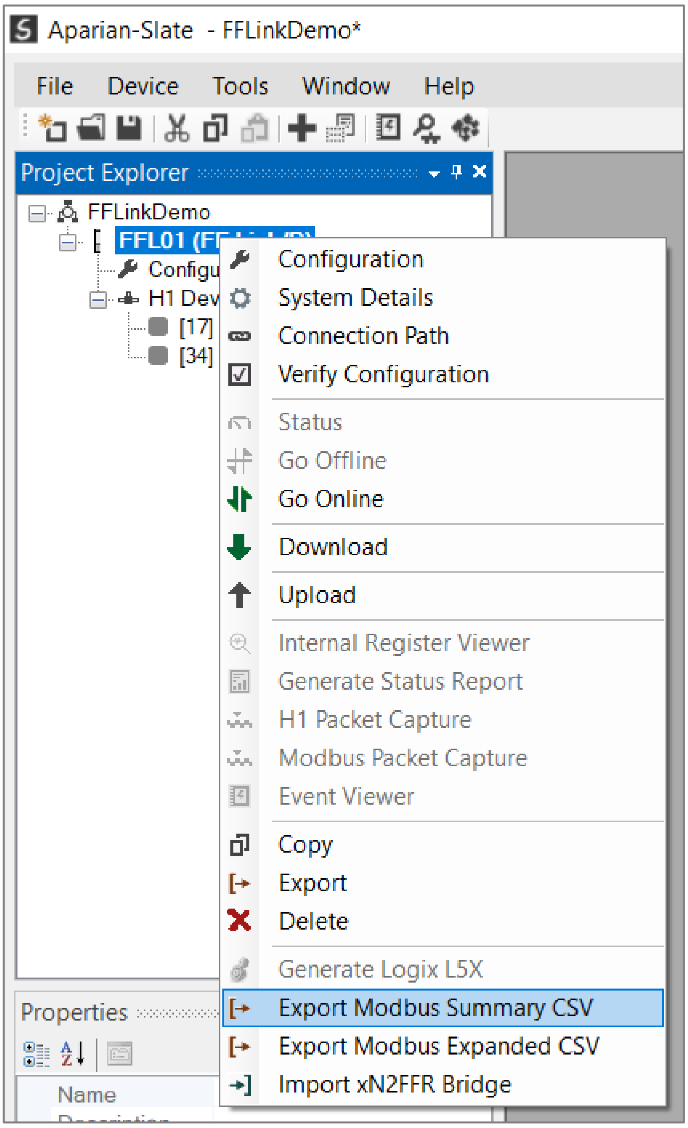
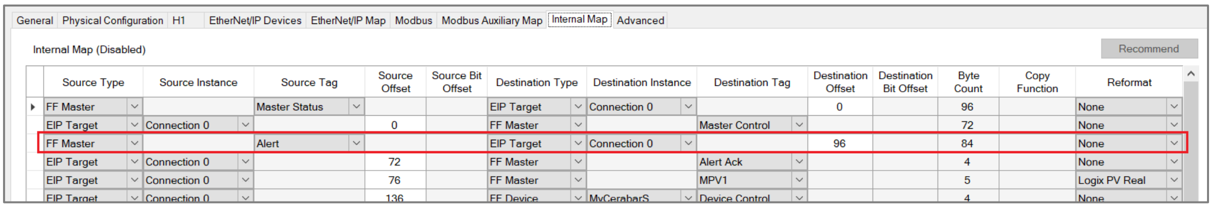


Figure 4.30 – Export Modbus Summary Report

4.5. H1 ALERT (ALARM) MANAGEMENT

The FF Link can buffer up to 255 H1 alerts. H1 alerts will be buffered in volatile memory which means if the module is power cycled with the events still buffered, then the events will be lost. It is therefore important for the primary interface to unload the H1 alerts from the FF Link module as soon as possible.

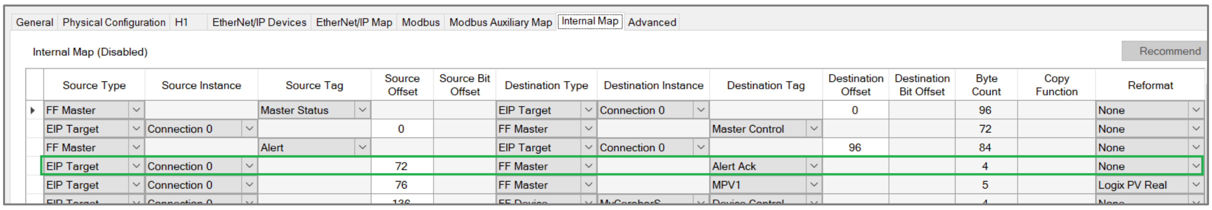
Events will be displayed in the format shown in section 4.6.1.3 and will be mapped to the primary interface using the Internal Mapping in the configuration. The number of buffered events will be shown in the FF Master Alert data.



Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			EIP Target	Connection 0		0		96		None
EIP Target	Connection 0		0		FF Master		Master Control			72		None
FF Master		Alert			EIP Target	Connection 0		96		84		None
EIP Target	Connection 0		72		FF Master		Alert Ack			4		None
EIP Target	Connection 0		76		FF Master		MPV1			5		Logix PV Real
EIP Target	Connection 0		136		FF Device	MyCerabarS	Device Control			4		None

Figure 4.31 – H1 Alert Data

Once the H1 Alert data has been read, the primary interface must write the ***Alert Index*** number to the ***FF Master Alert Ack*** using the Internal Mapping in the configuration. See section 4.6.1.6 for the format of the Alert Ack.



Source Type	Source Instance	Source Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Tag	Destination Offset	Destination Bit Offset	Byte Count	Copy Function	Reformat
FF Master		Master Status			EIP Target	Connection 0		0		96		None
EIP Target	Connection 0		0		FF Master		Master Control			72		None
FF Master		Alert			EIP Target	Connection 0		96		84		None
EIP Target	Connection 0		72		FF Master		Alert Ack			4		None
EIP Target	Connection 0		76		FF Master		MPV1			5		Logix PV Real

Figure 4.32 – H1 Alert Acknowledge

Once the Alert Ack – Alert Index is the same as the Alert – Alert Index, the FF Link will remove the H1 alert from the buffer and, if there are more events, present the next event in the FF Master – Alert mapping and decrease the buffer count by one.

4.5.1. EtherNet/IP Target (Logix)

When using EtherNet/IP Target as the primary interface, then Slate will automatically create the required Alert Extraction Logix AOI when generating the L5X file.

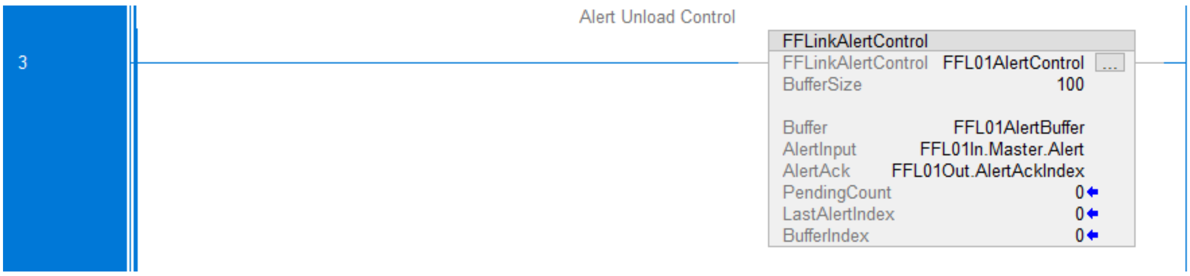


Figure 4.33 – EtherNet/IP - Alert Extraction

The Alert Extraction Add-On-Instruction (AOI) “FFLinkAlertControl”, controls the extraction of alert data from the FF Link’s internal buffer and populates a circular buffer in the Logix controller.

The circular buffer is an array of a specific Alert UDT (FFLinkAlertEvent). The array length is specified in the FF Link’s configuration ***Logix Alert Buffer Size*** parameter.

4.5.2. MODBUS

When using ***Modbus TCP Master*** or ***Modbus TCP Slave*** as the ***Primary Interface***, the Internal Mapping will allocate Modbus Holding Registers to the Alert and Alert Ack objects. It is the responsibility of the primary interface to ensure that Alerts are read from the Holding Registers mapped to the Alert Source and to write the Alert Index in the Alert Data to the Alert Ack Modbus Holding Registers.

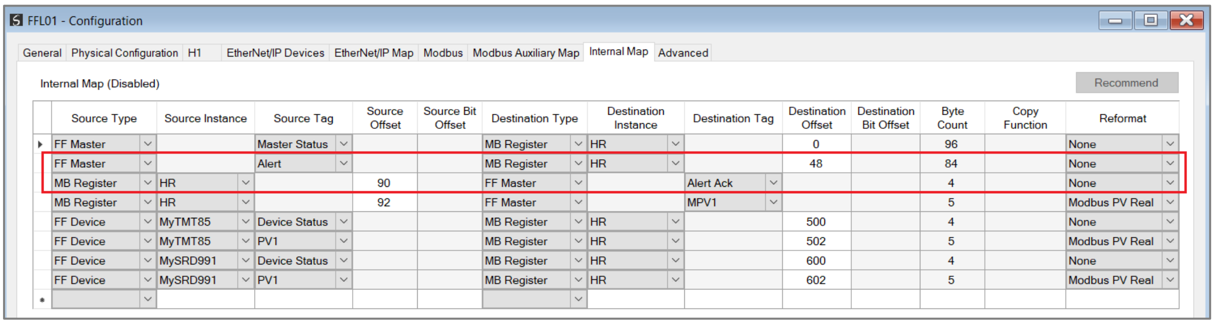


Figure 4.34 – Modbus TCP Master/Slave - Alert Internal Mapping

4.5.3. ETHERNET/IP ORIGINATOR

When using ***EtherNet/IP Originator*** as the ***Primary Interface***, the Internal Mapping will have to be manually configured by the user to ensure that the EtherNet/IP devices can access the Alert and Alert Ack data. It is then the responsibility of the primary interface device(s) to ensure that Alerts are read and to write the Alert Index in the Alert Data to the Alert Ack once the Alert data has been read.

4.6. INTERNAL MAP H1 DATA FORMATS

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

NOTE: When using EtherNet/IP Target, Modbus TCP Slave, or Modbus TCP Master, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map all the required data in the Internal Map.

4.6.1.1. FF MASTER - MASTER STATUS

Below is the format of the FF Master Status:

Parameter	Data Type	Byte Offset	Description
ConfigValid	BOOL	0.0	The configuration in the module is valid.
DuplicateNode	BOOL	0.1	Duplicate H1 Node detected that is the same as the FF Link H1 node.
ActiveLAS	BOOL	0.2	The local FF Link module is the FF LAS on the H1 network.
EIPOwned	BOOL	0.3	Indication that the module is owned by an EtherNet/IP originator when the module is operating as an EtherNet/IP target.
EIPOriginatorCommsOk	BOOL	0.4	EtherNet/IP communication is ok when the module is configured as an EtherNet/IP Originator.
MBCommsOK	BOOL	0.5	Modbus communication is ok when the module is configured as a Modbus TCP Master or Modbus TCP Slave.
PLCModeRun	BOOL	0.6	When operating as an EtherNet/IP Target, this shows the state of the Controller that owns the FF Link over EtherNet/IP.
RedundancyEnabled	BOOL	0.7	H1 Master Redundancy has been enabled for the FF Link.
RedundancyStatus	BOOL	1.0	The local FF Link is the active H1 Master on the H1 bus when running in H1 Master Redundancy mode.
PowerConditionerActive	BOOL	1.1	The internal power conditioner of the FF Link is active and will supply power to the H1 bus.
TerminatorActive	BOOL	1.2	The internal terminator (of the FF Link) for the H1 bus is active and will terminate the H1 bus.
PortTripped	BOOL	1.3	When the FF Link is supplying the power to the H1 bus, this will indicate if the H1 port has tripped on over-current.
PortProtection	BOOL	1.4	This is the H1 Bus port protection status.
Reserved	BOOL[19]	1.5 – 3.7	Reserved for future use.
BusVoltage	REAL	4	Bus Voltage (Volts)
BusCurrent	REAL	8	Bus Current supplied by the internal Power Conditioner (mA)
CPUTemperature	REAL	12	The internal temperature of the module’s CPU.
ConfigCRC	INT	16	The signature of the configuration currently executing on the module.
ActiveNodeCount	SINT	18	Number of H1 devices configured, online, and exchanging data.
Reserved	SINT	19	Reserved for future use.
SwitchOverTimeOut	INT	20	When operating as a Redundant H1 Master, this is the configured “dead” time for the H1 Master switch over.

AlertPendingTotalCount	INT	22	Number of H1 alarms that have been buffered but not yet unloaded by the primary interface.
DeviceLiveList	BOOL[256]	24	Indicates the nodes that are online on the local H1 network. Each bit represents a node. When the specific bit is set '1' then the device is online and when the bit is off '0' the device is not on the H1 network. Bit 0 – Node 0 Online Bit 1 – Node 1 Online ……… Bit 255 – Node 255 Online
DeviceDataExchangeActive	BOOL[256]	56	Indicates the nodes that are exchanging data on the local H1 network. Each bit represents a node. When the specific bit is set '1' then the device is exchanging data while when the bit is off '0' the device is not exchanging data on the H1 network. Bit 0 – Node 0 Exchanging Data Bit 1 – Node 1 Exchanging Data ……… Bit 255 – Node 255 Exchanging Data
MasterPVVCRMismatch	BOOL[64]	88	When the FF Link is producing data on the H1 network and the primary interface is EtherNet/IP Target, then this bit will indicate that the L5X mapping used in Logix matches the configuration.

Table 4.1 – FF Master Status Format

4.6.1.2. FF MASTER - ADVANCED STATUS

Below is the format of the FF Advanced Status:

Parameter	Data Type	Byte Offset	Description
EthPort1LinkUp	BOOL	0.0	The link status of Ethernet Port 1.
EthPort1Mirror	BOOL	0.1	Indicates that Port 1 has been configured for Port Mirroring.
EthPort2LinkUp	BOOL	0.2	The link status of Ethernet Port 2.
EthPort2Mirror	BOOL	0.3	Indicates that Port 2 has been configured for Port Mirroring.
DLREnabled	BOOL	0.4	Device Level Ring has been enabled.
DLRRingMode	BOOL	0.5	Indicates that the Device Level Ring is in Ring mode.
DLRRingFault	BOOL	0.6	Indicates a fault in the Device Level Ring.
NTPEnabled	BOOL	0.7	Network Time Protocol has been enabled.
NTPLocked	BOOL	0.8	Network Time Protocol service is locked to an NTP server.
FWRevMajor	SINT	4	Major module firmware revision.

FWRevMinor	SINT	5	Minor module firmware revision.
FWRevMicro	SINT	6	Micro module firmware revision.
DIPSwitch	SINT	7	Current DIP switch configuration.
ConfigCRC	INT	8	The signature of the configuration currently executing on the module.
MACAddress	SINT[6]	12	The module’s MAC address.
PortTripCounter	DINT	20	The number of times the H1 Port has tripped on over-current.
UpTime	DINT	24	The number of seconds since the module booted-up.
DTYear	INT	28	The modules internal Date Time - Year
DTMonth	SINT	30	The modules internal Date Time - Month
DTDay	SINT	31	The modules internal Date Time - Day
DTHour	SINT	32	The modules internal Date Time - Hour
DTMinute	SINT	33	The modules internal Date Time - Minute
DTSecond	SINT	34	The modules internal Date Time - Second

Table 4.2 – FF Advanced Status Format

4.6.1.3. FF MASTER - ALERT

NOTE: See the H1 Alert Management section 4.5 for details regarding H1 Alert unloading.

Below is the format of the FF Master Alert:

Parameter	**Data Type**	**Byte Offset**	**Description**
Pending Alert Count	INT	0	Number of alarms that have been buffered by the FF Link, but not unloaded by the primary interface.
Reserved	SINT[6]	2	Reserved for future use.
First event no unloaded			
TimeStamp	LINT	8	Timestamp of when the event occurred. The format is the number of microseconds since 1 Jan 1970.
StationNumber	INT	16	The H1 device node number that reported the alert
Alert Number	INT	18	Event notification number generated by the H1 device.
Alert Object Index	INT	20	The Object Index for the alert generated by the H1 device.
StandardType	SINT	22	The type of alarm: 0 – Undefined

			1 LowLimit 2 HighLimit 3 CriticalLowLimit 4 CriticalHighLimit 5 DeviationLow 6 DeviationHigh 7 Discrete 8 BlockAlarm 9 UpdateStaticData 10 WriteProtectChanged 11 UpdateLink 12 UpdateTrend 13 Ignore 14 Reset 15 FailAlarm 16 OffSpecAlarm 17 MaintAlarm 18 CheckAlarm 19 SIFErrorAlarm
AlertKey	SINT	23	The Alert Key from the block in the H1 device which generated the alert.
Alert Index	DINT	24	The Alert Index which is used to confirm the event (via the Alert Ack) has been unloaded. This will indicate to the FF Link that the event has been unloaded and the next buffered event can be displayed.
BlockTag	String32	28	The tag name of the block that generated the alert in the specific H1 device.
ManufactureType	SINT	64	The manufacture type of the alert generated by the H1 device.
MessageType	SINT	65	The message type of the alert generated by the H1 device.
Priority	SINT	66	The priority of the alert generated by the H1 device.
Reserved	SINT	67	Reserved for future use.
BlockIndex	INT	68	The index number of the block that generated the alert in specific H1 device.
RelativeIndex	INT	70	The index number of the parameter (relative to the block index) that generated the alert in specific H1 device.
UnitIndex	INT	72	The unit index of the alert generated by the H1 device.
StaticRevision	INT	74	The static revision of the alert generated by the H1 device.
ValueDiscrete	INT	76	If the alert standard type is a discrete alarm, then this will provide the integer value.
SubCode	INT	78	The sub code of the alert generated by the H1 device.

ValueReal	REAL	80	If the alert standard type is an analog alarm, then this will provide the floating-point value.
Raw Alert Data	SINT[128]	84	If there is additional custom alert data available, then it will be in the Raw Alert Data. The size of this raw data is configured in the H1 configuration as the *Additional Raw Data*.

Table 4.3 – FF Master Alert

4.6.1.4. FF MASTER - MASTER CONTROL

Below is the format of the FF Master Control:

Parameter	**Data Type**	**Byte Offset**	**Description**
Redundancy Control	SINT	0	Bit 0 – Force the FF Link to become the Active LAS. **NOTE:** This parameter is only relevant for H1 Master Redundancy operation and will be managed by the generated Logix AOI.
Reserved	SINT	1	Reserved for future use.
OtherMasterConfigCRC	INT	2	The CRC of the other H1 Master used for H1 Master Redundancy. This will allow the local H1 Master to determine if the two H1 Masters have the same configuration. **NOTE:** This parameter is only relevant for H1 Master Redundancy operation and will be managed by the generated Logix L5X.
Reserved	DINT	4	Reserved for future use.
MasterPvVcr	SINT[64]	8	The VCR number of the H1 data to be produced by the FF Link. This is used to verify that the generated Logix L5X mapping matches the FF Link configuration. **NOTE:** This parameter will be managed by the generated Logix L5X.

Table 4.4 – FF Master Control

4.6.1.5. FF MASTER - PROCESS VARIABLE

Below is the raw format of the FF Master Process Variable (PV).

NOTE: When using EtherNet/IP Target or Modbus (Master/Slave) primary interfaces, then the format for the PVs will be specific to that interface to ensure alignment. See the Internal Mapping setup section 3.5.4 for details regarding the reformat options.

NOTE: When using EtherNet/IP Target, Modbus TCP Slave, or Modbus TCP Master, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

Parameter	Data Type	Byte Offset	Description
Status	SINT	0	The raw status of the Process Variable sent. This will be derived from the reformatted data.
Process Variable	SINT / REAL	1	The process variable data to be sent. If the data type is a Discrete, then the process variable data type is SINT. If the data type is an Analog, then the process variable data type is a REAL.

Table 4.5 – FF Master Process Variable (PV)

4.6.1.6. FF MASTER - ALERT ACK

Below is the format of the FF Master Alert Ack:

Parameter	Data Type	Byte Offset	Description
Alert Index Acknowledge	DINT	0	The Alert Index that must match the Alert Index number in the Alert data to inform the FF Link that the next Alert can be loaded into the Alert Data.

Table 4.6 – FF Master Alert Ack

4.6.1.7. FF DEVICE - DEVICE STATUS

Below is the format of the FF Device Status:

Parameter	Data Type	Byte Offset	Description
Online	BOOL	0.0	The H1 device is online.
Data Exchange Active	BOOL	0.1	The H1 device is exchanging data.
Ident Mismatch	BOOL	0.2	The H1 device online does not match the configured H1 device.
Device Error	BOOL	0.3	The H1 device has an error.
System Update Busy	BOOL	0.4	The H1 device is busy reading and updating the required system parameters.
AP Update Busy	BOOL	0.5	The H1 device is busy reading and updating the required application parameters.
MapChecksumMismatch	BOOL	0.6	The Logix L5X mapping does not match the configured device.

			NOTE: This is only relevant to the EtherNet/IP Target primary interface.
Reserved	BOOL[9]	0.7 – 1.7	Reserved for future use.
Device Node Address	INT	2	The H1 node address of the H1 device.

Table 4.7 – FF Device Status

4.6.1.8. FF DEVICE - DEVICE CONTROL

Below is the format of the FF Device Control:

Parameter	Data Type	Byte Offset	Description
MappingCRC	INT	0	The L5X mapping checksum which is used to verify the Logix Mapping. **NOTE:** This is only relevant to the EtherNet/IP Target primary interface. This will be updated by the L5X file created.
Device Node Address	INT	2	The L5X mapping device node address which is used to verify the Logix Mapping. **NOTE:** This is only relevant to the EtherNet/IP Target primary interface. This will be updated by the L5X file created.

Table 4.8 – FF Device Control

4.6.1.9. FF DEVICE - PROCESS VARIABLE

NOTE: When using EtherNet/IP Target or Modbus (Master/Slave) primary interfaces, then the format for the PVs will be specific to that interface to ensure alignment. See the Internal Mapping setup section 3.5.4 for details regarding the reformat options.

NOTE: When using EtherNet/IP Target, Modbus TCP Slave, or Modbus TCP Master, it is recommended to use the Auto Target Mapping feature (in the H1 tab configuration). This will automatically map and reformat all the required data in the Internal Map.

Below is the raw format of the FF Device Process Variable (PV):

Parameter	Data Type	Byte Offset	Description
Status	SINT	0	The raw status of the Process Variable sent. This will be derived from the reformatted data.

Process Variable	SINT / REAL	1	The process variable data to be sent. If the data type is a Discrete then the process variable data type is SINT. If the data type is a Analog then the process variable data type is a REAL.

Table 4.9 – FF Device Process Variable (PV)

4.7. INTERNAL DATA SPACE (IDS)

The FF Link Status and Control will be mapped to a specific Internal Data Space register offset. The Internal Status will be written to IDS offset 0. The internal status can be mapped to another destination using the internal mapping.

4.7.1. INTERNAL STATUS

The Internal Status structure is shown below.

Field	**Byte Offset**	**Parameter Size**	**Data Type**	**Description**
Uptime	0	4	DINT	The number of seconds since the module was powered up.
Date Year	4	2	INT	The module UTC Year.
Date Month	6	1	SINT	The module UTC Month.
Date Day	7	1	SINT	The module UTC Day.
Time Hour	8	1	SINT	The module UTC Hour.
Time Minute	9	1	SINT	The module UTC Minute.
Time Second	10	1	SINT	The module UTC Second.
Device Temperature	11	4	REAL	Current temperature of the device in °C.
Hardware MAC address	15	6	SINT[6]	The MAC address of the module.
Ethernet Port1 Status	21	1	SINT	Status of Ethernet Port 1.
Link Up	*21.0*	-	*BIT*	*0 – The Link is down* *1 – The Link is up*
Ethernet Port2 Status	22	1	SINT	Status of Ethernet Port 2.
Link Up	*22.0*	-	*BIT*	*0 – The Link is down* *1 – The Link is up*

DLR Status	23	1	SINT	DLR Status when DLR is enabled
Enabled	*23.0*	-	*BIT*	*0 – Disabled* *1 – Enabled*
Topology	*23.1*	-	*BIT*	*0 – Linear* *1 – Ring Topology*
Status	*23.2*	-	*BIT*	*0 – Normal* *1 – Fault*
NTP Status	24	1	SINT	NTP Status when NTP is enabled
Enabled	*24.0*	-	*BIT*	*0 – Disabled* *1 – Enabled*
Connection to Server	*24.1*	-	*BIT*	*0 – Not Locked to a NTP Server* *1 – Locked to a NTP Server*
Firmware Major Rev	25	1	SINT	Application Firmware Major Rev
Firmware Minor Rev	26	1	SINT	Application Firmware Minor Rev
Firmware Micro Rev	27	1	SINT	Application Firmware Micro Rev
Modbus Slave Comms Status	28	1	SINT	0 – Modbus Slave Communication Fail 1 – Modbus Slave Communication Ok
Modbus Master Slave Status	29	256	SINT[256]	Each byte represents a Modbus Node address for the Modbus Master interface. 0 – The Modbus Slave Comms Fail 1 – The Modbus Slave Comms Ok

Table 4.10 – IDS Status Register Format

5. LEGACY MASTER MIGRATION

The FF Link supports a configuration migration option for the previous generation 1788-EN2FFR and 1788-CN2FFR modules. To migrate the FF configuration, the user must export the 1788-xN2FFR configuration from the Add-On Profile (AOP).

The user can select whether to migrate the entire 1788-xn2FFR configuration including the master configuration and all the H1 devices (*.hbx), or just an individual H1 device (*.hdx).

To export the entire bridge, select the ***Export Device*** button (as shown below).

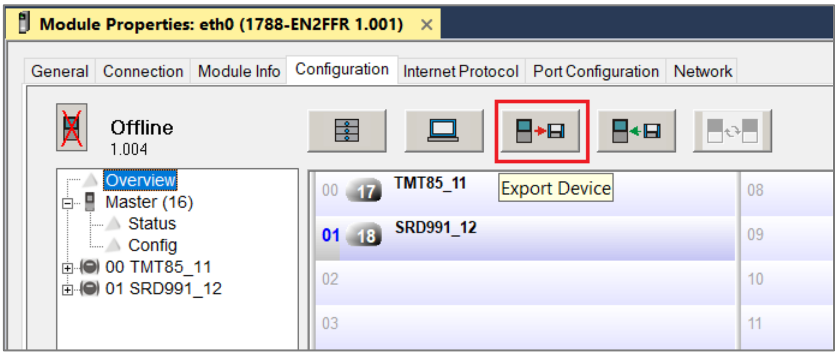


Figure 5.1 – Export 1788-xN2FFR configuration – Bridge

This will create an “hbx” file that can be imported into Slate.

Alternatively, to export an individual H1 device, right-click on the selected device on the ***Overview*** screen and select the ***Export Device*** option.

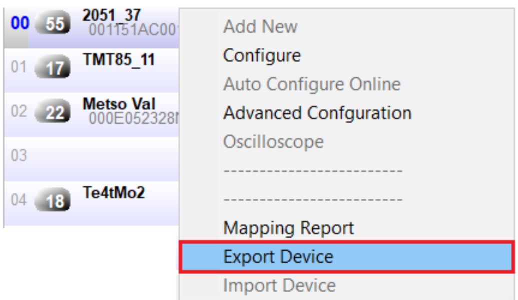


Figure 5.2 – Export 1788-xN2FFR configuration – Device

This will create an “hdx” file that can be imported into Slate.

In Slate, create a new project with an FF Link module. The, right-click on the module and select the ***Import xN2FFR Bridge*** option. Select the previously exported “hbx” or “hdx” file and press ***Ok***.

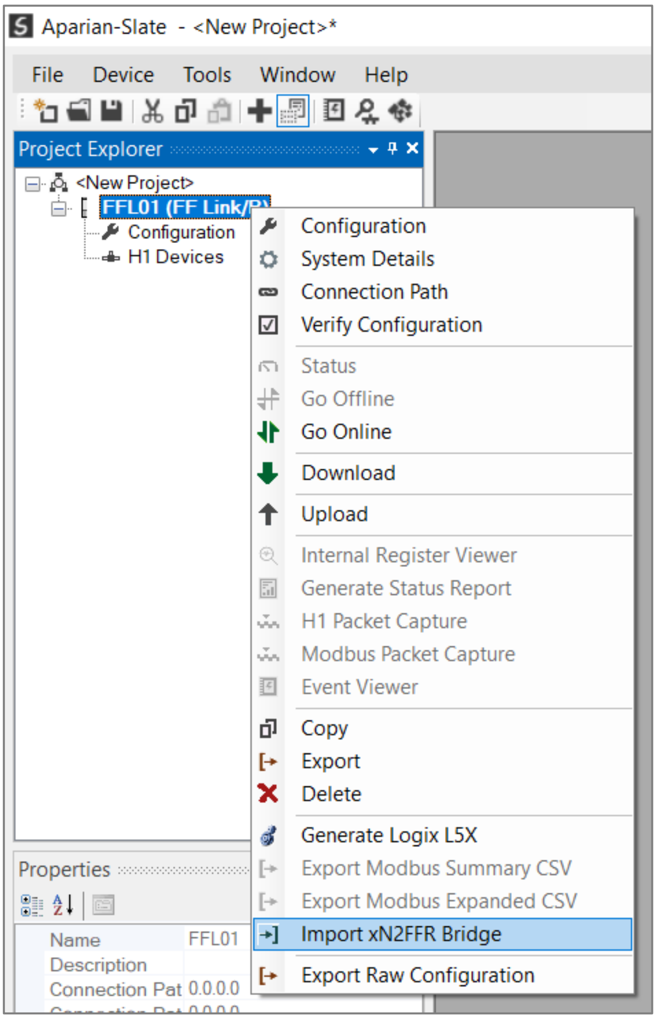


Figure 5.3 – Import 1788-xN2FFR configuration

The file will be processed and the ***Legacy xN2FFR Import*** window will show a summary of the master and H1 device configuration items found.

The import utility will attempt to match the devices found in the legacy configuration with DD files in Slate’s DD catalog. If no matching DD files could be found, then the user will need to add the required DD files to the Slate DD catalog.

The user can then select which of the H1 devices to import.

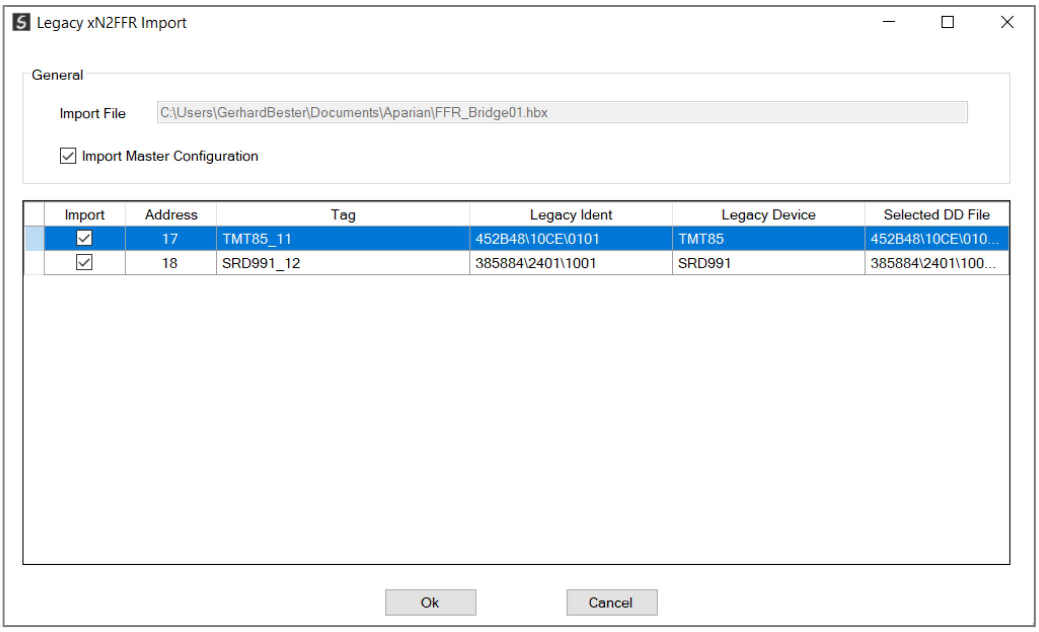


Figure 5.4 – Import 1788-xN2FFR device selection

Then press the ***Ok*** button. The H1 devices will appear in the FF Link H1 Devices tree once the devices from the legacy configuration have been imported.

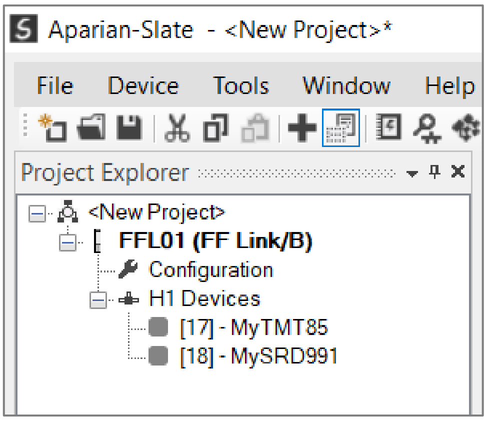


Figure 5.5 – H1 devices imported

The block and parameter configuration from the legacy 1788-xN2FFR configuration will also have been imported as best as possible to replicate the legacy configuration.

NOTE: If the import did not provide the exact configuration as per the legacy configuration (due to DD file mismatch, etc.), then it is the user’s responsibility to ensure the imported H1 device’s configuration is updated to match.

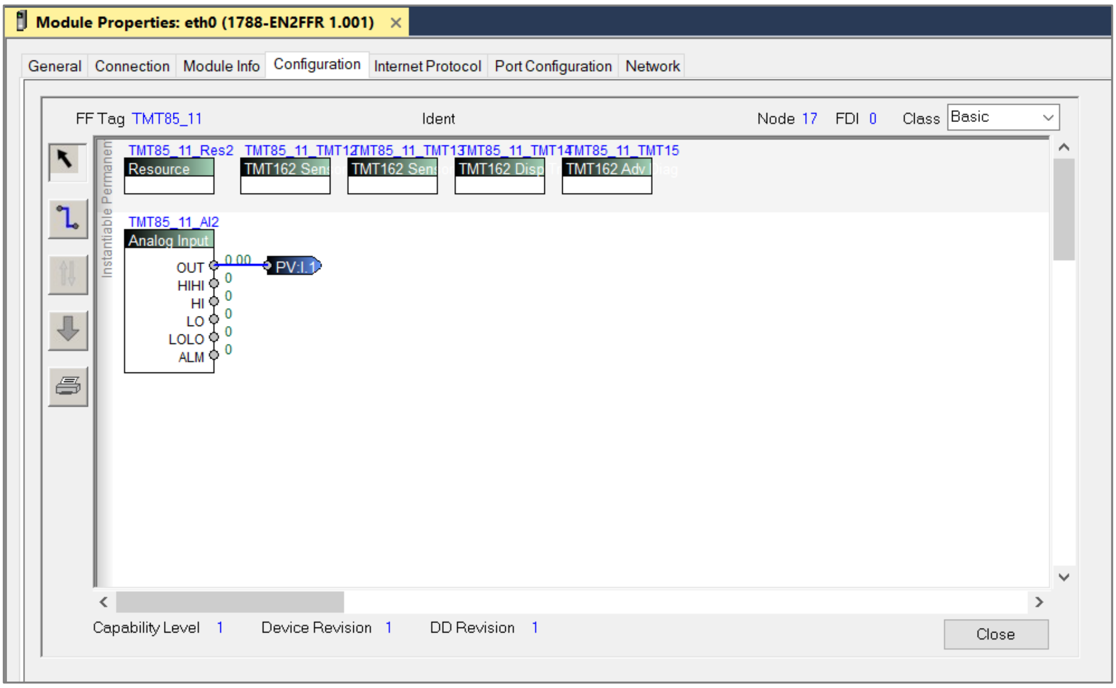


Figure 5.6 – Exported H1 device block configuration

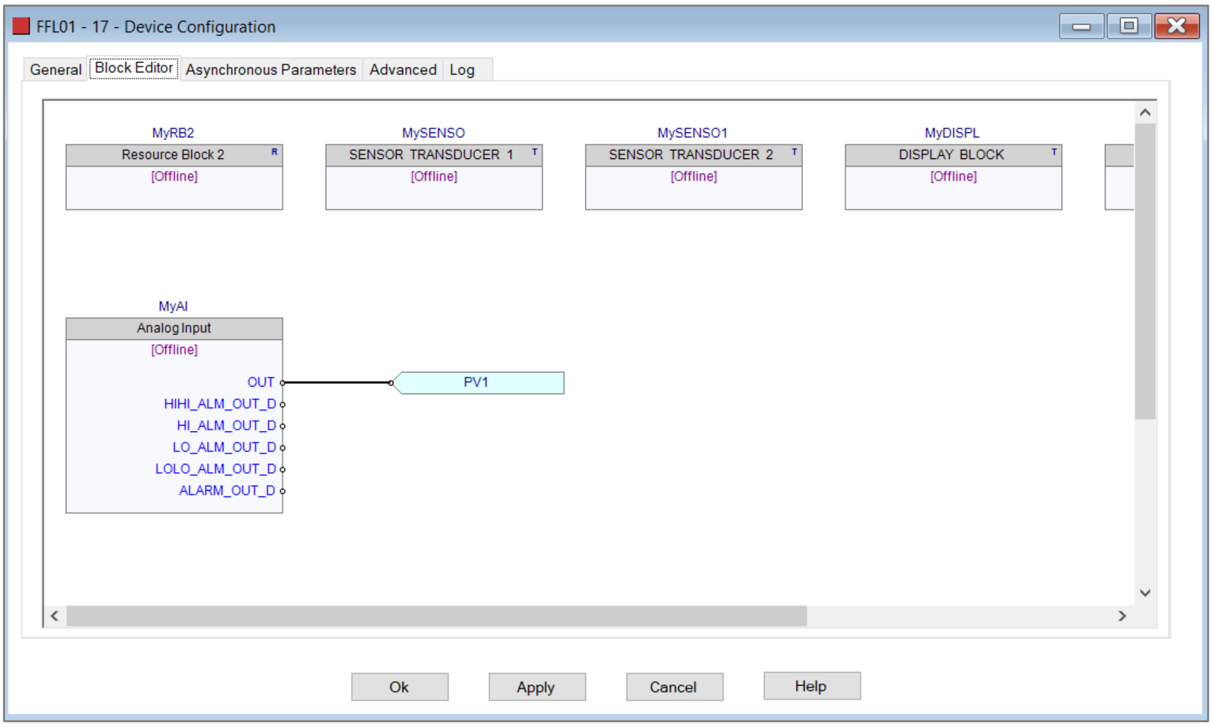


Figure 5.7 – Imported H1 device block configuration

6. FF LINK MASTER REDUNDANCY

The FF Link module supports H1 Master redundancy allowing two FF Link modules to be connected to the same H1 segment. The FF Link modules have the same configuration and operate in a **one-active-one-standby** strategy.

Using FF Link redundancy removes the single point failure associated with:

- Ethernet connection failure
- FF Link power failure
- H1 cable fault (open circuit)

NOTE: The FF Link redundancy functionality is available only when selecting ***EtherNet/IP Target*** as the Primary Interface. It is not available when using Modbus TCP Master, Modbus TCP Slave or EtherNet/IP Originator.

6.1. REDUNDANCY STRATEGY

The FF Link redundancy strategy is based on two FF Link masters with the identical configuration. One in the ***Active*** state and the other in the ***Standby*** state.

The master selection between which FF Link module ("A" or "B") is the ***Active*** is generally made by redundancy Add-On-Instruction (AOI). However, because the switch-over of the H1 Master functionality needs to happen asynchronously to the Logix AOI execution and module RPI, the Standby module will automatically take-over the Master role after it sees no valid traffic on the H1 bus for the LAS claim time.

Once the Logix AOI detects the switch-over it will adjust the master selection to the new Active H1 Master. This switch-over (confirmation) needs to occur within the ***Switch-Over Timeout*** parameter to prevent the new Active master reverting to Standby and a disruption to the H1 bus.

The necessary Logix code required to manage the redundancy, including the AOI is automatically generated by the Slate configuration software.

6.2. REDUNDANT ARCHITECTURE

The figure below shows a typical redundant FF Link architecture. Both FF Link modules are connected to the same H1 segment.

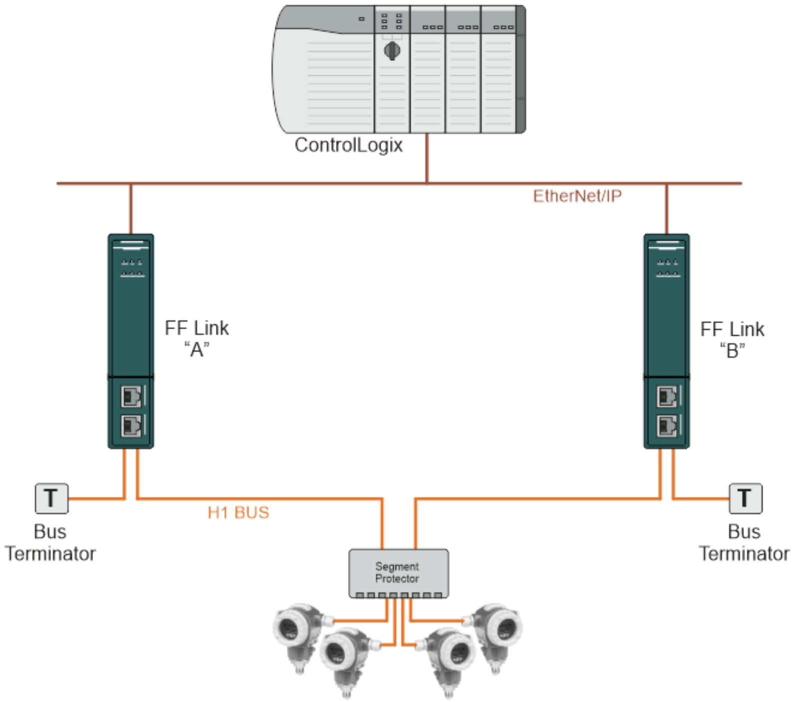


Figure 6.1 - Typical Redundant Architecture

The FF Link modules can be connected to the same Ethernet network and connected to Logix via the same Ethernet bridge module. For added redundancy, it is also possible to connect the FF Link module with separate ControlLogix Ethernet bridge modules using either the same or different physical Ethernet switches.

When using the FF Link's internal power conditioner, the required bus power will be shared by the two FF Link modules when both are powered and available. Should one FF Link module be disconnected or powered down, then the other FF Link module will supply all the required bus power.

NOTE: Although the H1 bus power will be shared between the two FF Link modules, the bus current will not necessarily be shared equally.

When using FF Link redundancy, it is not recommended to use the internal Bus Terminator functionality, because removing power from one of the modules will reduce the total bus terminators to one making the H1 segment unstable.

It is also recommended that the two FF Link modules are fed from different 24V power supplies, and where possible, these power supplies should be fed from different AC sources.

6.3. FF LINK CONFIGURATION

The configuration of a redundant FF Link is similar to that of a standalone with a few notable exceptions.

6.3.1. GENERAL CONFIGURATION

In the ***General*** tab of the FF Link configuration, set the ***Mode*** to ***Redundant Master***. Then configure the IP Addresses of both the FF Link modules “A” and “B”.

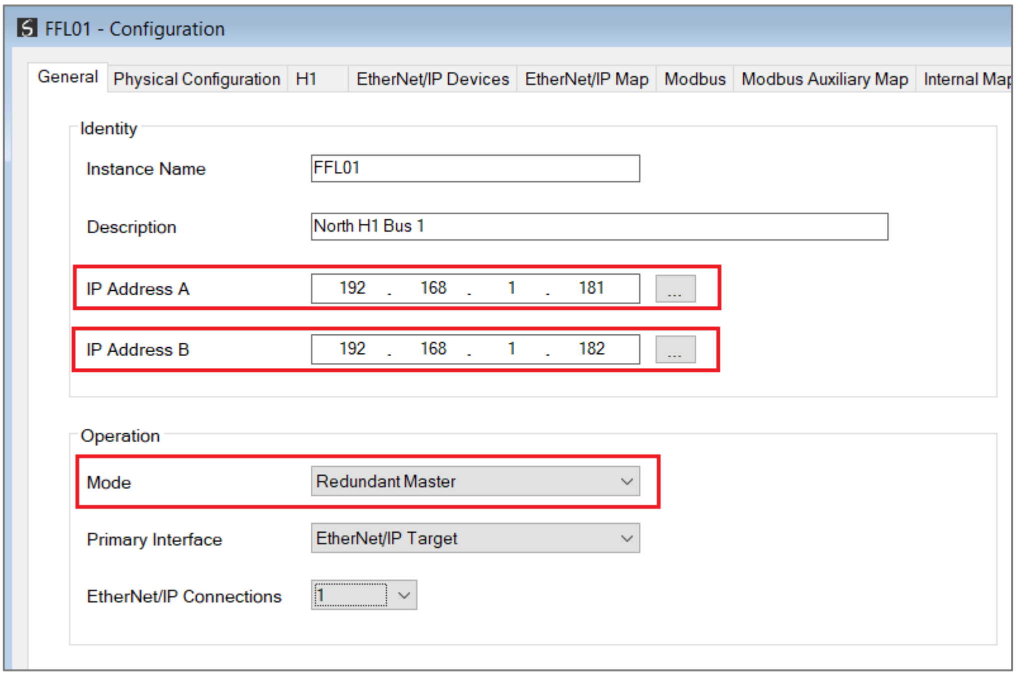


Figure 6.2 – Redundant Configuration - General

6.3.2. PHYSICAL CONFIGURATION

In the ***Physical Configuration*** tab, set the ***Switch-Over Timeout*** parameter. By default, the ***Auto* Recommend** option is enabled (which will automatically calculate the ***Switch-Over Timeout*** parameter). The internal ***Bus Termination*** should not be used when using redundancy.

NOTE: It is recommended to not manually set the ***Switch-Over Timeout*** but to rather have the value auto calculated.

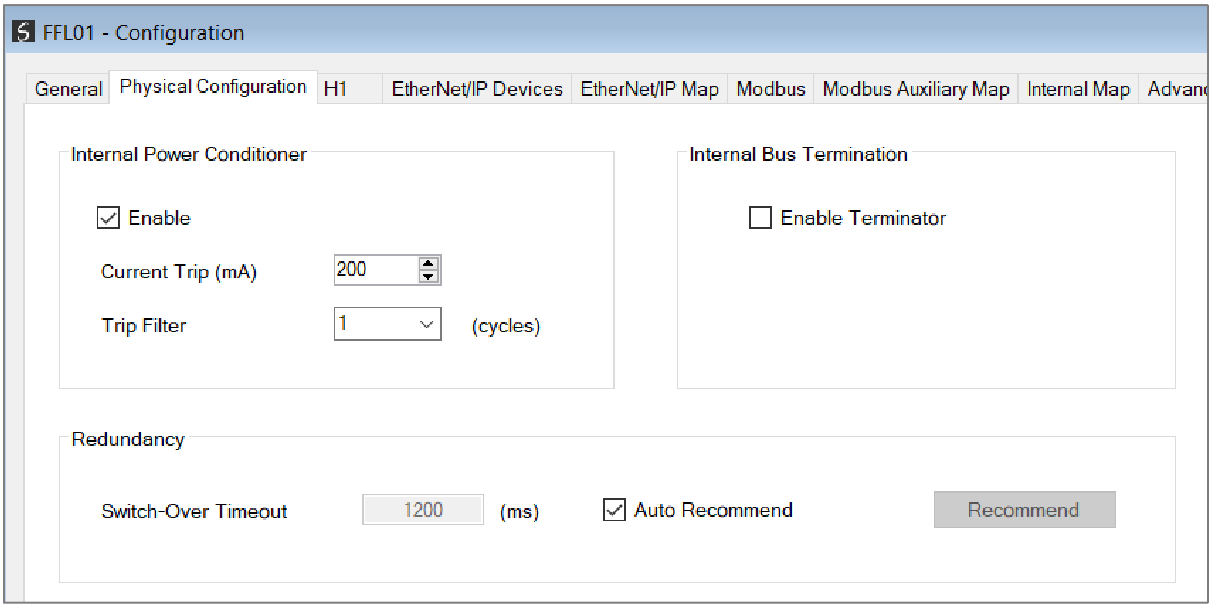


Figure 6.3 – Redundant Configuration - Physical Configuration

6.3.2.1. SWITCH-OVER TIMEOUT

The ***Switch-Over Timeout*** is used by the module to override the Active / Standby command from the Logix AOI.

The ***Switch-Over Time*** parameter is in milliseconds and should be the greater of 1000ms and 4 x the module RPI.

The automatic calculation (Auto Recommended) of this parameter takes into account the free bandwidth within the Macro Cycle, by looking at the number of configured compels and required token passes.

NOTE: The ***Recommend*** button function assumes the RPI is not a limiting factor, which may be inadequate when using larger RPI values. If the default RPI of 100ms is not used, then this value may need to be adjusted manually.

NOTE: Failure to configure the ***Switch Timeout*** correctly will result in unexpected behaviour during a redundancy switch.

6.4. LOGIX CONFIGURATION

The addition of a redundant FF Link module pair to the Logix IO tree follows the same procedure as a Standalone FF Link module, except two instances of the module are added.

It is important to follow the correct naming convention to ensure the Logix code generated by Slate is valid. The “A” module must have the same name as specified in Slate, with an “A” suffix, similarly, the “B” module must have the same name as specified in Slate, with a “B” suffix.

In the example below, the instance names are as follows:

Slate Module (pair) name:	**FFL01**
Logix IO Tree Module A:	**FFL01A**
Logix IO Tree Module B:	**FFL01B**

Ensure the IP addresses are configured correctly for the A and B modules and that the RPIs for all connections are identical.

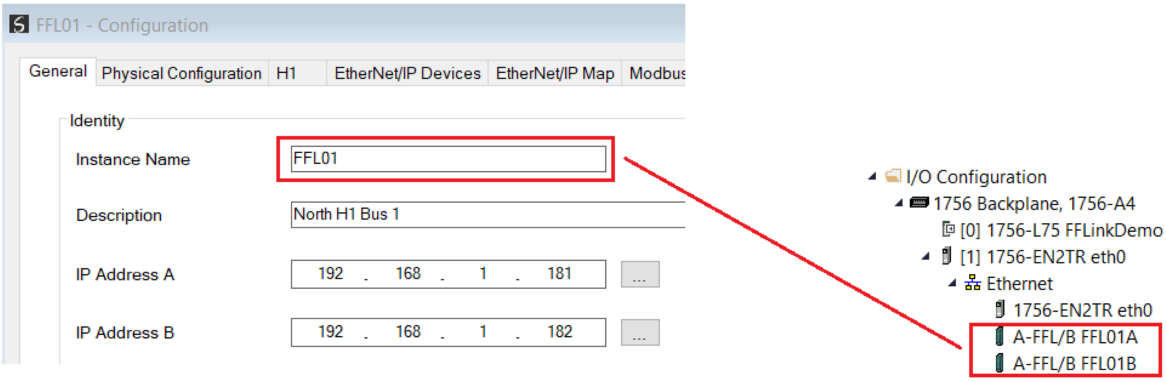


Figure 6.4 – Logix IO Tree – Redundant FF Link pair

After the FF Link modules have been added into the Logix IO tree, the mapping code can be imported. As in the case of the Standalone Master mode, the Logix code can be generated in Slate by right-clicking on the module and selecting the ***Generate Logix L5X*** option.

In addition to the H1 device mapping code, the **FFLinkRedundancyMaster** AOI will be instantiated. This AOI is responsible for controlling the redundancy and mapping the first IO connection.

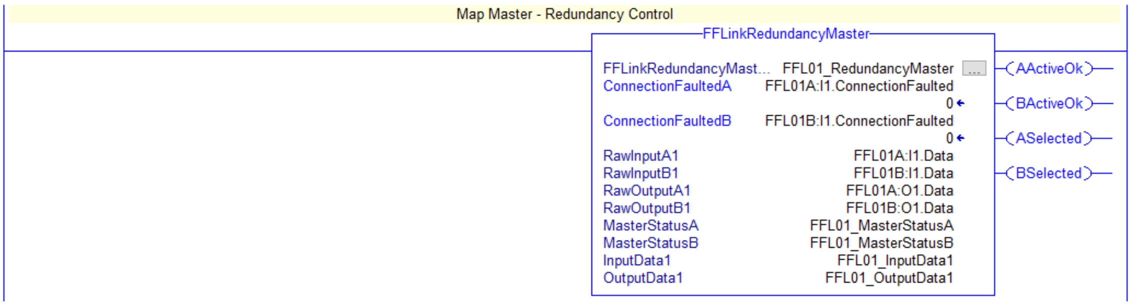


Figure 6.5 – Logix – Redundant Master AOI

If the FF Link configuration is making use of more than one EtherNet/IP Class 1 connection, then a **FFLinkRedundancyConnectionMap** AOI will be instantiated for each additional class 1 connection.

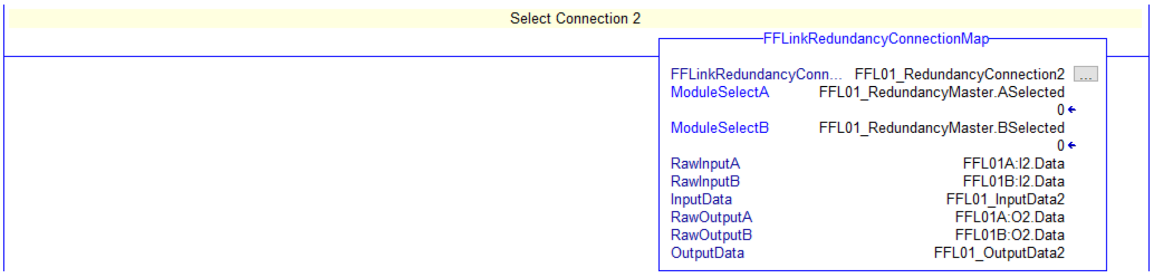


Figure 6.6 – Logix – Redundant Connection Map AOI

6.5. OPERATION

The operation of the FF Link module in redundancy mode is similar to that in standalone mode.

Once the Slate configuration has been finalised, it can be downloaded to the FF Link module pair. Before downloading it is important to confirm the connection paths to the two modules by right-clicking on the module pair and selecting the ***Connection Path*** option.

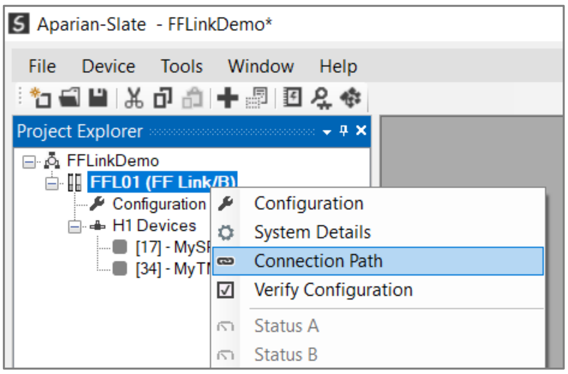


Figure 6.7 – Select Connection Path

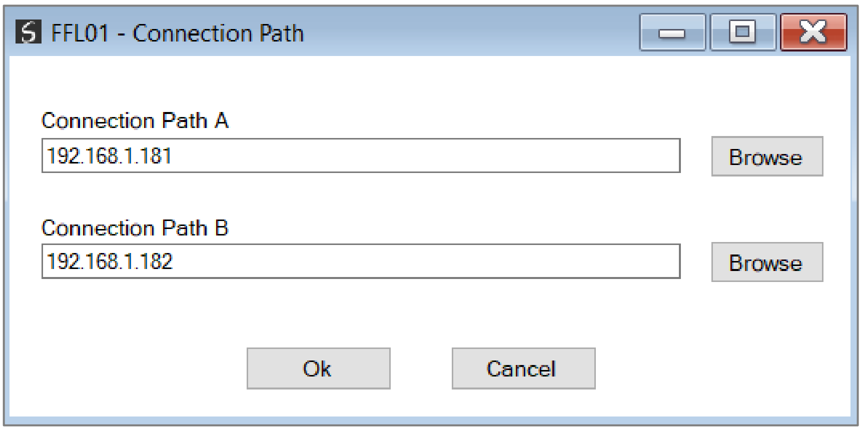


Figure 6.8 –Connection Path

The ***Download to Both*** option, can then be selected, which will transfer the configuration to both the “A” and “B” FF Link modules.

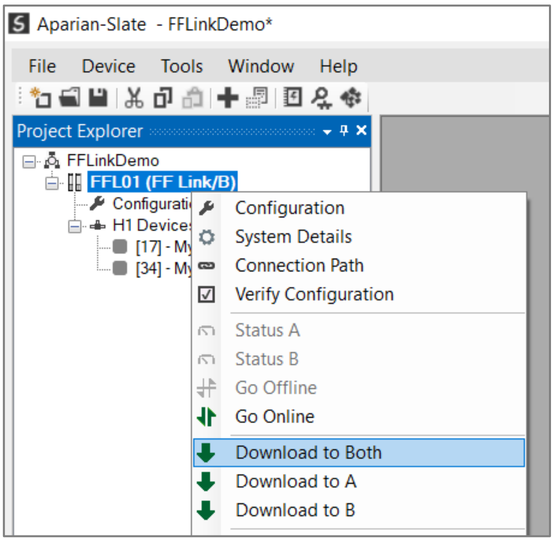


Figure 6.9 – Download to Both

7. DEVICE FIRMWARE UPDATE

The FF Link 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.

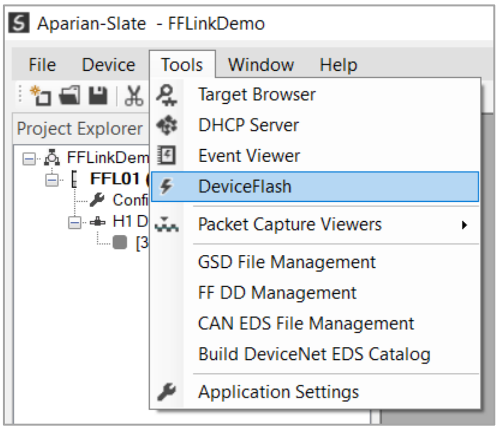


Figure 7.1 – Select DeviceFlash utility from Slate

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

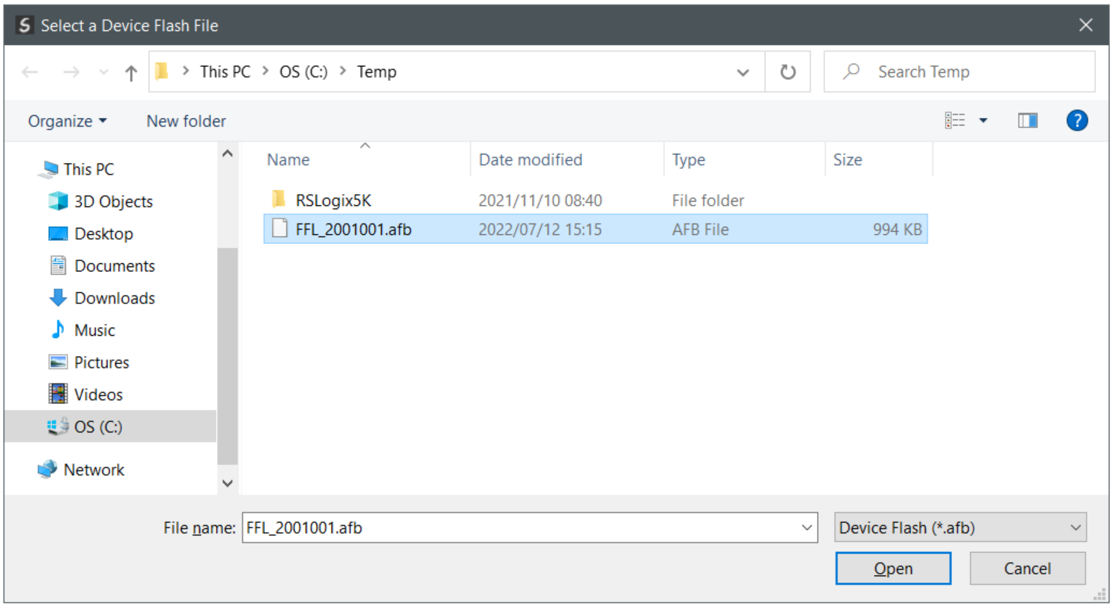


Figure 7.2 – Select the binary file

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

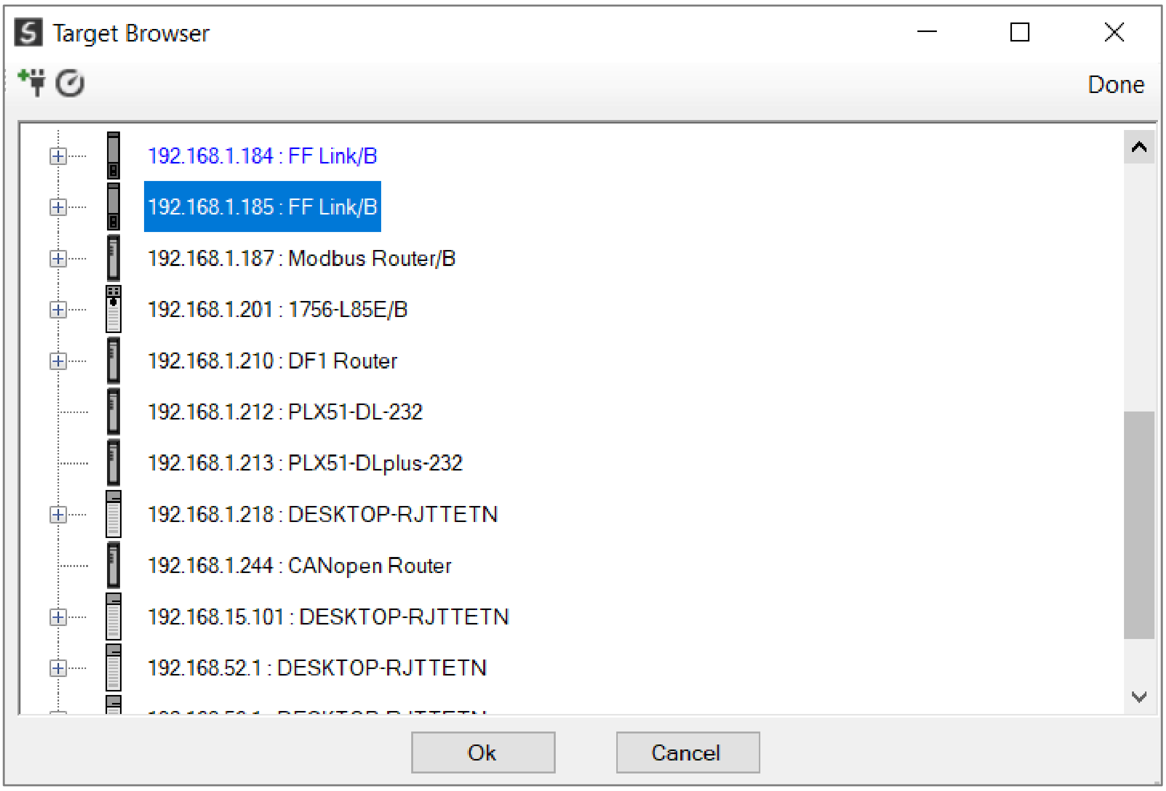


Figure 7.3 – Select the device to be updated

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

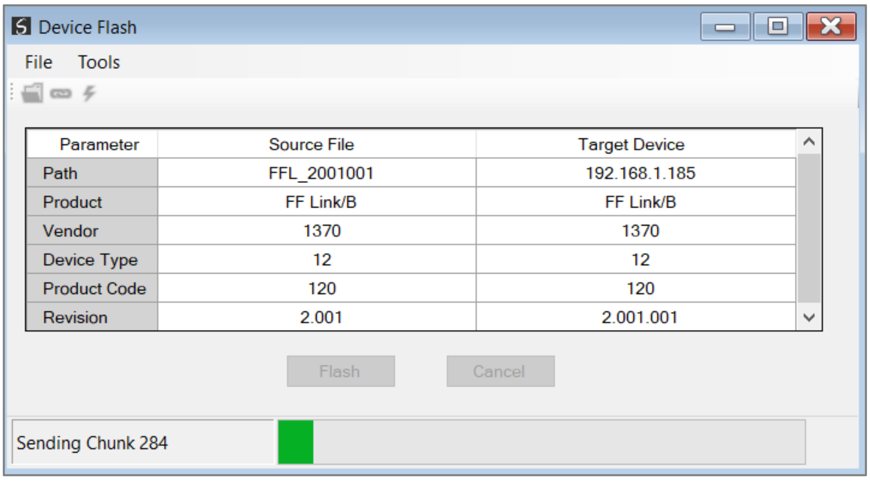


Figure 7.4 – Firmware update busy

- Once the firmware update has successfully completed, the ***Target Device*** column will have a green background.

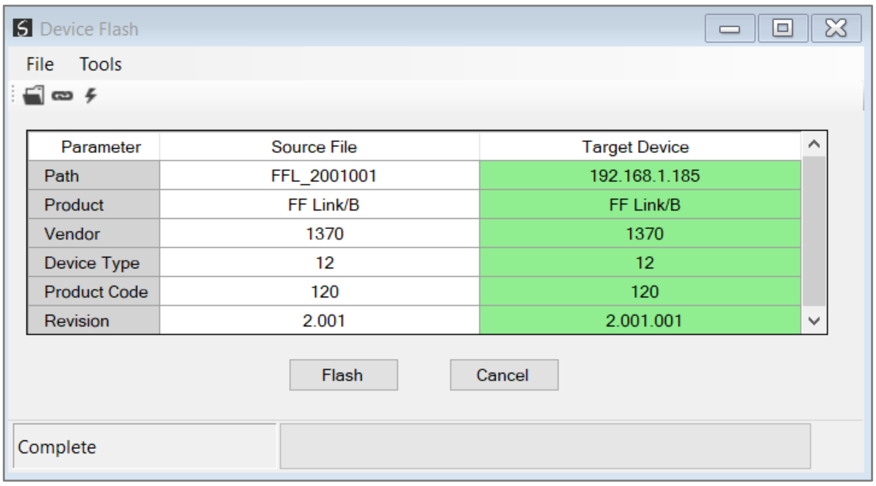


Figure 7.5 – Firmware update successfully completed.

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.

8. DEVICE TYPE MANAGER (DTM)

The FF Link supports FDT / DTM technology, allowing the user to configure any field device using its DTM (Device Type Manager) in any standard FDT Frame (Field Device Tool). To use a device DTM with the FF Link, the Aparian DTM pack must first be installed.

8.1. INSTALLATION

Installation of the Aparian DTM pack is achieved by executing the following installer:
Aparian DTM Pack 1.005 Setup.msi

The installation wizard will guide the user through the installation process.

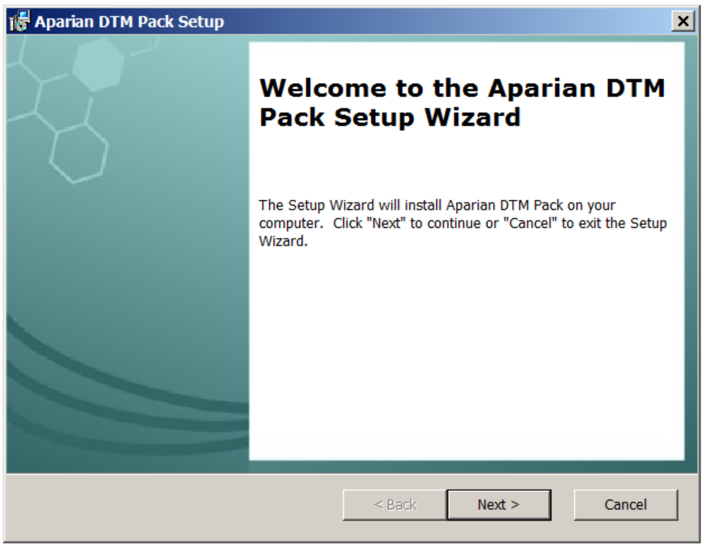


Figure 8.1 – Aparian DTM Pack Installation

8.2. CONFIGURATION

Once the DTM pack is installed, the selected FDT Frame would need to have its DTM Catalogue updated. The steps required for this action are slightly different for each FDT frame. Typically, one selects the ***DTM Catalogue*** or ***Device Catalogue*** and select ***Refresh*** or ***Rebuild***.

The FF Link provides two DTM options:

- **FF Link** (Direct) – Includes a built-in CommDTM with direct EtherNet/IP communication
- **FF Link-FT** – Requires the FactoryTalk Linx CommDTM (Rockwell Automation)

After the catalogue has been updated, then a new project can be created using either one of the FF Link DTMs.

With the exception of the initial instantiation, the **FF Link (Direct)** DTM and **FF Link-FT** DTM operate in an identical manner.

8.2.1. Configuration – FF Link (Direct) DTM

To use the **FF Link (Direct)** DTM in a new project, select the ***Add Device*** function and then select the ***FF Link*** DTM. The example below makes use of PACTware FDT frame.

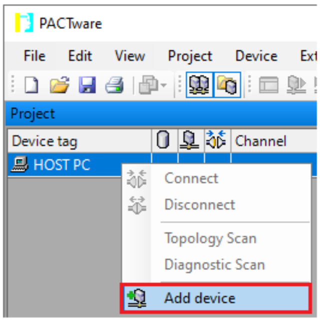


Figure 8.2 – Adding new device

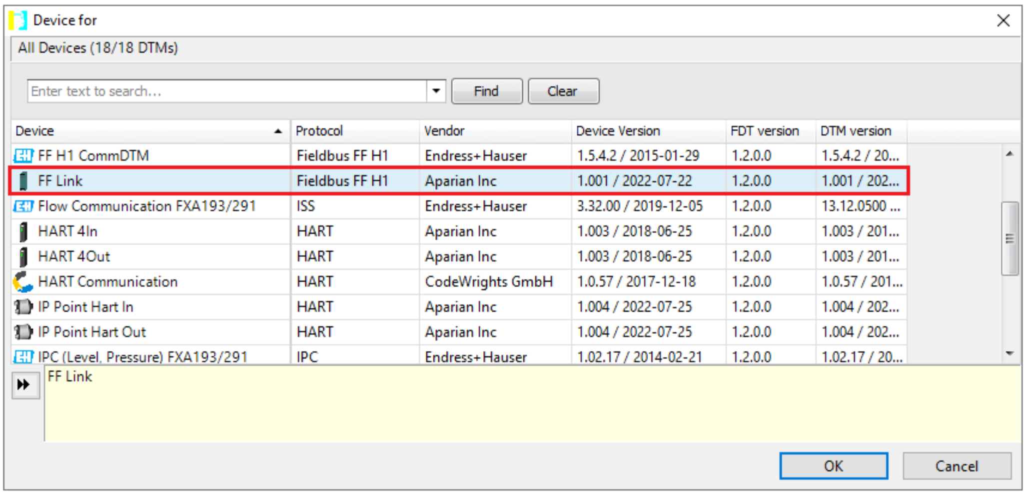


Figure 8.3 – Selecting FF Link DTM

After instantiating the FF Link DTM, select the ***Parameter*** option.

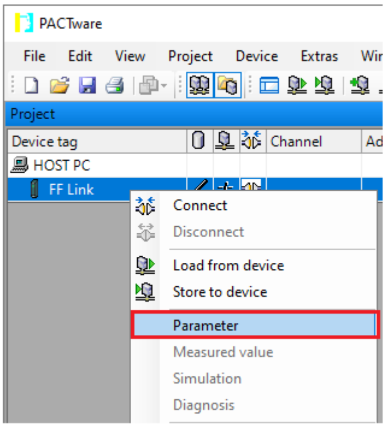


Figure 8.4 – Select Parameter option

The FF Link DTM's configuration allows the ***CIP Path*** to the FF Link to be configured. This is typically the IP address of the FF Link, but can include a more complex CIP path when, for example, routing through a ControlLogix chassis is required.

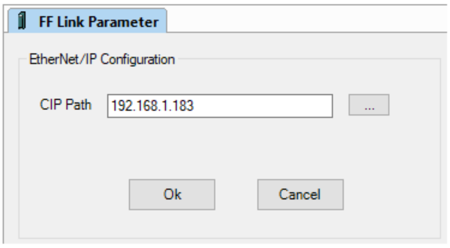


Figure 8.5 – FF Link CIP Path

The path can either be entered manually or the ***Browse*** "…" button can be used to open the ***Target Browser***, and then the FF Link can be selected.

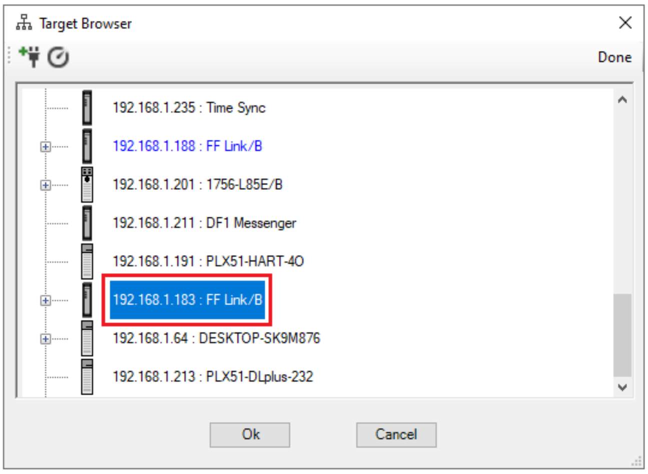


Figure 8.6 – Target Browser

The FF Link DTM is now ready to have device DTMs added under it.

8.2.2. CONFIGURATION – FF LINK-FT DTM

To use the FF Link-FT DTM in a new project, the Rockwell Automation **FactoryTalk Linx CommDTM** must first be instantiated. To do this select the ***Add Device*** function and then select the ***FactoryTalk Linx CommDTM***.

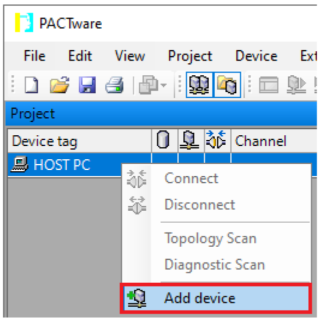


Figure 8.7 – Adding new device

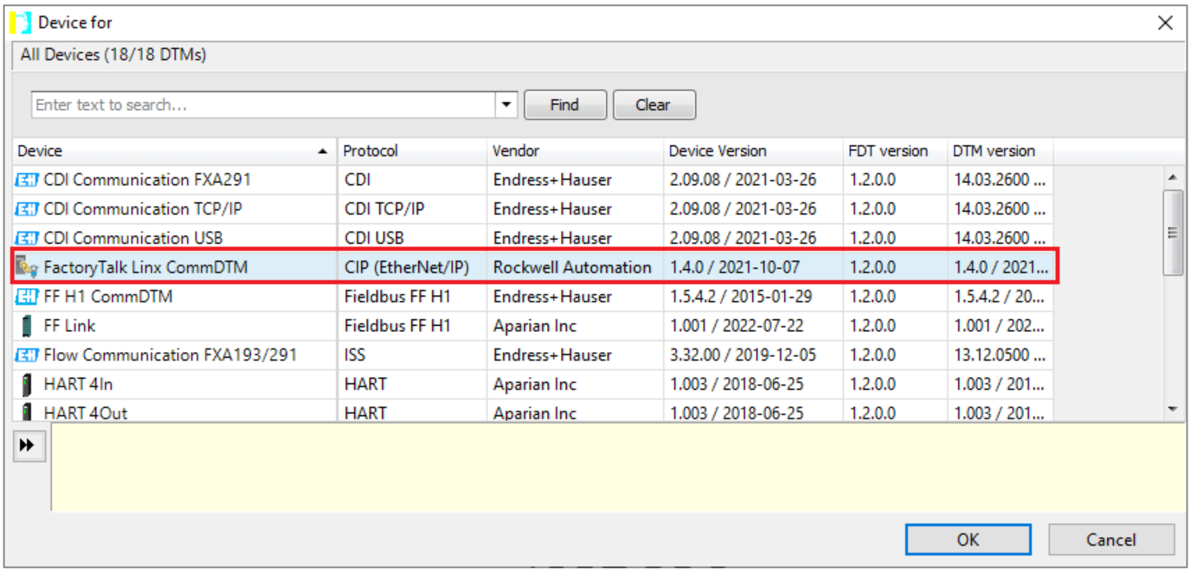


Figure 8.8 – Selecting FactoryTalk Linx CommDTM

Once added, right-click on the **FactoryTalk Linx CommDTM** and again select the ***Add Device*** option.

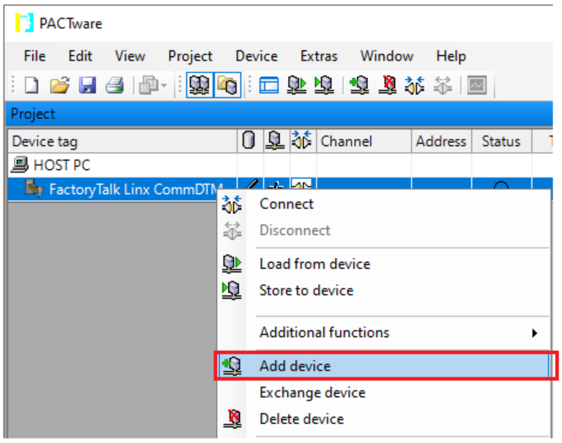


Figure 8.9 – Adding new device under FT Linx CommDTM

Then select the ***FF Link-FT*** DTM.

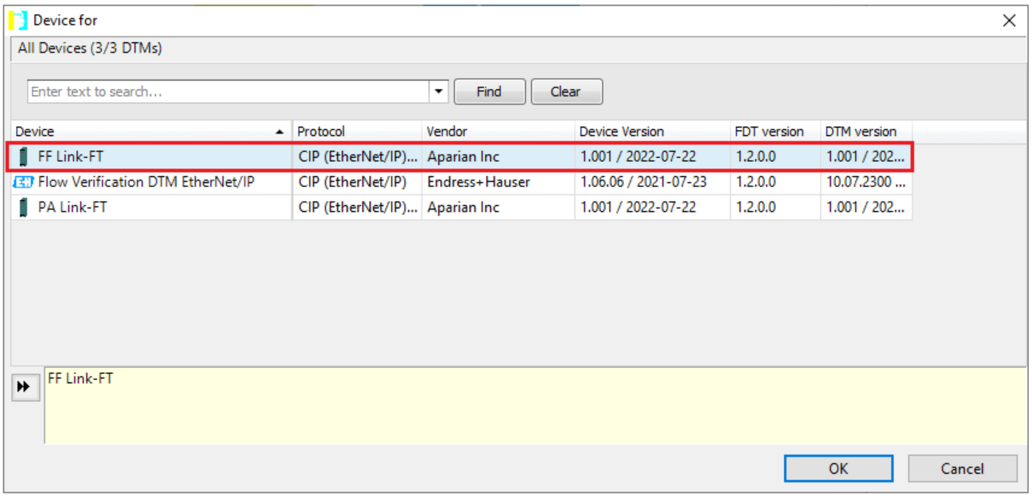


Figure 8.10 – Select FF Link-FT DTM

Once the DTM has been selected, the (FTLinx) Path to the FF Link device will need to be entered.

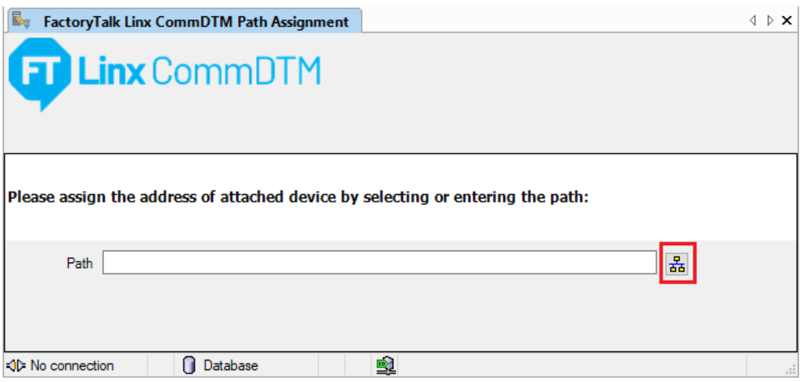


Figure 8.11 – Select FT Linx CommDTM - Path

Select the **RSWho Browse** button located to the right of the **Path**.

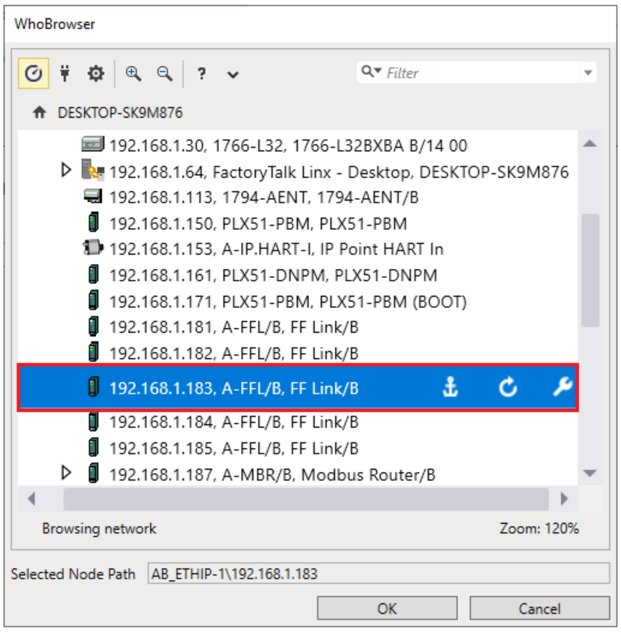


Figure 8.12 – FT Linx WhoBrowser

Select the ***FF Link*** in the WhoBrowser and click on the ***Ok*** button.

The FF Link-FT DTM is now ready to have device DTMs added under it.

8.2.3. Adding Device DTMs

After either the **FF Link (Direct)** DTM or **FF Link-FT** DTM has been configured, the child Device DTMs can be added by right-clicking on the FF Link (or FF Link-FT) DTM and selecting ***Add Device***.

The user can then select the matching device DTM.

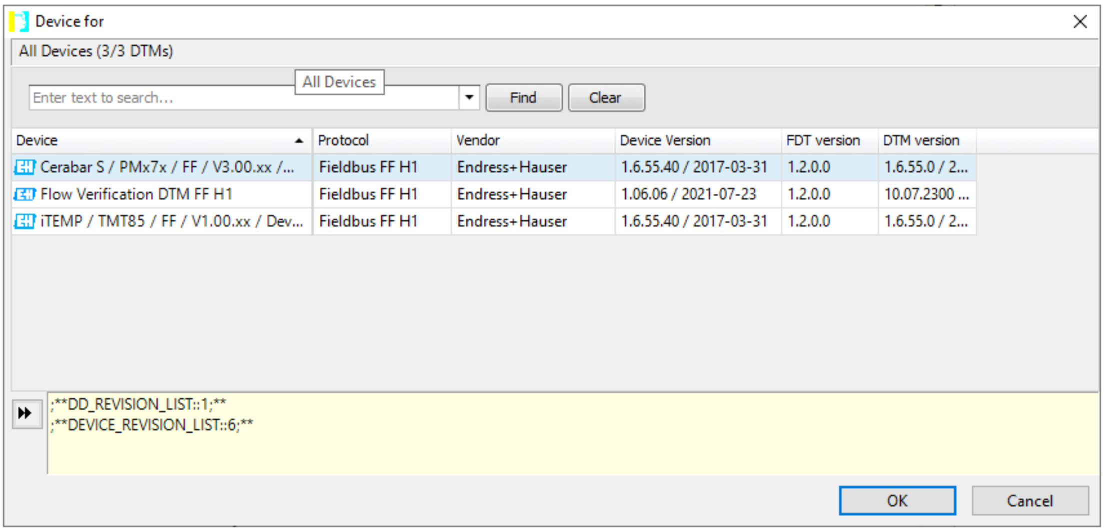


Figure 8.13 – Device DTM Selection

Once the child Device DTM has been added, a configuration window opens to set the ***Node Address***.

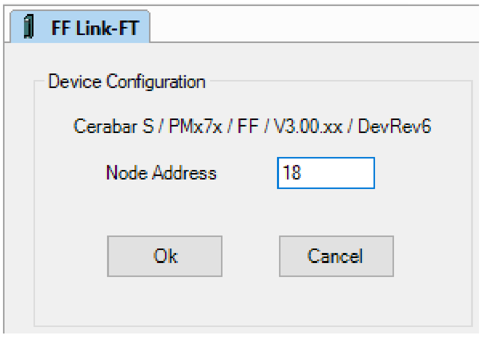


Figure 8.14 – Device DTM Node Address

8.3. OPERATION

After the FDT project has been configured, the DTMs can be placed online by selecting the ***Online*** or ***Connect*** option.

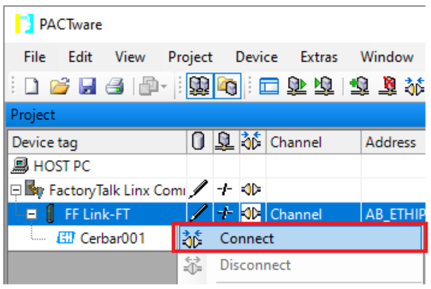


Figure 8.15 – DTM Connect

Once the **FF Link (Direct)** DTM or **FF Link-FT** DTM is online (connected) a number of diagnostic pages can be opened by selecting the ***Measured Value*** option.

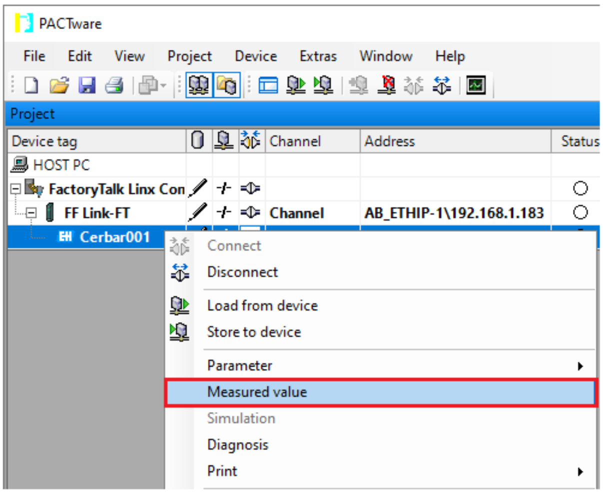


Figure 8.16 – Select Measured Value

The General page provides basic status information for the FF Link module.

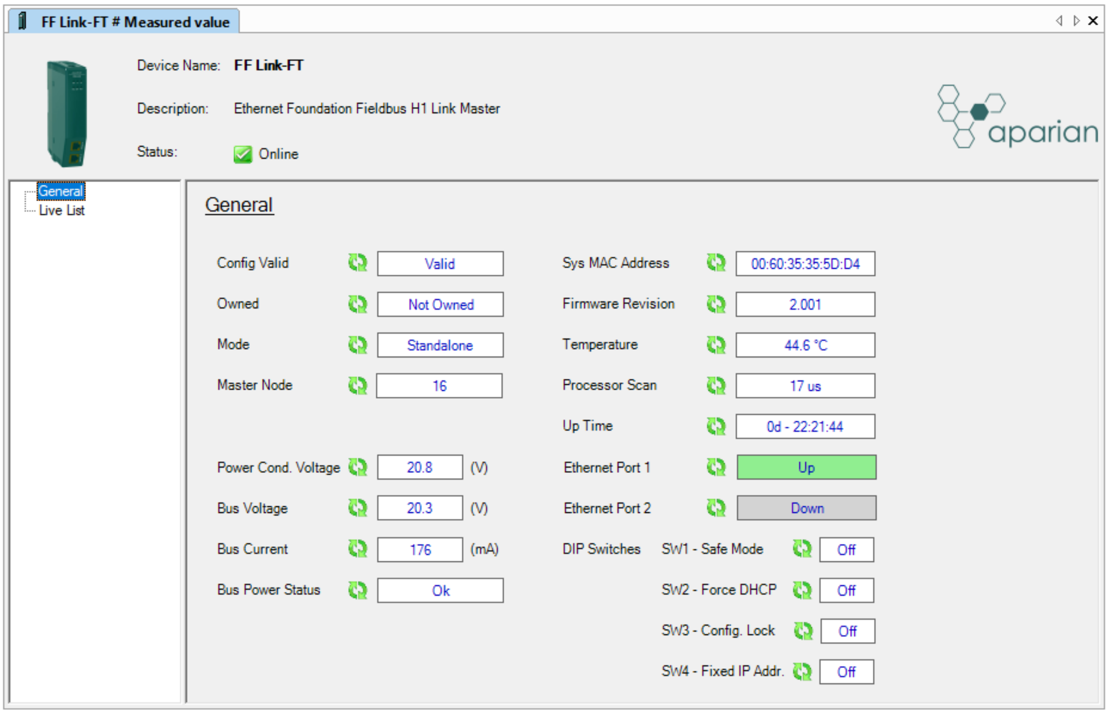


Figure 8.17 – FF Link DTM - General Status Page

The Live List page shows the state of the field devices on the H1 network.

Figure 8.18 – FF Link DTM - Live List Page

A field Device DTM under the FF Link DTM can also be brought online by selecting the ***Online*** or ***Connect*** option.

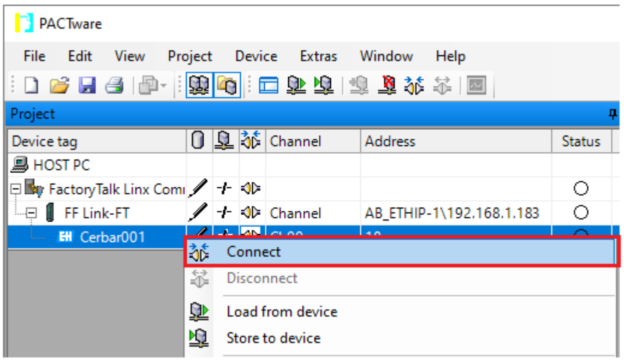


Figure 8.19 – Field Device DTM Connect

Depending on the device DTM, a number of function windows, for example, online parameters, diagnostics and measure variables, can be displayed.

These are accessed by right-clicking on the device DTM and selecting the required function.

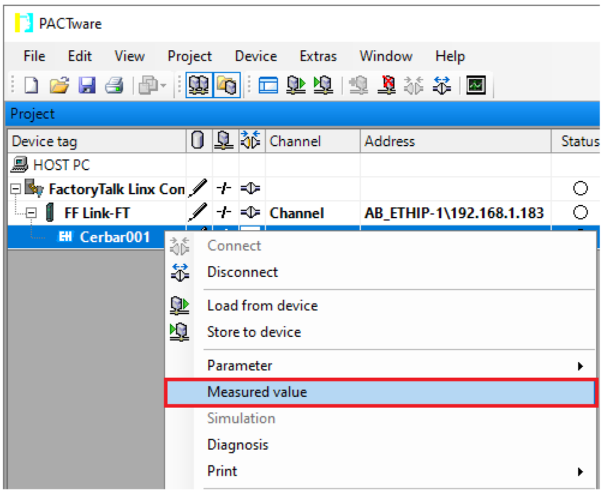


Figure 8.20 – Device DTM - Selecting Measured Value

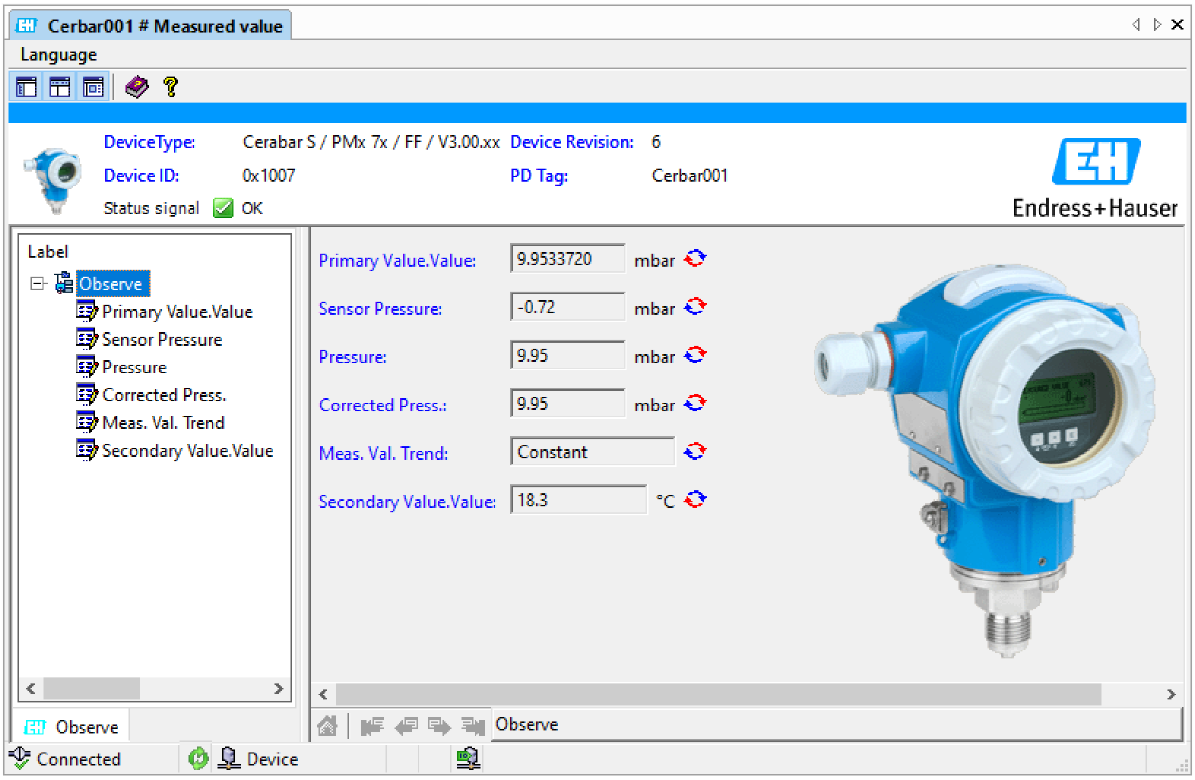


Figure 8.21 – Device DTM – Measure Value

9. DIAGNOSTICS

9.1. LEDS

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

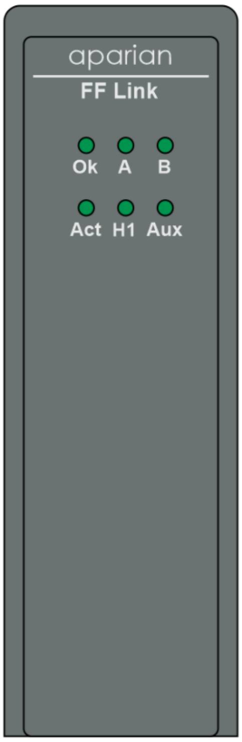


Figure 9.1 - FF Link LEDs

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 green (flashing), then the module has booted and is running correctly **without** any application configuration loaded. If the LED is green (solid), then the module has booted and is running correctly **with** application configuration loaded.
A / B	The Ethernet LED will light up when an Ethernet link has been detected (by plugging in a connected Ethernet cable). The LED will flash every time traffic was detected. This module has two Ethernet ports A and B. Each LEDs represents each specific port.

Act	This LED will indicate the H1 **LAS** status Solid Red – A duplicate H1 node was detected. Solid Green – The local FF Link is the active LAS. Off - The local FF Link is the standby LAS.
H1	This LED indicates the **activity** of the H1 network Solid Red – There H1 Bus has been shorted when the FF Link is supplying power. Flashing Red – A corrupted or incorrect H1 packet was received. Flashing Green – A valid H1 packet was received. Off – No H1 packets are being received.
Aux	The activity LED is used for the activity on the Primary Interface (e.g. EtherNet/IP or Modbus TCP). Every time a valid packet is received from the Primary Interface the LED will toggle green. The LED will toggle red if an incorrect or corrupted packet is received.

Table 9.1 - Module LED operation

9.2. MODULE STATUS MONITORING

The FF Link provides 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 Slate environment, the FF Link must be online. If the module is not already Online (following a recent configuration download), then right-click on the module and select the ***Go Online*** option.

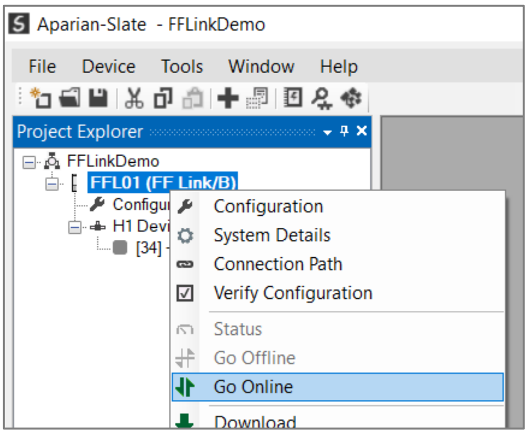


Figure 9.2 - Selecting to Go Online

The Online mode of the module/s is indicated by the icon in the Project Explorer tree.

9.2.1. Project Explorer – Non-Redundant Mode

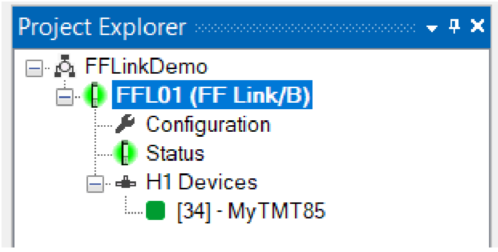


Figure 9.3 - Project Explorer - Non-Redundant

The status of the FF Link module in non-redundant mode is illustrated in the Project Explorer tree as follows:

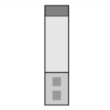
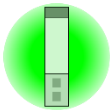
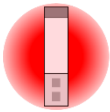
Icon	Description
	Offline
	Online
	Offline – Module was disconnected when previously online.

Table 9.2 – Project Explorer – Non-Redundant

9.2.2. Project Explorer – Redundant Mode

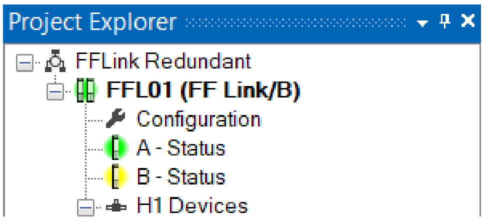


Figure 9.4 - Project Explorer – Redundant FF Link Pair

The status of the FF Link module pair in redundant mode is illustrated in the Project Explorer tree as follows:

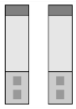
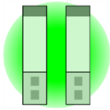
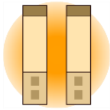
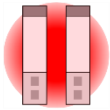
Icon	Description
	Offline
	Online (Both modules connected)
	Online – Partially (Only one of the modules connected)
	Offline – Modules were disconnected when previously online.

Table 9.3 – Project Explorer – Redundant Pair

The status of the individual FF Link modules in redundant mode is illustrated in the Project Explorer tree as follows:

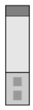
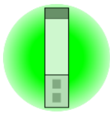
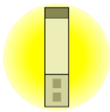
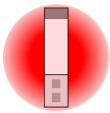
Icon	Description
	Offline
	Online and Active
	Online and Standby
	Offline – Module was disconnected when previously online.

Table 9.4 – Project Explorer – Redundant Module

9.2.3. FF Link Master

The Status monitoring window of the FF Link 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***.

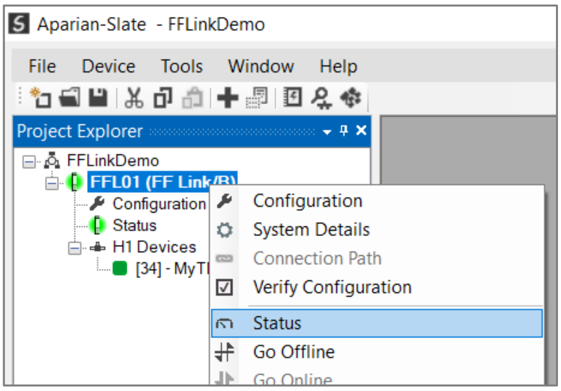


Figure 9.5 – Open Master Online Status

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

9.2.3.1. GENERAL

The ***General*** tab displays the following general parameters:

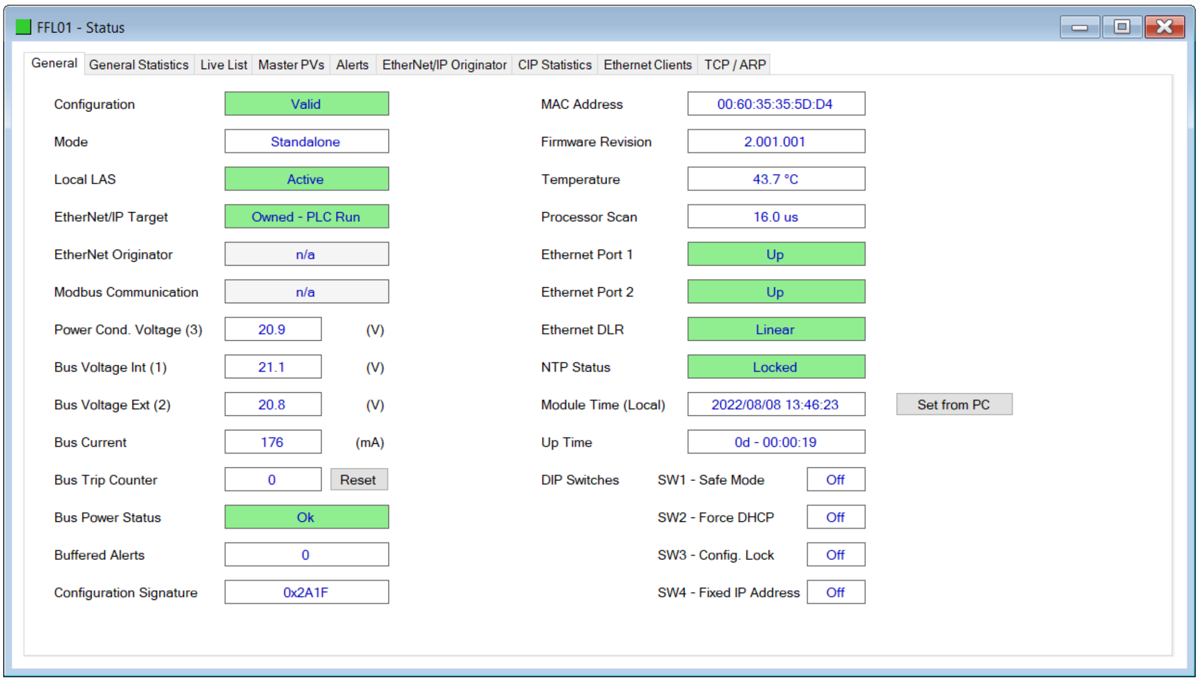


Figure 9.6 – Status monitoring - General

Parameter	Description
Config Valid	Indicates the state of the module's configuration: **Not Valid** The configuration is not valid. **Valid**

	The configuration is valid (standalone system) **Valid – Mismatch** The configuration is valid, but does not match that of its redundant partner. **Valid – Matched** The configuration is valid and matches that of its redundant partner.
Mode	This is the mode of operation of the module. The following states can be returned: **Quiet** This mode allows the user to connect the FF Link to an active bus and run an H1 packet capture. In this mode the FF Link will not communicate on the H1 Bus but rather only listen. **Standalone** In this mode the FF Link is the H1 Master on the H1 network. **Redundant - Active** The module is the **Active** H1 Master in a redundant configuration. **Redundant - Standby** The module is the **Standby** H1 Master in a redundant configuration.
Local LAS	The status of the LAS on the local FF Link **Active** The local FF Link is the active Link Active Scheduler (LAS) **Standby** The local FF Link is the backup Link Active Scheduler (LAS)
EtherNet/IP Target	Indicates whether or not the module is currently owned (Class 1) by a Logix Controller and the current mode of the controller. **Not Owned** No connection from Logix controller **Owned – PLC Prog** Controller connected but not in Run mode **Owned – PLC Run** Controller connected and in Run mode
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 Communication	When the module is operating as a **Modbus TCP Slave**, this parameter will indicate that the module has received a valid Modbus request within the Modbus inactivity time. When the module is operating as a **Modbus TCP Master**, this parameter will indicate that all the mapping items in the Modbus Auxiliary Map are executing correctly.
Power Cond Voltage	The internal isolated Power Conditioner voltage.

	NOTE: This voltage is still active even when the Power Conditioner is disabled.
Bus Voltage Int	The internal Bus Voltage. Measured before the bus protection circuitry.
Bus Voltage Ext	The external Bus Voltage. Measured on the port side of the bus protection circuitry.
Bus Current	The current supplied by the internal power conditioner when enabled. **NOTE**: This current is the sum of the external bus current and the internal FF Link MAU consumption. The latter accounts for 15-20 mA.
Bus Trip Counter	The number of occurrences of the power conditioner being tripped due to the total bus current exceeding the configured (Current Trip) limit.
Buffered Alerts	The number of H1 Alerts that have been buffered but has not yet been unloaded by the Primary Interface.
Configuration Signature	The signature of the configuration currently executing on the module.
MAC Address	Displays the module's unique Ethernet MAC address.
Firmware Revision	The application firmware revision currently executing.
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 1/2	This is the status of each Ethernet port. **Down** The Ethernet connector has **not been** successfully connected to an Ethernet network. **Up** The Ethernet connector has successfully connected to an Ethernet network. **Mirror Enabled** The Ethernet port is mirroring the traffic on the other Ethernet port.
Ethernet DLR (Device Level Ring)	The status of the Ethernet DLR. **Disabled** Device Level Ring functionality has been disabled. **Linear** The DLR functionality has been enabled and the Ethernet network architecture is linear. **Ring – Fault** The DLR functionality has been enabled and the Ethernet network architecture is ring, but there is a fault with the network. **Ring – Ok**

	The DLR functionality has been enabled and the Ethernet network architecture is ring and is operating as expected.
NTP Status	The status of the local NTP Client. **Disabled** The NTP time synchronization has been disabled. **Locked** NTP time synchronization has been enabled and the FF Link has locked onto the target time server. **Not Locked** NTP time synchronization has been enabled and the FF Link has not locked onto the target time server.
Up Time	Indicates the elapsed time since the module was powered-up.
DIP Switch Position	The status of the DIP switches when the module booted. SW1 – Safe Mode SW2 – Force DHCP SW3 – Configuration Lock SW4 – Fixed IP Address

Table 9.5 - Parameters displayed in the Status Monitoring – General Tab

9.2.3.2. GENERAL STATISTICS

The ***General Statistics*** tab displays the following general parameters:

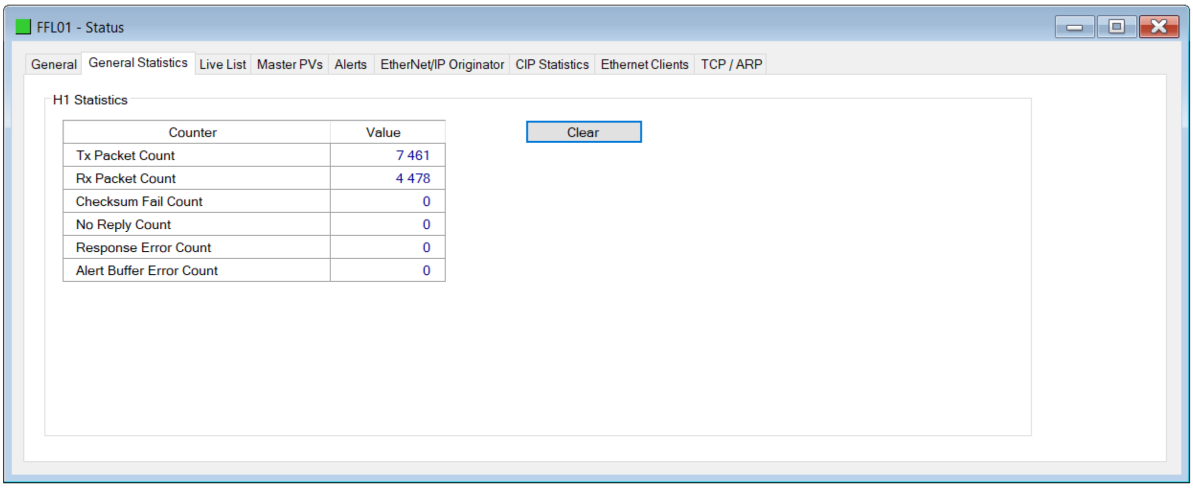


Figure 9.7 – Status monitoring – General Statistics

Parameter	**Description**
Tx Packet Count	The number of H1 packets transmitted.
Rx Packet Count	The number of H1 packets received.

Checksum Failed Packet Count	The number of H1 packets that had a failed checksum.
No Reply Count	The number of H1 requests from the FF Link where the station did not respond.
Response Error Count	The number of H1 responses that were incorrect or have failed (e.g., FMS response with an error).
Alert Buffer Error Count	The number of Alert errors occurred (e.g., Alert Buffer Overflow, Alert received while Alert Extraction is disabled, etc.)

Table 9.6 - Status Monitoring – General Statistics Tab

9.2.3.3. LIVE LIST

The ***Live List*** tab in the FF Link status monitoring provides the user with an overview of all H1 devices connected to the H1 network. If an H1 device has been configured, then the Live List will indicate if the configured DD and Device revisions match that of the live H1 device. If an H1 device has not been configured in the FF Link H1 Device tree, then it will indicate that it is online but not configured.

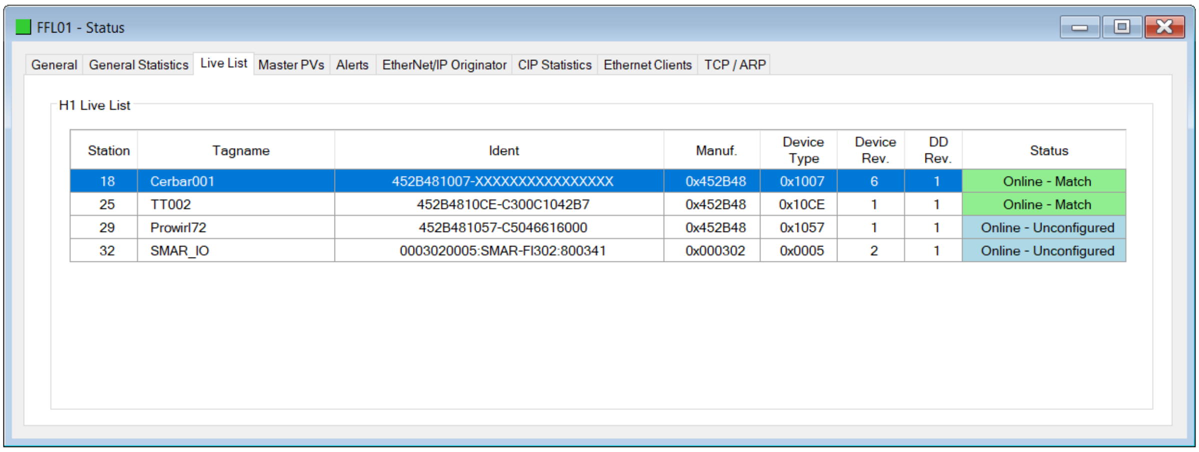


Figure 9.8 – Status monitoring – Live List

From the Live List the H1 device Station Address and Tag can be changed. See section 3.15.1 for more details. H1 Device can also be added to the FF Link H1 Device tree. See section 3.6.2 for more details.

9.2.3.4. MASTER PVS

The ***Master PVs*** tab displays all the H1 process variables that are sent from the FF Link to H1 Devices. These process variables will be updated by the Primary Interface.

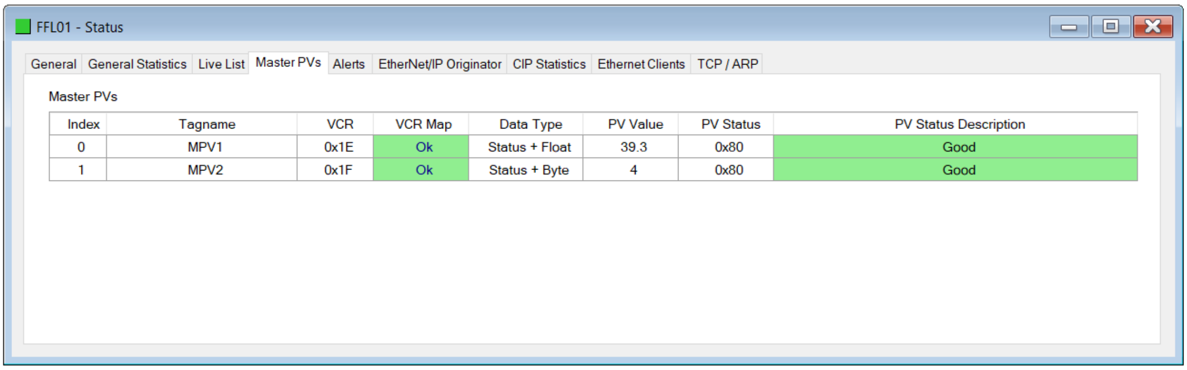


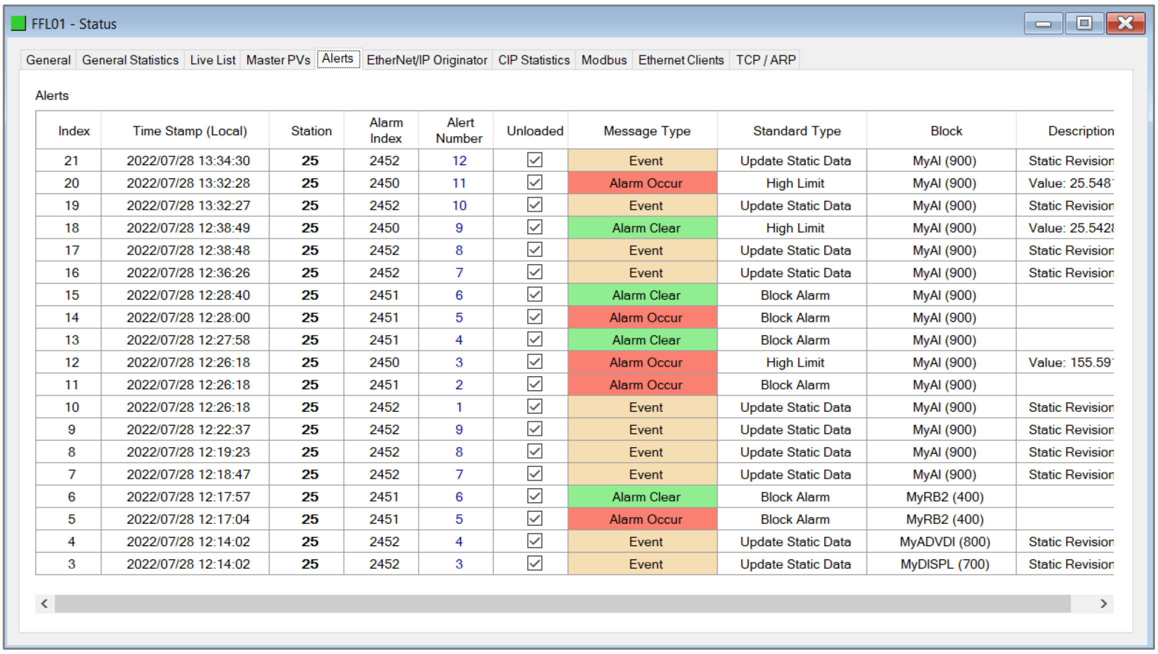
Figure 9.9 – Status monitoring – Master PVs

Parameter	**Description**
Index	The PV Index is assigned in the order that is was instantiated.
Tagname	Name of the consumer connector used in the block editor in one of the field devices.
VCR	The VCR number used on the H1 network.
VCR Map	When using EtherNet/IP target, this will indicate if the mapping in Logix is valid. **NOTE:** This is not applicable for Modbus TCP Master/Slave or EtherNet/IP Originator.
Data Type	The PV data type either: **Status + Float** or **Status + Byte**.
PV Value	The process variable value that was updated by the Primary Interface.
PV Status	The process variable status that was updated by the Primary Interface.

Table 9.7 - Status Monitoring – Master PVs

9.2.3.5. ALERTS

The ***Alerts*** tab displays the most recent 255 H1 Alerts received by the FF Link (regardless if the alert has been unloaded or not by the Primary Interface).



Index	Time Stamp (Local)	Station	Alarm Index	Alert Number	Unloaded	Message Type	Standard Type	Block	Description
21	2022/07/28 13:34:30	25	2452	12	☑	Event	Update Static Data	MyAI (900)	Static Revision
20	2022/07/28 13:32:28	25	2450	11	☑	Alarm Occur	High Limit	MyAI (900)	Value: 25.548
19	2022/07/28 13:32:27	25	2452	10	☑	Event	Update Static Data	MyAI (900)	Static Revision
18	2022/07/28 12:38:49	25	2450	9	☑	Alarm Clear	High Limit	MyAI (900)	Value: 25.542
17	2022/07/28 12:38:48	25	2452	8	☑	Event	Update Static Data	MyAI (900)	Static Revision
16	2022/07/28 12:36:26	25	2452	7	☑	Event	Update Static Data	MyAI (900)	Static Revision
15	2022/07/28 12:28:40	25	2451	6	☑	Alarm Clear	Block Alarm	MyAI (900)	
14	2022/07/28 12:28:00	25	2451	5	☑	Alarm Occur	Block Alarm	MyAI (900)	
13	2022/07/28 12:27:58	25	2451	4	☑	Alarm Clear	Block Alarm	MyAI (900)	
12	2022/07/28 12:26:18	25	2450	3	☑	Alarm Occur	High Limit	MyAI (900)	Value: 155.59
11	2022/07/28 12:26:18	25	2451	2	☑	Alarm Occur	Block Alarm	MyAI (900)	
10	2022/07/28 12:26:18	25	2452	1	☑	Event	Update Static Data	MyAI (900)	Static Revision
9	2022/07/28 12:22:37	25	2452	9	☑	Event	Update Static Data	MyAI (900)	Static Revision
8	2022/07/28 12:19:23	25	2452	8	☑	Event	Update Static Data	MyAI (900)	Static Revision
7	2022/07/28 12:18:47	25	2452	7	☑	Event	Update Static Data	MyAI (900)	Static Revision
6	2022/07/28 12:17:57	25	2451	6	☑	Alarm Clear	Block Alarm	MyRB2 (400)	
5	2022/07/28 12:17:04	25	2451	5	☑	Alarm Occur	Block Alarm	MyRB2 (400)	
4	2022/07/28 12:14:02	25	2452	4	☑	Event	Update Static Data	MyADVDI (800)	Static Revision
3	2022/07/28 12:14:02	25	2452	3	☑	Event	Update Static Data	MyDISPL (700)	Static Revision

Figure 9.10 – Status monitoring – H1 Alerts

Parameter	Description
Index	Index number of the H1 Alert in the rolling buffer.
Time Stamp	The time stamp of the H1 Alert when it occurred. Note that this time stamp is read from the H1 device that created the H1 Alert.
Station	The H1 station number of the H1 device that created the H1 alert.
Alert Index	The object index of the H1 Alert.
Alert Number	The H1 Alert Number that is used by the Primary Interface to acknowledge the Alert.
Unloaded	Indication if the Alert has been unloaded by the Primary Interface.
Message Type	H1 Alert Message Type. One of the following options: **Undefined** **Event** **Alarm Clear** **Alarm Occur** **Bit Clear** **Bit Occur**
Standard Type	H1 Alert Standard Type. One of the following options: **Undefined** **Low Limit** **High Limit** **Critical Low Limit** **Critical High Limit** **Deviation Low**

	Deviation High **Discrete** **Block Alarm** **Update Static Data** **Write Protect Changed** **Update Link** **Update Trend** **Ignore** **Reset** **Fail Alarm** **Off Spec Alarm** **Maint. Alarm** **Check Alarm** **SIF Error Alarm**
Block	The block in the H1 device that generated the Alert.
Description	A description of the Alert.

Table 9.8 - Status Monitoring – Alerts

9.2.3.6. ETHERNET/IP EXPLICIT

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

NOTE: This tab is only applicable when the module has the ***Primary Interface*** set to ***EtherNet/IP Originator***.

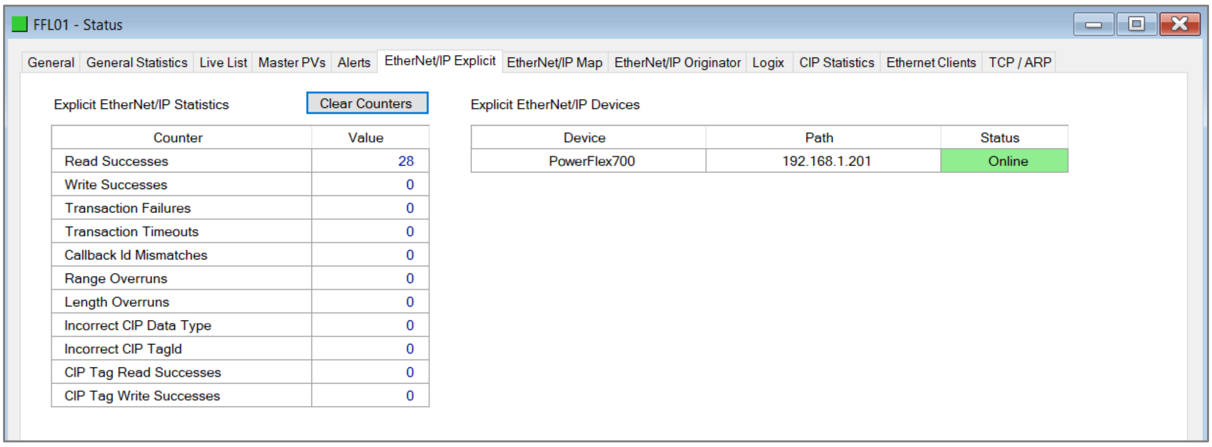


Figure 9.11 - Status monitoring – EtherNet/IP Explicit

Statistic	**Description**
Read Successes	The number of successful reads from the target EtherNet/IP device.
Write Successes	The number of successful writes to the target EtherNet/IP device.

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

Table 9.9 – EtherNet/IP Explicit Statistics

9.2.3.7. ETHERNET/IP MAP

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

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

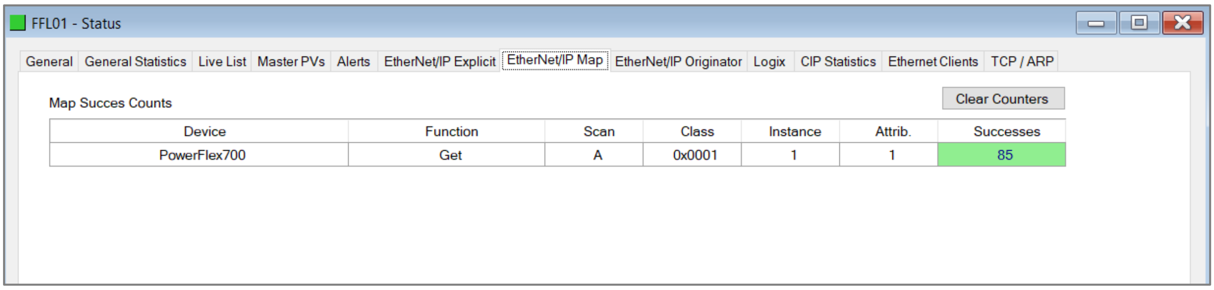


Figure 9.12 - Status monitoring – EtherNet/IP Map

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

9.2.3.8. ETHERNET/IP ORIGINATOR

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

NOTE: This tab is only applicable when the module has the ***Primary Interface*** set to ***EtherNet/IP Originator***.

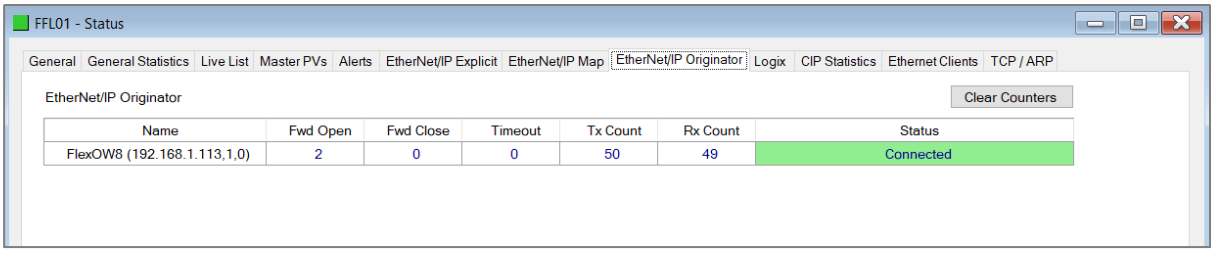


Figure 9.13 - Status monitoring – EtherNet/IP Originator

Statistic	**Description**
Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Class 1 cyclic communication. **Offline** The device it offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection reject, for example:  
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 times this connection was closed due to timeouts.
Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.

Table 9.10 – EtherNet/IP Class 1 status and statistics

9.2.3.9. LOGIX

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

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

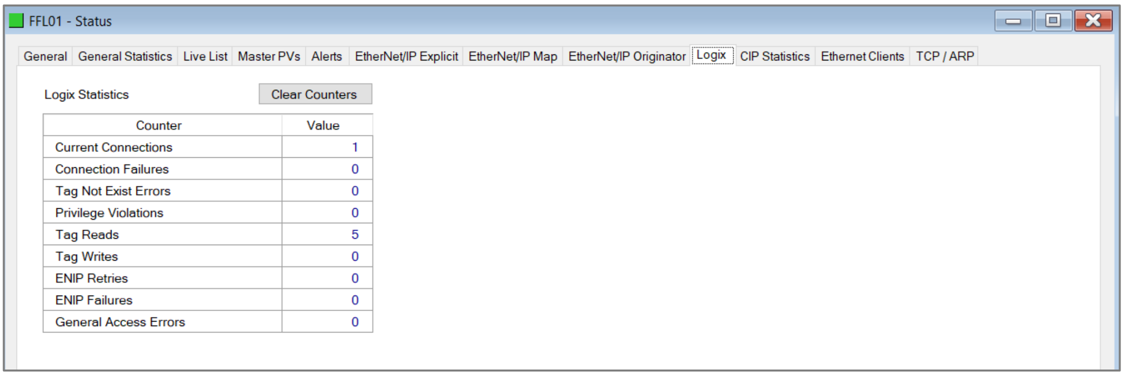


Figure 9.14 - Status monitoring – Logix Statistics

Parameter	**Description**
Current Connections	The number of current open class 3 connections.
Connection Failures	The number of failed attempts at establishing a class 3 connection with a Logix controller.
Tag Not Exist Errors	The number of tag read, and tag write transactions that failed due to the destination tag **not existing**.
Privilege Violations	The number of tag read, and tag write transactions that failed due to a **privilege violation** error. Note: This may be caused by the ***External Access*** property of the Logix tag being set to either ***None*** or ***Read Only***.
Tag Reads	The number of tag read transactions executed by the module.**Error! Bookmark not defined.**
Tag Writes	The number of tag write transactions executed by the module.
ENIP Retries	This count increases when no response is received from the Logix Controller within the ENIP timeout.
ENIP Failures	This count increases when the ENIP Retry Limit is reached, following successive no response been received from the Logix Controller.
Tag Access General Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 9.11 – Logix Statistics Tab

9.2.3.10. CIP STATISTICS

The ***CIP Statistics*** tab displays the Ethernet CIP statistics.

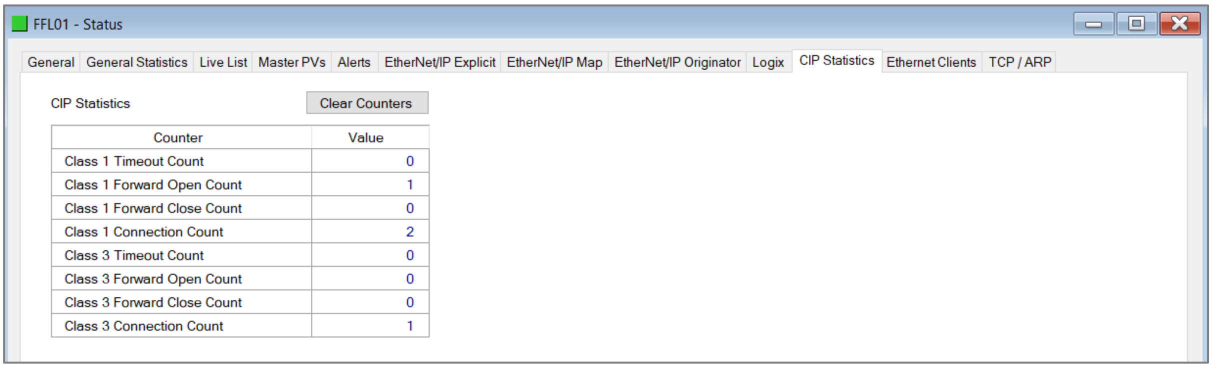


Figure 9.15 - Status monitoring – CIP Statistics

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

Table 9.12 – CIP statistics

9.2.3.11. MODBUS STATISTICS

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

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

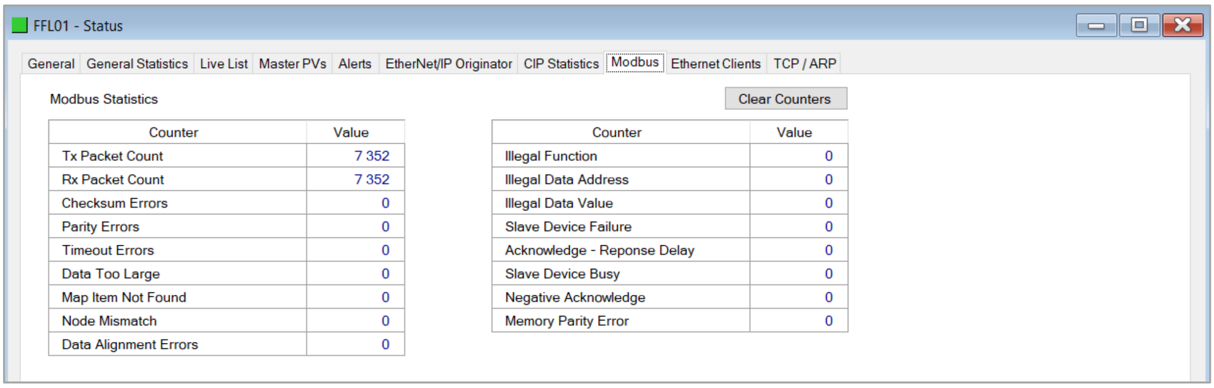


Figure 9.16 – FF Link Status monitoring – Modbus Statistics

The Modbus tab displays the following parameters:

Statistic	Description
Tx Packet Count	The number of Modbus packets sent by the module.
Rx Packet Count	The number of Modbus packets received by the module.
Checksum errors	The number of corrupted Modbus packets received by the module.
Parity errors	The number of bytes with parity errors received by the module.
Timeout Errors	The number of message response timeouts the module has encountered.
Data Too Large	The number of Modbus requests or responses where the data was too large to process.
Register Not Valid	The number of Modbus requests containing an invalid register.
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.
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.

Table 9.13 – Modbus statistics

9.2.3.12. ETHERNET CLIENTS

The ***Ethernet Clients*** tab displays details of the Ethernet and EtherNet/IP clients connected to the FF Link.

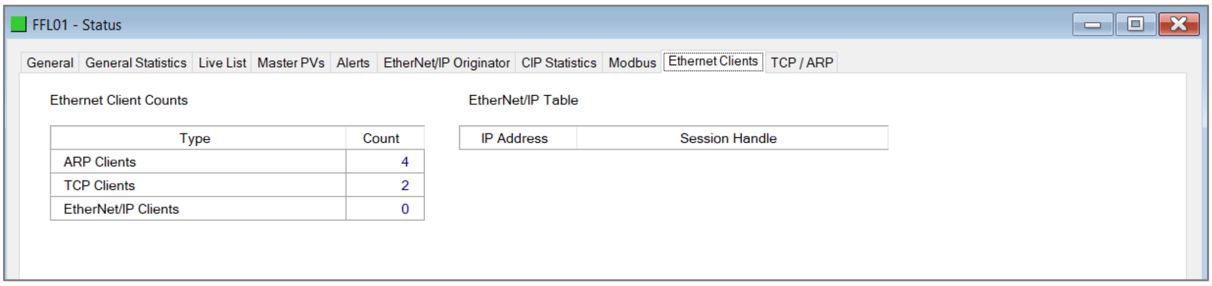


Figure 9.17 – FF Link Status monitoring – Ethernet Client Statistics

9.2.3.13. TCP/ARP

The ***TCP/ARP*** tab displays details of the internal Ethernet ARP and TCP lists of the FF Link.

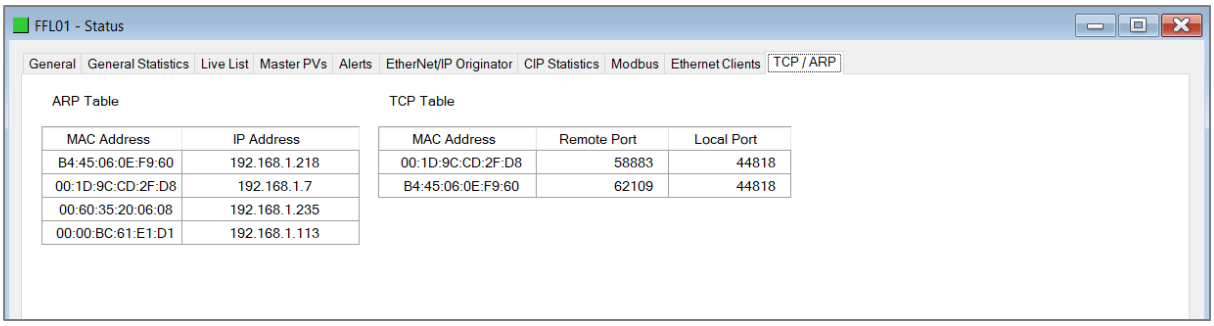


Figure 9.18 – FF Link Status monitoring – Ethernet TCP / ARP Statistics

9.2.4. H1 DEVICE STATUS

The Status monitoring window of each H1 device connected to the FF Link can be opened by right-clicking on the specific H1 device in Slate tree and selecting ***Status***.

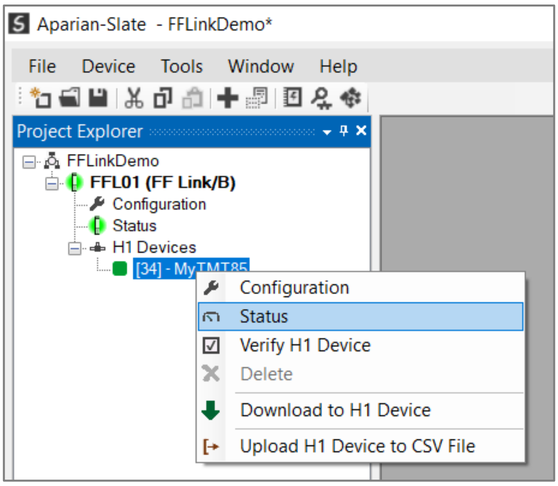


Figure 9.19 - Selecting H1 device online Status

The device status window contains multiple tabs to display the current status of the specific H1 device.

9.2.4.1. GENERAL

The ***General*** tab displays the following general parameters:

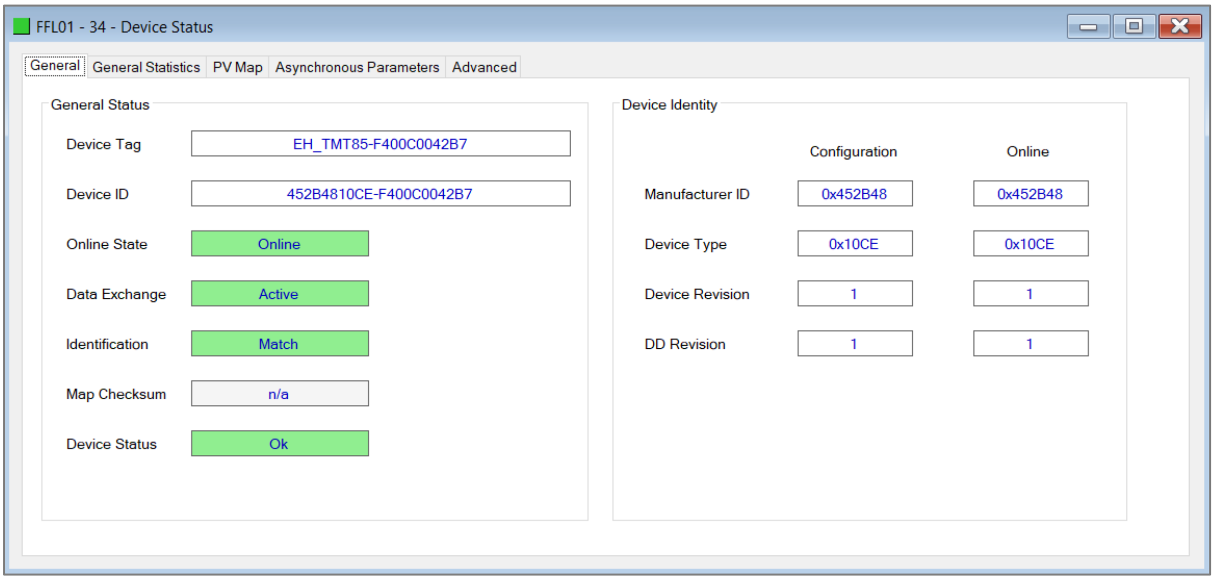


Figure 9.20 – Device Status monitoring - General

Parameter	**Description**
General Status	
Device Tag	The online Device Tag reported by the H1 device
Device ID	The online unique Device ID reported by the H1 device.

Online State	The online state of the H1 device, either: **Online** – Communicating on the H1 bus **Offline** – Not available on the H1 bus.
Data Exchange	Indicates if the H1 device is participating in data exchange: **Active** – Data Exchanging is Active **Not Active** – Date not being exchanged
Identification	Indicates whether the configured device for that Station Address matches that of the online device. **Match** – Matching device **Mismatch** – A mismatch between the configured and actual device.
Map Checksum	Indicates whether the mapping checksum in the FF Link master matches that of the Logix configuration: **Match** – The mapping matches. **Mismatch** – The mapping does not match (The L5X Generation and import into Studio5000 is required.) **n/a** – Not applicable (***Primary Interface*** is not ***EtherNet/IP Target***)
Device Identity (Configuration and Online)	
Manufacturer ID	The unique Manufacturer ID displayed in hexadecimal.
Device Type	The unique Device Type ID displayed in hexadecimal.
Device Revision	The revision of the H1 device.
DD Revision	The revision of the H1 device’s matching Device Description file.

Table 9.14 - Device Status Monitoring – General Tab

9.2.4.2. STATISTICS

The ***General Statistics*** tab displays the following device-specific H1 statistics:

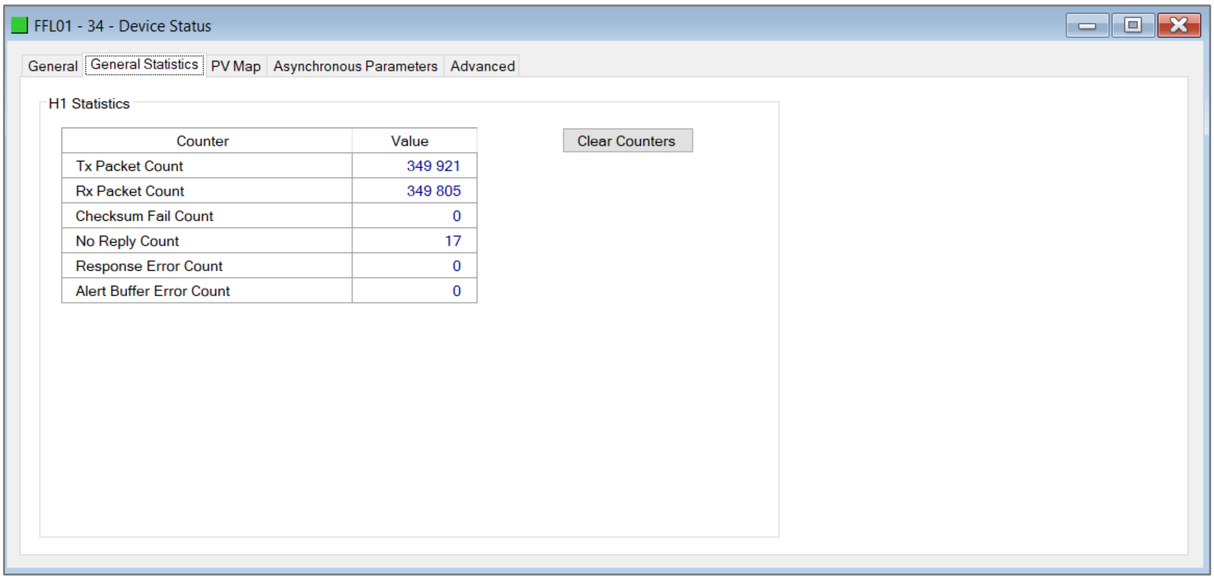


Figure 9.21 – Device Status monitoring - Statistics

Parameter	Description
Tx Packet Count	The number of H1 packets transmitted to this device.
Rx Packet Count	The number of H1 packets received from this device.
Checksum Failed Packet Count	The number of H1 packets that had a failed checksum.
No Reply Count	The number of H1 requests from the FF Link where this station did not respond.
Response Error Count	The number of H1 responses from this device that were incorrect or have failed (e.g., a FMS response with an error).
Alert Buffer Error Count	The number of Alert errors occurred (e.g., Alert Buffer Overflow, Alert received while Alert Extraction is disabled, etc.)

Table 9.15 - Device Status Monitoring – Statistics Tab

9.2.4.3. PV MAP

The ***PV Map*** tab displays the process variables configured and returned by the H1 device:

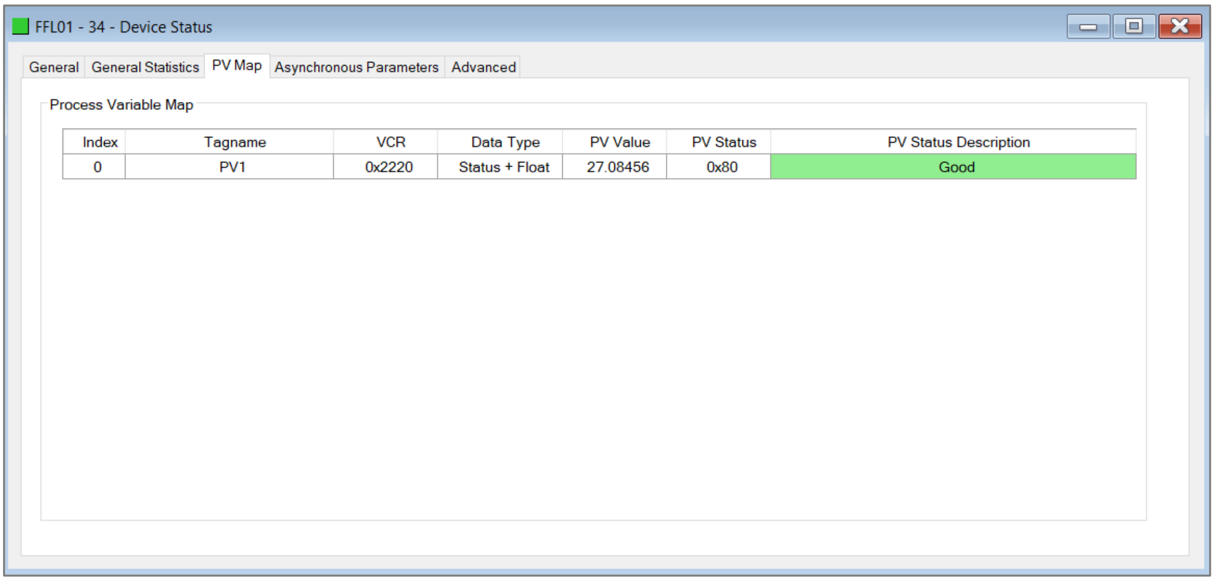


Figure 9.22 – Device Status monitoring – PV Map

Parameter	Description
Tagname	Name of the producer connector used in the block editor in one of the field devices.
VCR	The VCR number configured for the H1 device Process Variable on the H1 network.
Data Type	The PV data type, either **Status + Float** or **Status + Byte**.
PV Value	The process variable value that was received by the H1 device.
PV Status	The process variable status that was received by the H1 device.

Table 9.16 - Device Status Monitoring – PV Map

9.2.4.4. ASYNCHRONOUS PARAMETERS

The ***Asynchronous Parameters*** status page shows the status of each asynchronous parameter configured for the specific H1 device. See section 3.6.3.5 for more information.

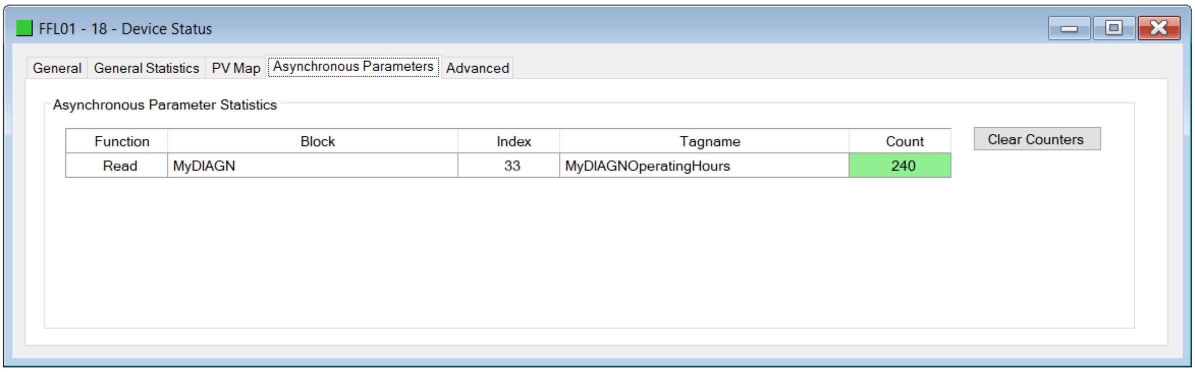


Figure 9.23 – Device Status monitoring – Asynchronous Parameters

Parameter	Description
Function	The function can either **read** data from the H1 device or **write** data to the H1 device.
Block	The name of the block used for this asynchronous parameter access.
Index	The (parameter) index in the block used for this asynchronous parameter access.
Tagname	The tagname assigned to the asynchronous parameter.
Count	The number of successful executions of the asynchronous parameter.

Table 9.17 - Device Status Monitoring – Asynchronous Parameters

9.2.4.5. ADVANCED STATUS

The Advanced Status page in the H1 device status window provides various information that could assist the user when operating the FF Link.

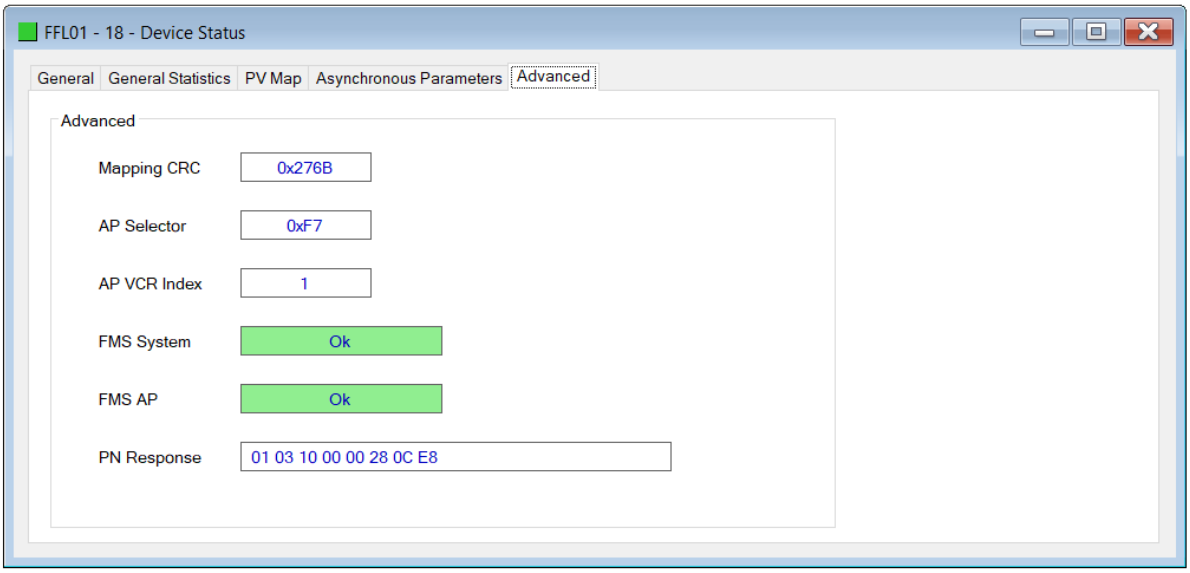


Figure 9.24 – Device Status monitoring – Advanced

Parameter	Description
Mapping CRC	The mapping CRC when using EtherNet/IP Target Logix mapping. **Note**: This is used to ensure the Logix mapping code is matched to the FF Link configuration.
AP Selector	The AP selector used for AP connections by the FF Link.
AP VCR Index	The AP VCR index used for AP connections by the FF Link.
FMS System	FMS System update status.
FMS AP	FMS AP update status.
PN Response	Most recent Probe Node response data.

Table 9.18 - Device Status Monitoring – Asynchronous Parameters

9.3. EXPLICIT MESSAGING UTILITY

Slate provides a utility to initiate explicit messages with an H1 device. This utility supports both read and write commands to either the System or AP VFD of the H1 device.

To open the utility, right-click on the H1 device and select the ***Explicit Messaging*** option.

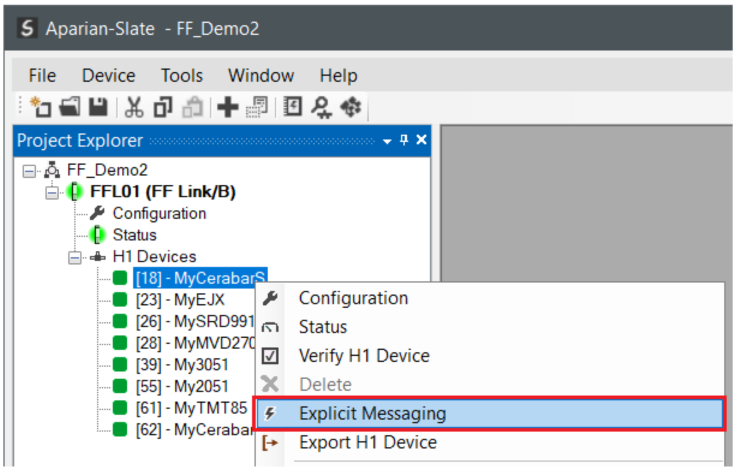


Figure 9.25 - Selecting Explicit Messaging

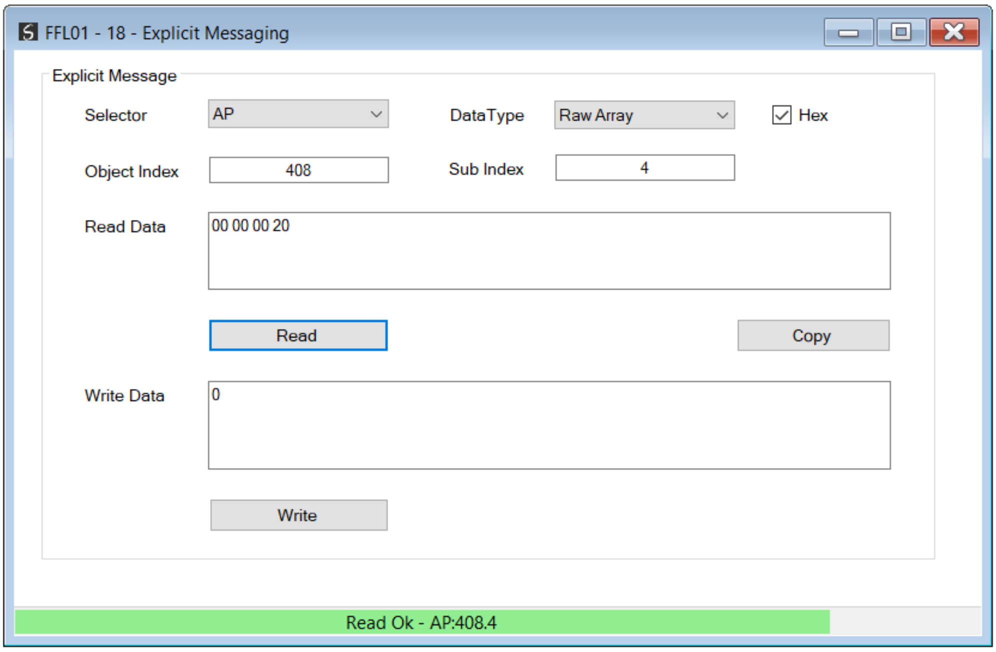


Figure 9.26 - Explicit Messaging utility

The utility provides the following fields:

Field	Description
Selector	The VFD selector, either System or AP (Application Process)
Object Index	The required Object Index.
Sub Index	The required Object Sub Index. Where not applicable, or to select the entire Object Index, set this field to 0.
Data Type	The Data Type or format of the Read Data Write Data.
Hex	This checkbox selects the hexadecimal format.
Read Data	The resulting data reply from a Read command, formatted using the Data Type and Hex options.
Write Data	The data to be written to the device using the Write command. The data is expected to be in the format specified by the Data Type and Hex fields.

Table 9.19 – Explicit Utility fields

Once the required data has been entered into the relevant fields, the user can then select the Read or Write command buttons.

The status of the message (Success or Error Command, will then be displayed in status bar at the bottom of the utility.

9.4. H1 PACKET CAPTURE

The module provides the capability to capture the H1 bus traffic for analysis. This will allow the user and the support team to view the packet stream. To invoke the capture of the module, right-click on the FF Link module in Slate and select the ***H1 Packet Capture*** option.

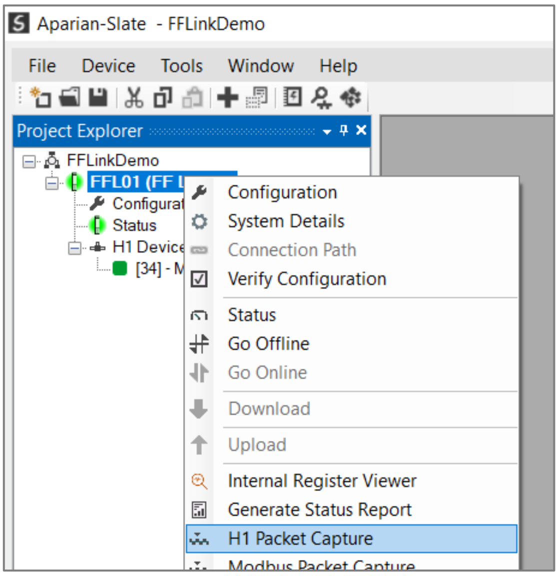


Figure 9.27 - Selecting H1 packet Capture

The ***H1 Packet Capture*** window will open and automatically start capturing all H1 packets.

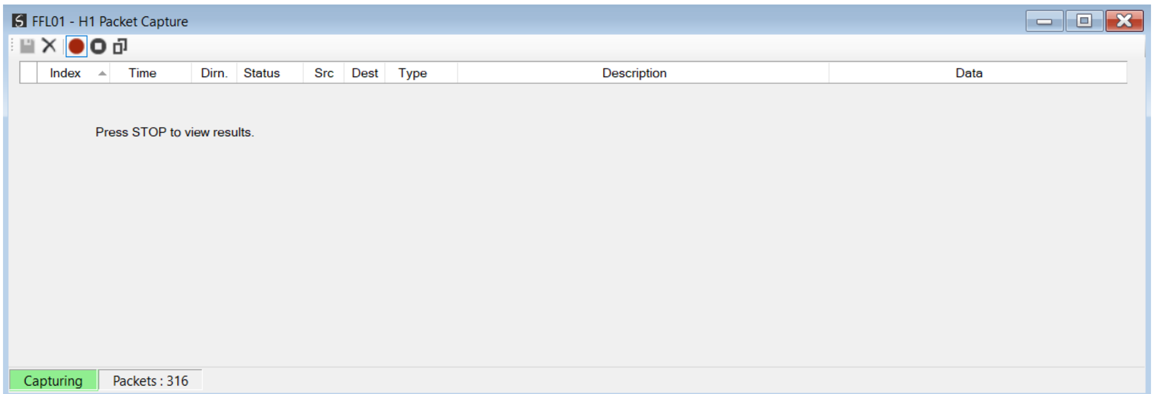


Figure 9.28 - H1 packet capture

When the capture process is stopped then the H1 capture will be presented as shown below.

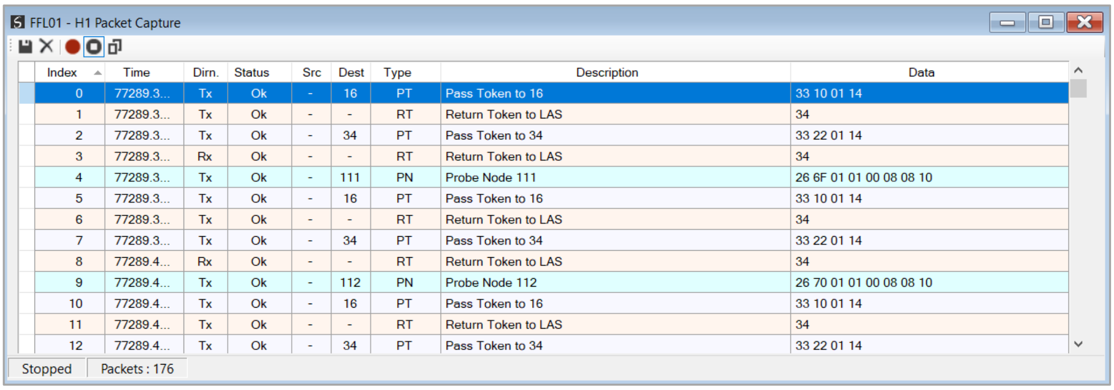


Figure 9.29 - H1 packet Capture complete

The captured H1 packets are tabulated as follows:

Field	Description
Index	The packet index incremented for each packet sent or received.
Time	The time is measured in microseconds (us) and formatted in seconds.
Dirn.	The direction of the packet, either transmitted (Tx) or received (Rx).
Status	The status of the packet. Received packets are checked for valid H1 constructs and valid checksums.
Src	H1 station address of the message source.
Dest	H1 station address of the message destination.
Type	The H1 function type (e.g. Token, Request, etc.)
Description	A more detailed description of the packet payload. Only applicable to specific packet types.
Data	The packet's raw data displayed in space delimited hex.

Table 9.20 - H1 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 H1 packet Capture files can be viewed by selecting the ***Foundation Fieldbus H1*** option under the ***Packet Capture Viewer*** option in the ***Tools*** menu.

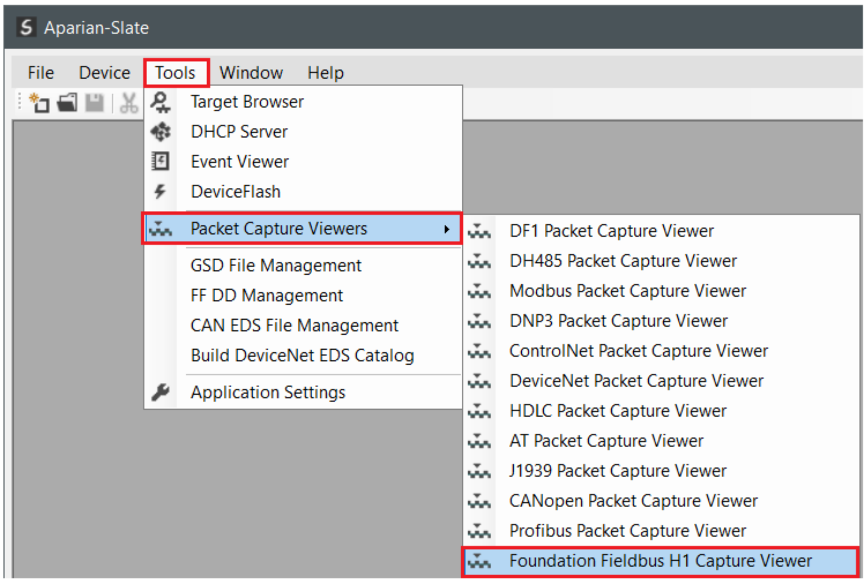


Figure 9.30 - Selecting the H1 packet Capture Viewer

9.5. MODBUS PACKET CAPTURE

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

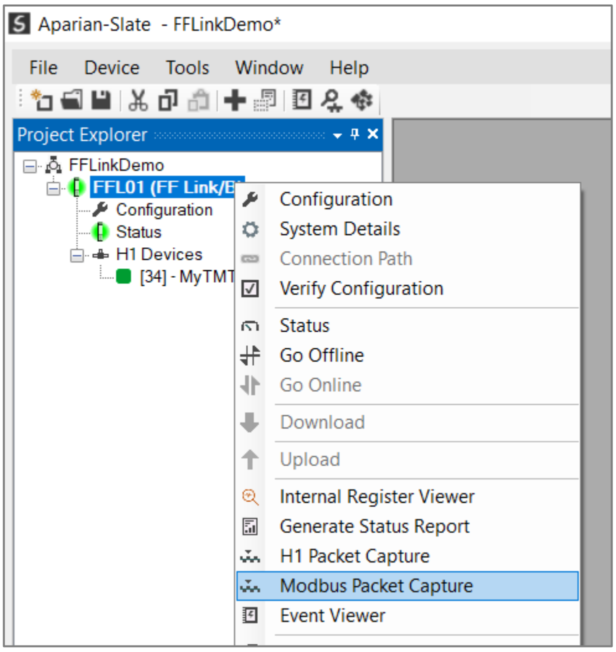


Figure 9.31 - Selecting Modbus Packet Capture

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

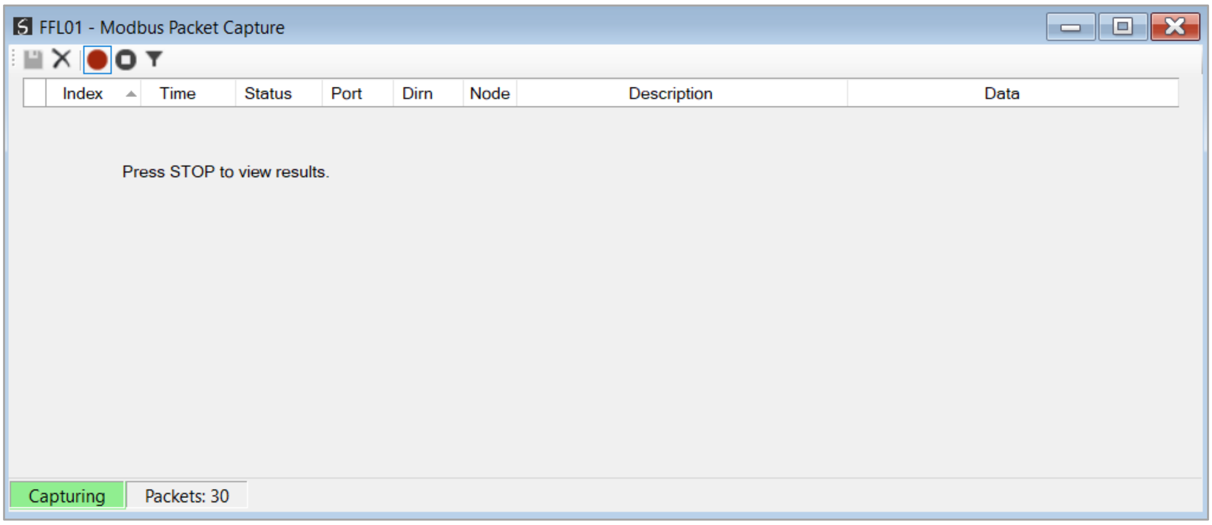


Figure 9.32 – Modbus packet capture

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

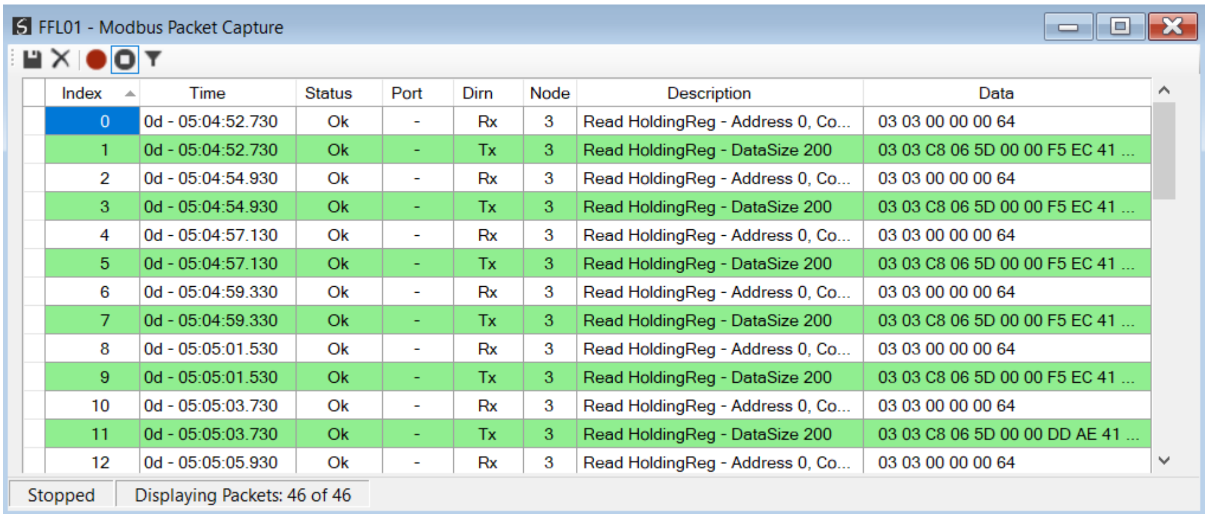


Figure 9.33 – Modbus Packet Capture complete

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)
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).

Node	The Source Node address for the packet
Description	Description of the packet that was received.
Data	The raw packet data.

Table 9.21 – 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 under the ***Packet Capture Viewer*** option in the ***Tools*** menu in the tools menu.

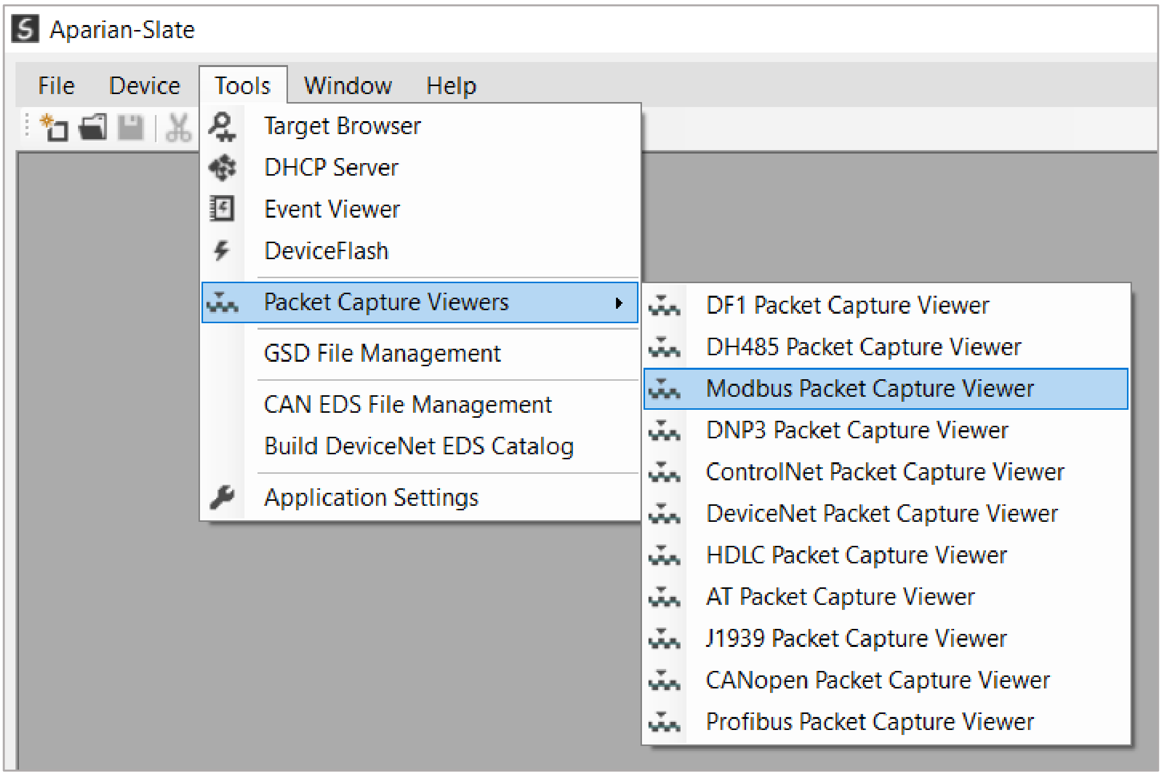


Figure 9.34 - Selecting the Modbus Packet Capture Viewer

9.6. MODULE EVENT LOG

The FF Link 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, right-click on the module in the project Explorer and select the ***Event Viewer*** option.

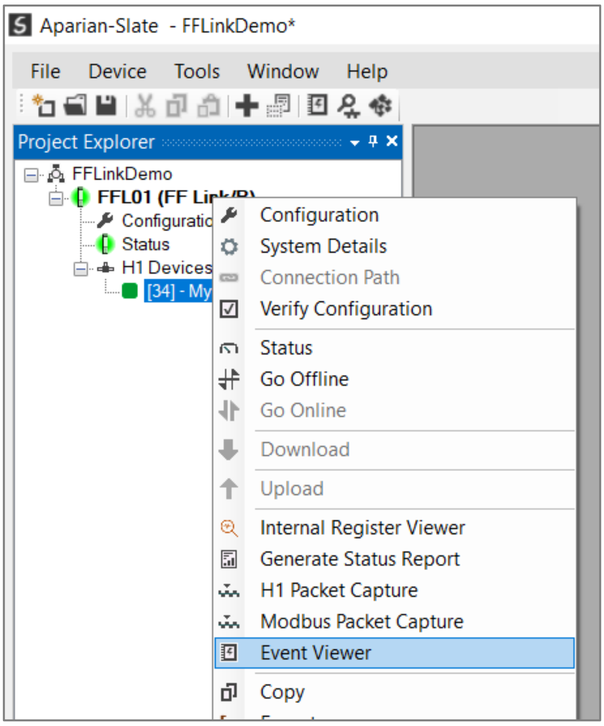
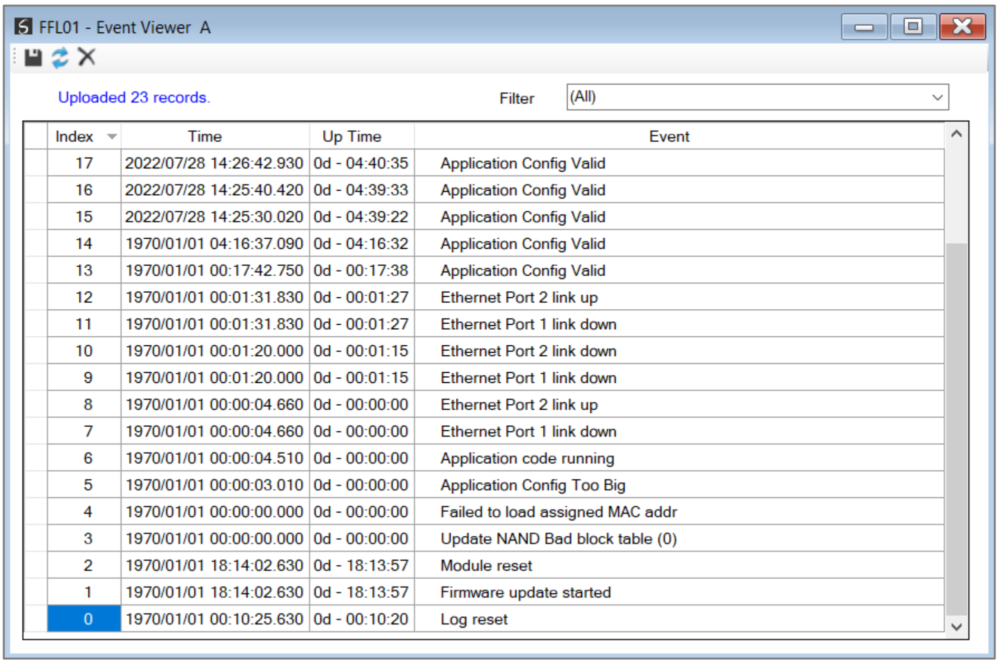


Figure 9.35. - Selecting the module Event Log

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



Index	Time	Up Time	Event
17	2022/07/28 14:26:42.930	0d - 04:40:35	Application Config Valid
16	2022/07/28 14:25:40.420	0d - 04:39:33	Application Config Valid
15	2022/07/28 14:25:30.020	0d - 04:39:22	Application Config Valid
14	1970/01/01 04:16:37.090	0d - 04:16:32	Application Config Valid
13	1970/01/01 00:17:42.750	0d - 00:17:38	Application Config Valid
12	1970/01/01 00:01:31.830	0d - 00:01:27	Ethernet Port 2 link up
11	1970/01/01 00:01:31.830	0d - 00:01:27	Ethernet Port 1 link down
10	1970/01/01 00:01:20.000	0d - 00:01:15	Ethernet Port 2 link down
9	1970/01/01 00:01:20.000	0d - 00:01:15	Ethernet Port 1 link down
8	1970/01/01 00:00:04.660	0d - 00:00:00	Ethernet Port 2 link up
7	1970/01/01 00:00:04.660	0d - 00:00:00	Ethernet Port 1 link down
6	1970/01/01 00:00:04.510	0d - 00:00:00	Application code running
5	1970/01/01 00:00:03.010	0d - 00:00:00	Application Config Too Big
4	1970/01/01 00:00:00.000	0d - 00:00:00	Failed to load assigned MAC addr
3	1970/01/01 00:00:00.000	0d - 00:00:00	Update NAND Bad block table (0)
2	1970/01/01 18:14:02.630	0d - 18:13:57	Module reset
1	1970/01/01 18:14:02.630	0d - 18:13:57	Firmware update started
0	1970/01/01 00:10:25.630	0d - 00:10:20	Log reset

Figure 9.36. – Module Event Log

The log can also be stored to a file for future analysis, by selecting the ***Save*** button in the tool menu.

To view previously saved files, use the ***Event Log Viewer*** option under the ***Tools*** menu.

9.7. WEB SERVER

The FF Link provides a web server allowing a user without Slate, Logix, or Modbus device to view various diagnostics of the module.

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

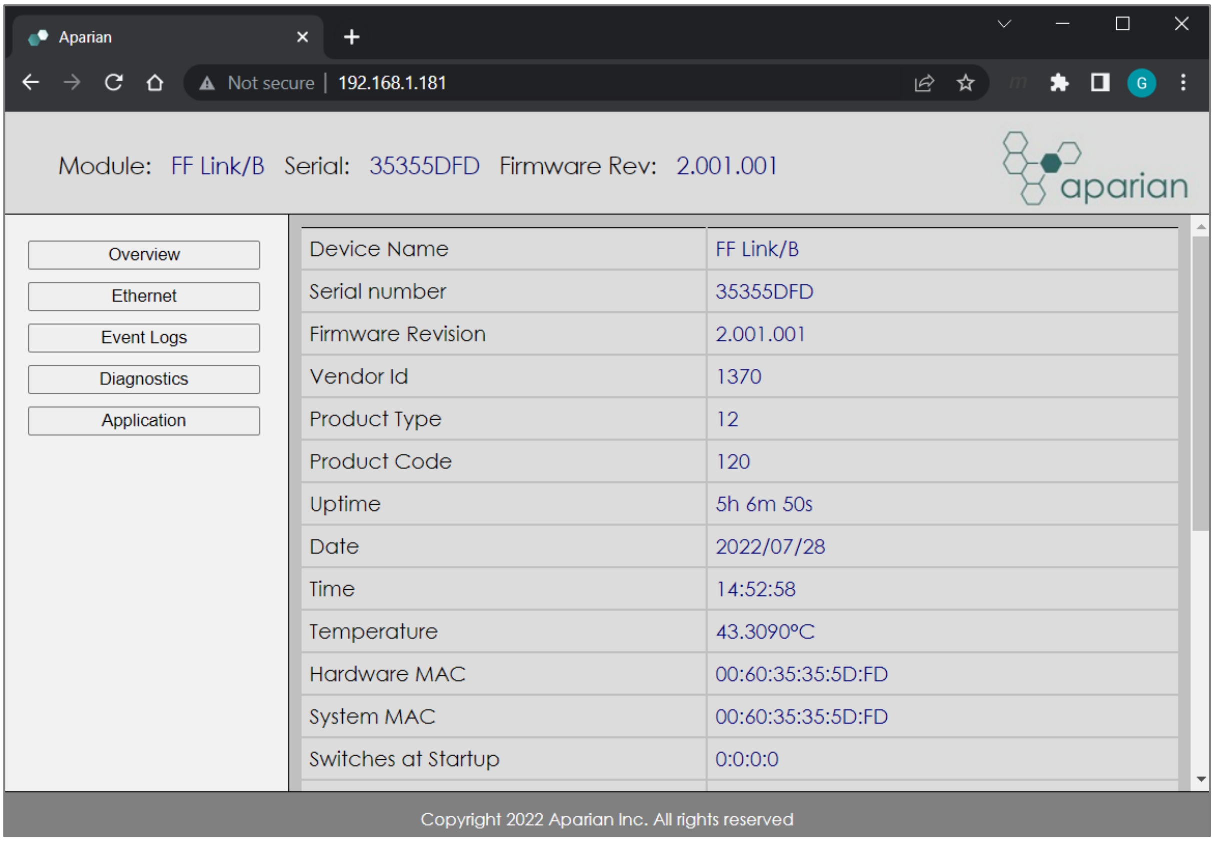


Figure 9.37 - Web interface

NOTE: The parameters and diagnostics in the webserver will match those in Slate status monitoring of the FF Link.

9.8. INTERNAL REGISTER VIEWER

The ***Internal Register Viewer*** utility is provided to view any of the Modbus Register as well as the Internal Data Space (IDS). The Internal Register Viewer can be opened by right-clicking on the module in Slate and selecting the ***Internal Register Viewer*** option.

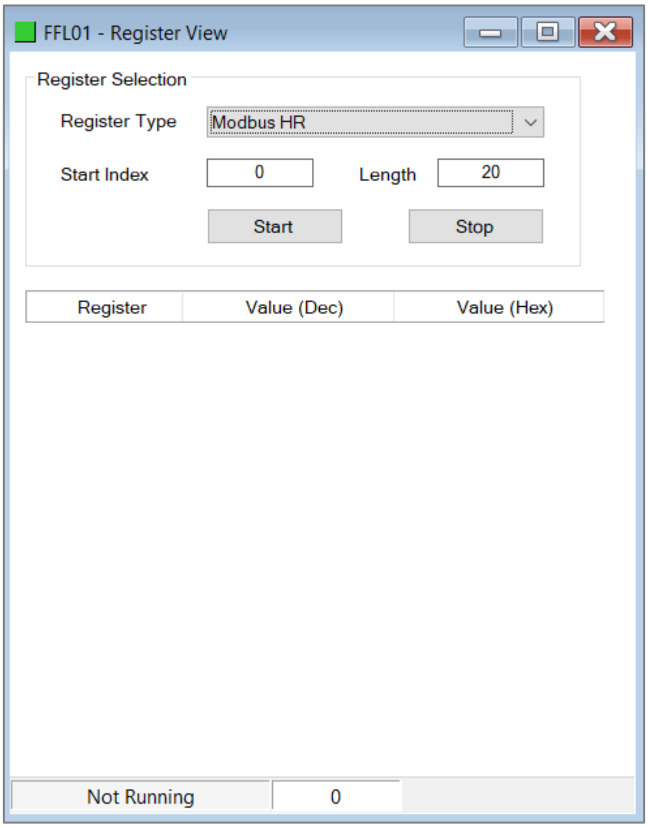


Figure 9.38 – Internal register viewer

The user can select to view ***Modbus Registers*** (***HR***, ***IR***, ***CS***, or ***IS***) or the ***Internal Data Space*** from the Register Type. Next the user will need to select the ***Start Index*** and the ***Length*** (number of elements to read).

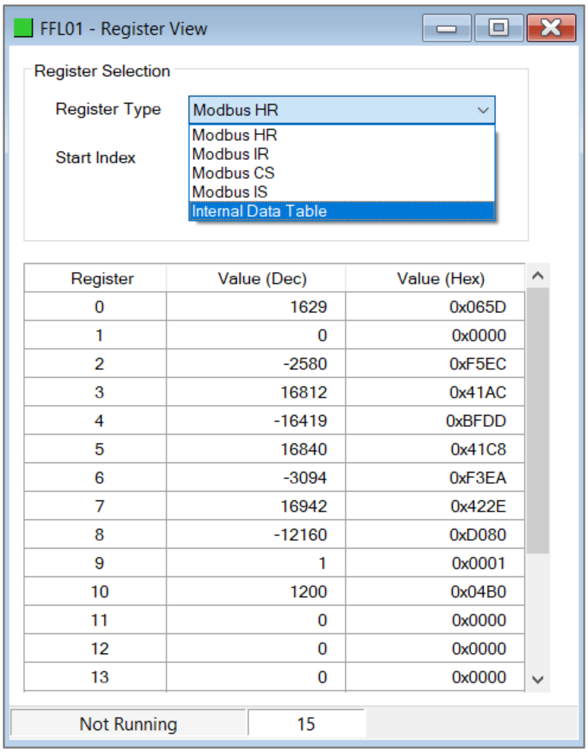


Figure 9.39 – Internal register viewer – Register Type Selection

The Internal Register Viewer will start reading and displaying the selected Register Type and Range once the user presses the ***Start*** button.

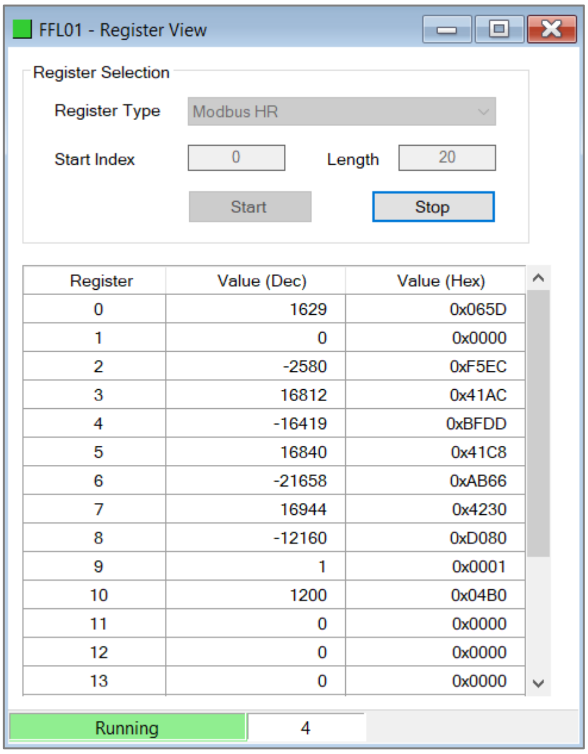


Figure 9.40 – Internal register viewer – Running

9.9. MODULE STATUS REPORT

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

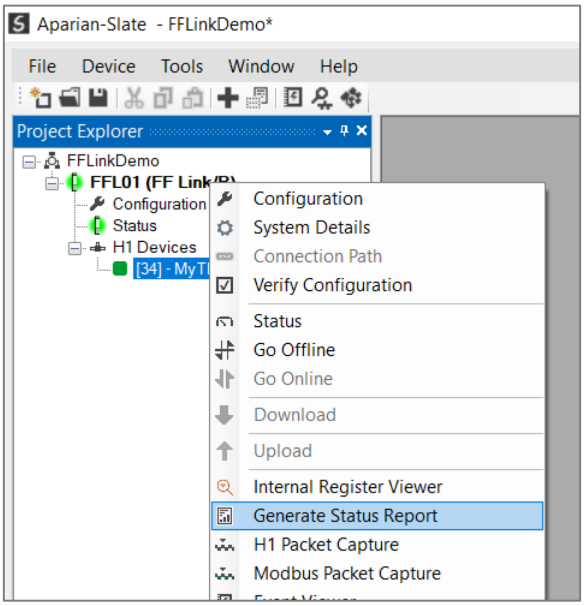


Figure 9.41 – Module - Generate Status Report

10.TECHNICAL SPECIFICATIONS

10.1. DIMENSIONS

Below are the enclosure dimensions. All dimensions are in millimetres.

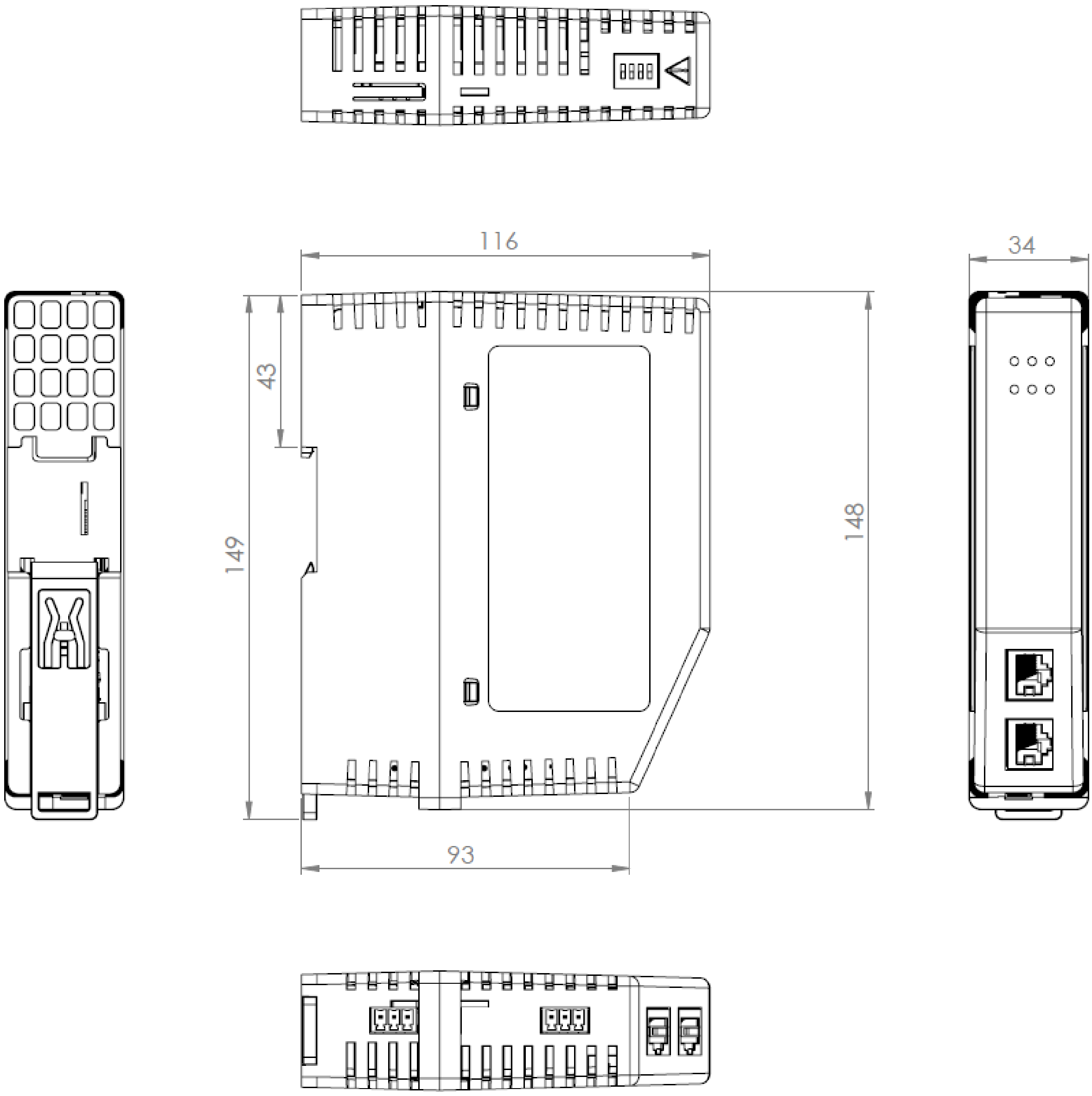


Figure 10.1 – FF Link enclosure dimensions

10.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 22 – 26 V DC
Power consumption	Maximum: 135 mA @ 24V => 3.3 W (No Bus Load) Maximum: 580 mA @ 24V => 14.0 W (Full Bus Load of 400 mA)
Connector	3-way terminal
Conductors	24 – 18 AWG
Enclosure rating	IP20, NEMA/UL Open Type
Temperature	-20 – 70 °C (Power Conditioner Load <= 220 mA, Full Bus Load <= 200 mA) -20 – 60 °C (Power Conditioner Load <= 320 mA, Full Bus Load <= 300 mA) -20 – 50 °C (Power Conditioner Load <= 420 mA Full Bus Load <= 400 mA)
Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 10.1 - Electrical specification

10.3. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 100
TCP connections	Max 100
CIP connections	Max 20
Communication rate	10/100Mbps
Duplex mode	Full/Half
Auto-MDIX support	Yes
Embedded switch	Yes, 2 x Ethernet ports
Device Level Ring (DLR)	Supported

Network Time Protocol (NTP)	Supported

Table 10.2 - Ethernet specification

10.4. ETHERNET/IP TARGET

Specification	Rating
Class 1 Connection Size	Max Input Size – 500 bytes (492 bytes mapped data) Max Output Size – 496 bytes (492 bytes mapped data)
Class 1 Connection Count	1 - 4
Class 3 Messaging Supported	Yes
UCMM Messaging Supported	Yes

Table 10.3 – EtherNet/IP Target specification

10.5. 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
Direct-To-Tag Logix Support	Yes

Table 10.4 – EtherNet/IP Originator specification

10.6. MODBUS TCP MASTER

Specification	Rating
Modes Supported	Modbus TCP
Max Modbus Slave device	20
Max Modbus Mapping	100
Mapping Ranges	Holding Register 0 – 65535 Input Register 0 – 65535 Input Status 0 – 65535

	Coil Status 0 – 65535
Base Offset	Modbus (Base 0) PLC (Base 1)
Configurable Modbus TCP Port	Yes
Data Re-formatting Supported	BB AA BB AA DD CC CC DD AA BB DD CC BB AA

Table 10.5 – Modbus Master specification

10.7. MODBUS TCP SLAVE

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

Table 10.6 – Modbus Slave specification

10.8. H1

Specification	Rating
Connector	3-way terminal
Conductor	24 – 18 AWG
LAS	Supported
H1 Alarming	Supported
Intrinsically Safe (IS) applications	Supported
Advanced H1 Scheduling	Supported
Maximum Slave Devices	32
Isolated	Yes

Internal Power Conditioner Voltage	Maximum: 23.0 V DC Minimum: 19.0 V DC
Internal Power Conditioner Current	Maximum: 420 mA – Max. Ambient Temperature <= 50 °C Maximum: 320 mA – Max. Ambient Temperature <= 60 °C Maximum: 220 mA – Max. Ambient Temperature <= 70 °C Note: Includes 20mA for internal MAU
Internal Termination	100 Ω - Software Enabled

Table 10.7 – H1 specification

10.9. CERTIFICATIONS

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

Table 10.8 – Certifications

11. FOUNDATION™ FIELDBUS H1

11.1. INTRODUCTION

Foundation Fieldbus is a digital, serial, bidirectional communication system used to connect intelligent field devices (sensors and actuators) and controllers. The protocol makes use of two networks, namely:

- H1 – Two-wire bus system
- HSE – High Speed Ethernet (Not applicable to FF Link)

The H1 bus combines power (nominally 24V) and data communications (31.25 kbit/s) on a single pair of conductors.

11.2. TOPOLOGY OF H1

H1 devices are connected in a bus structure. Up to 32 stations can be connected on one segment. A segment typically comprises a main trunk cable, with limited length spurs connecting the main trunk to each H1 device.

The maximum allowed length of the spur cables decrease as a function of their quantity. The bus is terminated by two bus terminators positioned at each end of the main bus trunk cable.

A power conditioner is required to inject power onto the H1 bus, without significantly affecting the impedance of the bus. When more than 32 stations are used, repeaters (line amplifiers) must be used to connect the individual bus segments.

11.3. H1 CABLE DESCRIPTION

Only one type of cable can be used for H1 network:

Parameter	Type A
Cable Type	Shielded, Twisted Pair
Conductor Cross-Section Area	>0.8 mm² (AWG 18)
Characteristic Impedance at 31.25kHz	100 Ω ±20%
Maximum DC Resistance (loop)	44 Ω/km
Maximum Attenuation at 39 kHz	3 dB/km
Maximum Capacitive Imbalance	2 nF/km
Maximum Overall Length	1 900 m

Table 9.2 – H1 network cable

The maximum spur length depends on the number of cable stubs or spurs.

Total Spur Count	Maximum Spur Length
1 - 12	120 m
13 - 14	90 m
15 - 18	60 m
19 - 24	30 m
25 - 32	1 m

Table 9.3 – H1 Cable Spur Length

12.APPENDIX

12.1. PROCESS VARIABLE (PV) STATUS

The Process Variable (PV) status is one byte in length and defined as shown below.
[See References: FF-890]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Quality		Quality Sub Status				Limit	

Table 9.3 – PV Status Enumeration

12.1.1. Quality

Value	Description
0	**Bad** - The value is not useful.
1	**Uncertain** - The quality of the value is less than normal, but the value may still be useful.
2	**Good (Non-cascade)** - The quality of the value is good. Possible alarm conditions may be indicated by the sub-status. Alarm indication applies only to the PV and primary output parameters not in the cascade path.
3	**Good (Cascade)** - The value may be used in control.

Table 9.3 – Status – Quality Enumeration

12.1.2. Quality Sub Status

Value	Description
BAD Sub-status	
0	**Non-specific** - There is no specific reason why the value is bad. Used for propagation.
1	**Configuration Error** - Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect.
2	**Not Connected** - Set if this input is not referenced in by a link object within the resource.
3	**Device Failure** - Set if the source of the value is affected by a device failure.
4	**Sensor Failure** - Set if the device can determine this condition. The Limits define which direction has been exceeded.
5	**No Communication, with last usable value** - Set if this value had been set by communication, which has now failed.
6	**No Communication, with no usable value** - Set if there has never been any communication with this value since it was last Out of Service.
7	**Out of Service** - The value is not reliable because the block is not being evaluated, and may be under construction by a configurator. Set if the block mode is O/S.
8	**Transducer in MAN** - The value is not reliable because the transducer block is in MAN mode.
9	**SIS**

UNCERTAIN Sub-status	
0	**Non-specific** - There is no specific reason why the value is uncertain. Used for propagation.
1	**Last Usable Value** - Whatever was writing this value has stopped doing so. (This happens when an input is disconnected by a configurer.)
2	**Substitute** - Set when the value is written when the block is not Out of Service.
3	**Initial Value** - Set when the value of an input parameter is written while the block is Out of Service.
4	**Sensor Conversion not Accurate** - Set if the value is at one of the sensor limits. The Limits define which direction has been exceeded. Also set if the device can determine that the sensor has reduced accuracy (e.g. degraded analyzer), in which case no limits are set.
5	**Engineering Unit Range Violation** - Set if the value lies outside of the range of values defined for this parameter. The Limits define which direction has been exceeded.
6	**Sub-normal** - Set if a value derived from multiple values has less than the required number of Good sources.
7	**Transducer in MAN** - The value is questionable because the transducer block is in MAN mode.
GOOD Sub-status	
0	**Non-specific** - There is no specific reason why the value is good. No error or special condition is associated with this value.
1	**Active Block Alarm** - Set if the value is good and the block has an active Block Alarm.
2	**Active Advisory Alarm** - Set if the value is good and the block has an active Alarm with a priority less than 8.
3	**Active Critical Alarm** - Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8.
4	**Unacknowledged Block Alarm** - Set if the value is good and the block has an unacknowledged Block Alarm.
5	**Unacknowledged Advisory Alarm** - Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8.
6	**Unacknowledged Critical Alarm** - Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8.
7	Reserved
8	**Initiate Fault State(IFS)** - The value is from a block that wants its downstream output block (e.g. DO) to go to Fault State. This is determined by a block option to initiate Fault State if the status of the primary input and/or cascade input goes Bad.
GOOD CASCADE Sub-status	
0	**Non-specific** - There is no specific reason why the value is good. No error or special condition is associated with this value.
1	**Initialization Acknowledge(IA)** - The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).
2	**Initialization Request(IR)** - The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.
3	**Not Invited (NI)** - The value is from a block which does not have a target mode that would use this input. This covers all cases other than Fault State Active, Local Override, and Not Selected. The target mode can be the next permitted mode of higher priority in the case of shedding a supervisory computer.
4	**Not Selected(NS)** - The value is from a Control Selector which has not selected the corresponding input. This tells the upper block to limit in one direction, not to initialize.
5	**Reserved**

6	**Local Override(LO)** - The value is from a block that has been overridden by a local key switch or is a Complex AO/DO with interlock logic active. The failure of normal control must be propagated to a PID block for alarm and display purposes. This also implies Not Invited.
7	**Fault State Active(FSA)** - The value is from a block that has Fault State active. The failure of normal control must be propagated to a PID block for alarm and display purposes. This also implies Not Invited.
8	**Initiate Fault State(IFS)** - The value is from a block that wants its downstream output block (e.g. AO) to go to Fault State. This is determined by a block option to initiate Fault State if the status of the primary input and/or cascade input goes Bad. See the status options table in Part 2 and 3.

Table 9.3 – Status – Quality Enumeration

12.1.3. Limit

Value	Description
0	**Not limited** - The value is free to move.
1	**Low limited** - The value is from a block that cannot generate or use a lower value because it is limited in that direction, either internally or by the transducer.
2	**High limited** - The value is from a block that cannot generate or use a higher value because it is limited in that direction, either internally or by the transducer.
3	**Constant** (high and low limited) - The value cannot change and is independent of the process.

Table 9.3 – Status – Limit Enumeration

13. REFERENCES

- FOUNDATION™ Function Block Application Process (FF-890)
- FOUNDATION™ Fieldbus System Engineering Guidelines (AG-181)
- FOUNDATION™ Fieldbus Technical Overview (FD-043)
- www.fieldcommgroup.org/technologies/foundation-fieldbus

14.INDEX

A

Adanced, 39, 52, 97, 98, 112
assembly instance, 88

C

Contact Us, 15

D

DD, 25
DHCP, 17, 20, 21, 22, 23
diagnostics, 8, 15, 179, 181, 212
dimensions, 216
DIN rail, 17, 18
DIP, 16, 17
DIP Switch, 188
Discovery, 125, 126, 185
Download, 114, 115, 124
DPV1, 79
DTM, 171, 172, 173, 176, 177, 178, 179, 180

E

Ethernet Bridge, 87
Ethernet/IP, 98
EtherNet/IP, 23, 98
EtherNet/IP Devices configuration, 97
EtherNet/IP Map configuration, 98, 99

F

FF Link, 128
FF LINK, 8
FF LINK Advanced configuration, 52, 112
FF Link general configuration, 39
FF LINK general configuration, 33
FF LINK H1 configuration, 36
FF LINK Modbus configuration, 110, 111
FF LINK parameters, 33
Field device general configuration, 57
Field device H1 configuration, 59

I

input assembly, 188, 200, 201
Instance Name, 33, 58
IP address, 124, 173
IP Address, 34
IP Point HART, 88

L

LED, 66, 67, 75, 79, 181, 182, 183, 184
Live List, 178, 179, 189, 190, 191
Logix Connection parameters, 88

M

MODBUS, 15, 197, 198
Modbus Router, 31, 32

O

Online, 125, 149, 177, 179, 182
output assembly, 128, 129, 154, 155

P

Packet Capture, 204, 205, 206, 207, 208
Project Explorer, 182, 184
Protocol, 20

R

requested packet interval (RPI), 88
Rockwell Automation, 25
RSLinx, 25
RSLogix 5000, 87, 88, 89

S

Safe Mode, 17
Slate, 8, 14, 20, 23, 31, 32
Slate Configuration Utility, 33
SLATE Configuration Utility, 35, 36, 53, 110, 111, 112
Slot, 186
Standalone, 34, 186
station address, 37, 58, 126, 127
statistics, 182, 193, 210
Support email, 15

T

Target Browser, 23, 24, 98

U

UDTs, 14

W

web server, 182, 212