DeviceNet Router/B

User Manual

A-DNTR/B

Document No. D154-007
07/2025
Revision 1.8

aparian

CONTENTS

Revision History

Revision	Date	Comment
1.0	26 October 2022	Initial document
1.1	17 Jan 2023	Update support contact details
1.2	25 Jan 2023	Minor wording update
1.3	13 Oct 2023	Corrected the maximum supported DeviceNet explicit device count (was 62 and should have been 63). Corrected the Input / Output Data Max (as a DeviceNet Target) in the specifications to 128 bytes (from 256 bytes).
1.4	8 November 2023	Added ATEX Conformance Mark Added UKCA Conformance Mark
1.5	27 November 2023	Updated EtherNet/IP connection configuration Updated DeviceNet Scanner device configuration
1.6	20 December 2023	Added PCCC specification
1.7	20 May 2025	Add duplicate IP address indication to LEDs in the Diagnostics section.
1.8	7 July 2025	Added Poll Ratio to the DeviceNet device settings when the module is operating as a DeviceNet Scanner.

1. PREFACE

1.1. INTRODUCTION TO THE DEVICENET ROUTER

This manual describes the installation, operation, and diagnostics of the Aparian DeviceNet Router Series B module. The DeviceNet Router/B, (hereafter referred to as the **module**) provides intelligent data routing between either EtherNet/IP or Modbus TCP/RTU and a DeviceNet network. This allows the user to integrate DeviceNet devices into a Rockwell Logix platform (e.g., ControlLogix or CompactLogix) or any Modbus Client or Server device with minimal effort.

The module can be configured to be either a DeviceNet Scanner or a DeviceNet Device allowing the user to not only integrate DeviceNet devices into a Logix or Modbus system, but to also allow the user to use EtherNet/IP or Modbus devices in an existing DeviceNet network (by using the DeviceNet Router/B in device mode).

DeviceNet Mode:

DeviceNet Scanner
When the module operates as a DeviceNet Scanner, it can connect to a maximum of 63 DeviceNet Devices. The input and output data from each DeviceNet Device can be mapped to any of the operating interfaces (EtherNet/IP Target, Modbus Server, Modbus Client, or EtherNet/IP Originator).

DeviceNet Device
When the module is configured to be a DeviceNet Device, it will allow the module to have input and output data sizes of up to 128 bytes.

Primary Interface:
The module can use one of five interface modes:

PCCC Client
The DeviceNet Router is able to asynchronously exchange data between a DeviceNet polling master (scanner) and an Ethernet PCCC device. The sizes of the DeviceNet's produced and consumed data are independently configurable from 0 to 128 bytes each.

The consumed (DeviceNet) data can then be mapped to a PLC5 type address file, e.g. N33, and then read by an Ethernet device e.g. a PanelView. Similarly, the produced data (DeviceNet) can also be mapped to a PLC5 type address file, to which an Ethernet device could write.

EtherNet/IP Target

As a EtherNet/IP target, the module can use one of two methods to read and write data to and from the DeviceNet network:

- **EtherNet/IP Class 1 connection**

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

EtherNet/IP Originator

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

- **EtherNet/IP Explicit Messaging**

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

- **Direct-To-Tag technology**

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

- **EtherNet/IP Class 1 connection**

  DeviceNet data (from either DeviceNet Scanner or Devices) can be mapped to a maximum of 10 EtherNet/IP devices using input and output class 1 cyclic connections. This will allow the DeviceNet Router/B to “own” the EtherNet/IP target device and exchange DeviceNet data using the EtherNet/IP device’s input and output assemblies.

Modbus Server

The diagnostics and DeviceNet data (from either DeviceNet Scanner or Devices) will be written to, or read from, the module’s internal Modbus Registers (Holding or Input Registers). These registers can be accessed by a remote Modbus Client using either Modbus TCP, Modbus RTU232, or Modbus RTU485.

Modbus Client

The diagnostics and DeviceNet data (from either DeviceNet Scanner or Devices) will be written to, or read from, the module’s internal Modbus Registers (Holding or Input Registers). The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Server devices and the module’s internal Modbus registers. The Modbus communication can be via Modbus TCP, Modbus RTU232, or Modbus RTU485.

The module and Slate will allow the user to parameterize each DeviceNet Device according to the parameters provided in the DeviceNet Device EDS file. These parameters can be saved in the DeviceNet Device’s non-volatile memory.

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

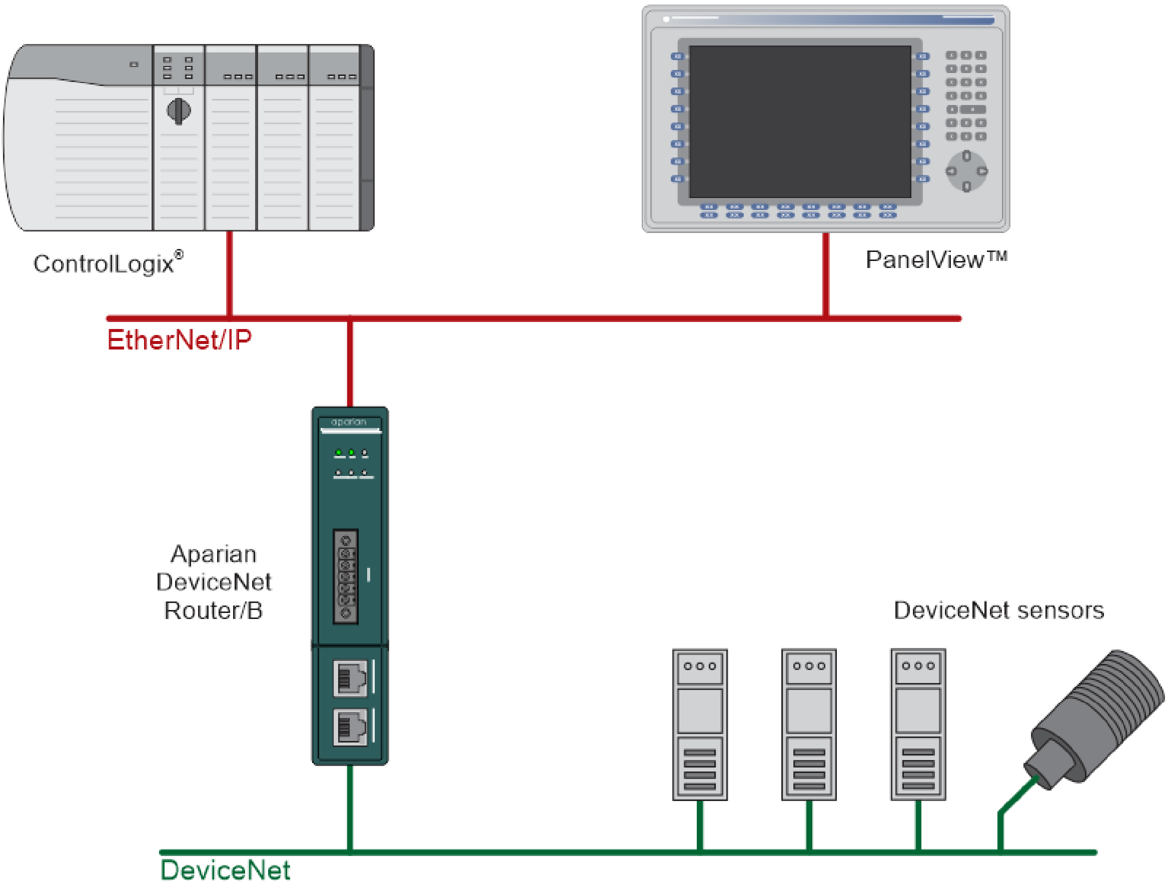


Figure 1.1. – Typical DeviceNet Scanner architecture using the DeviceNet Router/B

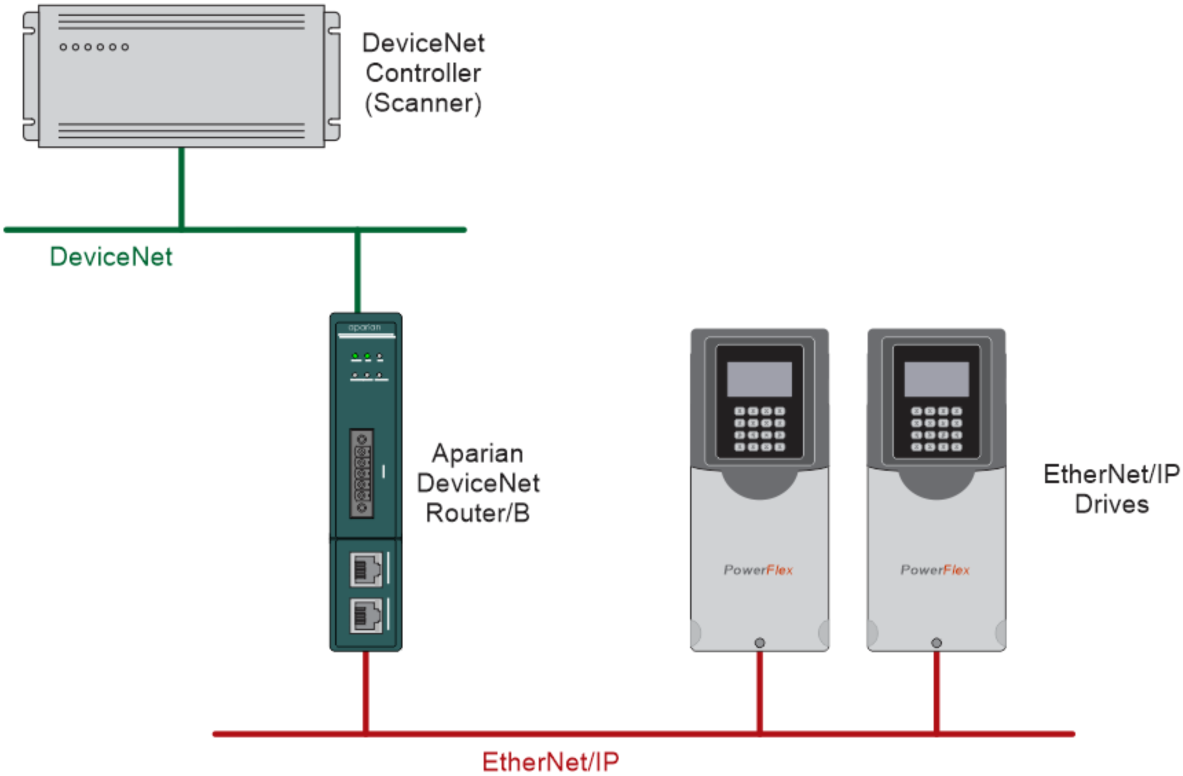


Figure 1.2. – Typical DeviceNet Device architecture using the DeviceNet Router

The module also provides a range of statistics, DeviceNet and Modbus packet capture functions, and internal Modbus and Data table reads to simplify the diagnostic process for remote diagnosis.

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

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

1.2. FEATURES

- Module can operate as a DeviceNet Scanner or Device.
- DeviceNet Scanner mode can configure and operate with up to 63 DeviceNet Devices with up to 256 bytes input and 256 bytes output per DeviceNet IO Device.
- DeviceNet Device mode can exchange up to 256 bytes of input and 256 bytes of output data with a DeviceNet Scanner.
- Supports DeviceNet Passthrough Messaging
- Supported DeviceNet Explicit Messaging
- Module has various Primary Interfaces:
  - PCCC Client for connecting new Ethernet only PanelViews to an existing DeviceNet network.
  - EtherNet/IP Target (Class 1 connection)
  - Modbus Server (TCP, RTU232, and RTU485)
  - Modbus Client (TCP, RTU232, and RTU485)
  - EtherNet/IP Originator (Class 1 connection with up to 10 EtherNet/IP devices and Explicit Messaging, including Direct-To-Tag Logix tag access, with up to 10 EtherNet/IP devices).
- Slate software provides a DeviceNet and Modbus packet capture utility for better diagnosis of issues.
- Supports all DeviceNet Baud Rates (125k, 250k, 500k).
- Dual Ethernet ports which support Device-Level-Ring (DLR).
- Network Time Protocol (NTP) supported for external time synchronization.
- Small form factor – DIN rail mounted.

1.3. ARCHITECTURE

The figures below provide an example of the typical network setup for connecting DeviceNet (device or scanner) to either EtherNet/IP or Modbus TCP/RTU232/RTU485.

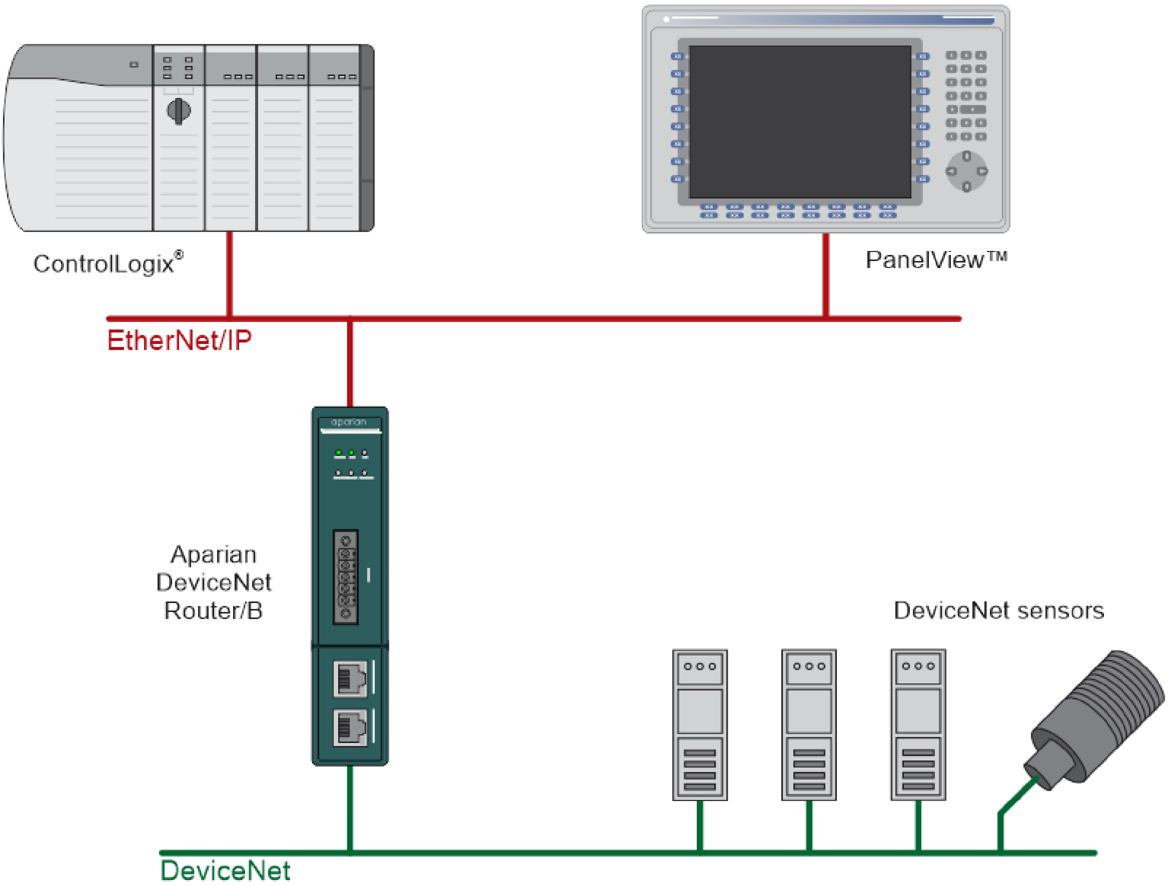


Figure 1.3. – Example of connecting DeviceNet Devices to a Logix Controller

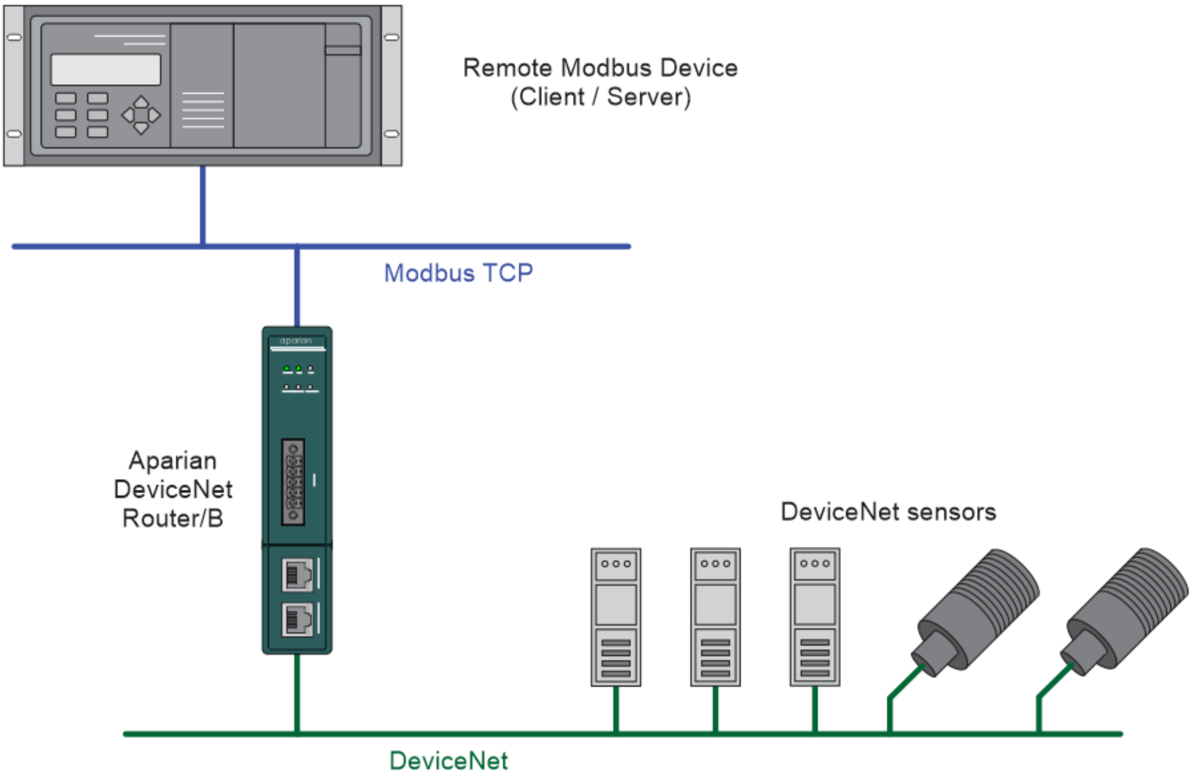


Figure 1.4. - Example of connecting DeviceNet Devices to a Modbus TCP Client or Server

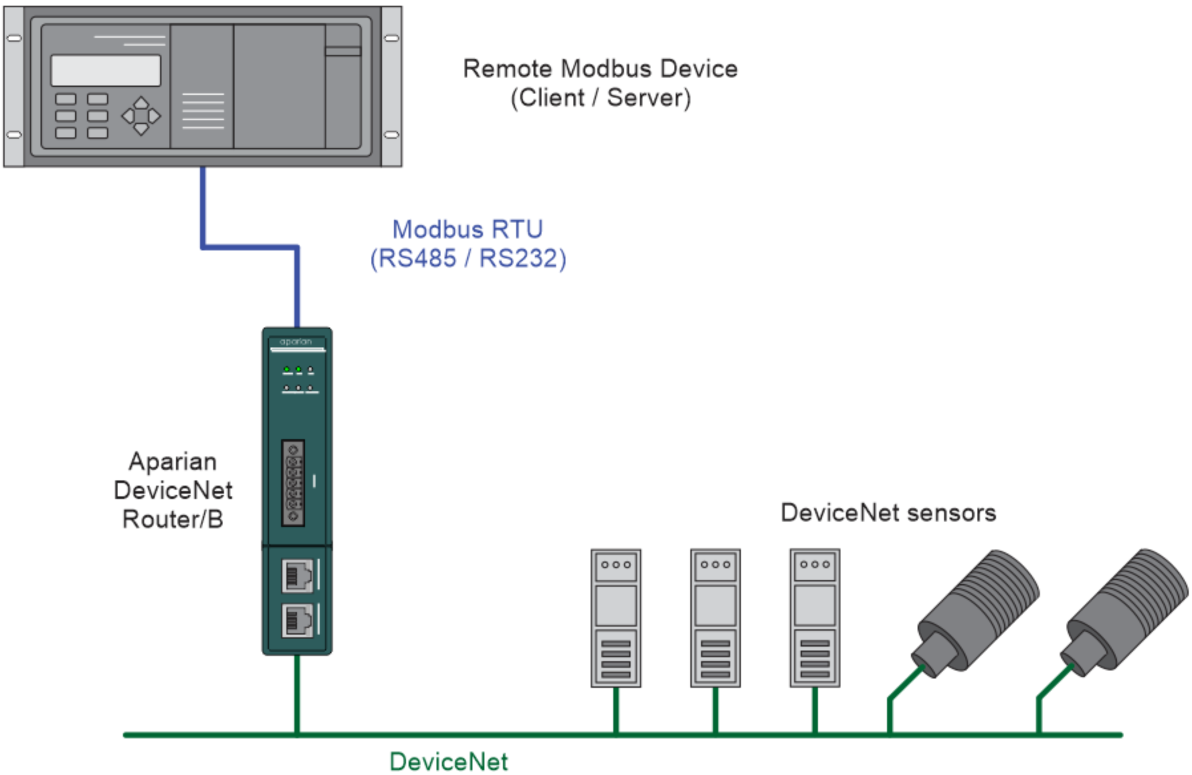


Figure 1.5. - Example of connecting DeviceNet Devices to a Modbus RTU Client or Server

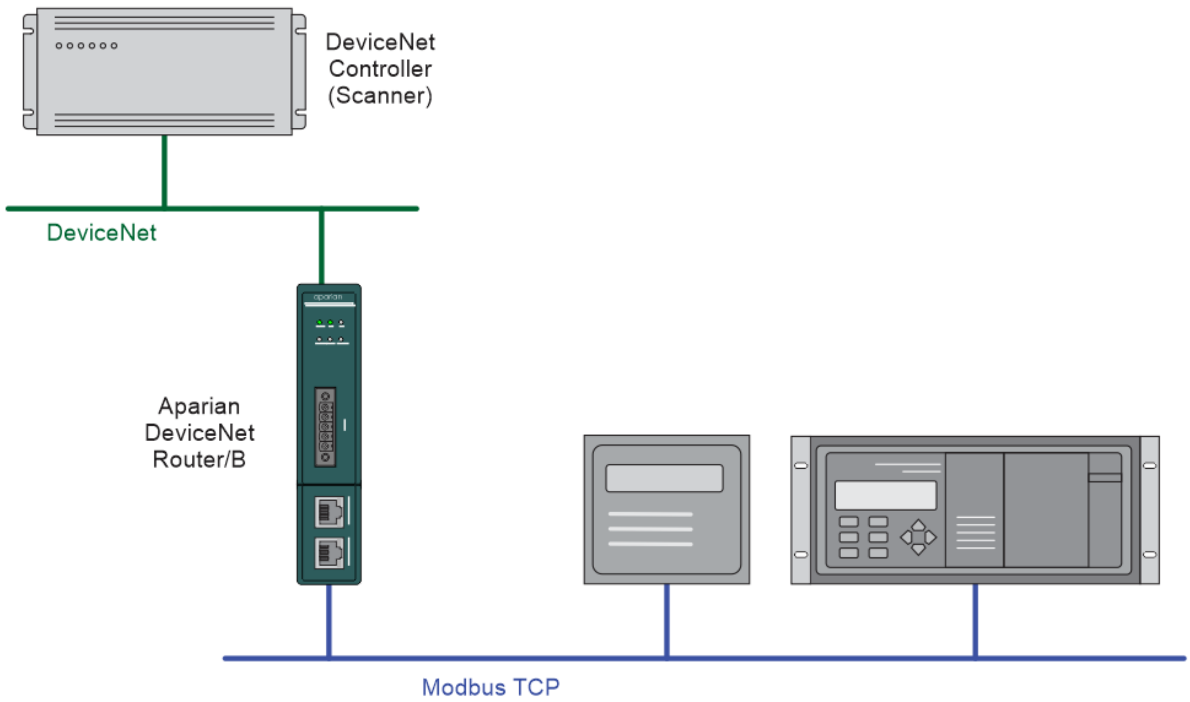


Figure 1.6. – Modbus TCP Device (Client or Server) operating as a DeviceNet Device

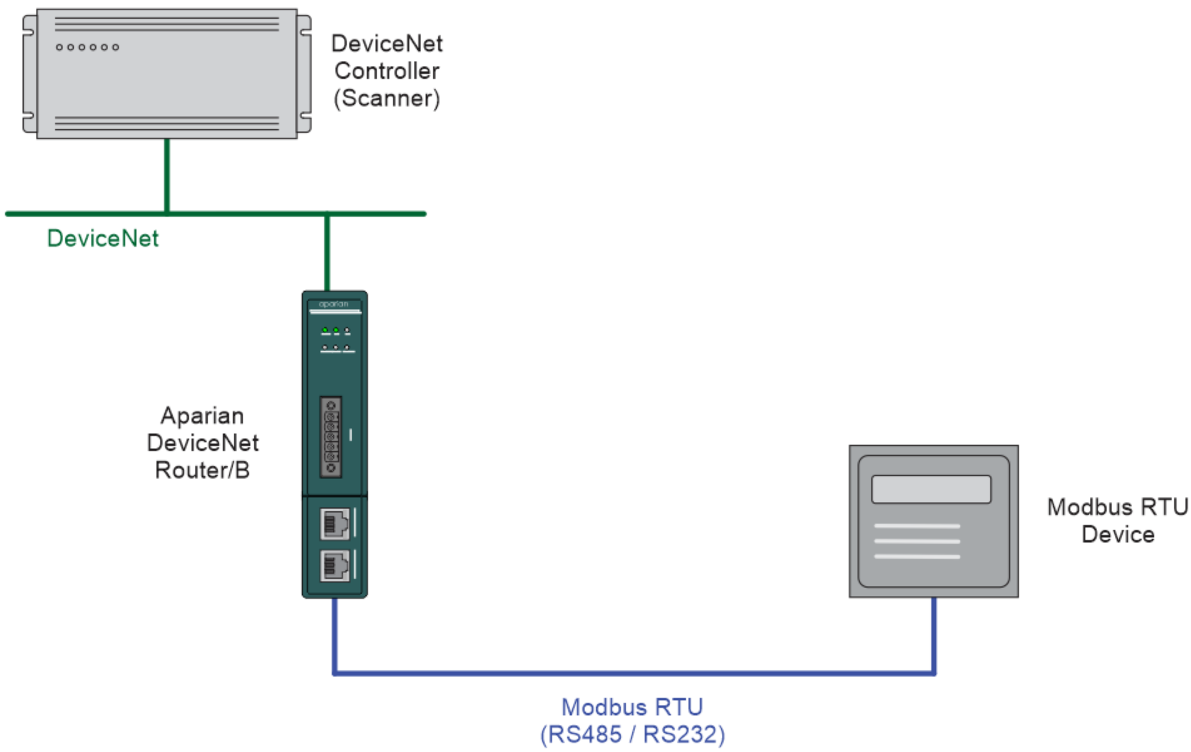


Figure 1.7. – Modbus RTU Device (Client or Server) operating as a DeviceNet Device

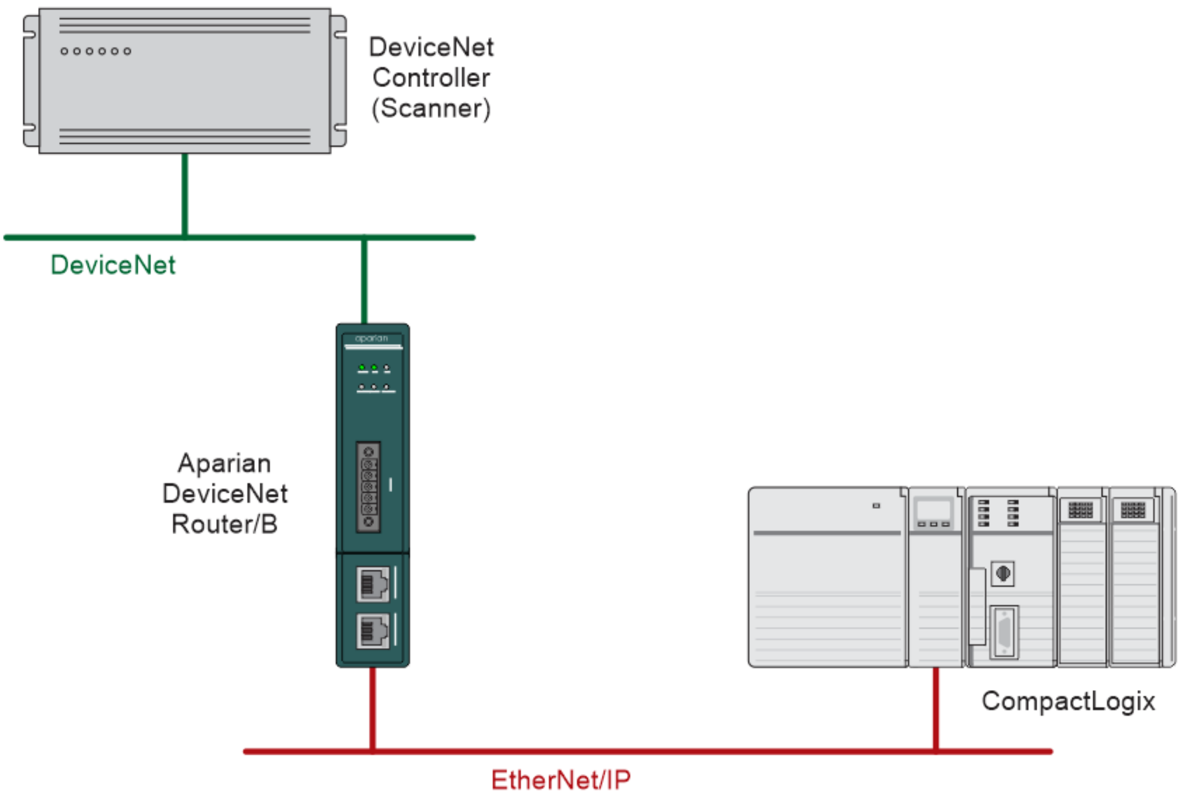


Figure 1.8. – Logix Controller operating as a DeviceNet Device via the DeviceNet Router

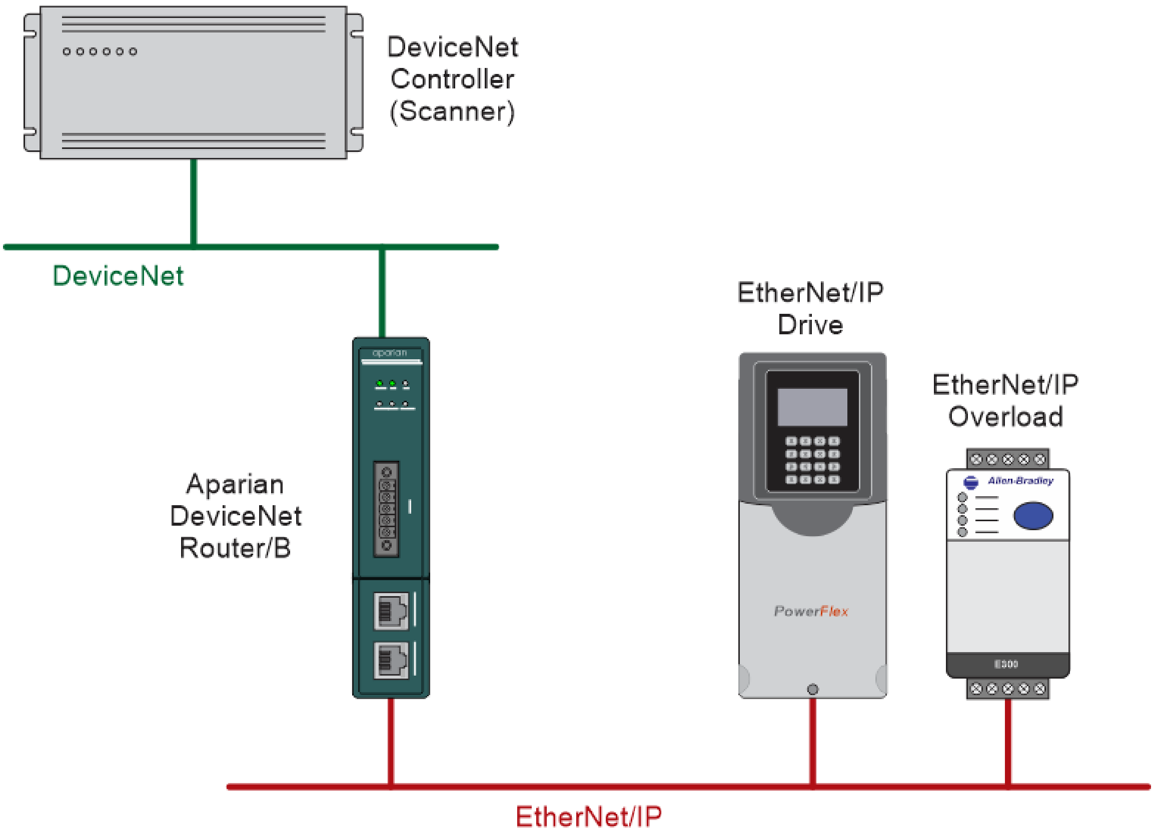


Figure 1.9. – EtherNet/IP Drive and Smart Overload operating as a DeviceNet Device

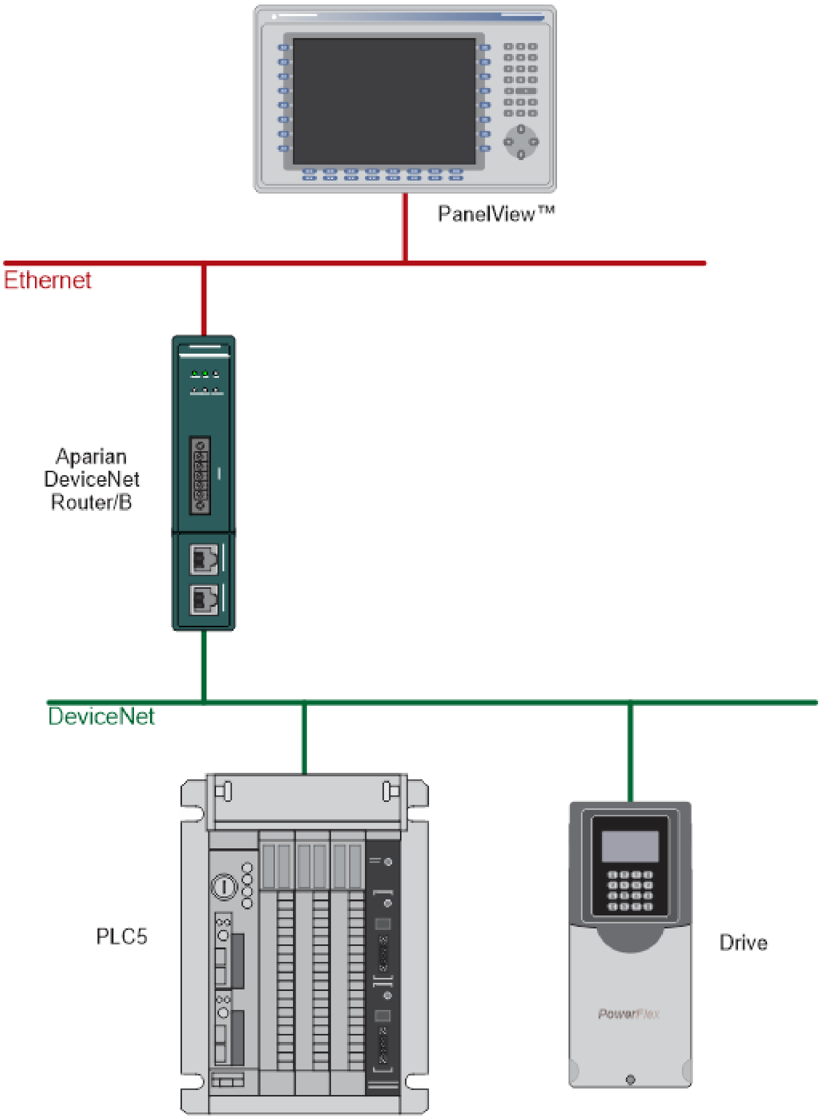


Figure 1.10. - Example of a typical network setup in PCCC Client Mode

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
DeviceNet Router/B User Manual DeviceNet Router/B Datasheet Example Code & UDTs	http://www.aparian.com/products/devicenetrouterb
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
DeviceNet	http://www.odva.org

Table 1.1. - Additional Information

1.5. SUPPORT

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

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

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

Table 1.2. – Support Details

2. INSTALLATION

2.1. MODULE LAYOUT

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

The port on the front of the module is the CAN port and can also be used to power the module.

NOTE: The module allows the user to provide power on both bottom and front power connectors and can be used for power supply redundancy.

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

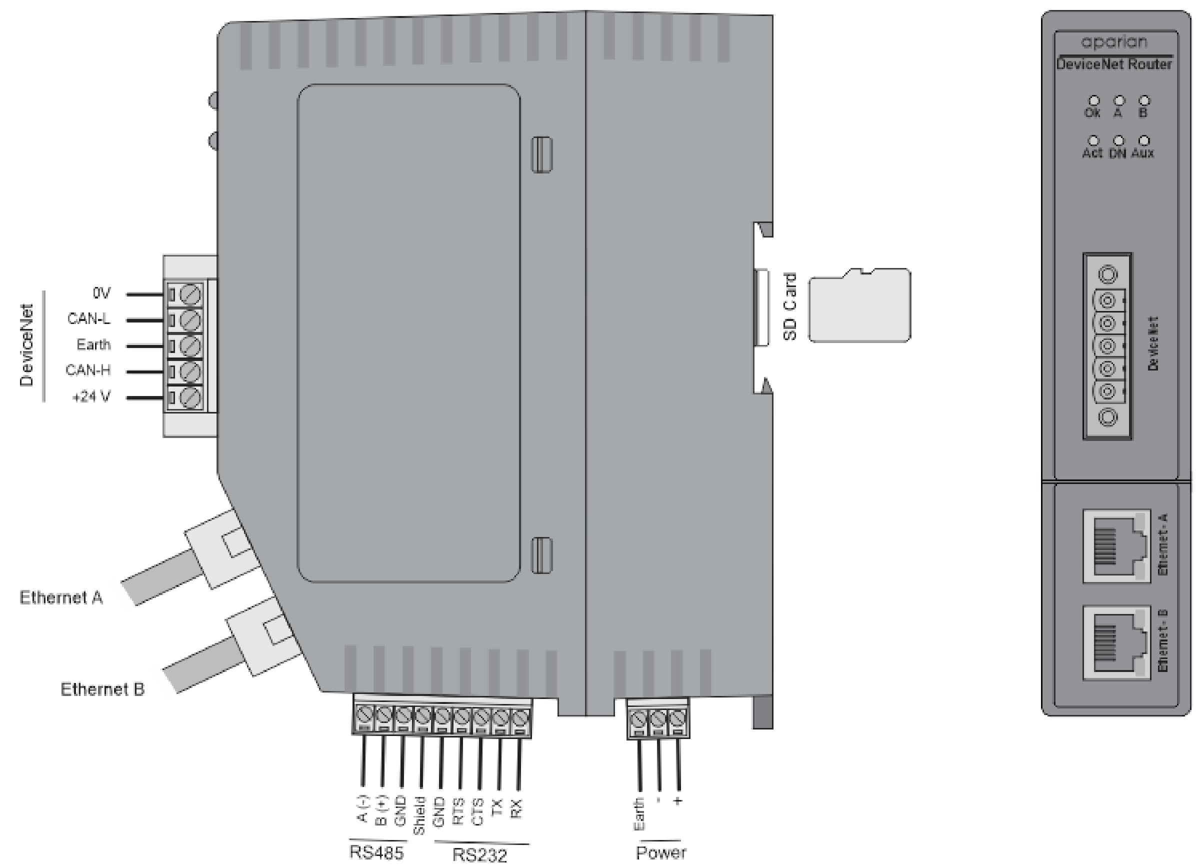


Figure 2.1 – DeviceNet Router/B side and front view

The module also supports an SD Card for disaster recovery which can be used to automatically update the configuration and/or firmware of a new module.

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

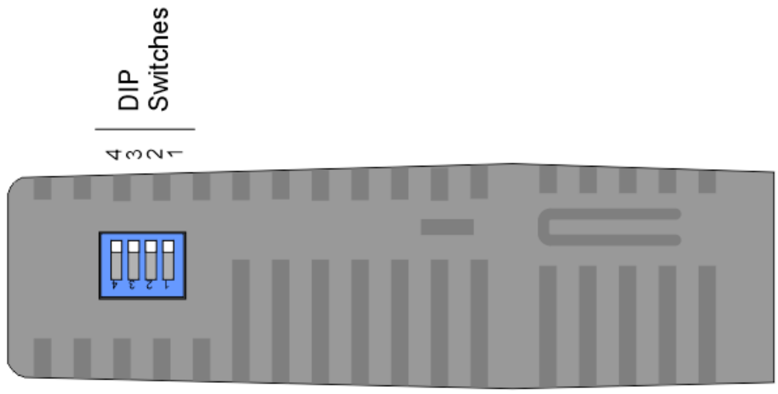


Figure 2.2 – DeviceNet Router/B top view

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

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

Table 2.1 - DIP Switch Settings

2.1. 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.

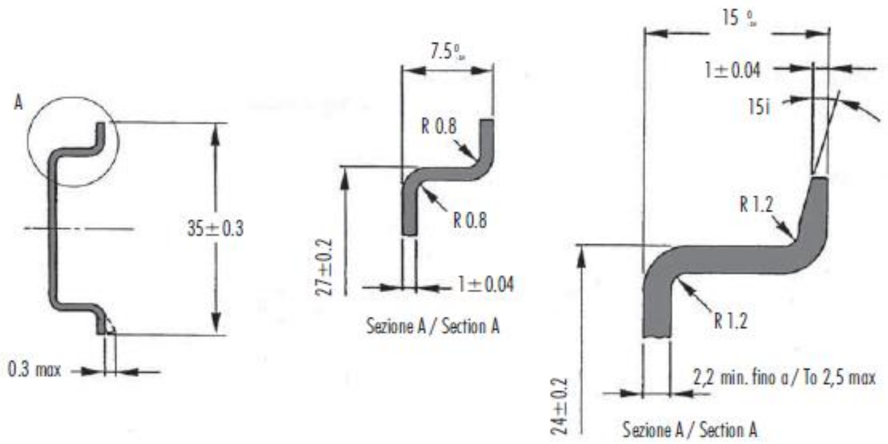


Figure 2.3 - DIN rail specification

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.

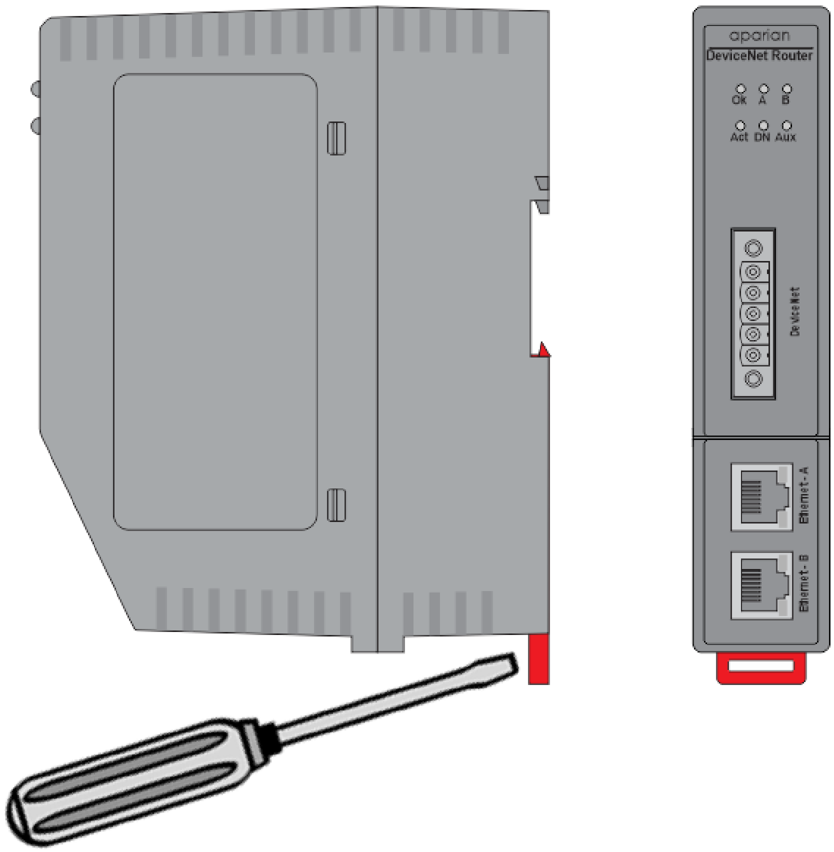


Figure 2.4 - DIN rail mouting

2.2. BOTTOM POWER

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

NOTE: The module allows the user to provide power on both bottom and front power connectors and can be used for power supply redundancy.

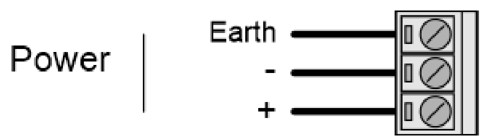


Figure 2.5 - Power connector

2.3. RS232/RS485 PORT

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

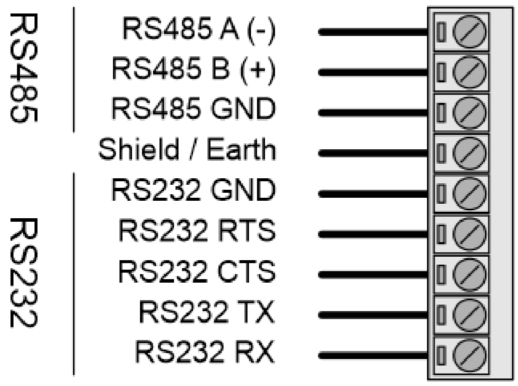


Figure 2.6 - RS232/RS485 connector

The RS485 port provides the standard A and B conductors. The RS232 port provides the standard communication conductors (RX, TX, and GND) as well as hardware handshaking lines for legacy systems (RTS – Request to Send, CTS – Clear to Send).

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

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

2.4. RS485 TERMINATION

All RS485 networks need to be terminated at the extremities (start and end point) of the communication conductor. The termination for the RS485 network can be enabled/disabled via the module configuration. Enabling the termination will connect an internal 125 Ohm resistor between the positive (B+) and negative (A-) conductors of the RS485 network.

2.5. ETHERNET PORTS

The Ethernet connectors should be wired according to industry standards. **Refer** to the additional information section in this document for further details. The module has an embedded switch connecting the two Ethernet ports.

2.6. CAN AND FRONT POWER

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

NOTE: The module allows the user to provide power on both bottom and front power connectors and can be used for power redundancy.

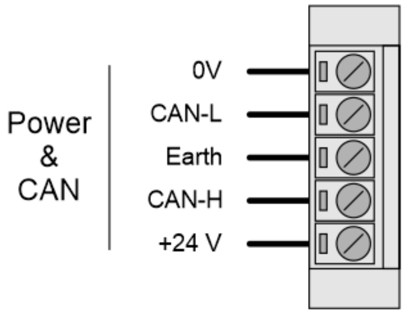


Figure 2.7 – CAN and Power connector

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.

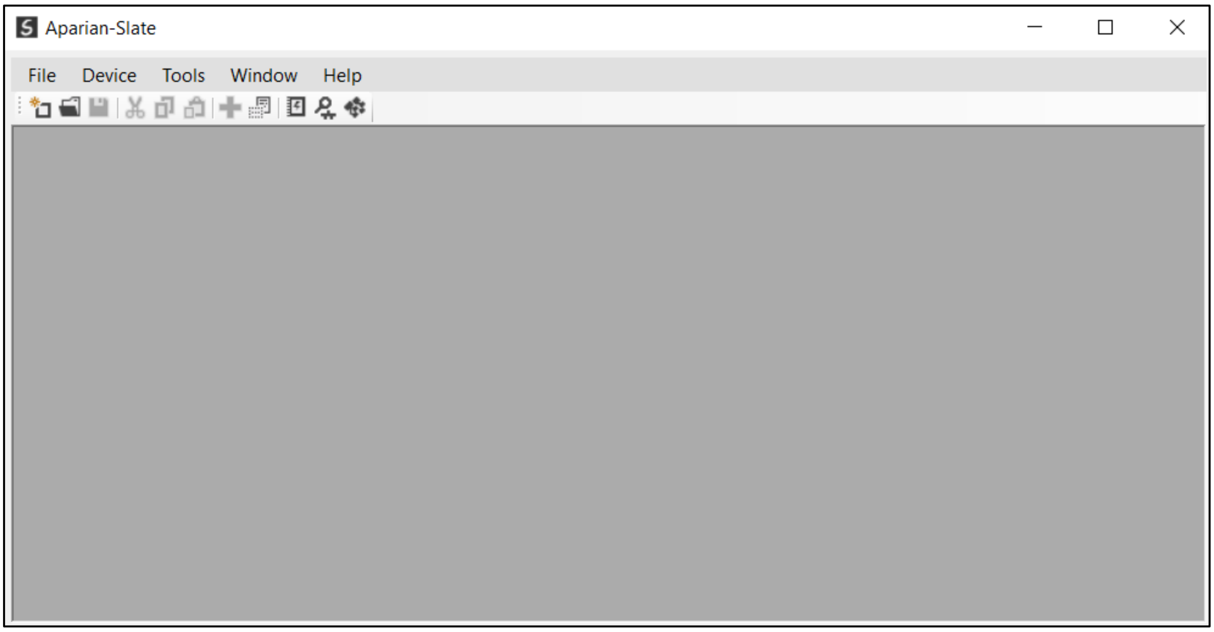


Figure 3.1. - Aparian Slate Environment

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.

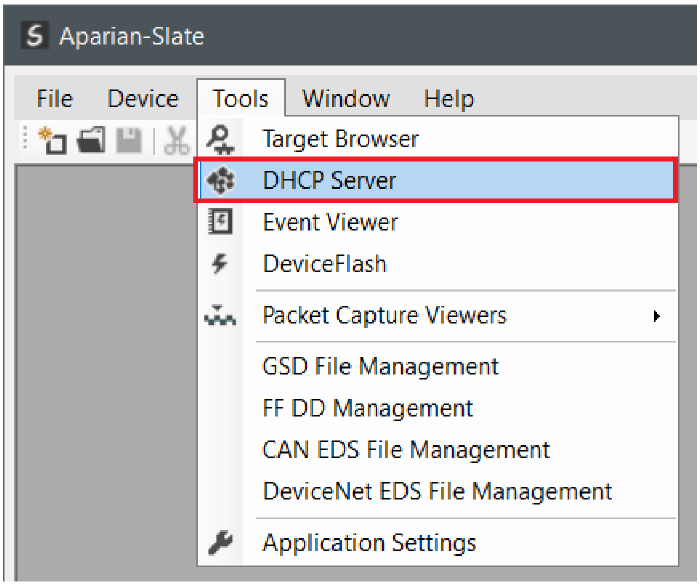


Figure 3.2. - Selecting DHCP Server

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

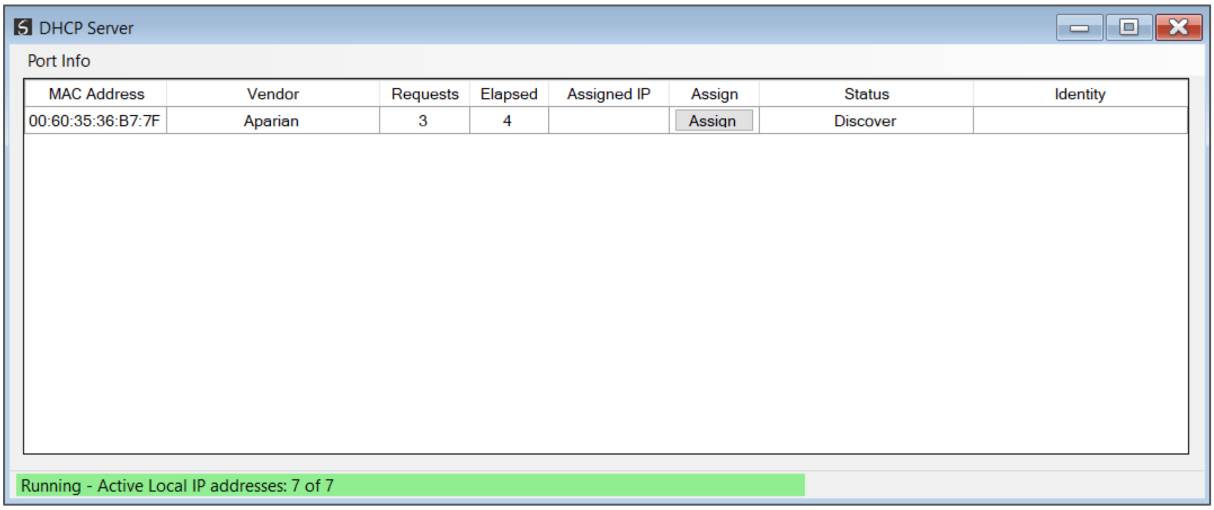


Figure 3.3. - DHCP Server

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.

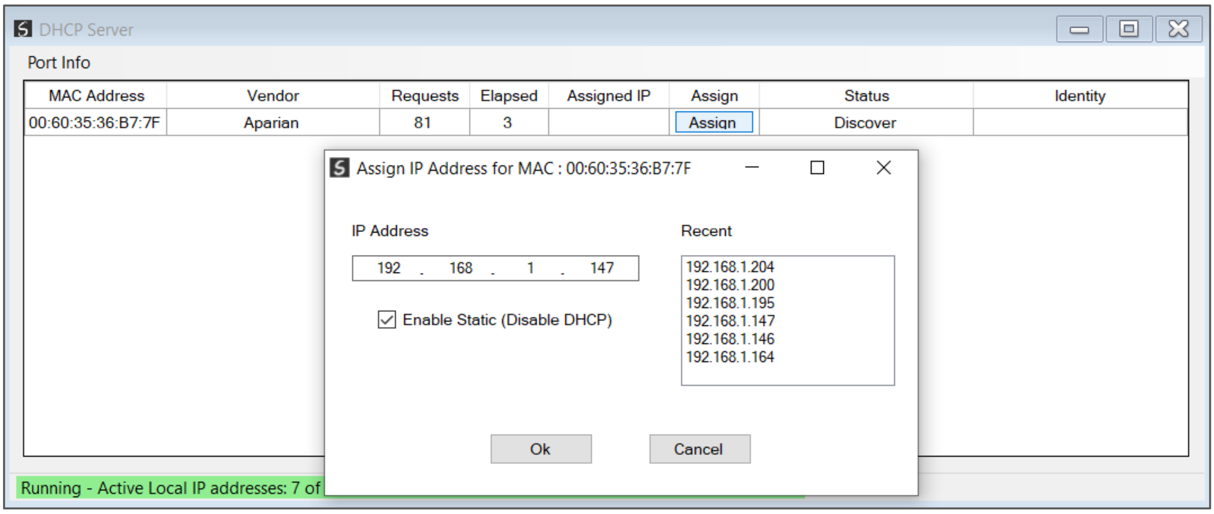


Figure 3.4. - Assigning IP Address

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.

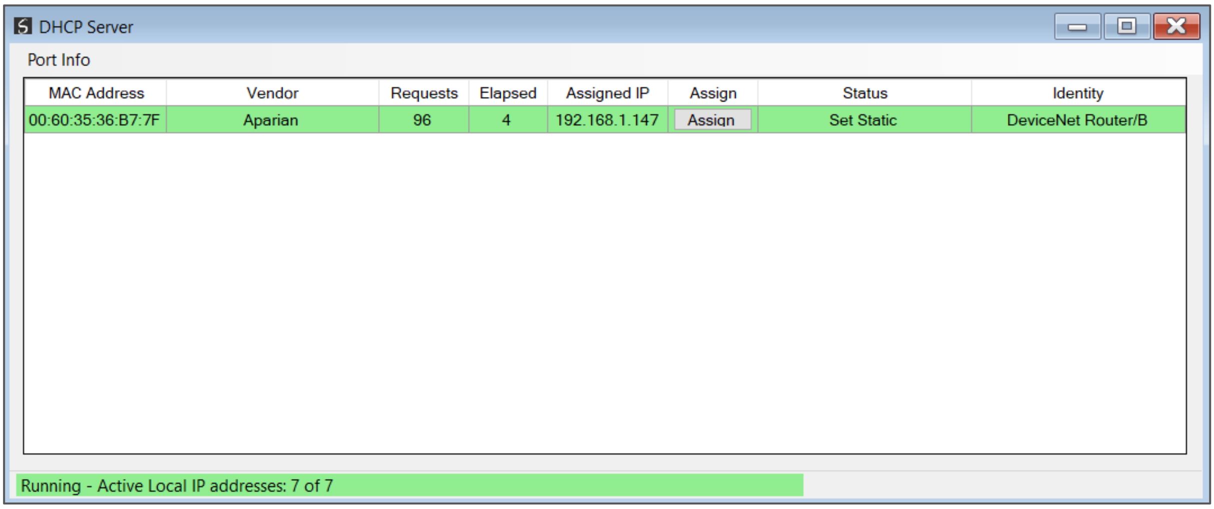


Figure 3.5. - Successful IP address assignment

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.

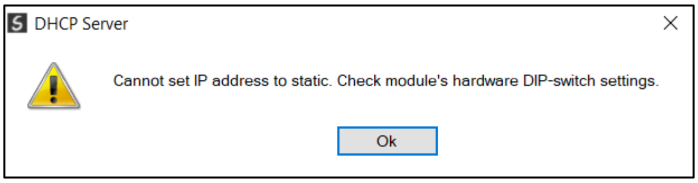


Figure 3.6. - Force DHCP warning

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.

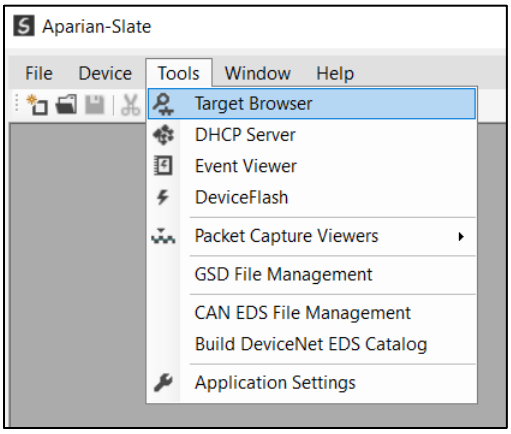


Figure 3.7. - Selecting the Target Browser

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

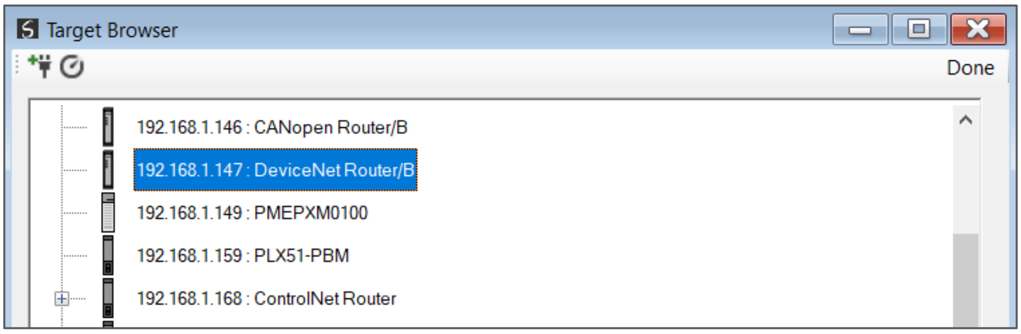


Figure 3.8. - Target Browser

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

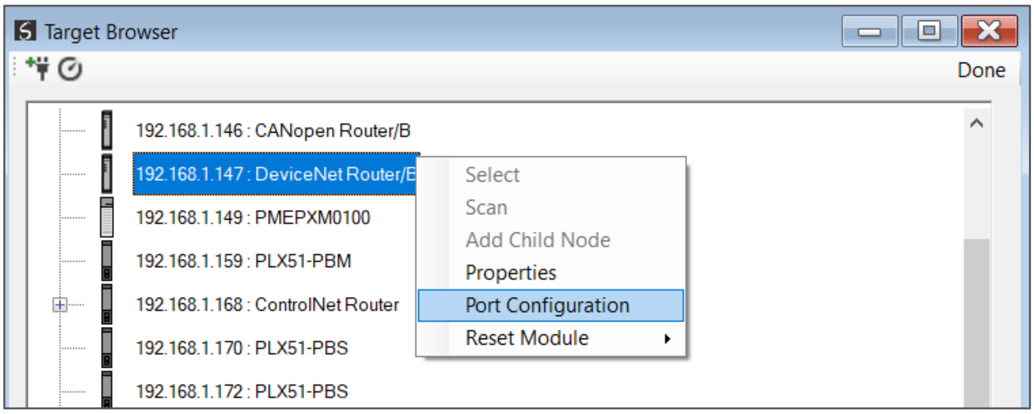


Figure 3.9. - Selecting Port Configuration

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

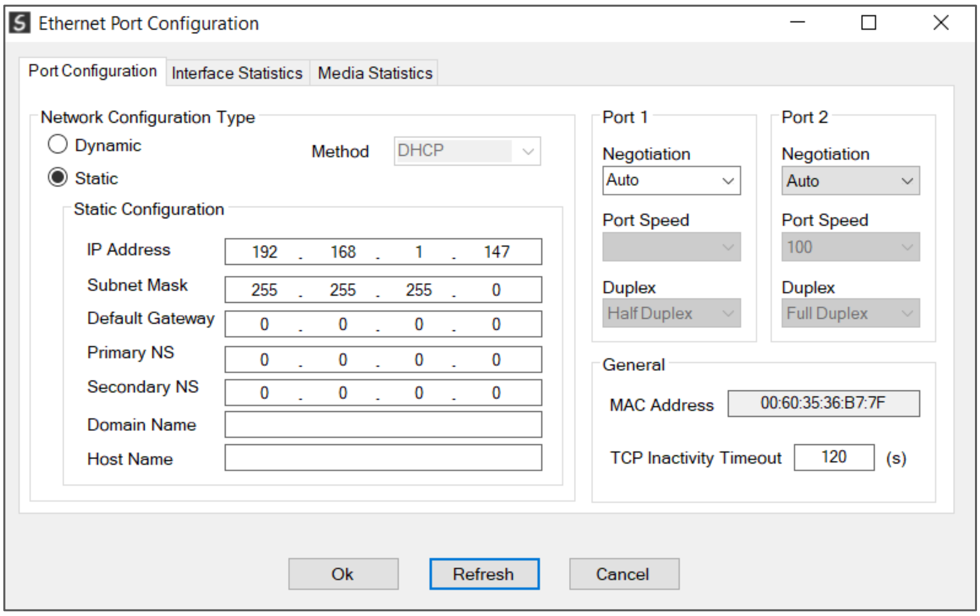


Figure 3.10. - Port Configuration

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

3.3. CREATING A NEW PROJECT

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

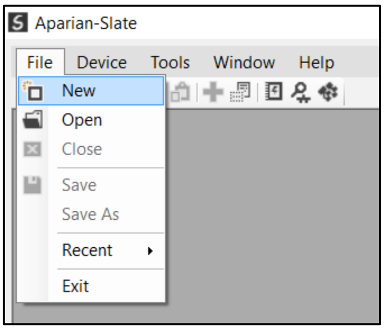


Figure 3.11. - Creating a new project

A Slate project will be created, showing the Project Explorer tree view. To save the project use the Save option under the File menu.

A new device can now be added by selecting Add under the Device menu.

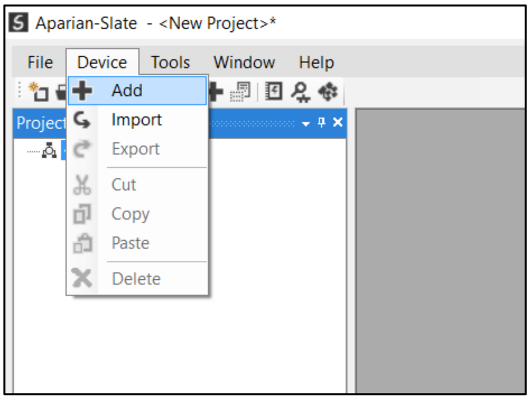


Figure 3.12. - Adding a new device

In the Add New Device window select the DeviceNet Router/B and click the Ok button.

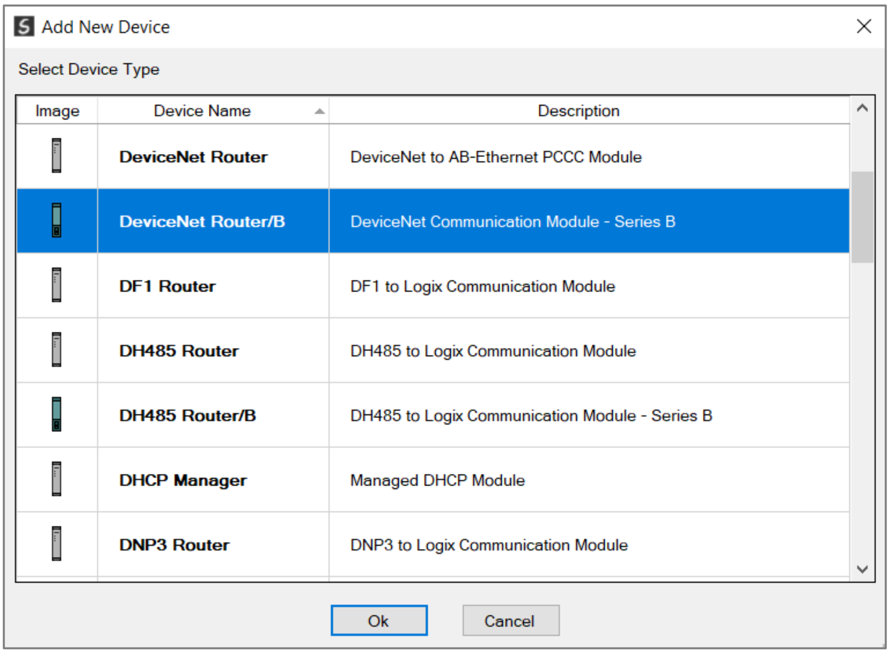


Figure 3.13 – Selecting a new DeviceNet Router

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

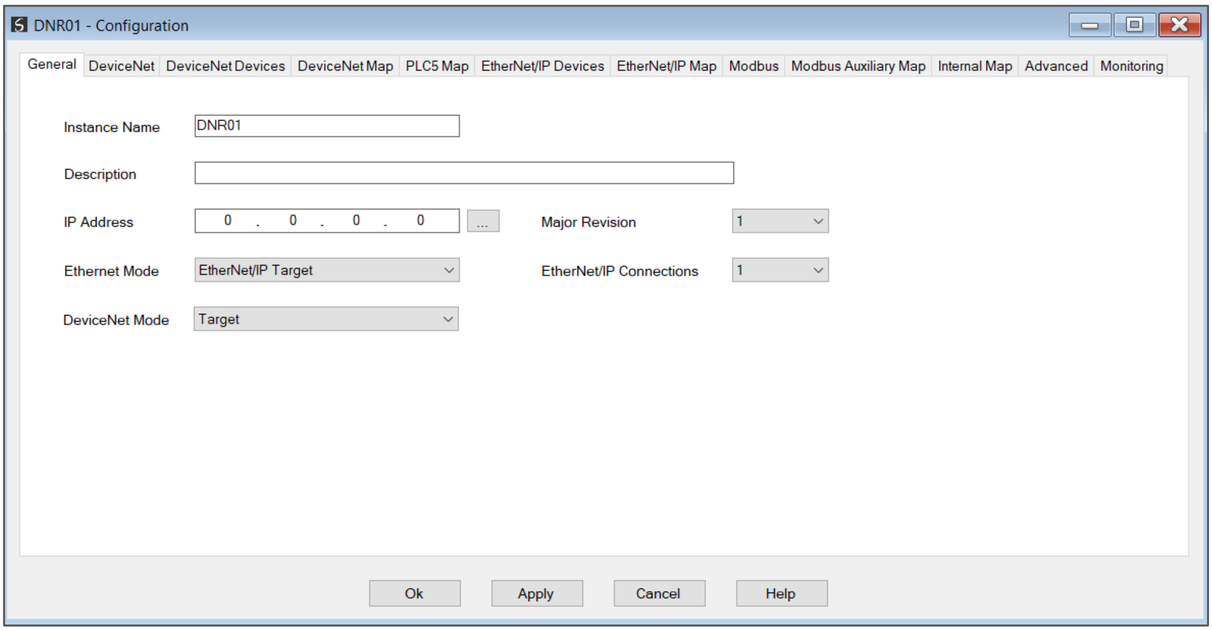


Figure 3.14. – DeviceNet Router/B configuration

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

3.4. GENERAL PARAMETERS

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

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

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

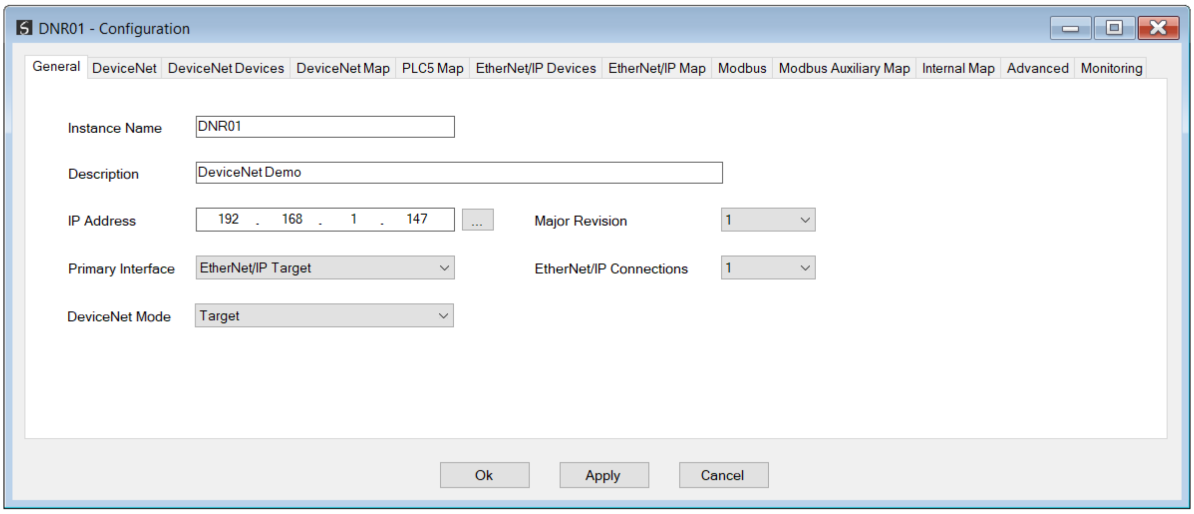


Figure 3.15 - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various DeviceNet Routers.
Description	This parameter is used to provide a more detailed description of the application for the module.
Major Revision	The major revision of the module
EtherNet/IP Connections	The number of connections the module will use when operating as an EtherNet/IP target.
IP Address	The IP address of the target module. The user can use the target browse button to launch the target browser to the select the DeviceNet Router/B on the network.
Primary Interface	**PCCC Client**

	This will allow the module to emulate a PLC5, providing an interface for Ethernet-only PanelViews and other legacy devices to a DeviceNet network. **NOTE: In this mode the DeviceNet mode will be forced to target.** **EtherNet/IP Target** A Logix controller can own the DeviceNet Router/B over EtherNet/IP using up to 4 class 1 connections. **Modbus Server** A Modbus Client can read and write data to the module which can then be mapped to one or more DeviceNet devices. The module can operate as a Modbus Server on Ethernet TCP, RTU232, and RTU485 **Modbus Client** A module can read and write data from various Modbus devices which can then be mapped to one or more DeviceNet devices. The module can operate as a Modbus Client on Ethernet TCP, RTU232, and RTU485 **EtherNet/IP Originator** As an EtherNet/IP originator, the module can use two methods to read and write data to and from an EtherNet/IP device (IO): *EtherNet/IP Class 1 Connection* The DeviceNet Router/B can own EtherNet/IP IO by using the Slate software to configure the IO connections. *EtherNet/IP Explicit Messaging* The DeviceNet Router/B can exchange data with up to 10 EtherNet/IP devices using explicit messaging.
DeviceNet Mode	**Target** A Logix controller can own the DeviceNet Router/B over DeviceNet using a cyclic DeviceNet connection (e.g. via a 1756-DNB) **Scanner** As a DeviceNet originator, the module can use two methods to exchange data with one or more DeviceNet devices: *Cyclic DeviceNet* The DeviceNet Router/B can own DeviceNet IO by using the Slate software to configure the IO connections and schedule the DeviceNet network. *DeviceNet Explicit Messaging* This DeviceNet Router/B can exchange data with up to 10 DeviceNet devices using explicit messaging over DeviceNet. **Listen-Only** In this mode the DeviceNet Router/B will only listen to the DeviceNet network allowing the user to capture the DeviceNet network traffic without affecting the current operation.

Table 3.1 - General configuration parameters

3.5. DEVICENET CONFIGURATION

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

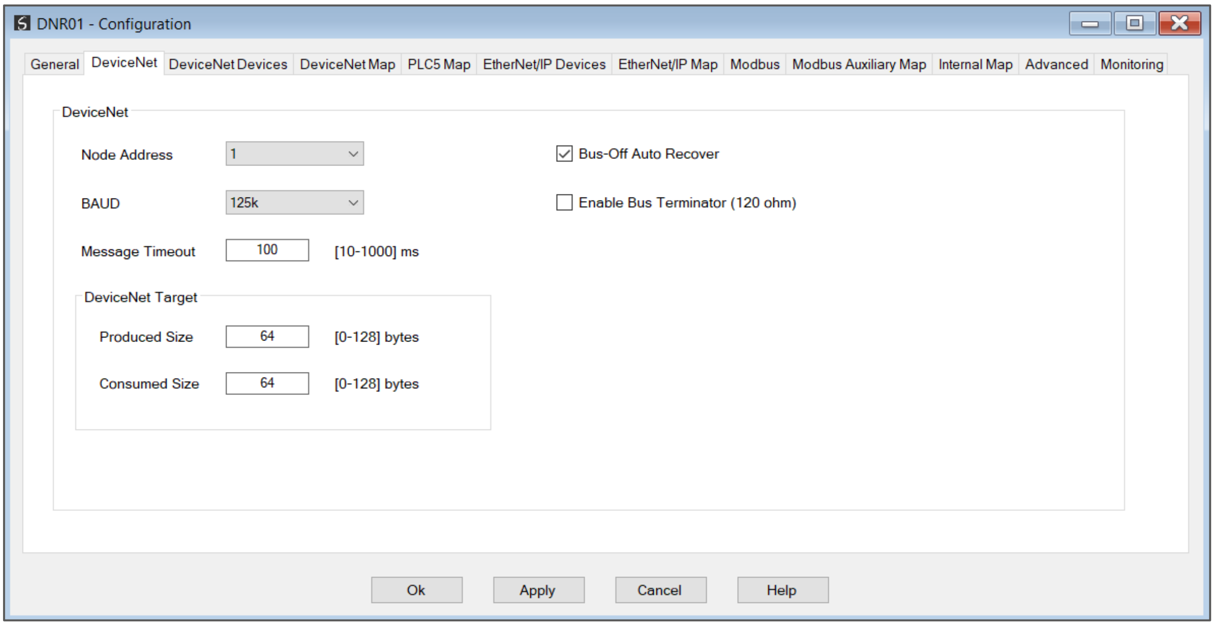


Figure 3.16 - General Configuration

The DeviceNet configuration consists of the following parameters:

Parameter	Description
Node Address	This is the node address of the DeviceNet Router/B on the DeviceNet network.
BAUD Rate	The DeviceNet bus BAUD rate. The following options are available: • 125k • 250k • 500k
Message Timeout	This is the DeviceNet message timeout which specifies how long the DeviceNet Router/B will wait for a response from a DeviceNet node before the response will be seen as a failed no-response.
Produced Size	When operating as a DeviceNet Target, this is the number of bytes that the DeviceNet Router/B will produce to the DeviceNet originator (i.e., the input assembly in Logix).
Consumed Size	When operating as a DeviceNet Target, this is the number of bytes that the DeviceNet Router/B will consume from the DeviceNet originator (i.e., the output assembly in Logix).
Bus Off Auto Recover	Enable or Disable automatic DeviceNet network recovery when the module has detected that the CAN bus network is off.

Enable Bus Terminator	Enables or disables the internal 120 Ohm terminator on the CAN bus network. The CAN bus network must be terminated at the two extremities of the network (i.e., the start and end of the network).

Table 3.2 - General configuration parameters

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

3.5.1. TARGET

A Logix controller can exchange data with the DeviceNet Router/B over DeviceNet using a cyclic DeviceNet connection when the DeviceNet Router/B is operating as a DeviceNet target. This will allow the DeviceNet Router/B to exchange data with the Logix controller using the input and output assembly of the DeviceNet bridge (e.g., 1756-DNB) where the DeviceNet Router/B data has been mapped. Data from EtherNet/IP or Modbus devices can be mapped to the Logix controller over DeviceNet.

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

3.5.1.1. DEVICENET CONFIGURATION – RSNETWORX

The DeviceNet IO messaging scheduling is typically configured using ***RSNetworx for DeviceNet*** and is required when the module is operating as a DeviceNet target.

Open RSNetWorx, create a new project and browse to the DeviceNet network. The software will scan the network for all the devices. Additional devices can be added (offline) if required.

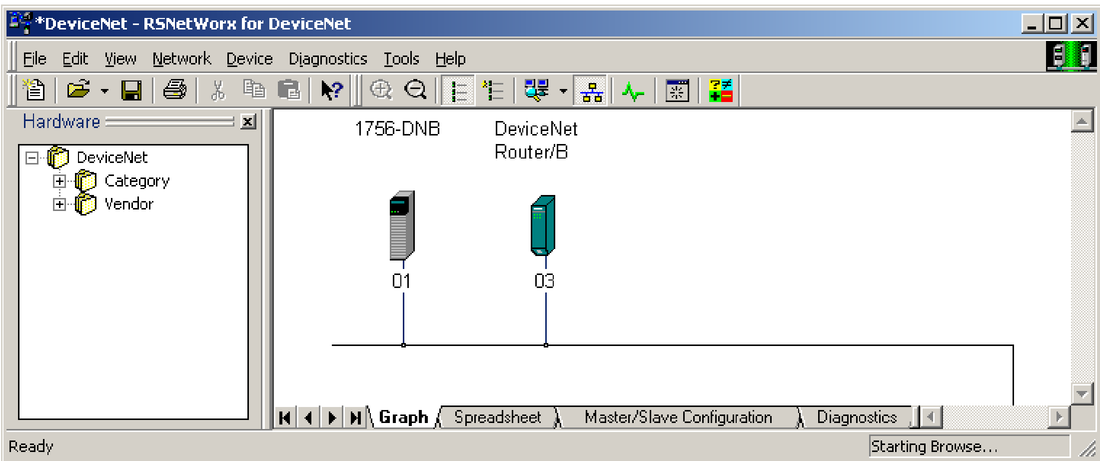


Figure 3.17 – RSNetWorx for DeviceNet

To schedule the master (scanner) module (e.g. 1771-SDN, 1756-DNB) right-click on the scanner module and select ***Properties***.

Select the ***Scanlist*** tab. The DeviceNet Router/B should be shown in the Available Devices (left) list box. Use the "**>**" (add) button to add it to the Scanlist (right).

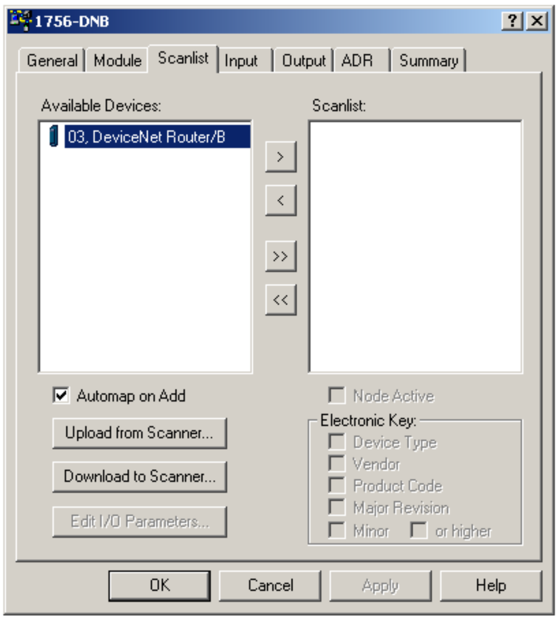


Figure 3.18 – Add device to Scanlist

NOTE: When the DeviceNet Router/B is added to the Scanlist, a dialog may appear, warning of no I/O Data. This is normal, because the DeviceNet Router's I/O data sizes are dynamically configured, and thus are not fixed in the EDS file.

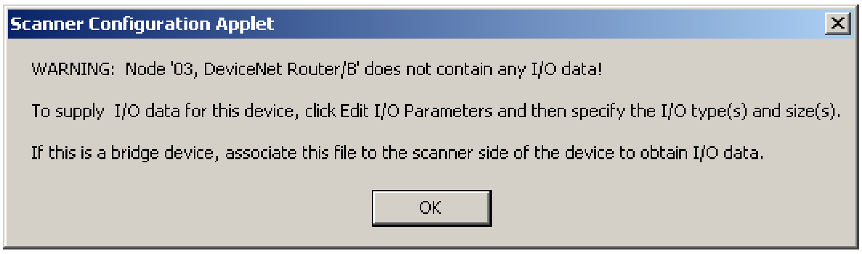


Figure 3.19 – I/O Data Warning

The DeviceNet Router/B will now appear in the Scanlist.

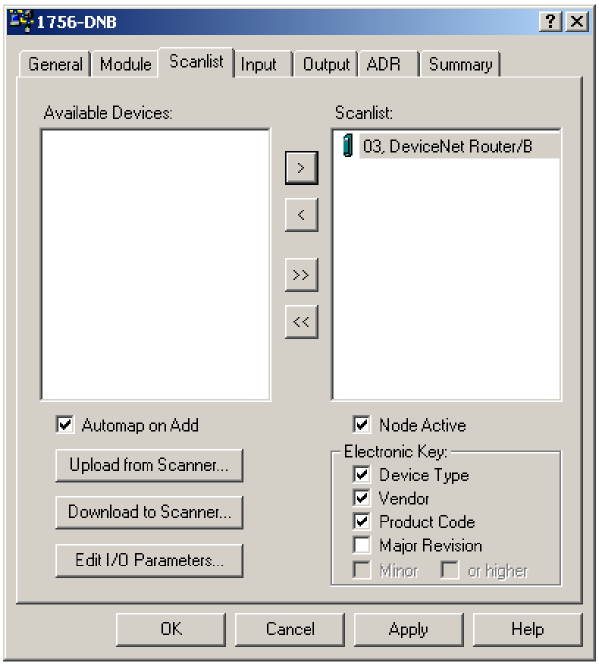


Figure 3.20 – Updated Scanlist

To configure the I/O data sizes, click on the ***Edit I/O Parameters*** button near the bottom of the ***Scanlist*** tab. The ***Edit I/O Parameters*** dialog will appear. The DeviceNet Router/B supports either ***Polled*** or ***Change of State*** (COS) data exchanges.

NOTE: Only one mode should be selected. Do not select both polled and Change of State.

A. POLLED METHOD

To schedule the data transfer using the polling method, select the Polled option.

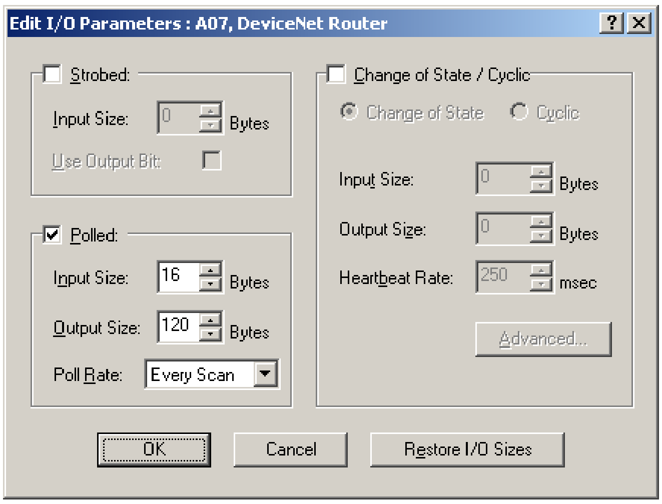


Figure 3.21 – Polled Configuration

The Input and Output sizes are specified in bytes, and should match the DeviceNet Router's configuration, as described the previous chapter.

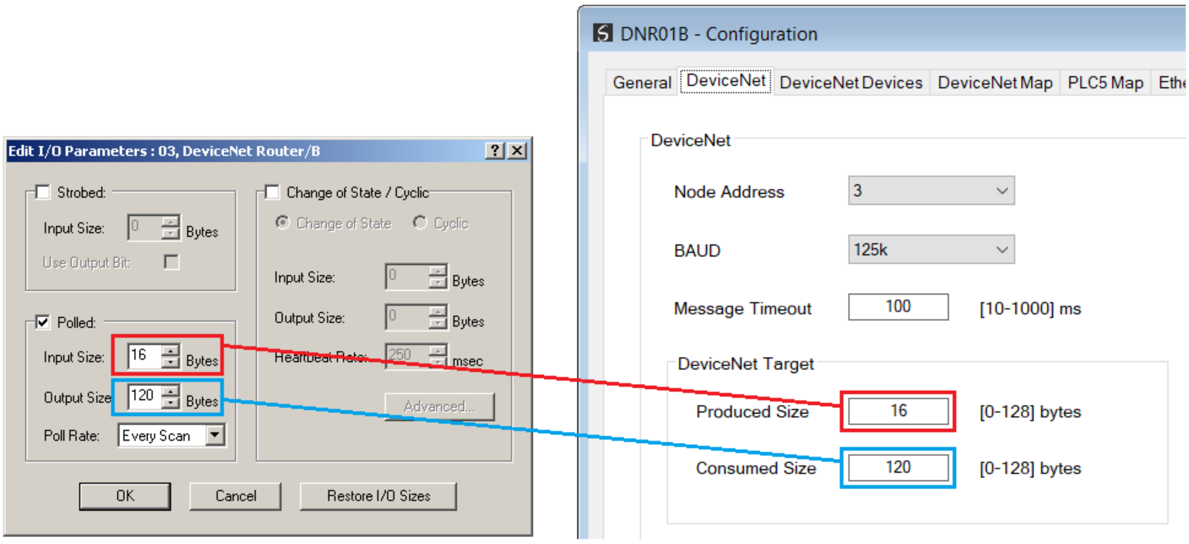


Figure 3.22 – Configuring Polled I/O Sizes

Once configured, click the Ok button.

NOTE: A dialog may again appear, warning that the configured size does not match that of the EDS file. This warning can be ignored.

B. CHANGE OF STATE METHOD

To schedule the data transfer using the ***Change of State*** method, select the "***Change of State / Cyclic***" option.

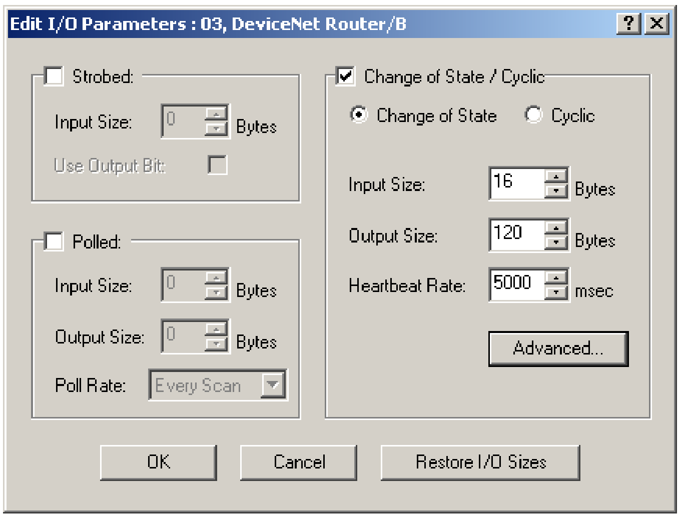


Figure 3.23 – Change of State Configuration

As with the polled configuration, the Input and Output sizes are specified in bytes, and should match the DeviceNet Router's configuration, as described earlier in this chapter.
Once configured, click the Ok button.

NOTE: A dialog may again appear, warning that the configured size does not match that of the EDS file. This warning can be ignored.

C. INPUT AND OUTPUT MAPPING

The scanner module typically transfers a large block of data to the host controller (PLC / PAC). The Input and Output mapping allows the user to specify where in this block the DeviceNet Router's data will appear. To map the Input data, select the ***Input*** tab.

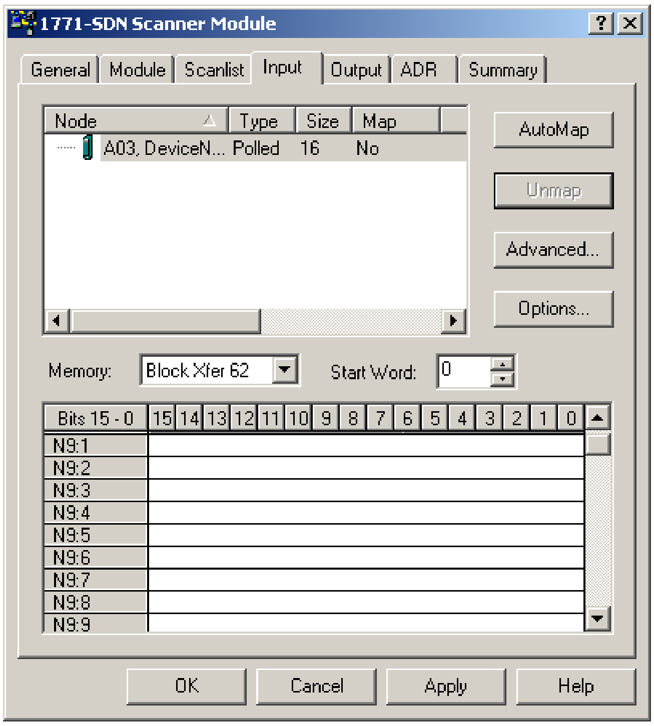


Figure 3.24 – Input Mapping

Select the ***DeviceNet Router/B*** in the "Node" items and click on the ***Advanced*** button. The ***Advanced Mapping*** dialog will appear.

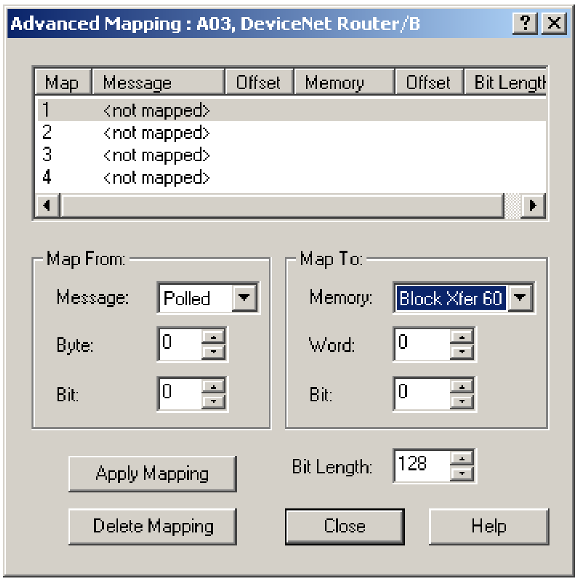


Figure 3.25 – Advanced Mapping – Input

Depending on the previously selected exchange method, the "Map From" Message, will either be ***Polled*** or ***COS***. In the "Map To" group box, enter the appropriate ***Memory*** (Xfer block), ***Word*** and ***Bit*** Offset. Not that the mapping ***Length*** is in bits. In this example, we enter 128 (16 bytes * 8).

Click the ***Apply Mapping*** to accept the configuration. The configured mapping will be illustrated in the lower section of the Input tab.

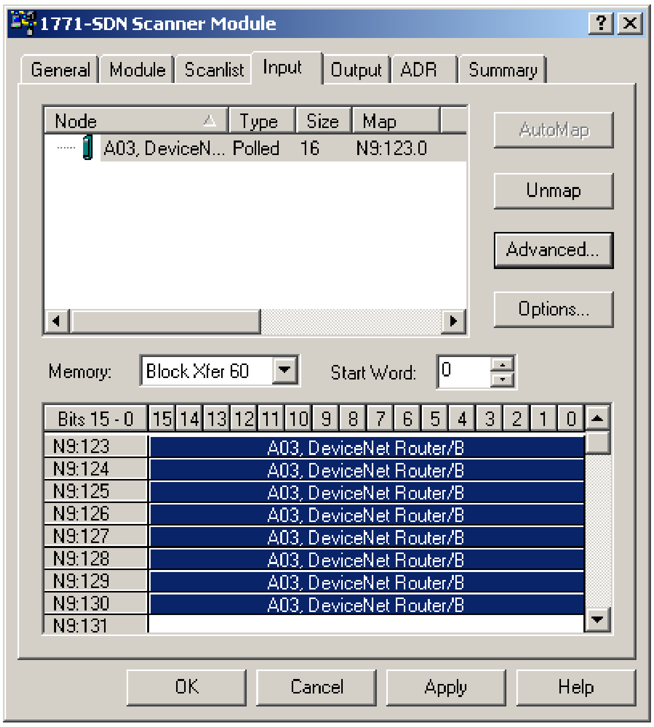


Figure 3.26 – Mapped Input Data

The output data is mapped in a similar method, by selecting the ***Advanced*** button on the ***Output*** tab.

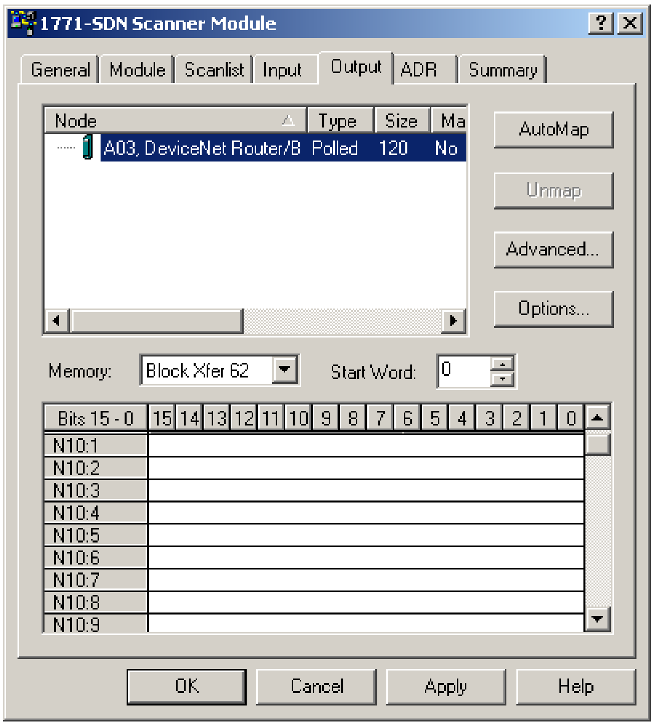


Figure 3.27 – Output Mapping

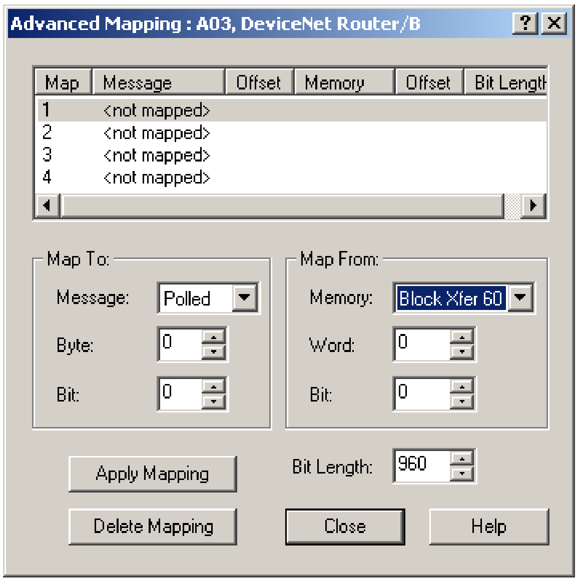


Figure 3.28 – Advanced Mapping – Output

Depending on the previously selected exchange method, the "Map To" Message, will either be ***Polled*** or ***COS***. In the "Map From" group box, enter the appropriate ***Memory*** (Xfer block),

Word and ***Bit*** Offset. Not that the mapping ***Length*** is in bits. In this example we enter 960 (120 bytes * 8).

Click the ***Apply Mapping*** to accept the configuration. The configured mapping will be illustrated in the lower section of the Output tab.

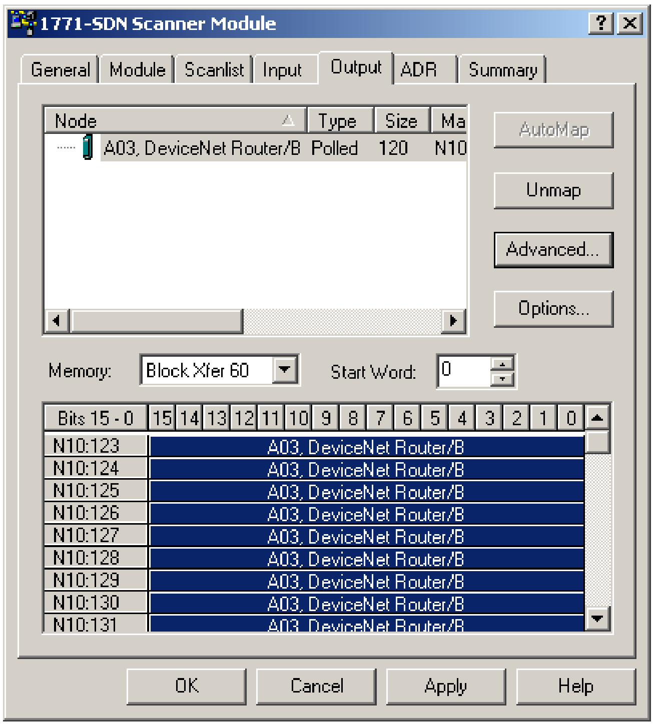


Figure 3.29 – Mapped Output Data

Once the mapping has been configured, select the Ok button on the scanner configuration dialog. The user will then be prompted to download the configuration changes to the scanner. Select the ***Yes*** option.

Figure 3.30 – Scanner Configuration Download Prompt

3.5.1.2. INTERNAL DATA SPACE MAPPING

When the module is operating as a DeviceNet Target, the data from the originator device (e.g. Logix Controller) can be mapped to the Ethernet interface using the Internal Map. The Internal

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

A. IDS COPY – DEVICENET TARGET SOURCE

When copying data from a connection originator (e.g. the output assembly of the 1756-DNB or 1771-SDN) to the Ethernet interface, the source type needs to be DNet Target.

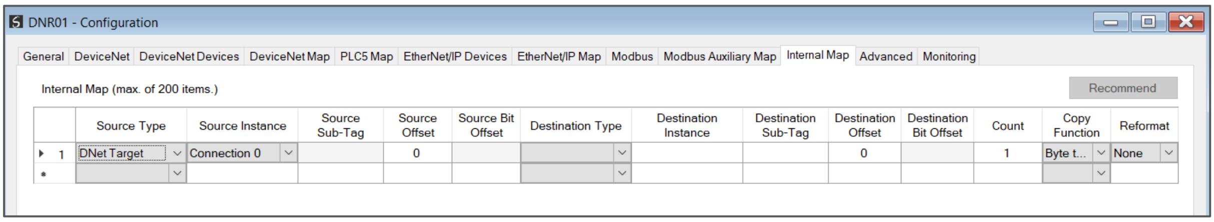


Figure 3.31 – IDS Copy – DeviceNet Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset of the consumed data from the DeviceNet originator (e.g., 1756-DNB) from where the data must be copied. The Count is the number of **bytes** that will be copied.

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

B. IDS COPY – DEVICENET TARGET DESTINATION

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

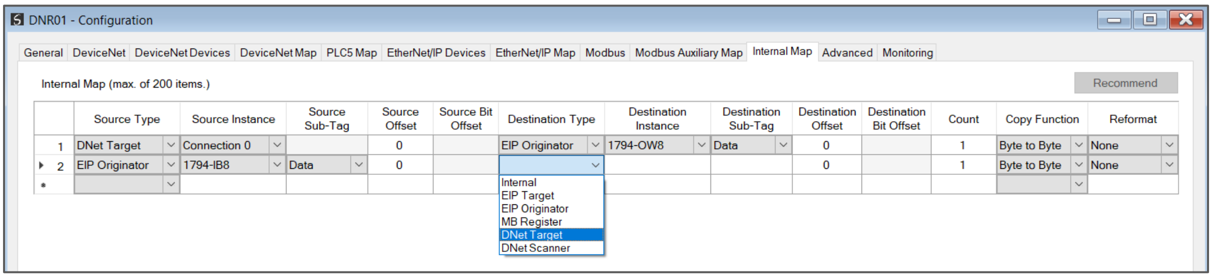


Figure 3.32 – IDS Copy – DeviceNet Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset of the produced data to the DeviceNet originator (e.g., 1756-DNB) from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.5.2. SCANNER

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

3.5.2.1. DEVICENET CYCLIC DEVICE CONNECTIONS

The DeviceNet Router/B can establish up to **63** cyclic DeviceNet connections to DeviceNet IO devices. Before the connection can be setup to the DeviceNet IO, the DeviceNet device EDS file must be loaded into the DeviceNet EDS catalog in the Slate software.

A. UPDATE DEVICENET EDS CATALOG

The user can add DeviceNet EDS files in one of two ways. The first is by using the ***Tools*** menu in the Slate software and selecting the ***DeviceNet EDS File Management*** option.

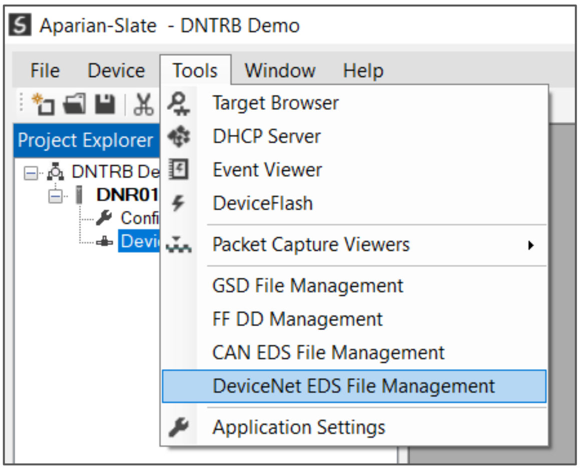


Figure 3.33 – Launching the DeviceNet EDS File Management

Next the user must select the *Add EDS File/s* option from the EDS Files tab in the DeviceNet EDS File Manager window. Next, the user must select the EDS file to be added. The EDS file will then be added to the DeviceNet EDS catalog in Slate.

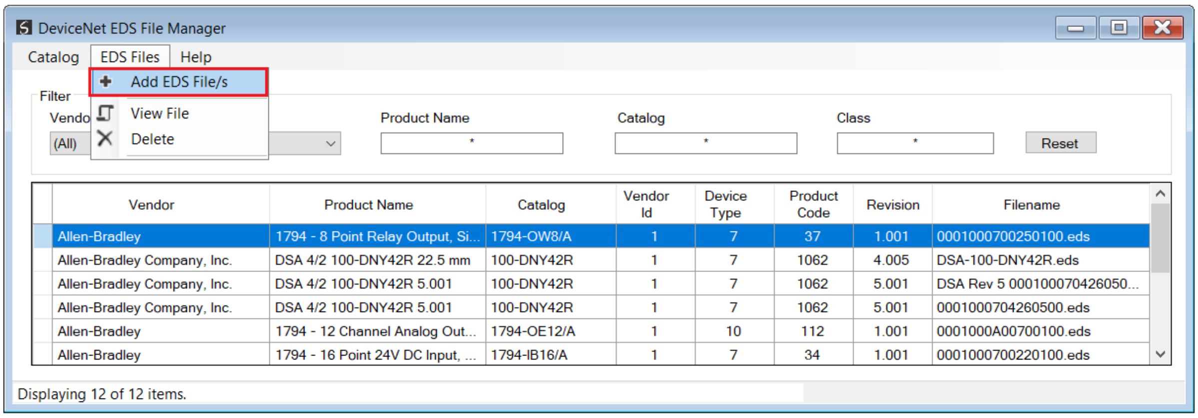


Figure 3.34 – Adding DeviceNet EDS file

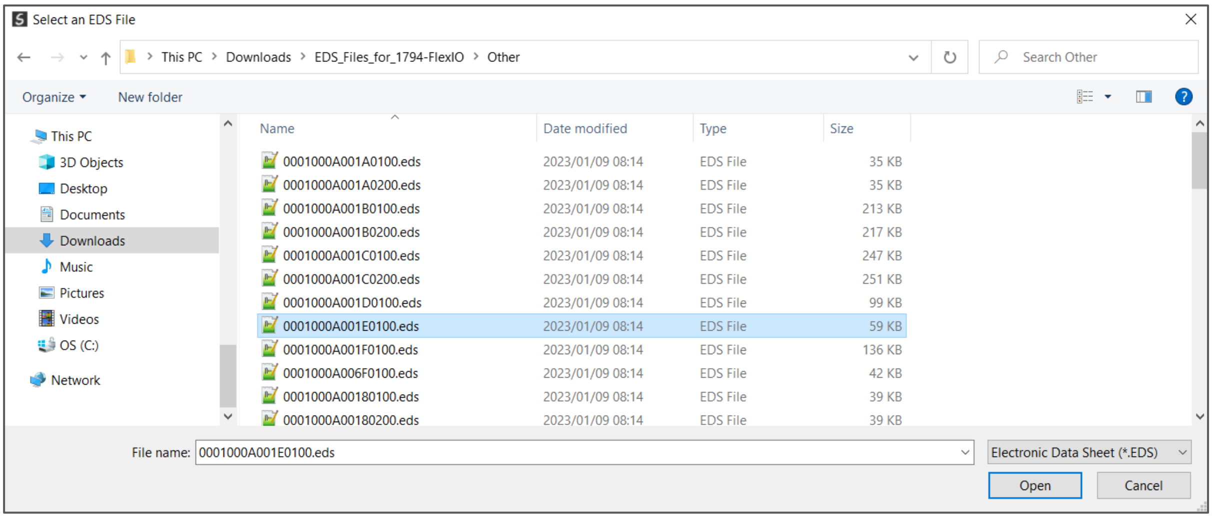


Figure 3.35 – Selecting DeviceNet EDS file

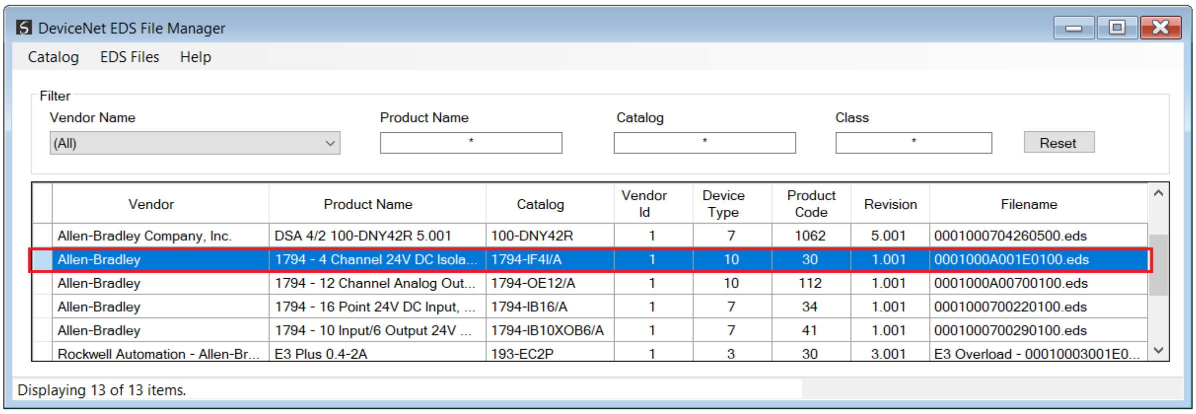


Figure 3.36 – DeviceNet EDS file added

Alternatively the user can add the DeviceNet EDS file when adding a DeviceNet IO device to the *DeviceNet Connections* IO tree in Slate.

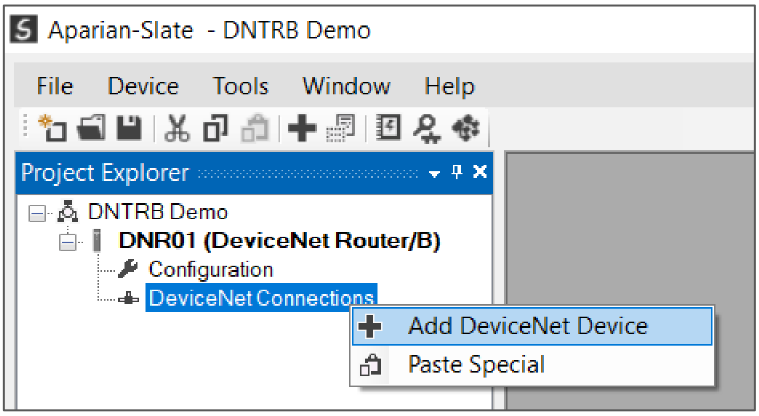


Figure 3.37 – DeviceNet Connection Add

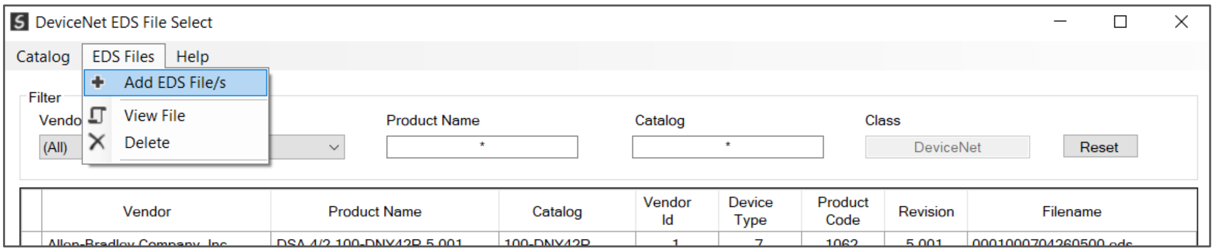


Figure 3.38 – Adding DeviceNet EDS file

B. *ADDING DEVICENET IO DEVICE - DISCOVERY*

The DeviceNet Router/B can scan the DeviceNet network to discover DeviceNet devices. This is done by going online with the module in Slate, and selecting the ***DeviceNet Discovery*** tab.

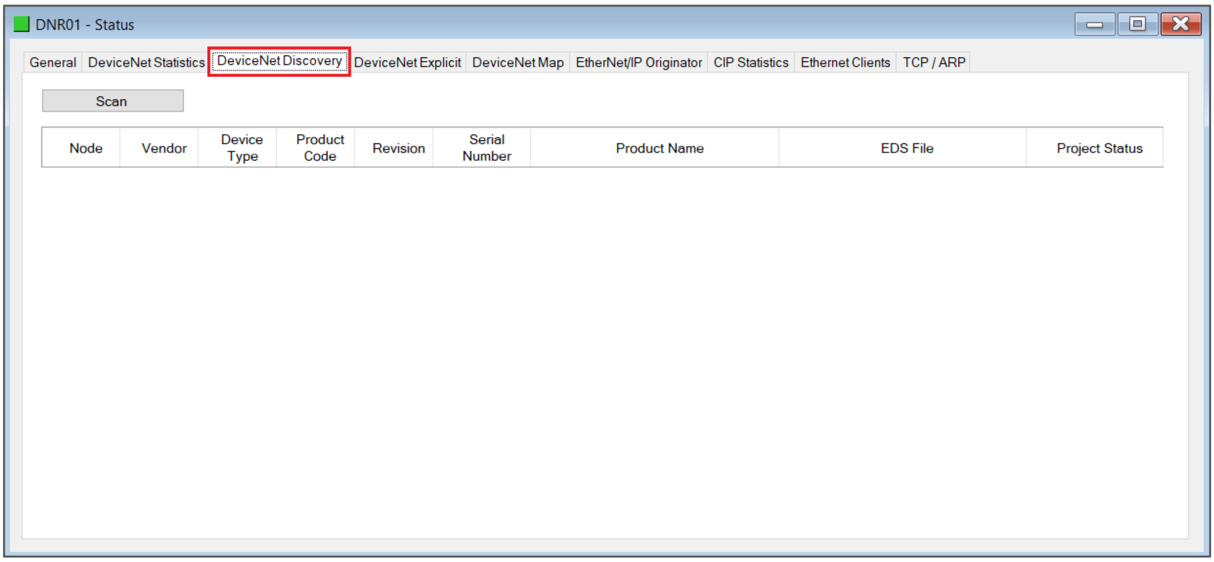


Figure 3.39 – DeviceNet Discovery

After the ***Scan*** button is pressed, the module will start scanning the DeviceNet network for devices. If a device has been found it will be listed in the window and indicate the status of the device.

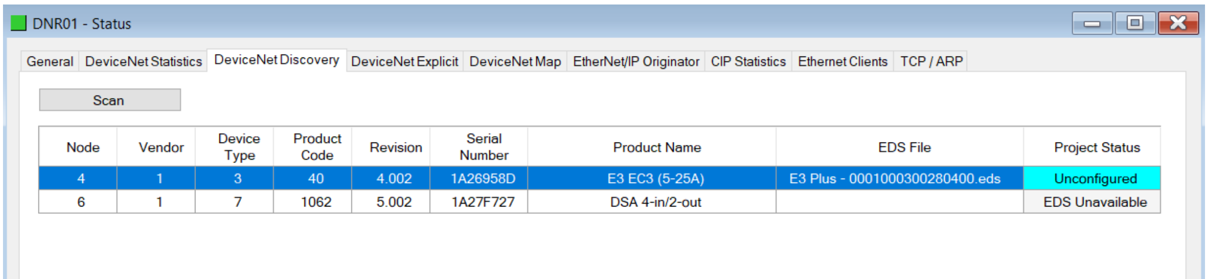


Figure 3.40 – DeviceNet Discovery – Found devices

If a matching EDS file was found for the DeviceNet device found, then it will be listed and the user can add the device to the *DeviceNet Connections* IO tree.

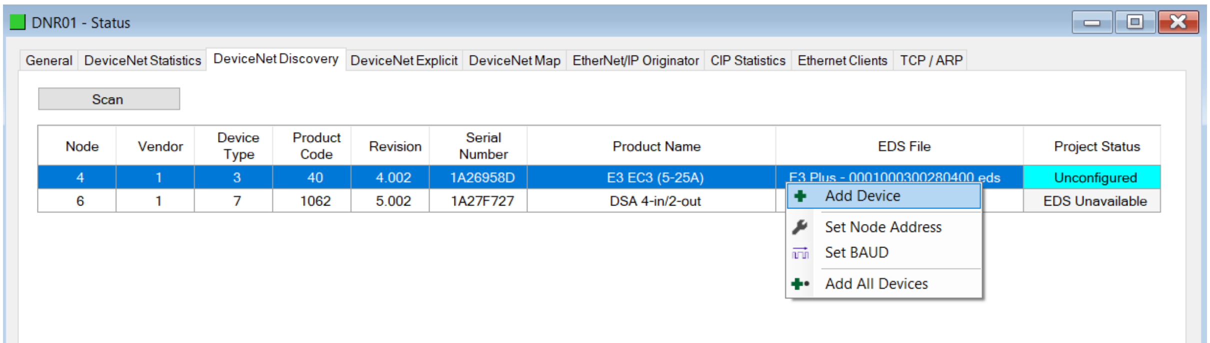


Figure 3.41 – DeviceNet Discovery – Add device

From here the user can follow the DeviceNet IO device Setup.

C. ADDING DEVICENET IO DEVICE - MANUAL

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

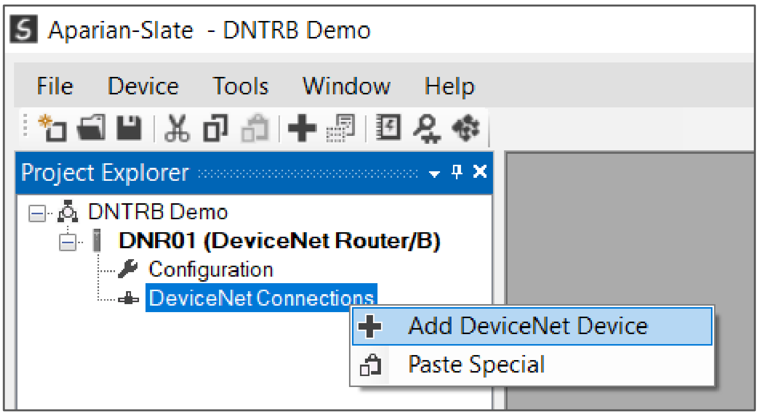


Figure 3.42 – Adding DeviceNet Cyclic IO Device Connection

Next the user will need to select the DeviceNet EDS file for the DeviceNet IO device.

NOTE: Only devices with direct connections (i.e., non-proxied devices) will be allowed to be added directly to the DeviceNet Connections tree. Devices that use proxied connections (e.g., 1794 Flex IO via the 1794 Flex adapter) will need to be added under the adapter.

From here the user can follow the DeviceNet IO device Setup.

D. DEVICENET IO DEVICE SETUP

The user will need to setup the connection parameters for the cyclic DeviceNet communication with the DeviceNet IO device.

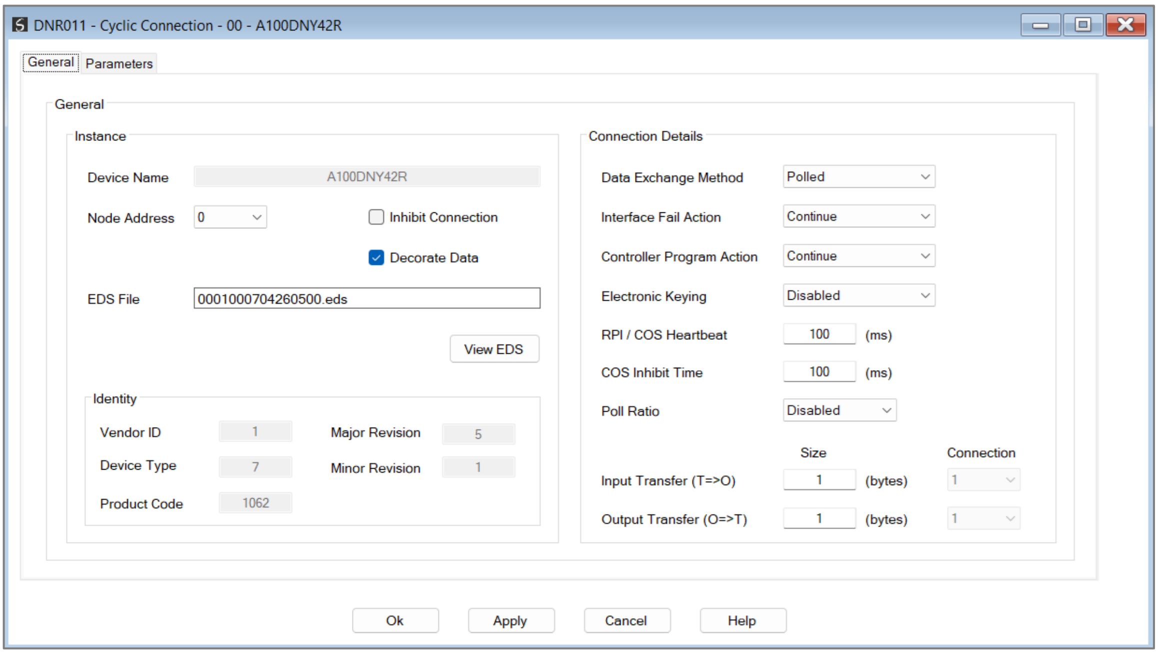


Figure 3.43 – IO Device Cyclic DeviceNet Connection Parameters

Parameter	Description
Device Name	
Device Name	The configurable instance name for the device to be added.
Node Address	The DeviceNet node address of the DeviceNet IO device.
Inhibit Connection	When selected, this option disables the cyclic connection and data exchange between the DeviceNet Router and this device.
Decorate Data	Used in the Logix L5X code generation. When checked, the device's EDS file content will be used to create a device specific UDT structure. When unchecked, the device data will be appear as a SINT array.
EDS File	The EDS file used for the connection parameters.
Identity	
Vendor ID	The Vendor ID extracted from the EDS File.
Device Type	The Device Type extracted from the EDS File.
Product Code	The Product Code extracted from the EDS File.
Major Revision	The Major Revision extracted from the EDS File.
Minor Revision	The Minor Revision extracted from the EDS File.
Connection Detail	
Data Exchange Method	The cyclic connection can be setup to communicate with the DeviceNet IO device in one of two ways: **Polled** When the connection has been setup for polled communication, then data will be exchanged between the DeviceNet Router/B and the DeviceNet IO device at the configured RPI. **Change of State** When the connection has been setup for Change of State (COS) communication, the data will only be exchanged between the DeviceNet Router/B and the DeviceNet IO device when there is a change in the data, thus reducing the traffic on the DeviceNet network. The ***COS Heartbeat*** parameter is required to keep the connection alive, should the data not change for an extended period. The ***COS Inhibit Time*** parameter is used to limit the data exchange rate to avoid flooding the DeviceNet network when the data from either the DeviceNet Router/B or DeviceNet IO device is changing at a rapid rate.
Interface Fail Action	When the Ethernet interface communication has failed, the Cyclic DeviceNet IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the DeviceNet IO to go into a pre-determined state when the communication to the controller (i.e., connection originator on EtherNet/IP) is lost.

Controller Program Action	When the module is configured as a EtherNet/IP Target and the Logix controller goes into a PROG state, the Cyclic DeviceNet IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the DeviceNet IO to go into a pre-determined state when the EtherNet/IP controller enters Program mode.
Electronic Keying	Electronic Keying can be used to ensure that the target device is the correct device type. **Disabled** Keying is not enabled, and no key information will be sent in the connection establishment. **Compatible** Keying has been enabled with compatibility enabled. This will allow devices with older firmware to also establish a connection. **Exact** Keying has been enabled and the exact device with specific firmware revision will allow the establishment of the connection.
RPI / COS Heartbeat	**RPI (when Data Exchange Method is Polled)** The requested packet interval (RPI) is the rate in milliseconds at which the data will be exchanged between the originator and the target device. **COS Heartbeat (when Data Exchange Method is Change of State)** The rate of exchange required to keep the connection alive should there have been no change in data.
COS Inhibit Time	**NOTE: Only relevant when Data Exchange Method is Change of State** The COS Inhibit Time parameter is used to limit the data exchange rate to avoid flooding the DeviceNet network when the data from either the DeviceNet Router/B or DeviceNet IO device is changing at a rapid rate.
Poll Ratio	**NOTE: Only relevant when Data Exchange Method is Polled** The Poll Ratio will determine the actual data exchange rate by dividing the RPI by the Poll Ratio. For example, if the RPI is set to 100ms and the Poll Ratio is set to 4, then the actual packet exchange will be 25ms (100ms / 4) with the expected packet rate set in the device still being 100ms (as this is required by certain devices). When the Poll Ratio is disabled, then the data exchange rate will be at the configured RPI.
Input Size (T=>O)	The size of the data (in bytes) that the DeviceNet IO device will produce.
Output Size (O=>T)	The size of the data (in bytes) that the DeviceNet IO device will consume.

Table 3.3 – Cyclic DeviceNet Connection Parameters

Each DeviceNet IO device can also have device specific parameters that can be modified as required. The user can access this by selecting the *Parameters* tab in the configuration window.

NOTE: To modify or read the parameters from a DeviceNet device, the user will need to be online with the DeviceNet Router/B in Slate.

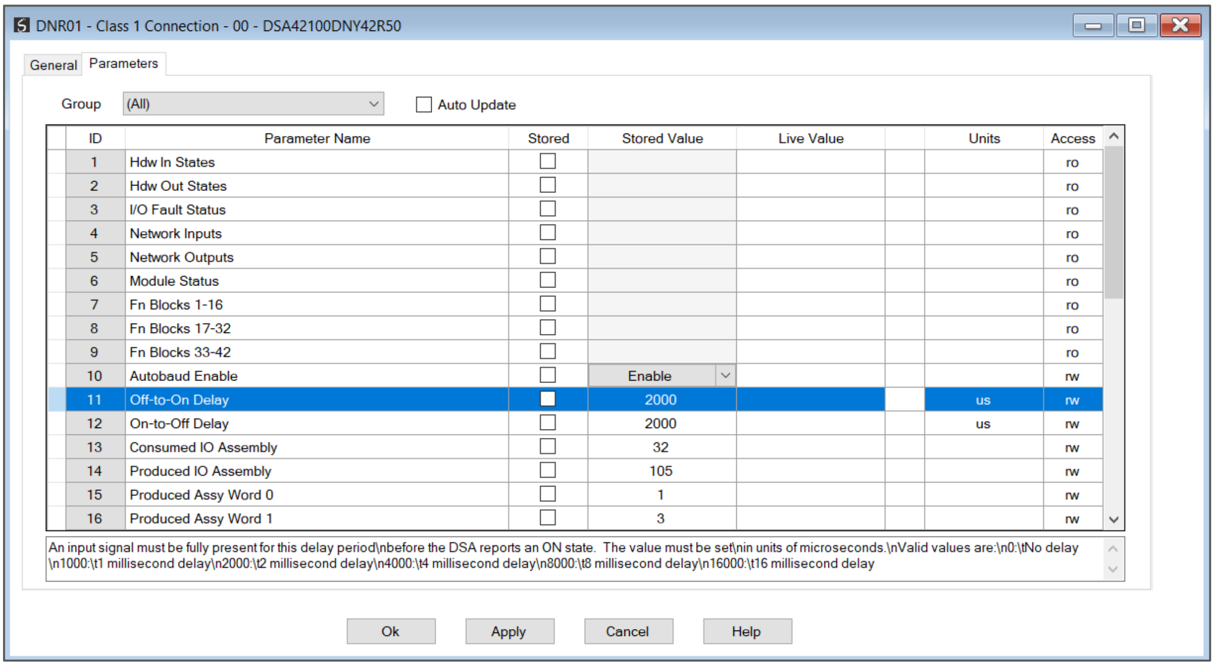


Figure 3.44 – IO Device Cyclic DeviceNet Parameters

Parameter	Description
Parameter Name	The name of the Parameter.
Stored	Option to store the value that has been changed in the project configuration. This will allow a user to retrieve the parameter values configured when uploading a project from the DeviceNet Router/B. See the operation section for more information regarding the DeviceNet IO device parameterization.
Stored Value	The value that was stored for the specific parameter. See the operation section for more information regarding the DeviceNet IO device parameterization.
Live Value	The current parameter value in the DeviceNet device. See the operation section for more information regarding the DeviceNet IO device parameterization.
Units	The engineering unit for the parameter value.
Access	**ro** – The parameter is read-only. **rw** – The user can read or write the parameter value.

Table 3.4 – Cyclic DeviceNet Parameters

E. ADDING DEVICENET FLEX IO

Flex (1794) IO modules can be added to the DeviceNet Router module using the 1794-ADN (DeviceNet Flex I/O Adapter).

NOTE: Before starting this process ensure that all the necessary EDS files (adapter and IO module) have been added.

The addition of the 1794-ADN under the ***DeviceNet Connections*** is no different to that of any other module, that is, by right-clicking on the *DeviceNet Connections* tree and selecting the ***Add DeviceNet Connection*** option.

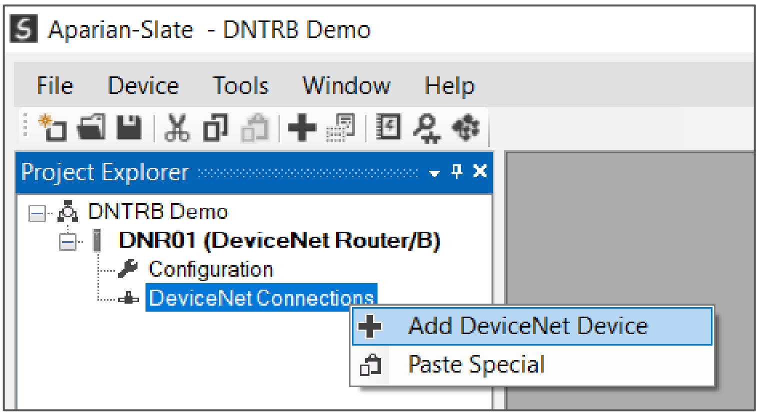


Figure 3.45 – Adding DeviceNet Flex Adapter Connection

In the DeviceNet EDS Selector window, select the **1794-ADN** EDS file.

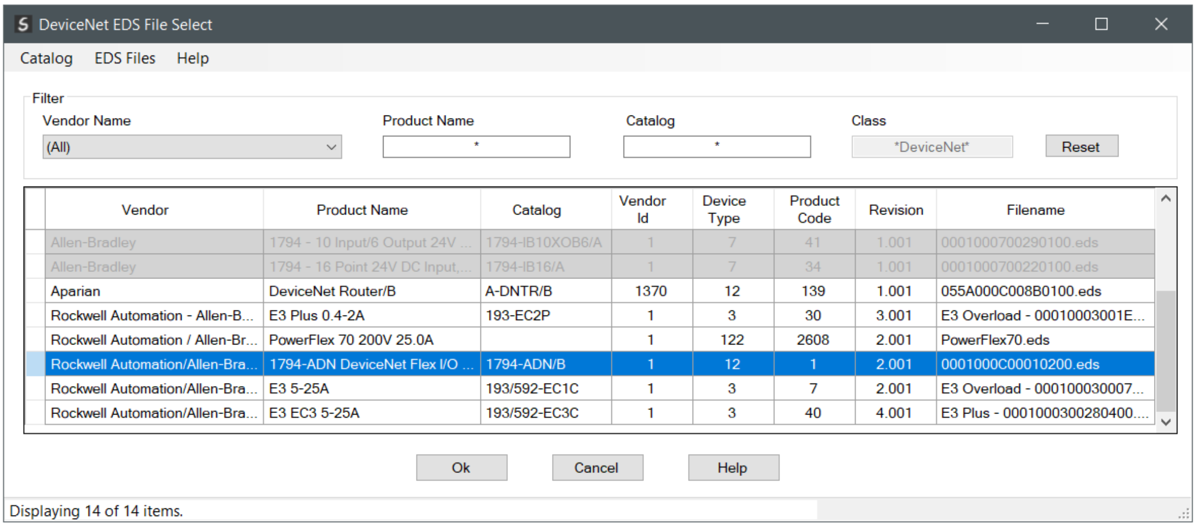


Figure 3.46 – Select the DeviceNet Flex I/O Adapter

Configure the correct DeviceNet node address etc., as described in the previous section, and press ***Ok***. The 1794-ADN will then be added to the tree.

To add a Flex IO module, right-click on the 1794-Adapter and select the ***Add Sub-Module*** option.

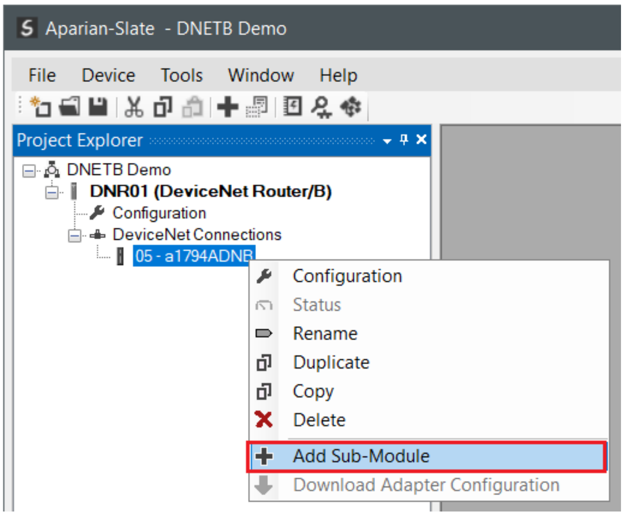


Figure 3.47 – Add Sub-Modules

The DeviceNet EDS Selector window will open with only the 1794 DeviceNet IO modules enabled.

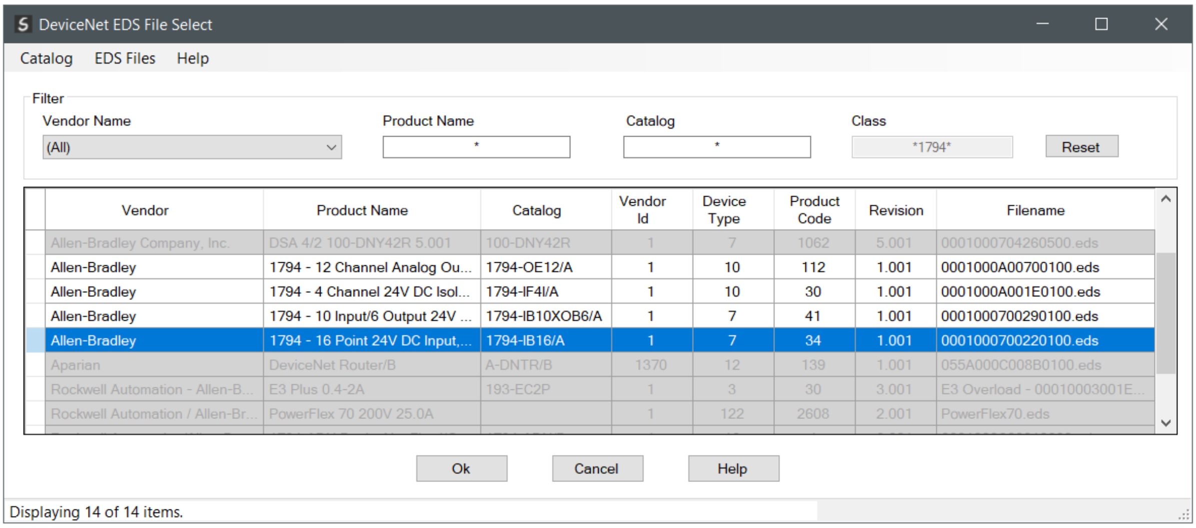


Figure 3.48 – Select the DeviceNet Flex IO Module

Select the Flex IO module to match the hardware. The configuration window for the Flex IO module is slightly different to a normal (direct) connection, for example, the ***Node*** number is replaced with a ***Slot*** number. Ensure that the ***Slot*** number is configured correctly.

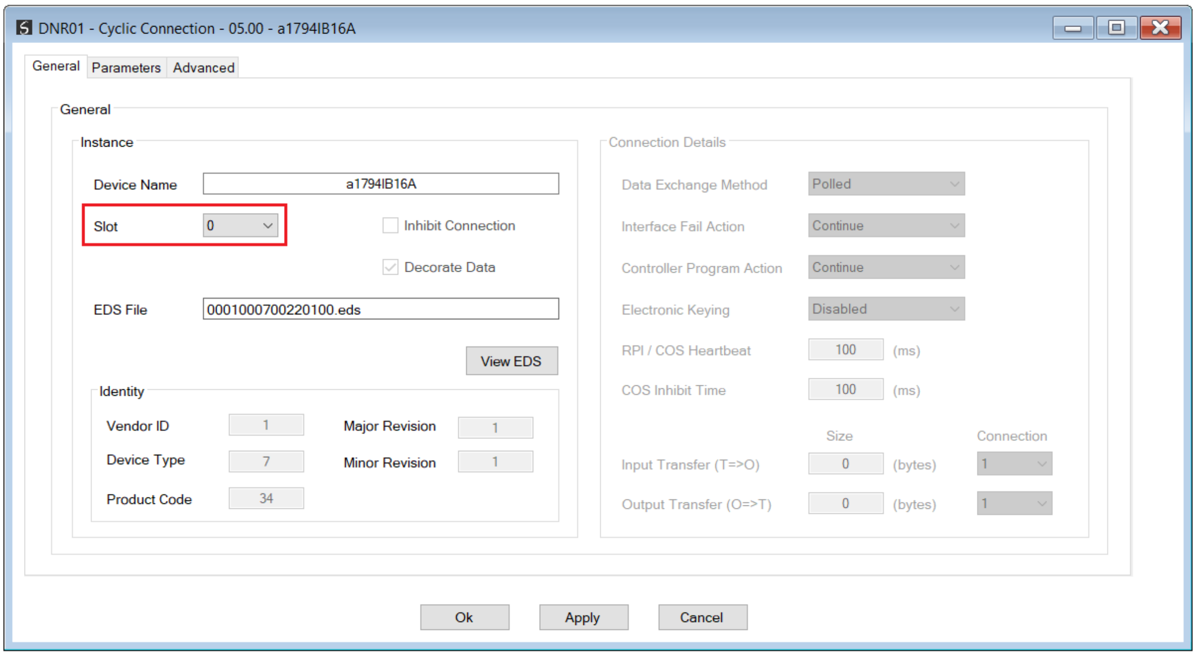


Figure 3.49 – DeviceNet Flex IO Module configuration

The ***Parameters*** tab can be used to configure the Flex IO module.

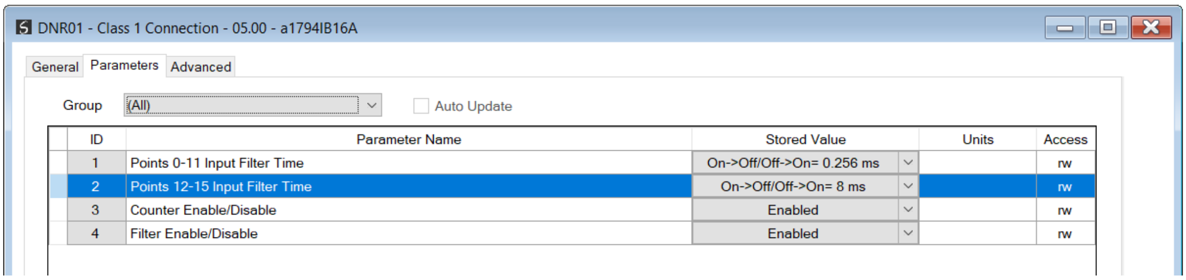


Figure 3.50 – DeviceNet Flex IO Parameter configuration

NOTE: The "Store" option is not shown because all of these parameters are automatically stored in the project.

The Flex IO module's configuration includes an ***Advanced*** tab, which allows the Proxied parameters to be configured. In most cases the default values, which are derived from the EDS file), can be used.

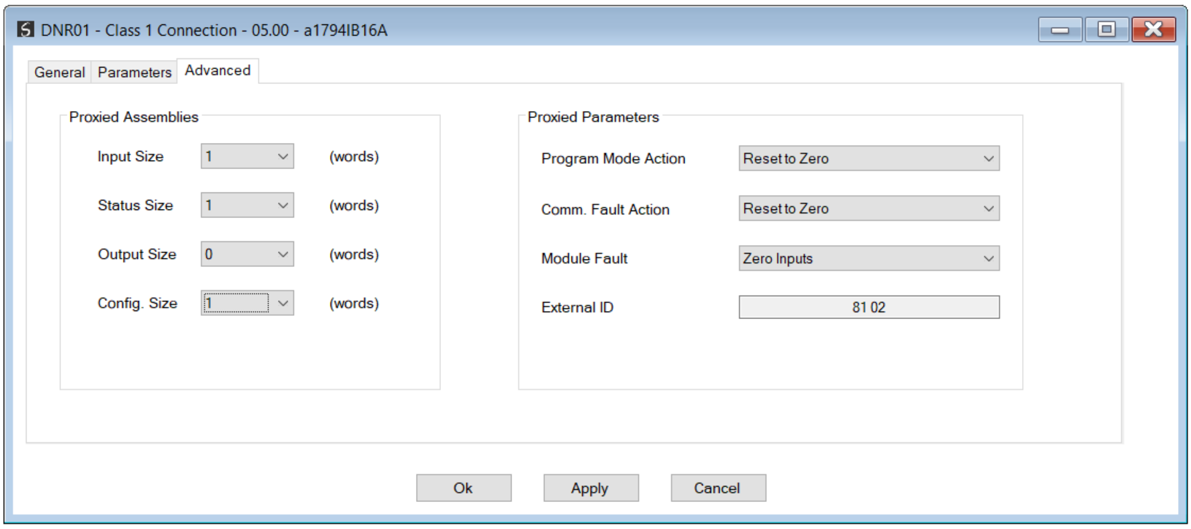


Figure 3.51 – DeviceNet Flex IO Module Advanced configuration

Parameter	**Description**
Proxied Assemblies	
Input Size	The size of the **Input** assembly in words (16bit)
Status Size	The size of the **Status** assembly in words (16bit)
Output Size	The size of the **Output** assembly in words (16bit)
Config. Size	The size of the **Configuration** assembly in words (16bit)
Proxied Parameters	
Program Mode Action	The action to be taken by the IO module when the DeviceNet Connection originator (controller) enters Program mode: **Reset to Zero**: Outputs are set to off **Hold Outputs in Last State**: Outputs do not change. **Use Safe State Outputs**: Outputs are set to safe state values.
Comm. Fault Action	The action to be taken by the IO module when the DeviceNet connection to the Adapter is interrupted: **Reset to Zero**: Outputs are set to off **Hold Outputs in Last State**: Outputs do not change. **Use Safe State Outputs**: Outputs are set to safe state values.
Module Fault	The action to be taken by the IO adapter in the event of the IO module faulting, or not being available: **Zero Inputs**: Inputs are set to off **Hold Inputs in Last State**: Inputs do not change.
External ID	The unique ID indicating the Flex IO module. This field is read only.

Table 3.5 – Advanced Configuration

After the Flex IO module's configuration has been accepted, the parent Flex adapter's (1794-ADN) connection sizes will be automatically readjusted to accommodate the Flex IO modules.

This process must be repeated to add all the required Flex IO modules. The Flex IO modules will be shown in the project tree under the Flex Adapter.

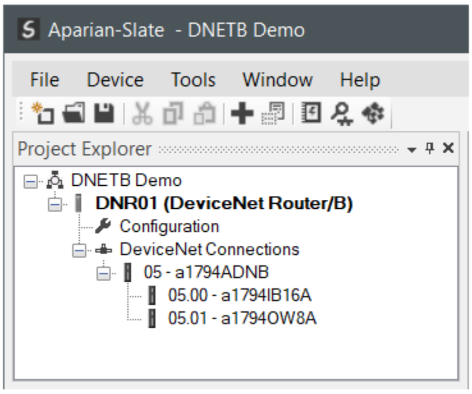


Figure 3.52 – DeviceNet Flex IO Adapter and Sub-Modules

Once the configuration has been downloaded to the DeviceNet Router, the Flex IO adapter's configuration must be downloaded.

Before the adapter configuration can be downloaded, the connection must be inhibited. To do this open the 1794-ADN configuration, check the Inhibit Connection option. This change will trigger another download to the DeviceNet Router.

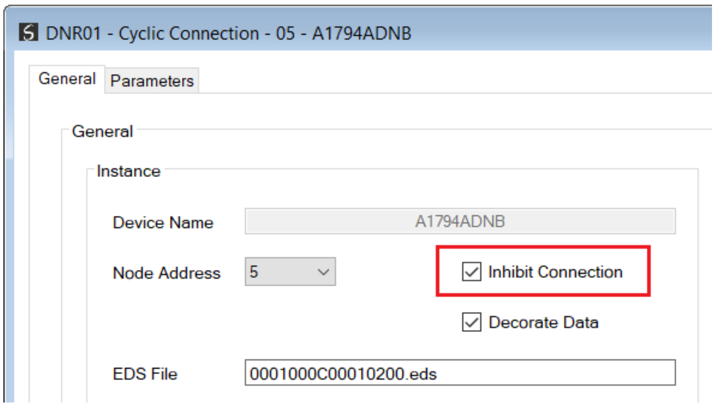


Figure 3.53 – DeviceNet Flex IO Adapter - Inhibit Connection

Once complete, right-click on the 1794-ADN adapter and select the ***Download Adapter Configuration*** option.

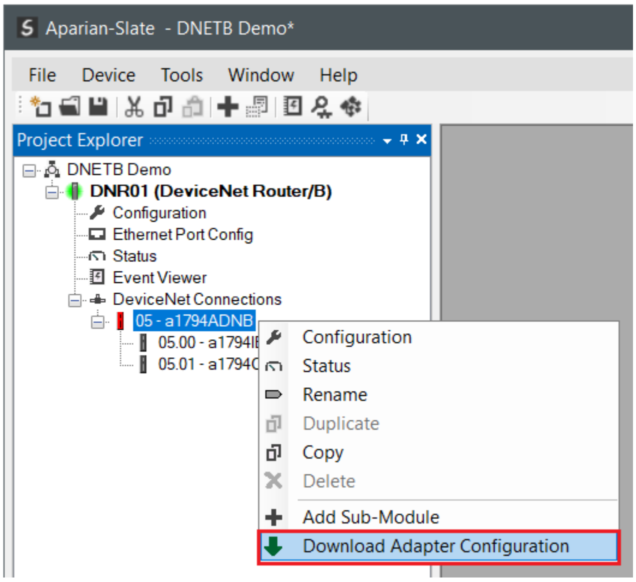


Figure 3.54 – DeviceNet Flex IO Adapter – Download Configuration

Once the download has completed, the ***Inhibit Connection*** can be unchecked, and the configuration redownloaded to the DeviceNet Router.

3.5.2.2. DEVICENET EXPLICIT MESSAGE DEVICE CONNECTIONS

Up to 63 DeviceNet devices can be added for explicit unscheduled messaging. The user will need to add each device as explained in the DeviceNet Devices section below. Once the DeviceNet devices have been added the user can then configure the required mapping for the DeviceNet Explicit messaging as shown in DeviceNet Map section below.

A. DEVICENET DEVICES

This tab is enabled when the DeviceNet Mode is set to *DeviceNet Scanner*.

The DeviceNet Devices configuration is shown in the figure below. Up to 63 DeviceNet devices can be configured with up to 200 DeviceNet mapped items allowing for explicit Unscheduled Unconnected Messaging (UCMM) to any of the 63 configured devices. The data from each DeviceNet device is written to, or read from, an Internal Data Space with a size of 100Kbytes. See the *Explicit DeviceNet Messaging* section for more details.

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

NOTE: A EDS file can be selected for the specific device in the Explicit Devices which will allow the user to select the specific parameters to be accessed in the *DeviceNet Map* from a drop down list without having to manually entering the message parameters.

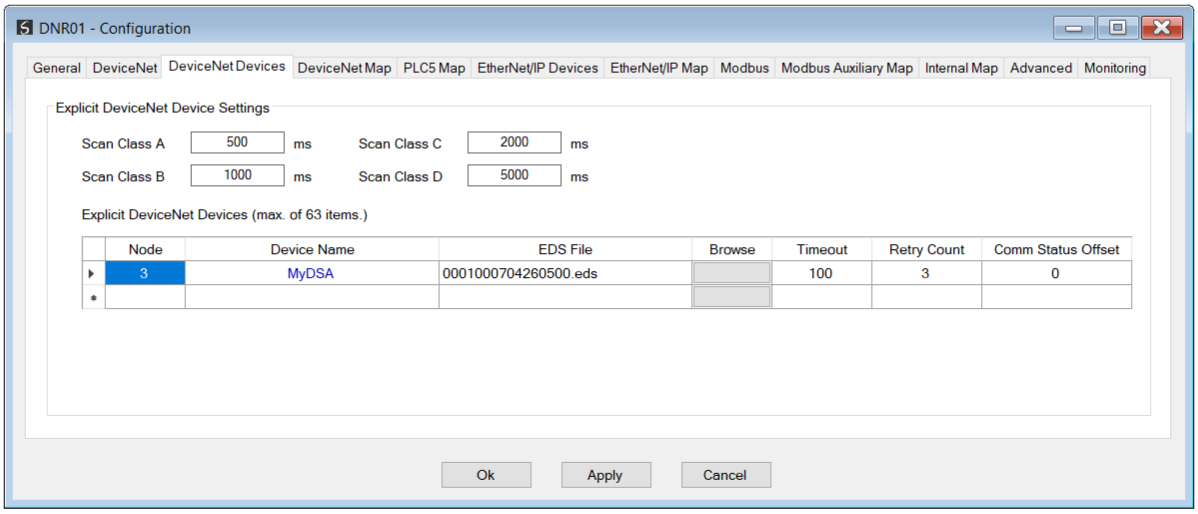


Figure 3.55 – DeviceNet Device configuration

The DeviceNet Devices configuration consists of the following parameters:

Parameter	Description
Scan Class A, B, C, D	The configurable update rates (in milliseconds) for each Scan Class in the DeviceNet Map.
Device List (per device)	
Node	The DeviceNet node number of the target DeviceNet device.
Device Name	The user assigned instance name for the specific device.
EDS File	The EDS file which can optionally be added. The user can browse the DeviceNet EDS catalog by clicking the Browse button next to the EDS File textbox.
Timeout	The amount of time (in milliseconds) the module will wait for a response from the target DeviceNet device.
Retry Count	The number of message retries before the target DeviceNet device is considered offline.
Comm Status Offset	This is the offset in the Internal Data Space (used to map DeviceNet device data) which provides the communication status of each DeviceNet device. The Communication Status is as shown below: Bit 0 - (1:Device Online , 0:Device Offline) Bit 1 to 7 – Reserved.

Table 3.6 – DeviceNet Devices - Configuration Parameters

B. DEVICENET MAP

This tab is enabled when the DeviceNet Mode is set to *DeviceNet Scanner*.

The DeviceNet Map configuration is shown in the figure below. Up to 200 DeviceNet mapped items (UCMM messages) can be configured to any of the 63 pre-configured DeviceNet devices. The data from each DeviceNet device is written to, or read from, the Internal Data Space with a size of 100Kbytes. See the *Explicit DeviceNet Messaging* section for more details.

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

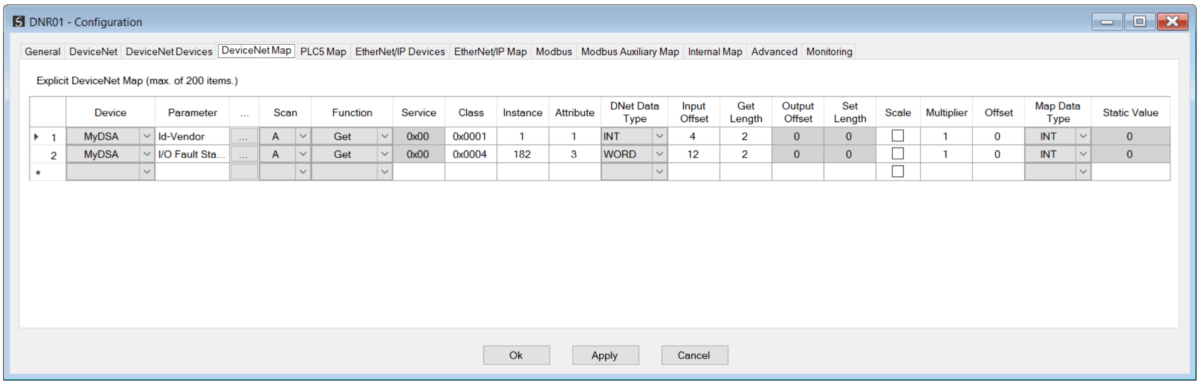


Figure 3.56 – DeviceNet Map configuration

The DeviceNet Map configuration consists of the following parameters:

Parameter	**Description**
Device	The device instance name configured in the previous DeviceNet Devices tab. The selected device will be used for executing the communication function.
Parameter	If the EDS file has been added for the selected DeviceNet device, then the user can browse the parameters of the device (by clicking the button next to the Parameter textbox) and selecting the required parameter. The message parameters (Class, Instance, Attribute, DataType, Scale, etc.) will automatically be populated once the parameter has been selected.
Scan	The user can select Scan Class **A**, **B**, **C** or **D** (which was configured in the DeviceNet 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.
Function	The user can select one of four functions. **Get**

	The module will read data from the target DeviceNet device by using the Get Single Attribute CIP function. The received data will be placed into the Internal Data Space at the ***Input Offset*** location configured in this tab. **Set** The module will write data to the target DeviceNet device by using the Set Single Attribute CIP function. The data to be written will be retrieved from the Internal Data Space at the ***Output Offset*** location configured in this tab. **Set Static** Similar to the Set function above, but the data to be written will be fixed (equal to the *Static Value*) parameter in this configuration window. This function will typically be used with the single (**S**) Scan class which means the DeviceNet Router/B module can be configured 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 ensure the custom service is supported by the target device.
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.
DNet Data Type	The data type of the DeviceNet parameter to be read or written.
Input Offset	The location in the Internal Data Space where the received data will be written. This will only be available for ***Get*** and ***Custom*** functions.
Get Length	The length of the data to be received. If the number of bytes received is more than the Get Length, then the data will **not** be written to the Internal Data Space. This will only be available for ***Get*** and ***Custom*** functions.
Output Offset	The location in the Internal Data Space from where the data to be written to the target device will be read. This will only be available for ***Set*** and ***Custom*** functions.
Set Length	The length of the data to be written. This will only be available for ***Set*** and ***Custom*** functions.
Scale	Certain DeviceNet parameters are scaled. These parameters will need to be adjusted when accessing the parameter to ensure the correct value is written to the DeviceNet IO device or read from the DeviceNet IO device. **NOTE:** When a DeviceNet parameter is selected from the EDS parameters list and a parameter is scaled, then the scaling will be automatically updated in the mapping.
Multiplier	The multiply scaling value.

	NOTE: When a DeviceNet parameter is selected from the EDS parameters list and a parameter is scaled, then the multiplier will be automatically updated in the mapping.
Offset	The offset scaling value. **NOTE:** When a DeviceNet parameter is selected from the EDS parameters list and a parameter is scaled, then the offset will be automatically updated in the mapping.
Map Data Type	The data type which the parameter value to which the parameter value will be converted. For example, if the DNet Data Type is a 16-bit WORD and the Map Data Type is a REAL, then the 16-bit value will be converted into a REAL value before updating (and vice versa).
Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.7 – DeviceNet Map configuration parameters

3.5.2.3. INTERNAL DATA SPACE MAPPING

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

A. IDS COPY – DEVICENET ORIGINATOR SOURCE

When copying data from a DeviceNet IO to the Primary Interface, the source type needs to be DNet Scanner.

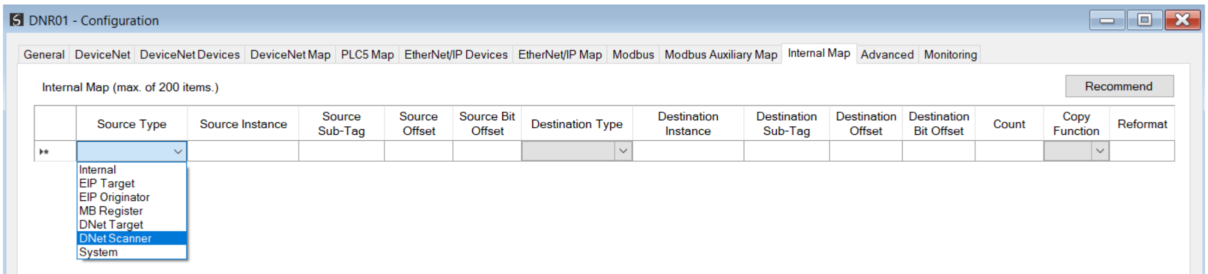


Figure 3.57 – IDS Copy – DeviceNet Scanner Source Type

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

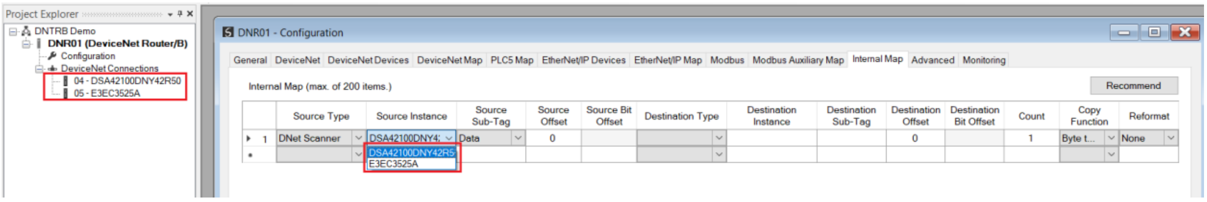


Figure 3.58 – IDS Copy – DeviceNet Scanner Source Instance

The Source Offset is the offset in the selected DeviceNet device Cyclic **Input** Assembly (data being produced by the DeviceNet IO device). The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

The user can select to copy either the Data, or the Status, from DeviceNet connection.

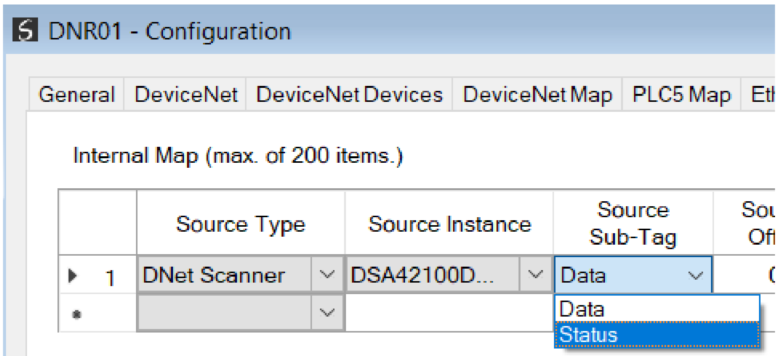


Figure 3.59 – IDS Copy – DeviceNet Originator Status

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

Parameter	Data Type	Description
DeviceNet Scanner Connection Status	DINT	Bit 0 – Online Bit 1 – Cyclic Data Exchange Ok Bit 2 – Device Mismatch
Node	SINT	The target DeviceNet device Node number.
Reserved	SINT	Reserved for future use.
Cyclic Communication Timeout Count	DINT	Number of times the cyclic connection has gone offline.
Cyclic Communication Tx Count	DINT	Number of cyclic connection bytes sent.
Cyclic Communication Rx Count	DINT	Number of cyclic connection byte received.

Table 3.8 – DeviceNet Scanner Connection Status

B. IDS COPY – DEVICENET TARGET DESTINATION

When copying data from the Primary Interface to a DeviceNet IO device's **Output** Assembly, the destination type needs to be DNet Scanner.

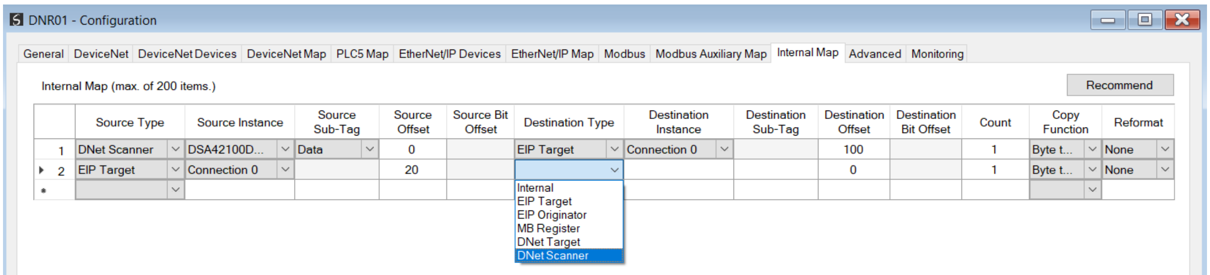


Figure 3.60 – IDS Copy – DeviceNet Scanner Destination Type

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

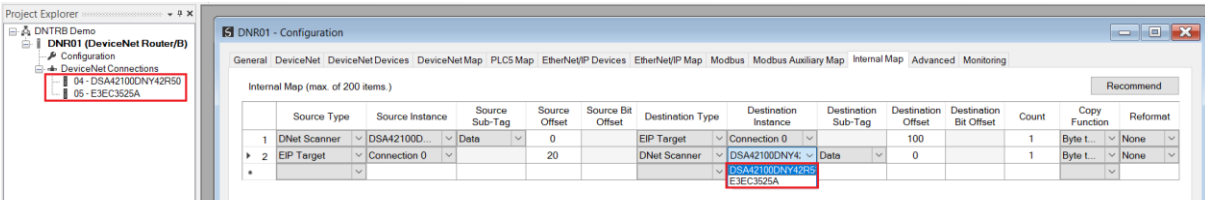


Figure 3.61 – IDS Copy – DeviceNet Scanner Destination Instance

The Destination Offset is the offset in the selected DeviceNet device Cyclic **Output** Assembly (data being consumed by the DeviceNet IO device). The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.6. PRIMARY INTERFACE

The DeviceNet Router/B module supports five different modes for the Primary Interface.

3.6.1. PCCC CLIENT

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

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

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

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

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

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

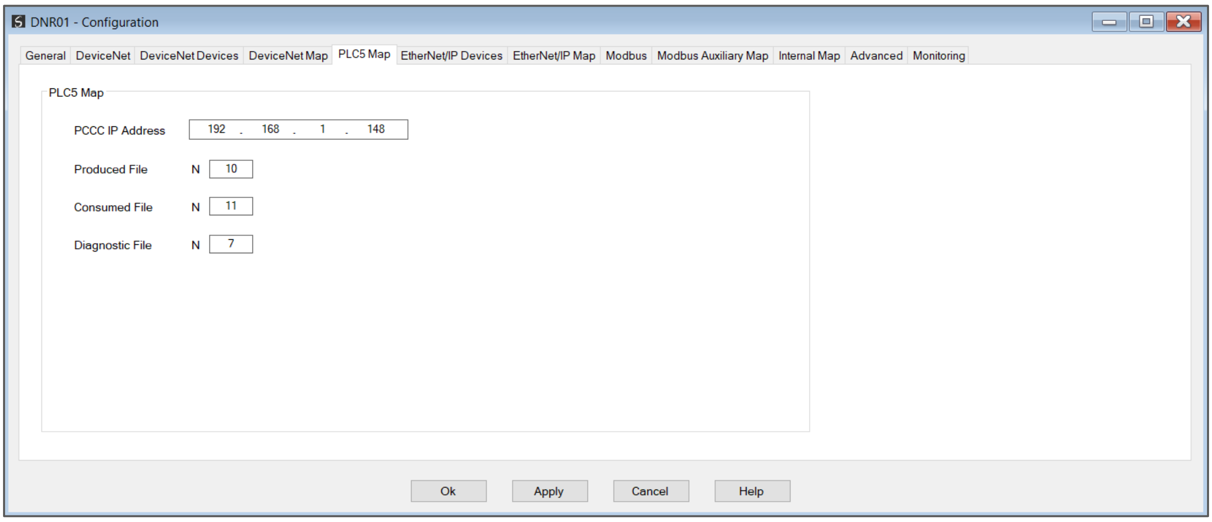


Figure 3.62 – PCCC Client Mode – IP Address Configuration

Parameter	**Description**
PCCC IP Address	The IP Address used to emulate a PLC5 controller. This IP address will be seen as the IP address of the PLC5 controller that the DeviceNet Router/B is emulating. Therefore, there will be two IP addresses on the network when in PLC5 Emulation mode. One for the actual target module and one for the emulated PLC5 controller.
Produced File	The “PLC5” data file emulation to be used for the DeviceNet produced data.
Consumed File	The “PLC5” data file emulation to be used for the DeviceNet consumed data.
Diagnostic File	The “PLC5” data file emulation to be used to expose the module’s diagnostic data.

Table 3.3 – DeviceNet configuration parameters

3.6.1.1. FTVIEW CONFIGURATION

The DeviceNet Router/B can be interfaced directly to FTView using PCCC (PLC5 Ethernet emulation). This is illustrated in the following example where a PanelView is configured to read data from the DeviceNet Router/B.

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

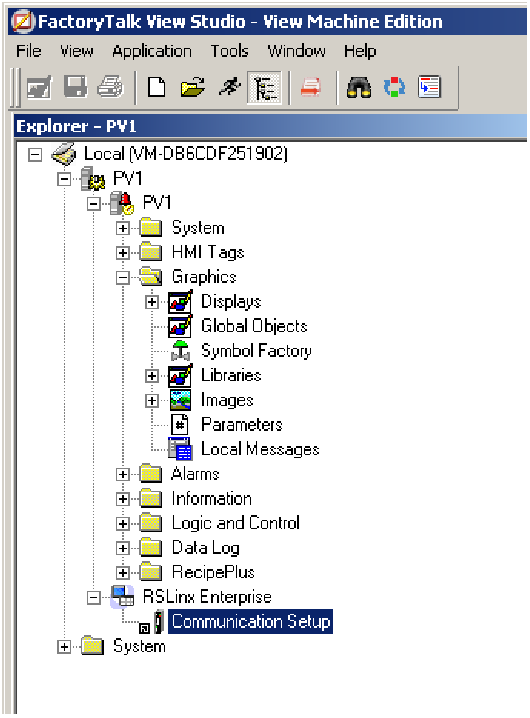


Figure 3.63 - FTView project

A. COMMUNICATION

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

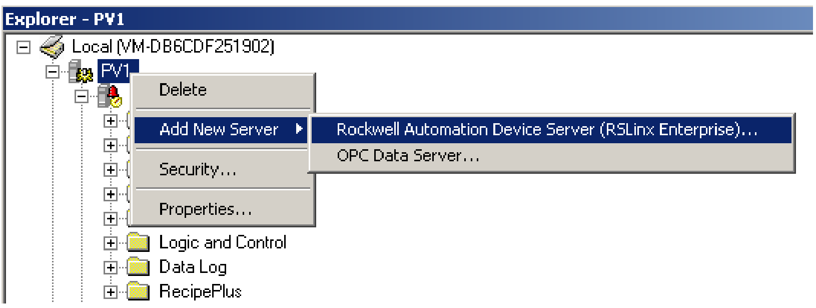


Figure 3.64 – Adding RSLinx Enterprise Server

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

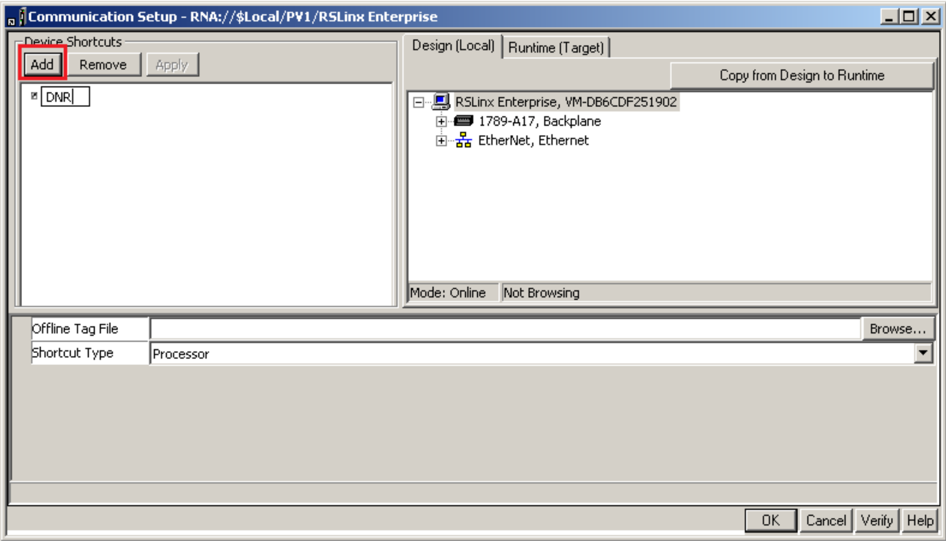


Figure 3.65 – Adding Device Shortcut

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

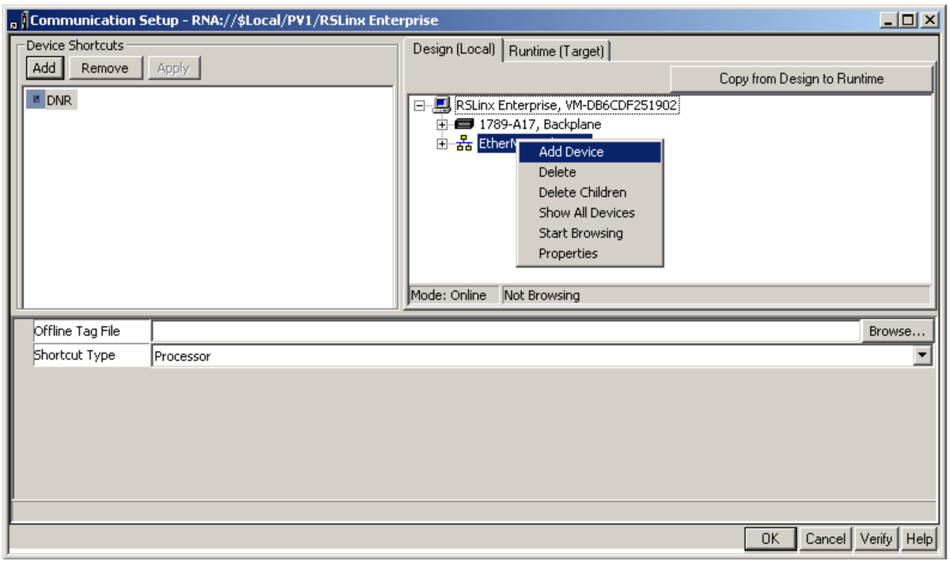


Figure 3.66 – Adding Ethernet Device

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

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

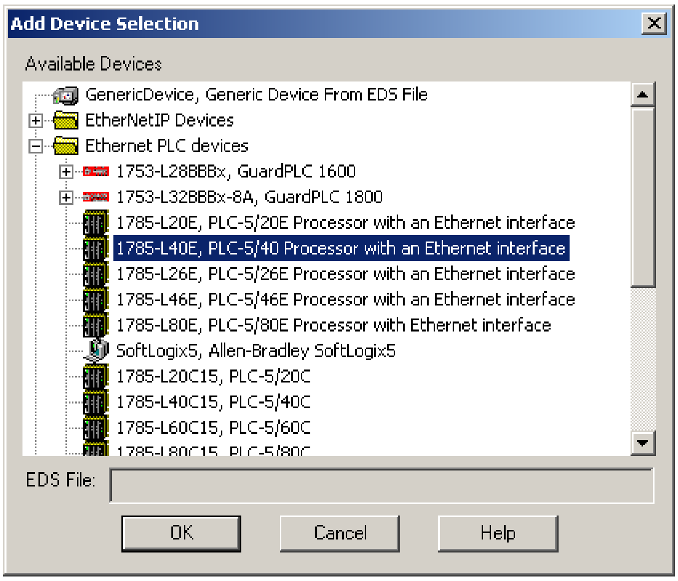


Figure 3.67 – Ethernet Device Selection

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

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

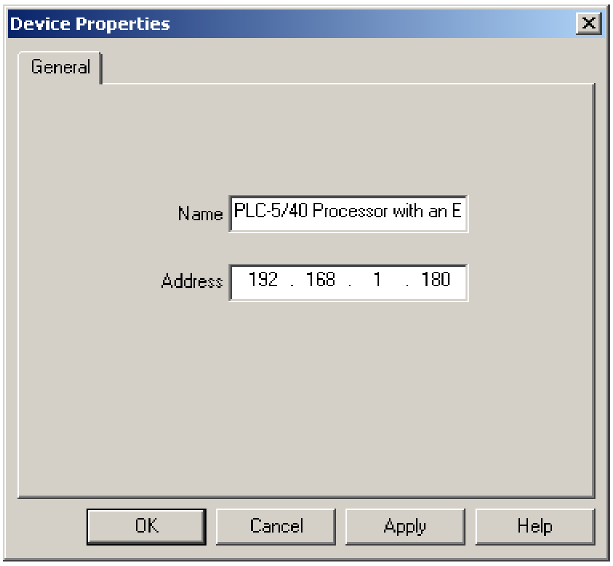


Figure 3.68 –Device Properties

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

B. ANIMATION

Once the communication has been correctly configured, objects can be linked to the DeviceNet Router/B data points. Create a new graphic Display by right-clicking on the ***Display*** item, under the ***Graphics*** section.

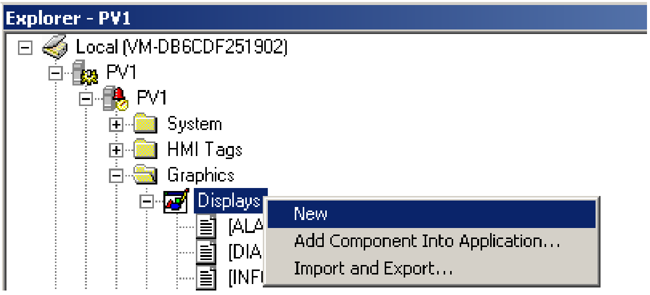


Figure 3.69 – Adding a Graphical Display

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

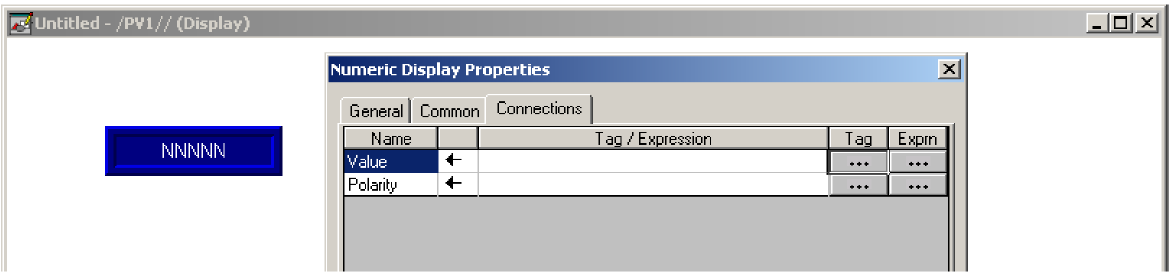


Figure 3.70 – Numeric Display Connections

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

NOTE: The DeviceNet Router/B must be online for the tag browsing option to work.

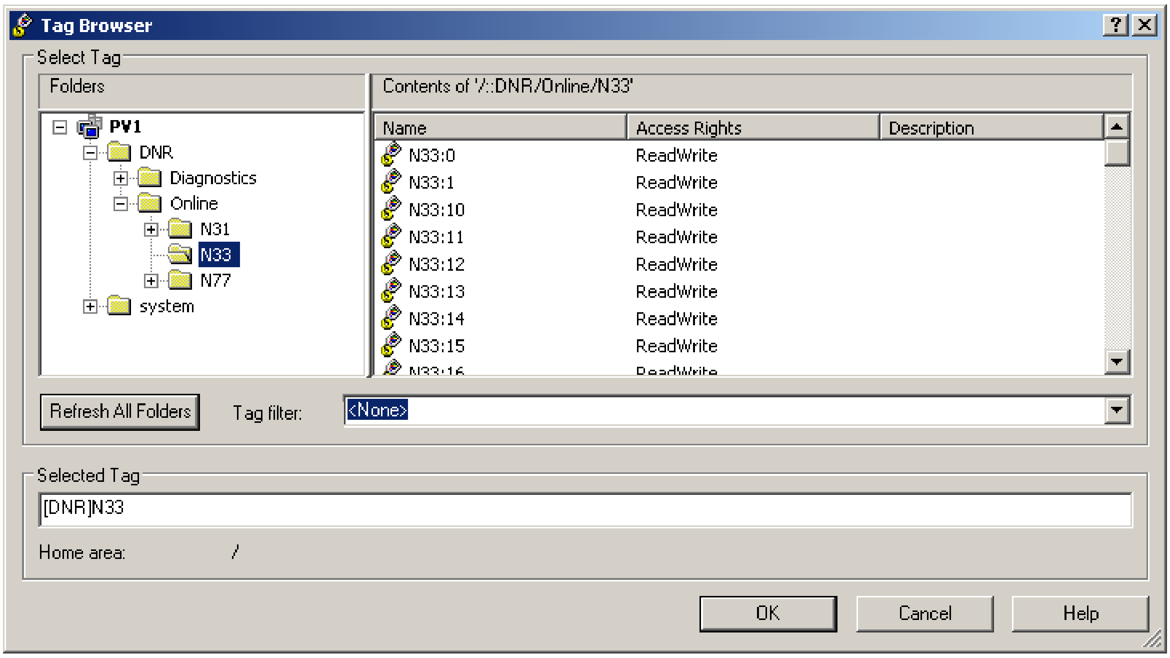


Figure 3.71 – Browsing Data Points

The data files, as configured in the DeviceNet Router's PLC5 Map tab, will appear under the ***Online*** section. Select the required data point to be connected to the Numeric Display. In this example, N31 and N33 represent the Input and Output data respectively and N77 represents the diagnostic data.

C. DIAGNOSTIC DATA

Various diagnostic items can be displayed in the FTView using the Diagnostic File. The diagnostic file number (e.g. N77) is configured in the DeviceNet Router's PLC5 Map tab. The meaning of each diagnostic data point is tabulated below.

Offset	Group	Description
0	DeviceNet	DeviceNet Polling Status Bit 0 – Connection Active - Poll Bit 1 – Connection Standby Bit 2 – Connection Active - COS
1		Rx Can Packet Count
2		Tx Can Packet Count
3		CAN CRC Errors
4		CAN Bit Errors
5		Can Stuff Errors
6		UCCM Connection Open
7		UCCM Connection Close
8		IO Connections
9		Poll Commands
10		Fragment Ack Errors

11		Explicit Fragment Error
12		Poll Fragment Error
13		Explicit Client Not Found
14		Duplicate Node Detected
15	PCCC - Ethernet	PCCC Connection Requests
16		PCCC Read Requests
17		PCCC Write Requests
18		PCCC Unsupported Command
19		PCCC Unsupported FNC Code
20		PCCC Client Not Found
21		PCCC Client Max Reached
22		PCCC File Not Found
23		Current Connections
24	Module	DeviceNet Router/B Internal Temperature

Table 3.9 - Diagnostic File

3.6.2. EtherNet/IP Target

A controller (e.g. Logix controller) can own the DeviceNet Router/B over EtherNet/IP using up to 4 Class 1 EtherNet/IP connections when the Primary Interface is set to EtherNet/IP target. This will allow the DeviceNet Router/B to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connections. Data from DeviceNet devices can be mapped to the Logix controller over EtherNet/IP.

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

3.6.2.1. STUDIO / LOGIX 5000 CONFIGURATION

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

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

NOTE: Logix version 21 and newer supports EDS AOPs.

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

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

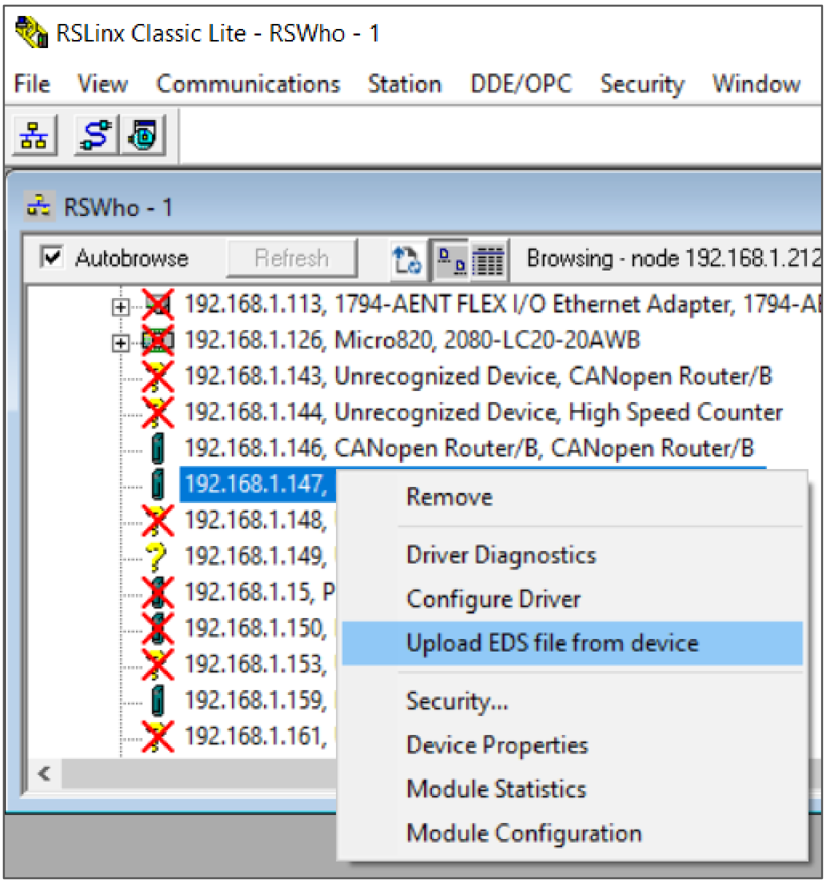


Figure 3.72 – EDS file upload from DeviceNet Router/B

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

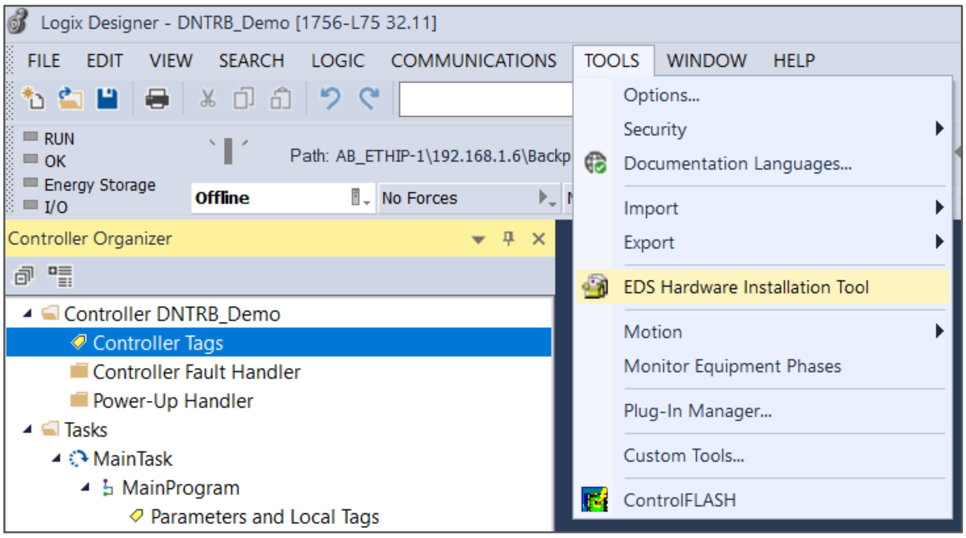


Figure 3.73 - EDS Hardware Installation Utility

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.

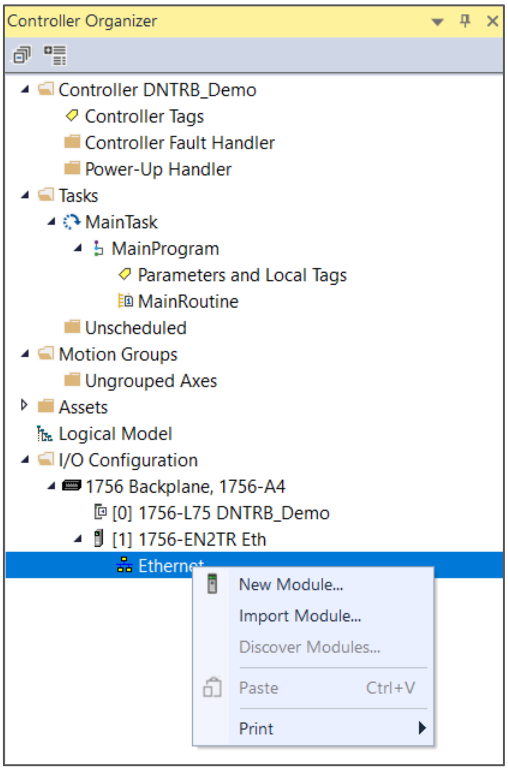


Figure 3.74 – Adding a module

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.

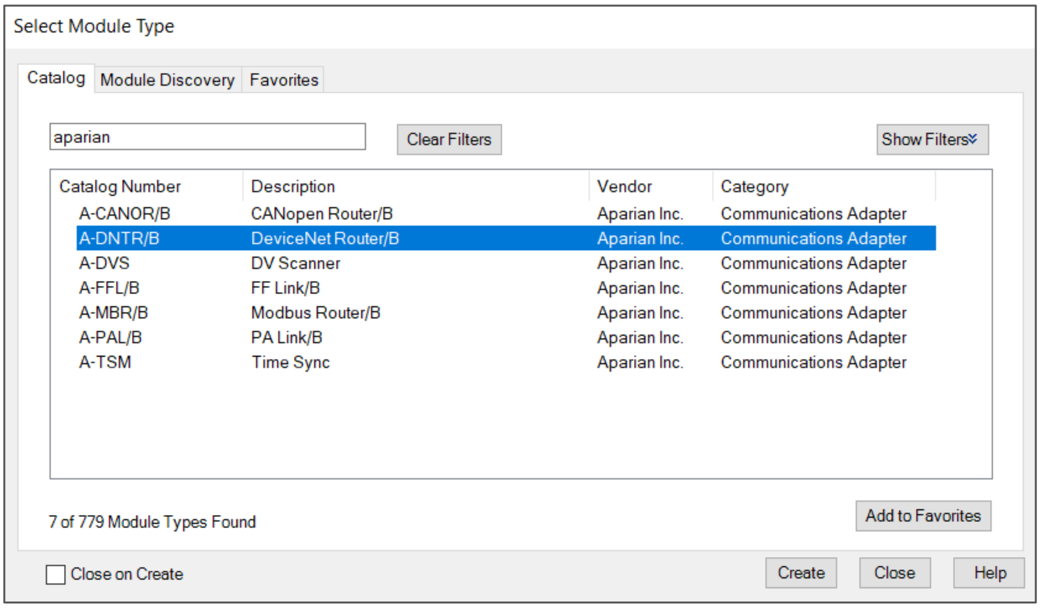


Figure 3.75 – Selecting the module

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

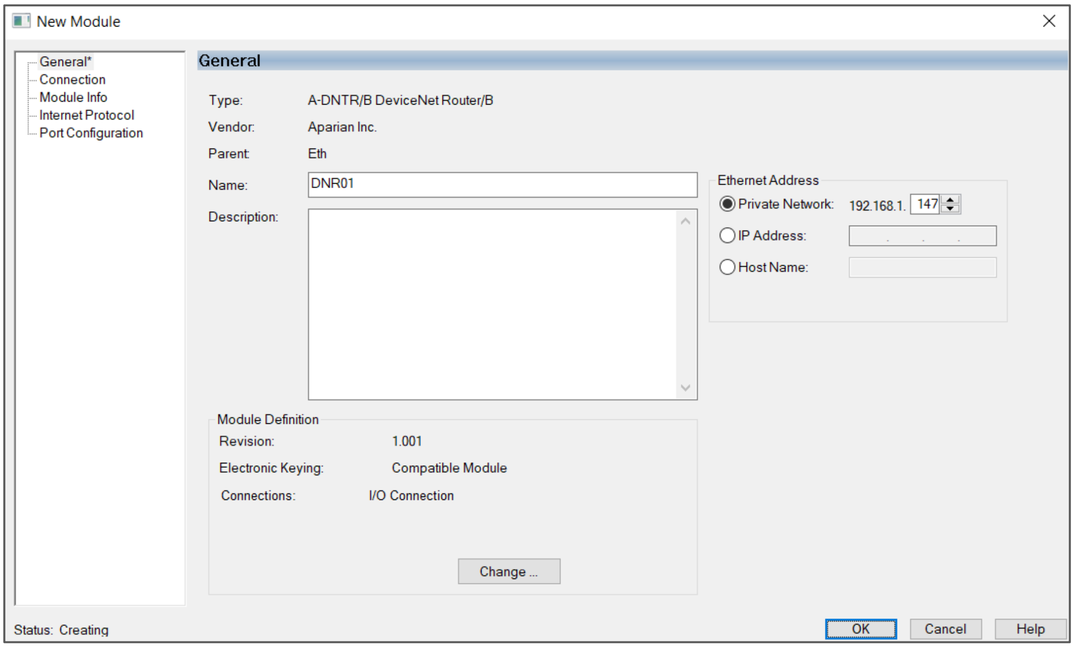


Figure 3.76 – Module instantiation

The DeviceNet Router/B supports up to 4 class 1 EtherNet/IP connections. The user will need to **ensure that the number of connections configured in the General tab of the module configuration matches the selected connection count in Logix**.

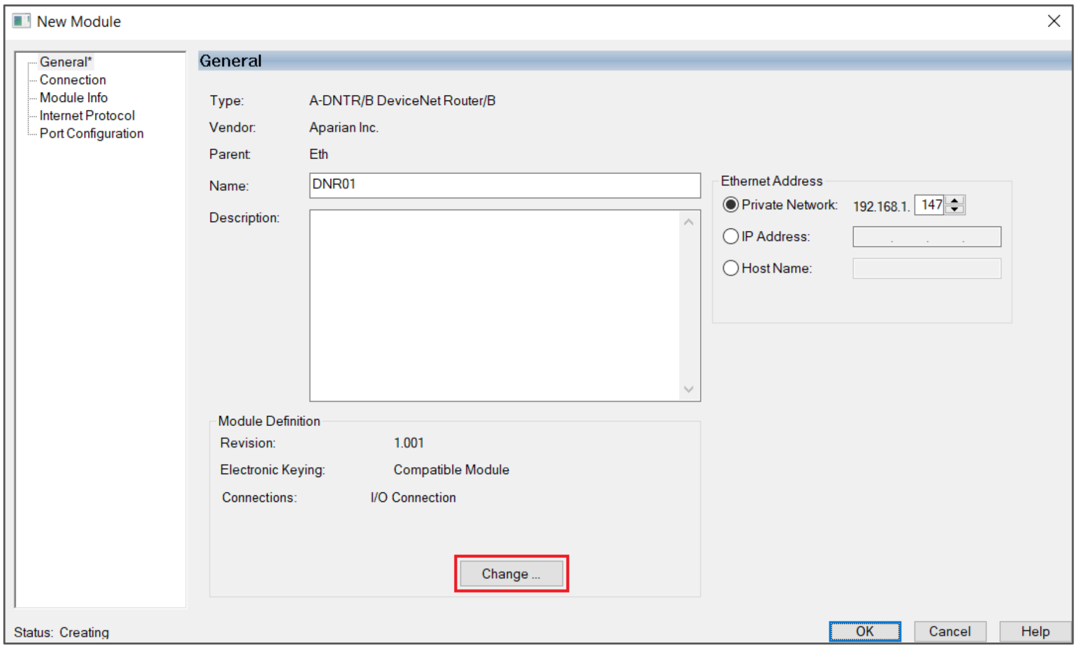


Figure 3.77 – Change number of IO Connections

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

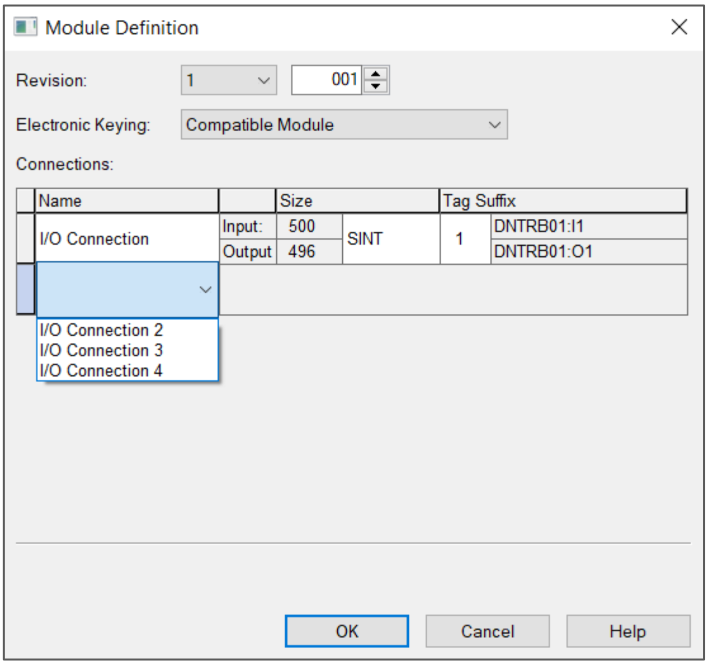


Figure 3.78 – Selection of IO Connections

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

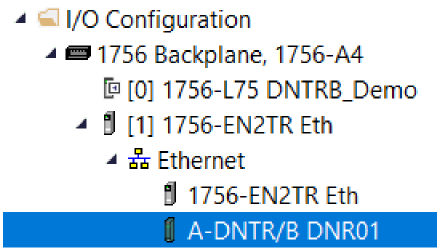


Figure 3.79 – Logix IO tree

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

B. LOGIX MAPPING

Slate will generate the required UDTs and Routines (based on the Internal Map) to map the required DeviceNet input and output data. The user will need to select the *Recommend* button in the Internal Mapping to auto populate the Internal Mapping which can then be used to generate the L5X file for Logix mapping, routines, and UDTs

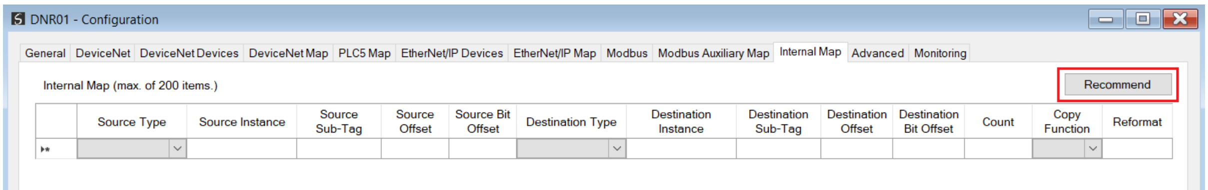


Figure 3.80 – Internal Mapping Recommend

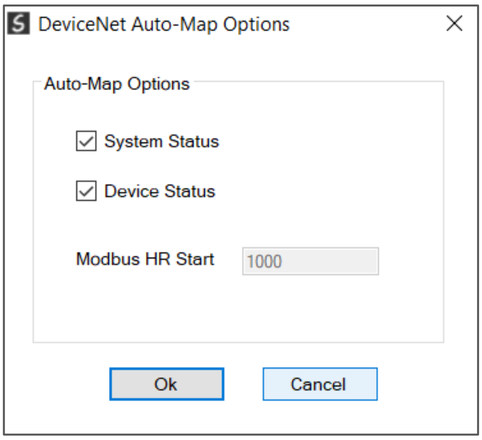


Figure 3.81 – Auto Map Options popup

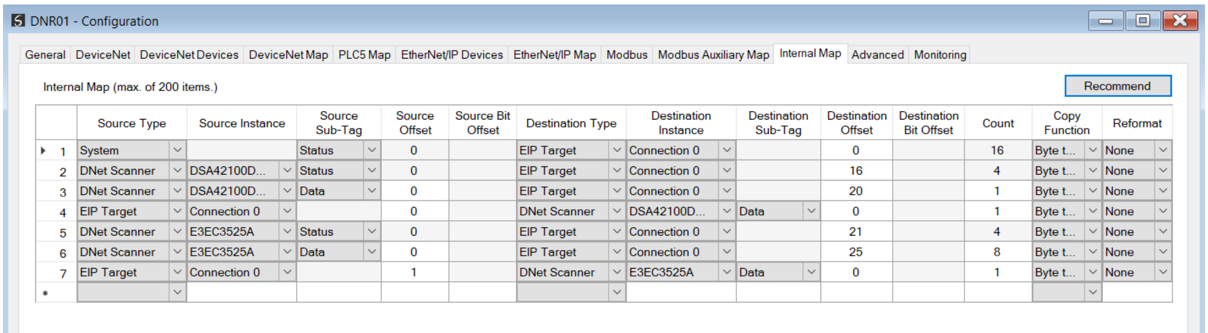


Figure 3.82 – Internal Mapping Auto Populated

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

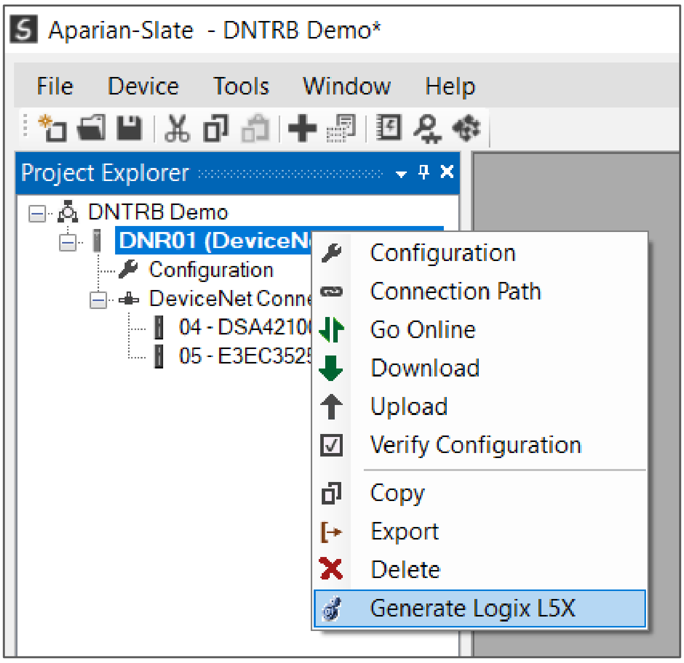


Figure 3.83 – Selecting Generate Logix L5X

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

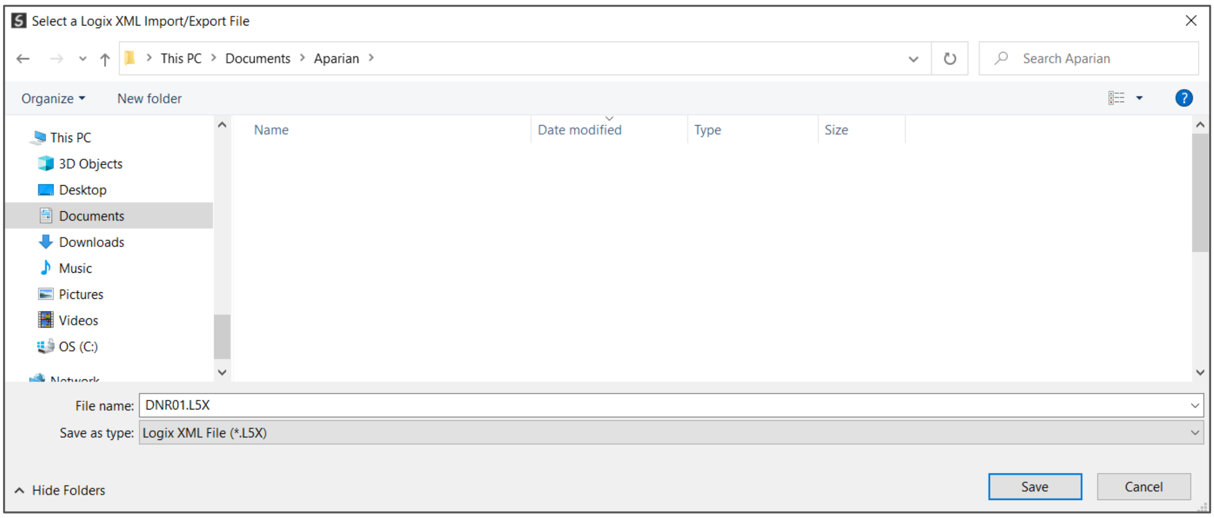


Figure 3.84 – Selecting the Logix L5X file name

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***.

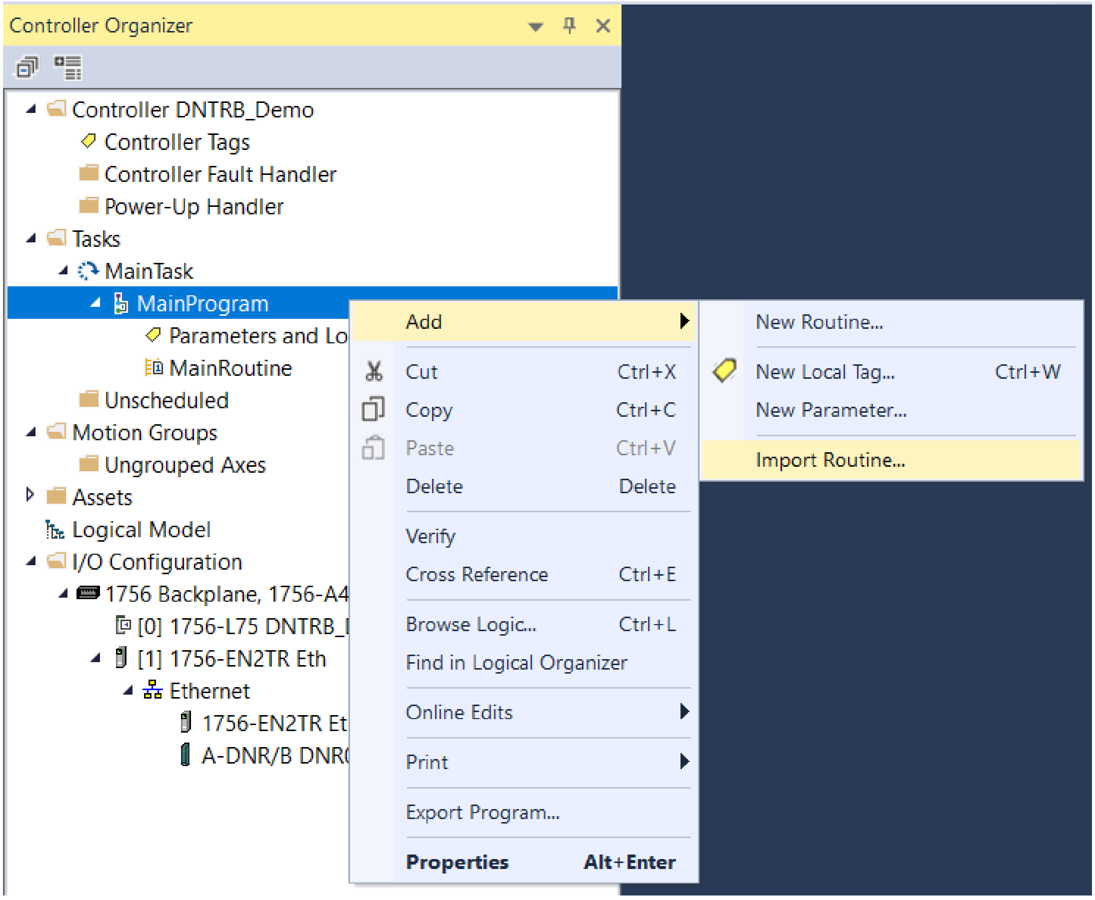


Figure 3.85 – Importing the L5X file into Studio 5000

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 3.86 – Calling the mapping routine

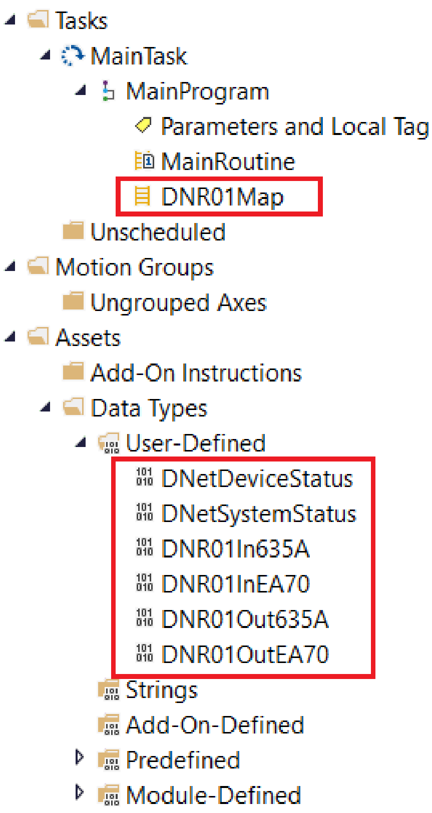


Figure 3.87 – Imported Logix Objects

A number of DeviceNet Router/B specific (UDT) tags are created.

The System tag displays the status and diagnostics information of the DeviceNet Router/B.

DNR01In	{...}	{...}		DNR01In635A
DNR01In.SystemStatus	{...}	{...}		DNetSystemStatus
DNR01In.SystemStatus.ConfigValid	0		Decimal	BOOL
DNR01In.SystemStatus.EIPOriginatorCommsOk	0		Decimal	BOOL
DNR01In.SystemStatus.ModbusOnline	0		Decimal	BOOL
DNR01In.SystemStatus.DnetOriginatorCommsOk	0		Decimal	BOOL
DNR01In.SystemStatus.EIPOwned	0		Decimal	BOOL
DNR01In.SystemStatus.DNetOwned	0		Decimal	BOOL
DNR01In.SystemStatus.PowerMainConnector	0		Decimal	BOOL
DNR01In.SystemStatus.PowerCANConnector	0		Decimal	BOOL
DNR01In.SystemStatus.DuplicateNodeNumber	0		Decimal	BOOL
DNR01In.SystemStatus.NTPOk	0		Decimal	BOOL
DNR01In.SystemStatus.ConfigCRC	16#0000		Hex	INT
DNR01In.SystemStatus.ActualBAUD	0		Decimal	SINT
DNR01In.SystemStatus.ActualNode	0		Decimal	SINT
DNR01In.SystemStatus.DNetTargetPollActive	0		Decimal	BOOL
DNR01In.SystemStatus.DNetTargetNullPoll	0		Decimal	BOOL
DNR01In.SystemStatus.DNetTargetCOSActive	0		Decimal	BOOL

Figure 3.88 – System Status tag

There is also a tag created for each configured DeviceNet IO device using cyclic communication. The structure of which comprises the following:

- Input Status – Status related to DeviceNet IO device
- Input Data – As specified in the DeviceNet IO configuration and internal mapping
- Output Data – As specified in the DeviceNet IO and internal mapping

DNR01In	{...}	{...}		DNR01In635A
DNR01In.SystemStatus	{...}	{...}		DNetSystemStatus
DNR01In.DSA42100DNY42R50Status	{...}	{...}		DNetDeviceStatus
DNR01In.DSA42100DNY42R50Status.Online	0		Decimal	BOOL
DNR01In.DSA42100DNY42R50Status.DataExchangeActive	0		Decimal	BOOL
DNR01In.DSA42100DNY42R50Status.DeviceMismatch	0		Decimal	BOOL
DNR01In.DSA42100DNY42R50Status.Node	0		Decimal	SINT
DNR01In.DSA42100DNY42R50Data	{...}	{...}	Decimal	SINT[1]
DNR01In.DSA42100DNY42R50Data[0]	0		Decimal	SINT
DNR01In.E3EC3525AStatus	{...}	{...}		DNetDeviceStatus
DNR01In.E3EC3525AData	{...}	{...}	Decimal	SINT[8]
DNR01Out	{...}	{...}		DNR01Out635A
DNR01Out.DSA42100DNY42R50Data	{...}	{...}	Decimal	SINT[1]
DNR01Out.DSA42100DNY42R50Data[0]	0		Decimal	SINT

Figure 3.89 – DeviceNet IO Device-Specific tags

NOTE: The Logix tags for the data being produced and consumed by the DeviceNet IO device will be SINT arrays with a size matching the connection sizes when the Decorate Data option has been unchecked.
Device specific UDTs will be created when this option has been checked.
With some devices, it may be useful to copy this data into application specific, or device specific, UDTs for better context.

3.6.2.2. INTERNAL DATA SPACE MAPPING

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

A. IDS COPY – ETHERNET/IP TARGET SOURCE

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

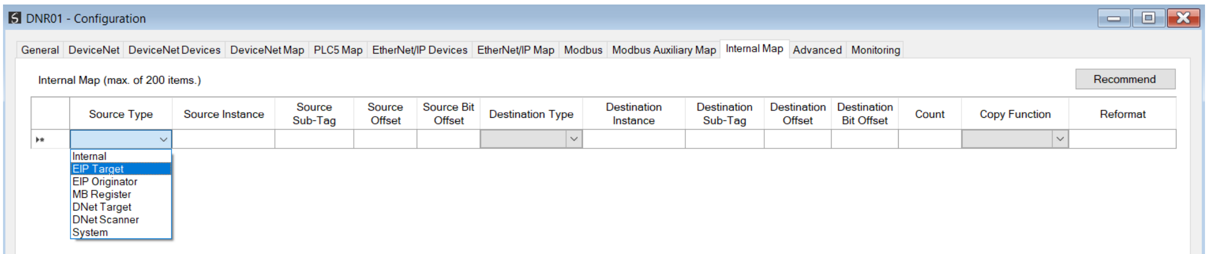


Figure 3.90 – IDS Copy – EtherNet/IP Target Source Type

The source instance will be the connection number, which can be connection 0 to 3, based on the number of connections configured. The Source Offset is the offset in the EtherNet/IP output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

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

B. *IDS COPY – ETHERNET/IP TARGET DESTINATION*

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

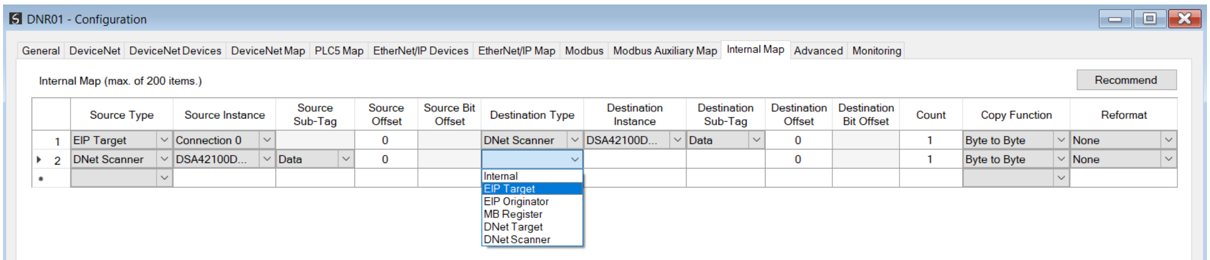


Figure 3.91 – IDS Copy – EtherNet/IP Target Destination Type

The destination instance will be the connection number, which can be connection 0 to 3, based on the number of connections configured. The Destination Offset is the offset of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.6.3. Modbus Server

The DeviceNet Router/B can operate as a Modbus Server for Modbus TCP, RTU232, and RTU485. A Modbus Client can read and write to the full Modbus Register range in the DeviceNet Router/B. The DeviceNet Router/B can operate as a Modbus Server for Modbus TCP, Modbus RTU232, and Modbus RTU485 simultaneously. The user will need to configure the relevant Modbus Parameters as shown below.

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

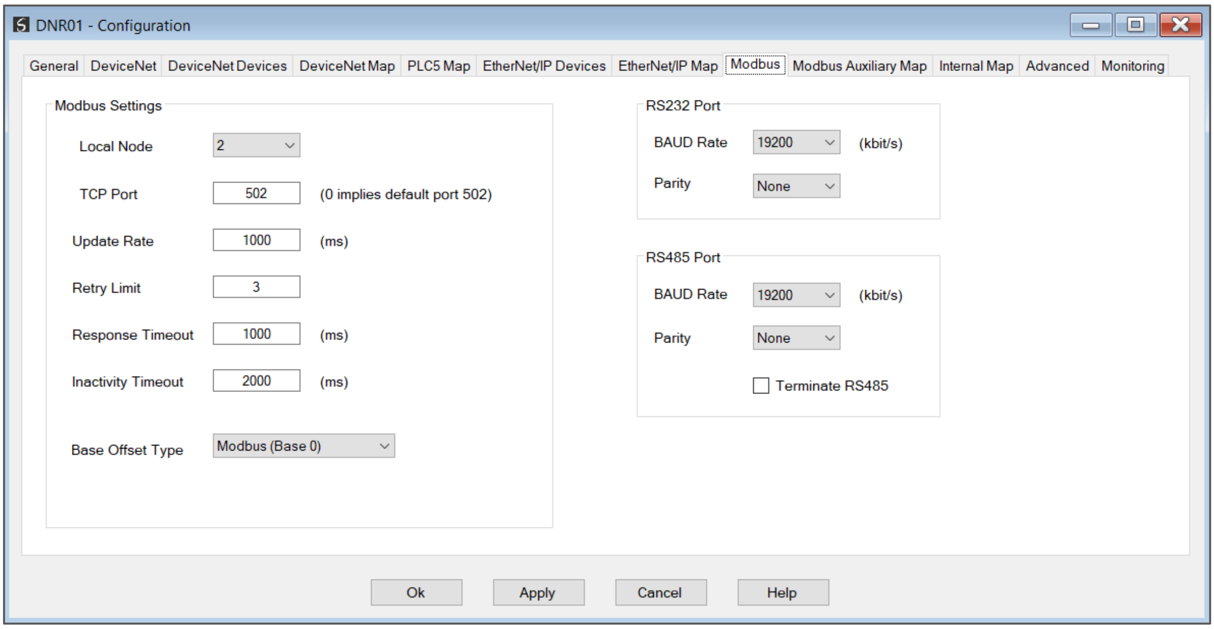


Figure 3.92 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node address assigned to the DeviceNet Router/B.
TCP Port	The TCP port to be used for the Modbus communication. By default, the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between Client requests to the target Modbus device. **(Modbus Client mode only)**
Retry Limit	The number of successive Modbus request retries before the request is set to have failed. **(Modbus Client mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **(Modbus Client mode only)**
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the DeviceNet Router/B indicates that the connection to the Modbus Client is no longer active. **(Modbus Server 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.
RS232 Port	
BAUD Rate	The RS232 serial port's BAUD rate. (Modbus RTU232)
Parity	The RS232 serial port's Parity configuration. (Modbus RTU232)
RS485 Port	
BAUD Rate	The RS485 serial port's BAUD rate. (Modbus RTU485)
Parity	The RS485 serial port's Parity configuration. (Modbus RTU485)
Terminate RS485	Enables the on-board 125Ω RS485 terminating resistor.

Table 3.10 – Modbus parameters

The Modbus Node Number will need to be configured in the parameters above to allow a Modbus Client to access the DeviceNet Router/B as a Modbus Server device.

3.6.3.1. INTERNAL DATA SPACE MAPPING

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

NOTE: The user can select the *Recommend* button in the Internal Map to auto map the DeviceNet parameters to recommended Modbus Registers.

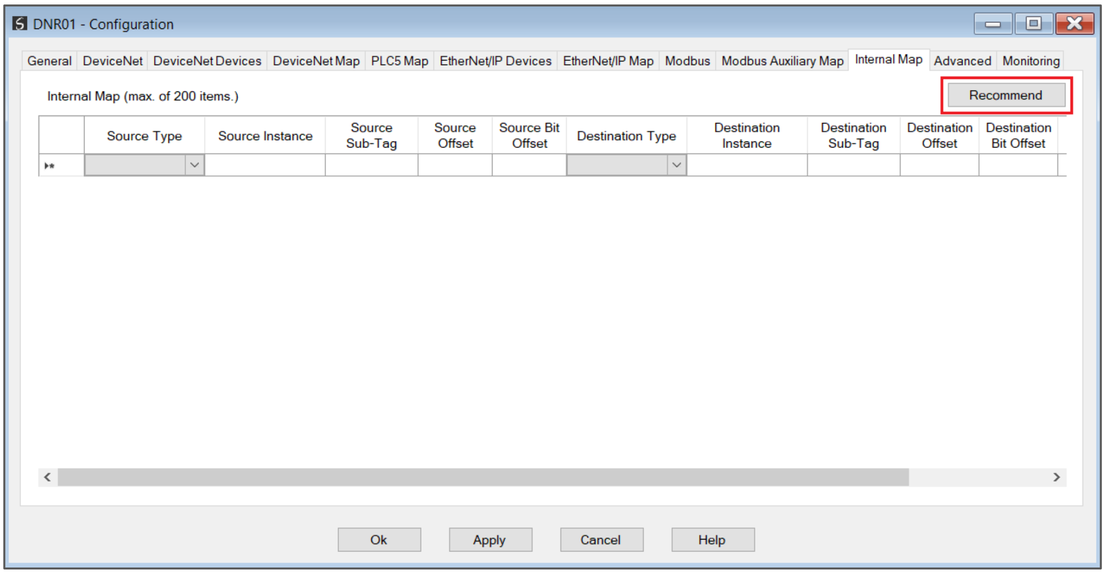


Figure 3.93 – Modbus Server – Internal Mapping Recommend

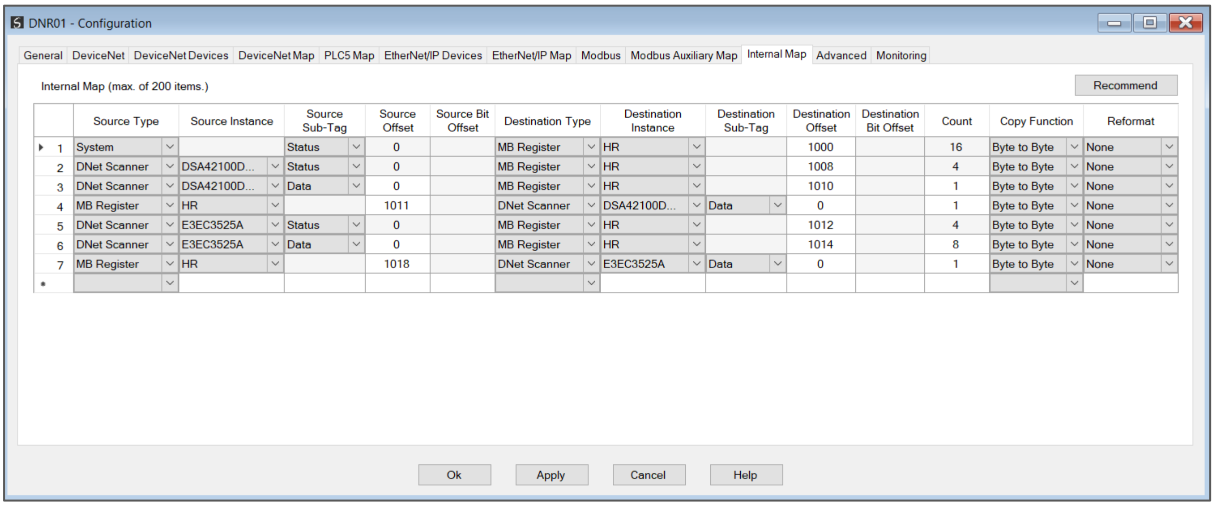


Figure 3.94 – Modbus Server – Internal Mapping Updated

A. IDS COPY – MODBUS SOURCE

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

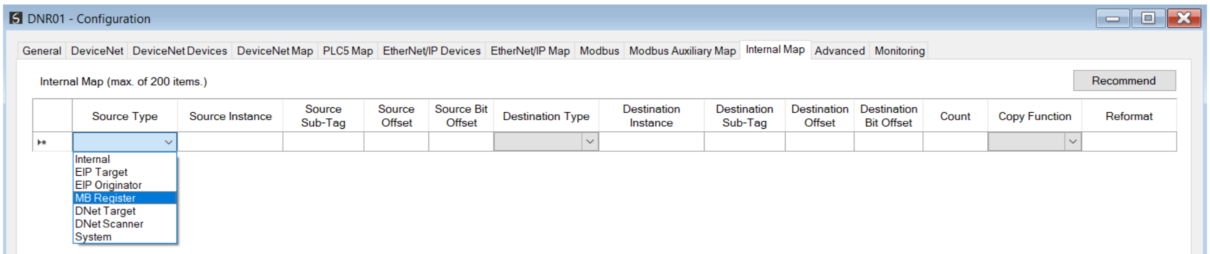


Figure 3.95 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

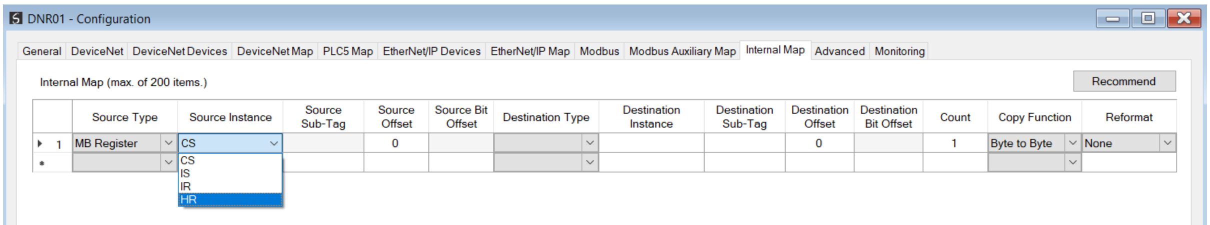


Figure 3.96 – IDS Copy - Modbus Source Instance

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

B. IDS COPY – MODBUS DESTINATION

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

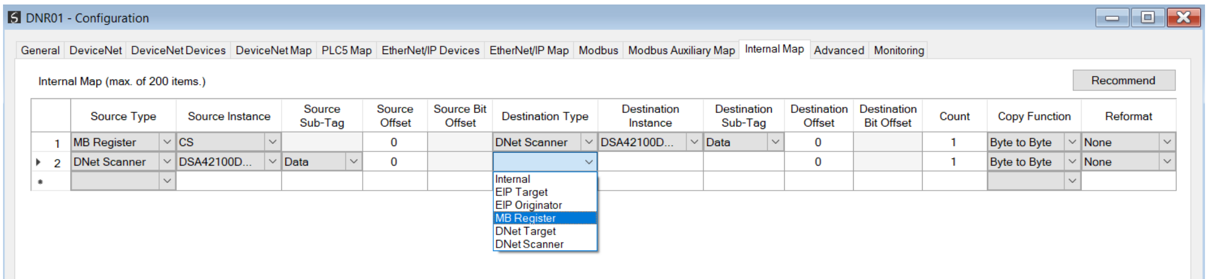


Figure 3.97 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

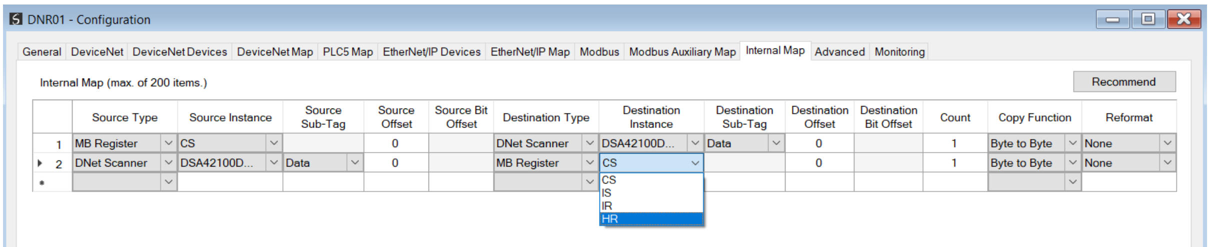


Figure 3.98 – IDS Copy - Modbus Destination Instance

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

3.6.4. Modbus Client

The DeviceNet Router/B can operate as a Modbus Client for Modbus TCP, RTU232, and RTU485. The user will need to configure the relevant Modbus Parameters as shown below followed by the configuration of the Modbus Auxiliary Map. This map will allow the user to configure various read and write functions to external Modbus Registers, to and from the internal Modbus registers.

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

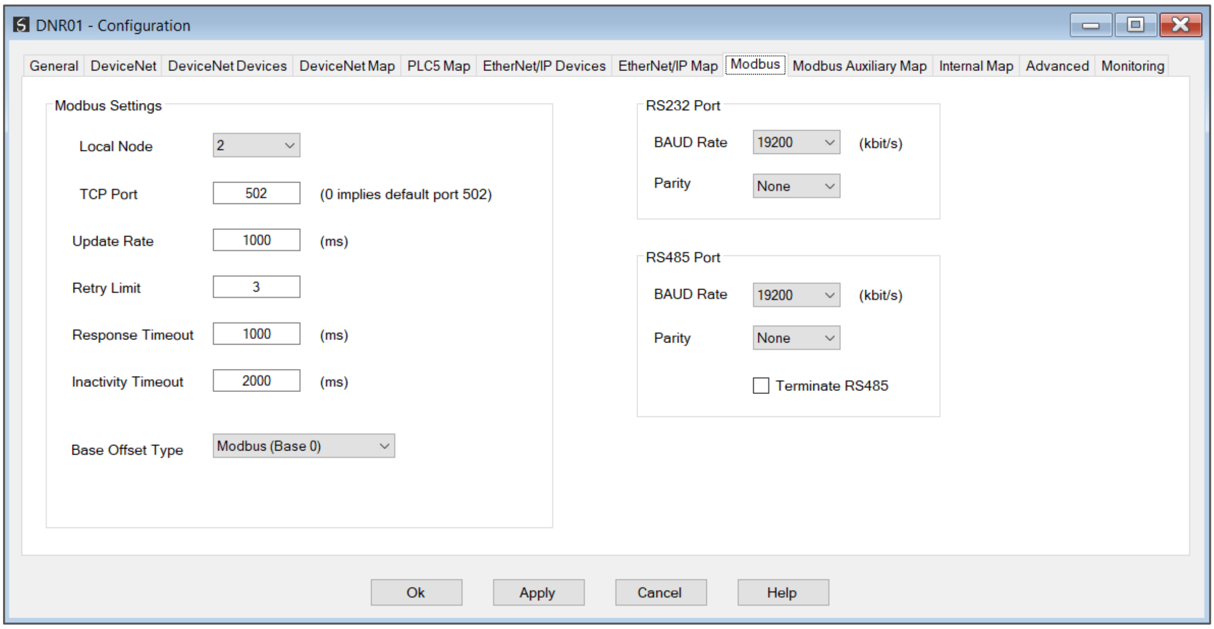


Figure 3.99 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node address assigned to the DeviceNet Router/B.
TCP Port	The TCP port to be used for the Modbus communication. By default the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between Client requests to the target Modbus device. **(Modbus Client mode only)**
Retry Limit	The number of successive Modbus request retries before the request is set to have failed. **(Modbus Client mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **(Modbus Client mode only)**
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the DeviceNet Router/B indicates that the connection to the Modbus Client is no longer active. **(Modbus Server 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.

RS232 Port	
BAUD Rate	The RS232 serial port's BAUD rate . (Modbus RTU232)
Parity	The RS232 serial port's Parity configuration. (Modbus RTU232)
RS485 Port	
BAUD Rate	The RS485 serial port's BAUD rate . (Modbus RTU485)
Parity	The RS485 serial port's Parity configuration. (Modbus RTU485)
Terminate RS485	Enables the on-board 125Ω RS485 terminating resistor.

Table 3.11 – Modbus parameters

3.6.4.1. MODBUS AUXILIARY MAP

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

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

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

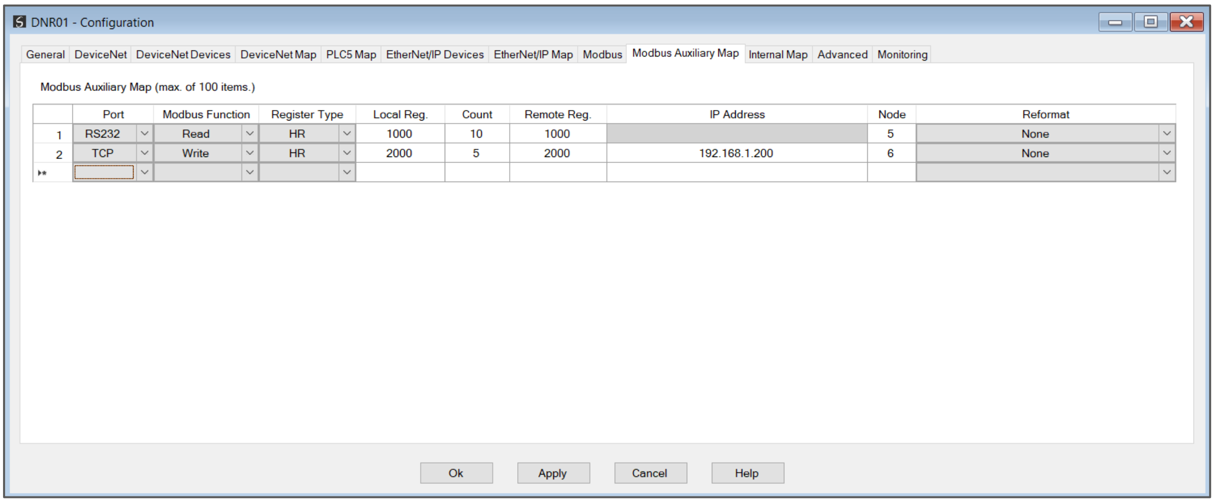


Figure 3.100 – Modbus Auxiliary Map Configuration

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Port	The external port to be used: **TCP** – Modbus TCP (Ethernet) **RS232** – Modbus RTU232 **RS485** – Modbus RTU485
Modbus Function	This is the Modbus function that is sent to the Modbus Server. **Read** – Read a Modbus Register (e.g. HR, IR, CS, or IS) from a Modbus Server. **Write** – Write a Modbus Register (e.g. HR or CS) to a Modbus Server.
Register Type	Modbus Register Type: **CS** – Coil Status **IS** – Input Status **IR** – Input Register **HR** – Holding Register
Local Reg.	The local (internal) DeviceNet Router/B Modbus register address.
Count	The number of Modbus elements to read or write.
Remote Reg.	The remote Server Modbus address register.
IP Address	The IP address of the remote Modbus Server.
Node	The Modbus Node address of the remote Modbus Server.
Reformat	Used to specify how the data is formatted before writing to, or after reading from, the Modbus Server. **None** – No reformatting applied. (AA BB CC DD). **BB AA** – 16bit Byte swap **BB AA DD CC** – 32bit Byte Pair Swap **CC DD AA BB** – Word Swap **DD CC BB AA** – Word and Byte Pair Swap

Table 3.12 – Modbus Auxiliary Map parameters

3.6.4.2. INTERNAL DATA SPACE MAPPING

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

NOTE: The user can select the ***Recommend*** button in the Internal Map to auto map the DeviceNet parameters to recommended Modbus Registers.

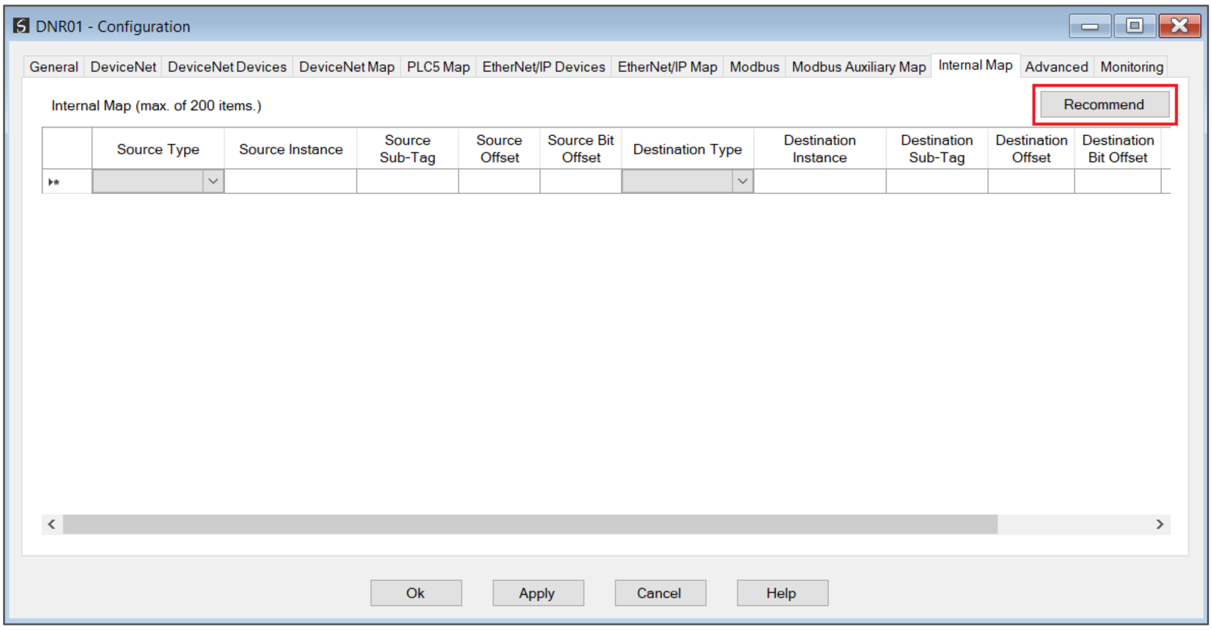
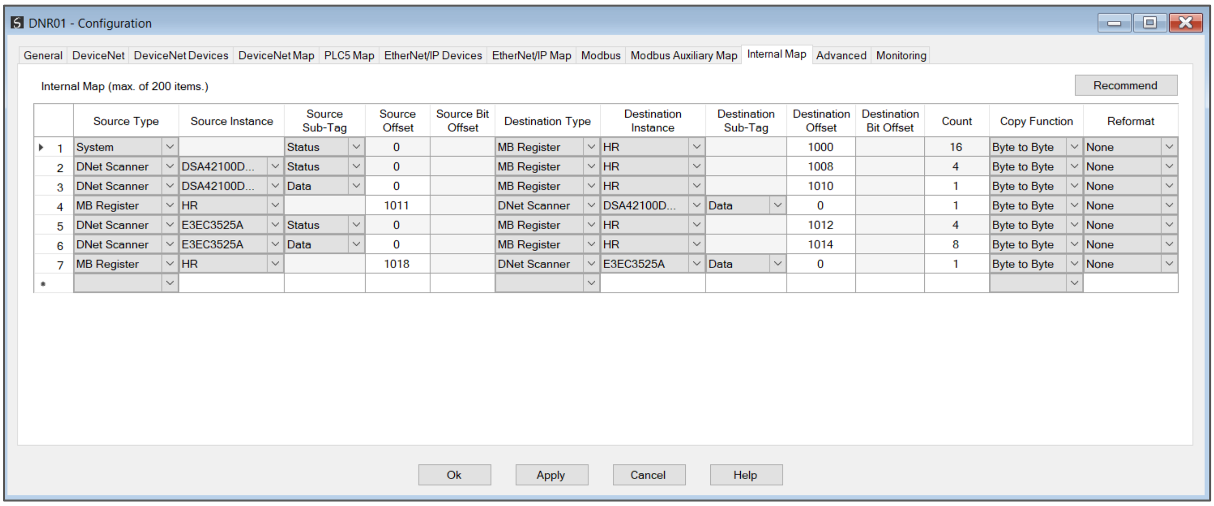


Figure 3.101 – Modbus Server – Internal Mapping Recommend



	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0		MB Register	HR		1000		16	Byte to Byte	None
2	DNet Scanner	DSA42100D...	Status	0		MB Register	HR		1008		4	Byte to Byte	None
3	DNet Scanner	DSA42100D...	Data	0		MB Register	HR		1010		1	Byte to Byte	None
4	MB Register	HR		1011		DNet Scanner	DSA42100D...	Data	0		1	Byte to Byte	None
5	DNet Scanner	E3EC3525A	Status	0		MB Register	HR		1012		4	Byte to Byte	None
6	DNet Scanner	E3EC3525A	Data	0		MB Register	HR		1014		8	Byte to Byte	None
7	MB Register	HR		1018		DNet Scanner	E3EC3525A	Data	0		1	Byte to Byte	None

Figure 3.102 – Modbus Server – Internal Mapping Updated

A. IDS COPY – MODBUS SOURCE

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

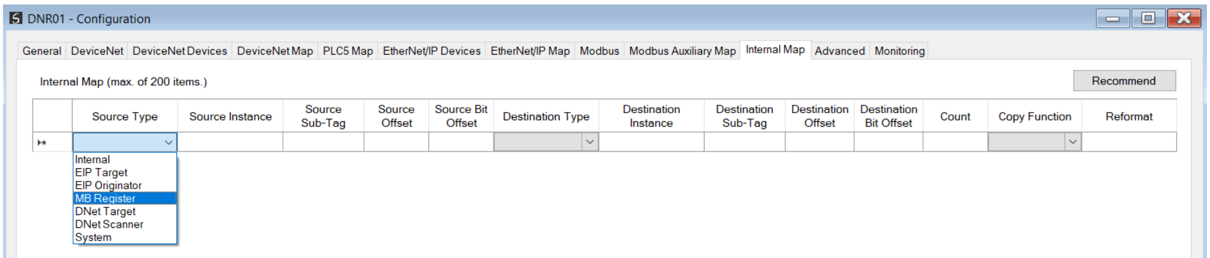


Figure 3.103 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

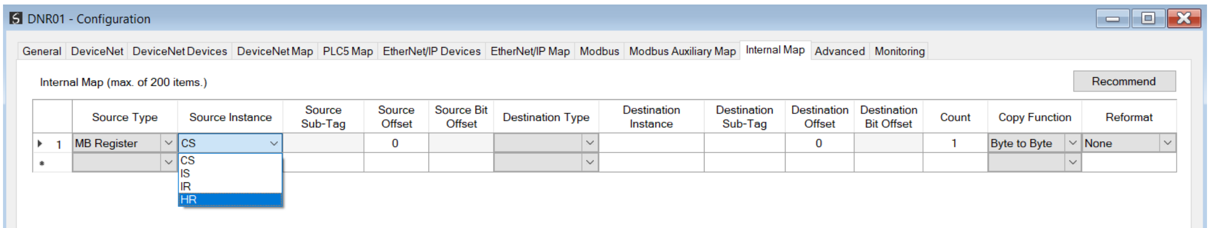


Figure 3.104 – IDS Copy - Modbus Source Instance

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

B. IDS COPY – MODBUS DESTINATION

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

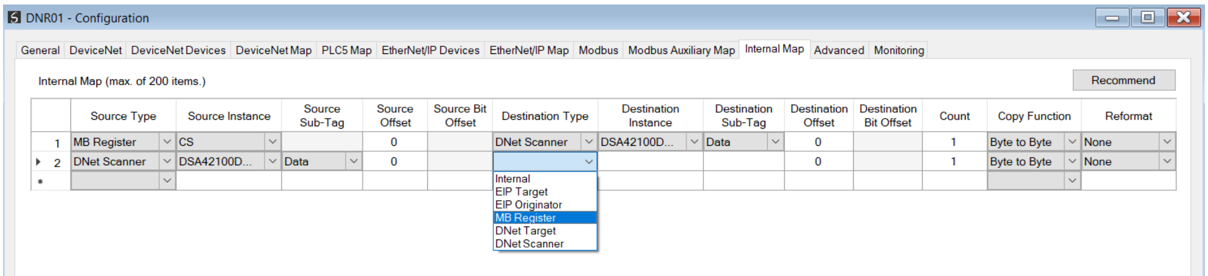


Figure 3.105 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

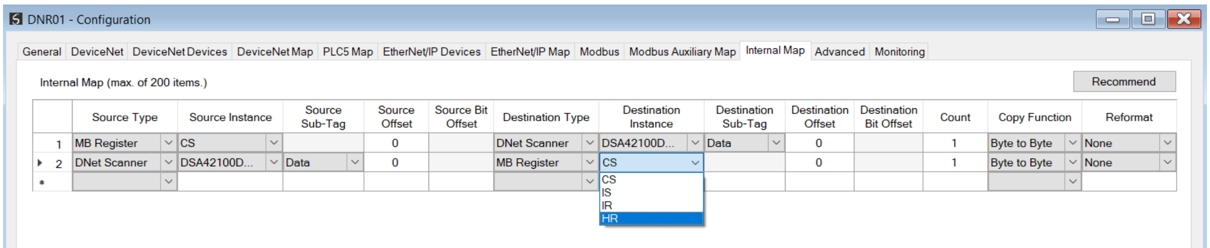


Figure 3.106 – IDS Copy - Modbus Destination Instance

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

3.6.5. EtherNet/IP Originator

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

3.6.5.1. EtherNet/IP Class 1 Device Connections

The DeviceNet Router/B can establish up to 10 cyclic Class 1 EtherNet/IP connections to EtherNet/IP devices. This can be done by either manually entering the connection data into the Connection Parameter window, or by importing the configuration from one or more of the following sources:

- Online Logix Controller
- Logix Controller L5X
- EDS File
- Connection Library

A. Manual Configuration

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

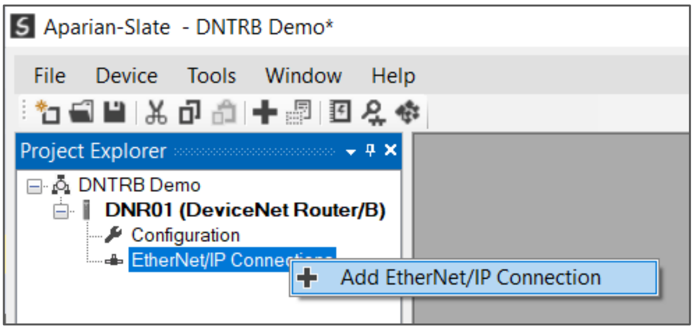


Figure 3.107 – Adding EtherNet/IP Class 1 Connection

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

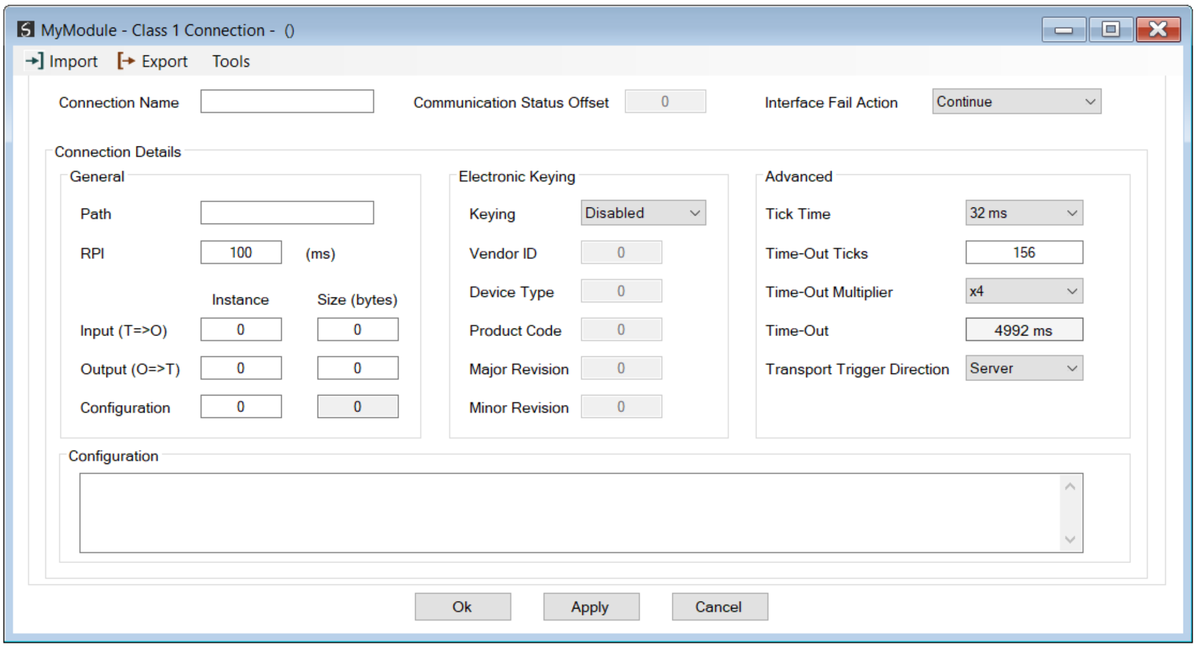


Figure 3.108 – EtherNet/IP Class 1 Connection Parameters

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

Parameter	Description
Connection Name	The instance name given to the Class 1 Connection.
Interface Fail Action	When the DeviceNet communication has failed, the EtherNet/IP IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the EtherNet/IP device to go into a pre-determined state when the communication to the controller (i.e., connection originator on DeviceNet) is lost.

General	
Path	The path to the target device. If the device is an Ethernet device, then this will just be the IP address of the module. If the device is, for example, a module in a backplane or via an adapter, then the user will need to enter the IP address of the bridge or adapter followed by the backplane port (for example 1) and the slot number of the device. Each item is separated by a comma. As an example, to connect to an Allen-Bradley Flex module (via the Flex Adapter at IP address 192.168.1.100) that is in slot 2 of the Flex backplane, the user will need to enter the following path: 192.168.1.100,1,2 (IP address, port (backplane), slot).
RPI	The requested packet interval (RPI) is the rate in milliseconds at which the data will be sent from the originator to the target and vice versa.
Input (T=>O) – Instance	The instance of the input assembly.
Input (T=>O) – Size (bytes)	The size in bytes of the input assembly.
Output (O=>T) – Instance	The instance of the output assembly.
Output (O=>T) – Size (bytes)	The size in bytes of the output assembly.
Configuration – Instance	The instance of the configuration assembly.
Configuration – Size (bytes)	The size in bytes of the configuration assembly. **NOTE:** This is a read-only value and will be equal to the number of bytes entered into the configuration window below.
Electronic Keying	
Keying	Electronic Keying can be used to ensure that the target device is the correct device type. **Disabled** Keying is not enabled, and no key information will be sent in the connection establishment. **Compatible** Keying has been enabled with compatibility enabled. This will allow devices with older firmware to also establish a connection. **Exact** Keying has been enabled and the exact device with specific firmware revision will allow the establishment of the connection.
Vendor ID	The Vendor ID of the target device.
Device Type	The Device Type of the target device.
Product Code	The Product Code of the target device.
Major Revision	The Major Revision of the target device.
Minor Revision	The Minor Revision of the target device.

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

Table 3.13 – EtherNet/IP Class 1 Connection Parameters

B. IMPORT FROM ONLINE CONTROLLER

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

PREPARATION

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

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

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

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

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

IMPORT CONNECTION PARAMETERS

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

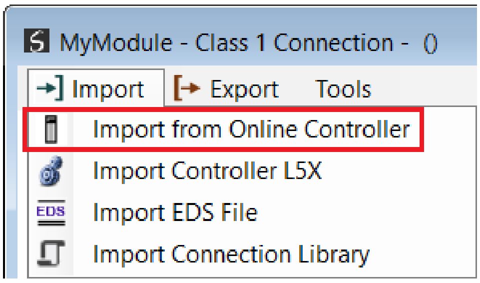


Figure 3.109 – Import from Online Controller

The Import Connection Parameters form will open.

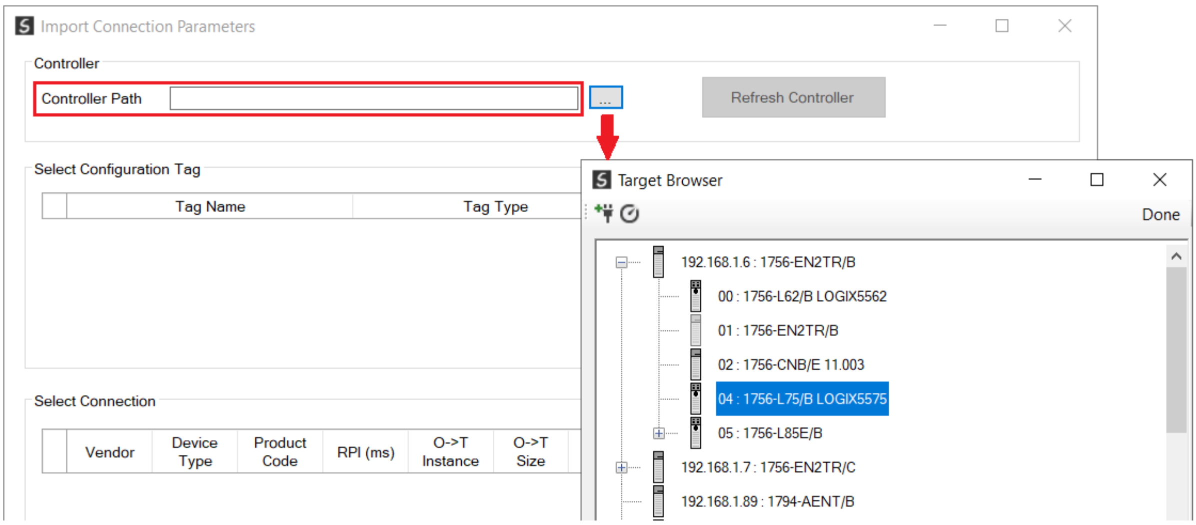


Figure 3.110 – Import Connection Parameters – Controller Path

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.

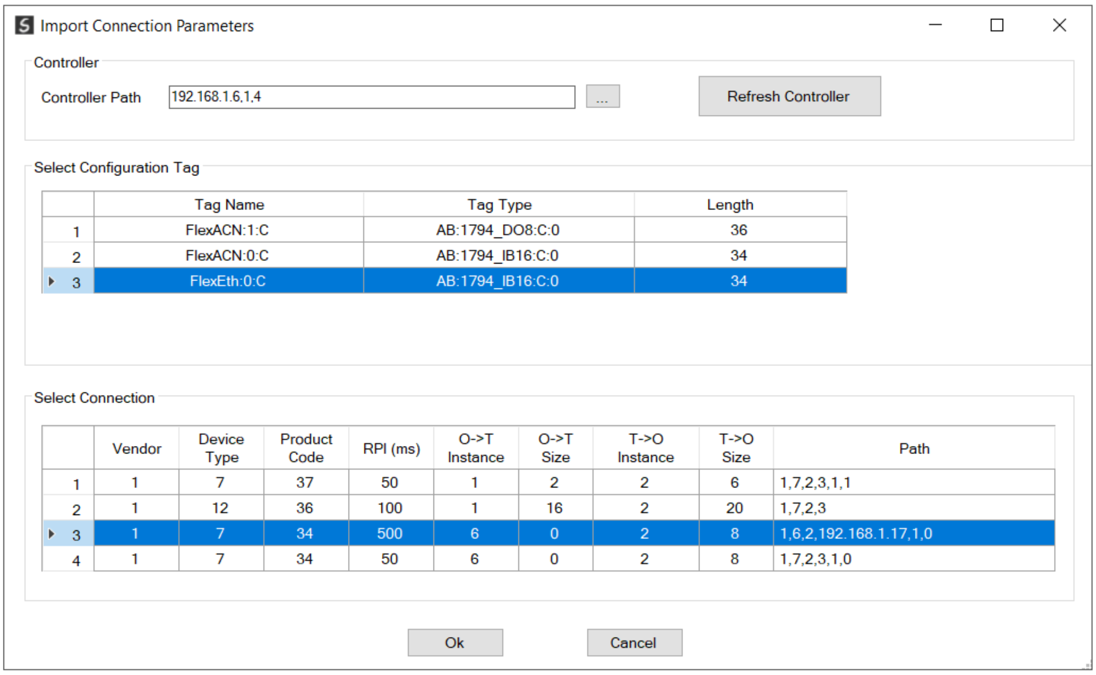


Figure 3.111 – Import Connection Parameters – Select Connection

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

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

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

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

C. IMPORT FROM CONTROLLER L5X FILE

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

PREPARATION

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

1. In Studio5000 create a new project and add the required EtherNet/IP device in the IO tree. If the device's profile supports configuration, then configure the device as required.
2. Save the Studio5000 project as an L5X file.

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

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

IMPORT L5X FILE

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

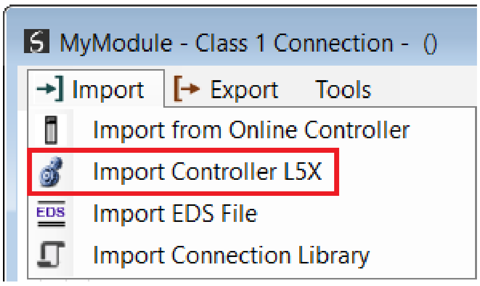


Figure 3.112 – Import from Controller L5X

The Import Connection Parameters form will open.

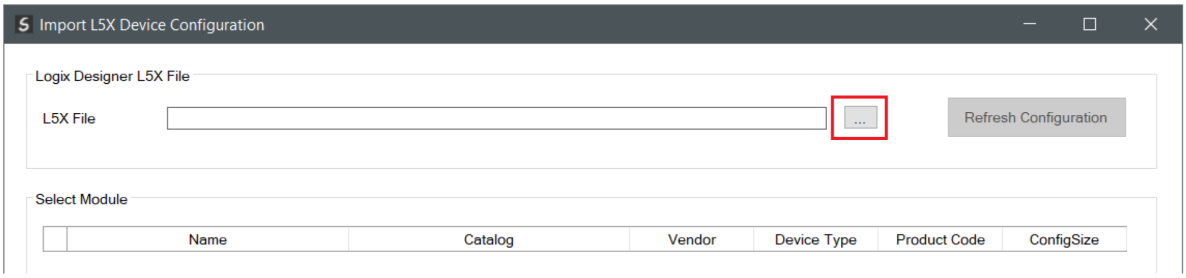


Figure 3.113 – Import L5X Device Configuration – Select L5X

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.

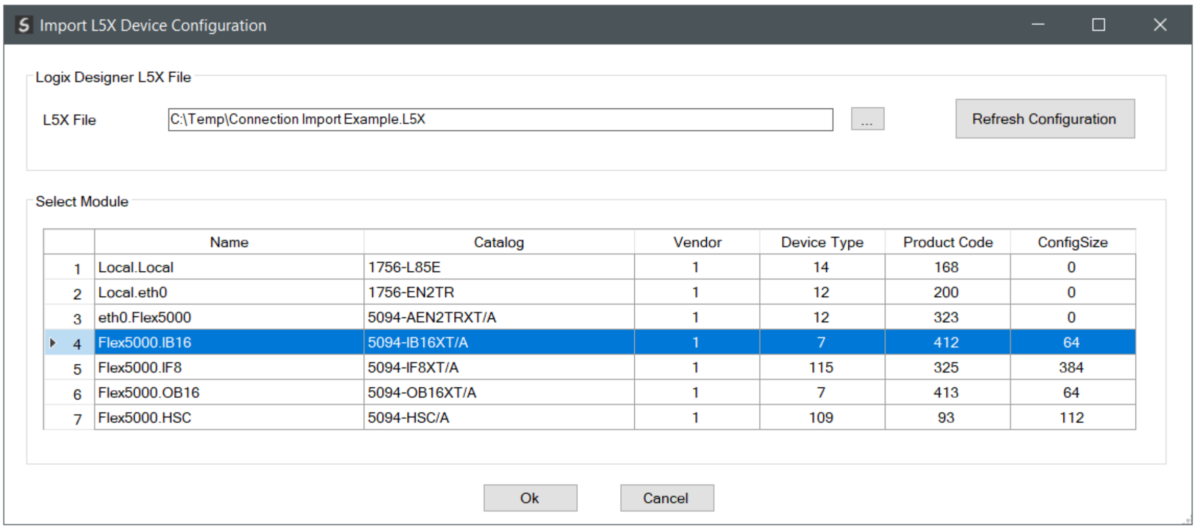


Figure 3.114 – Import L5X Device Configuration

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

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

D. IMPORT EDS FILE

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

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

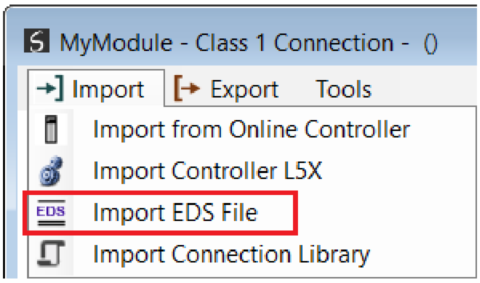


Figure 3.115 – Import EDS File

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

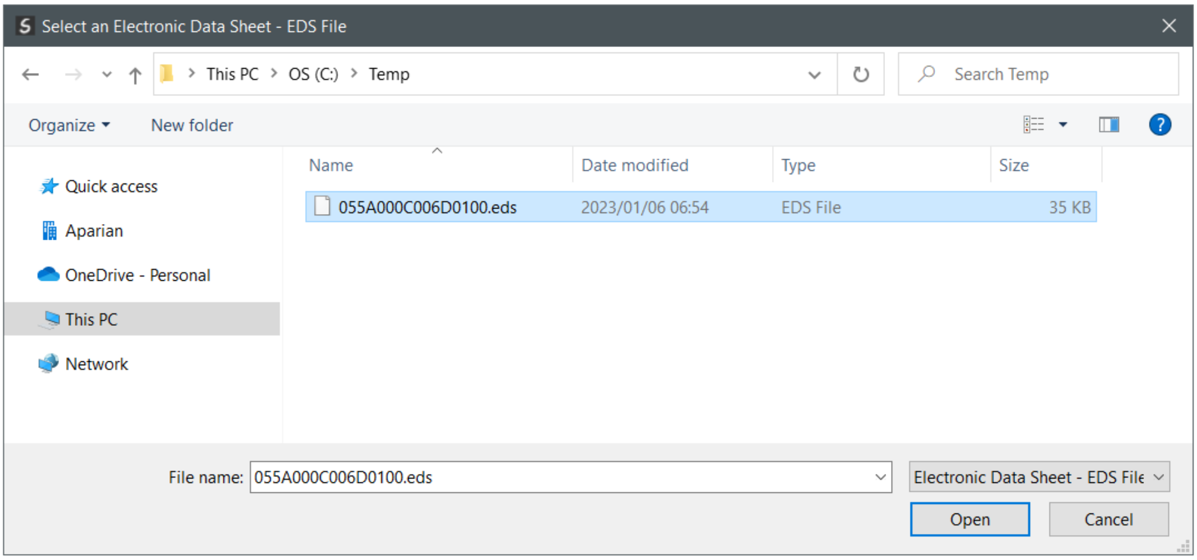


Figure 3.116 – Browse to EDS File

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.

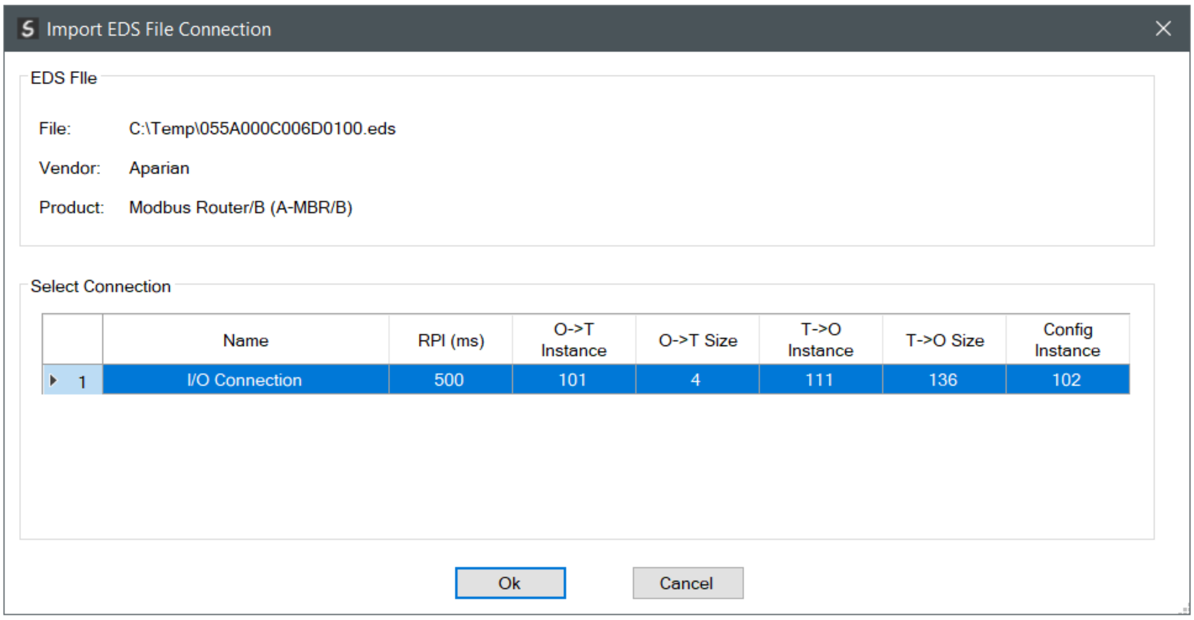


Figure 3.117 – Select Connection

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

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

E. IMPORT CONNECTION LIBRARY

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

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

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

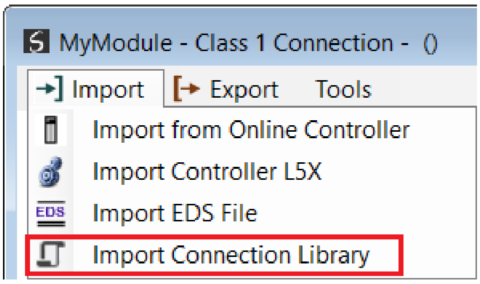


Figure 3.118 – Import Connection Library File

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

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

EXPORT LIBRARY FILE

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

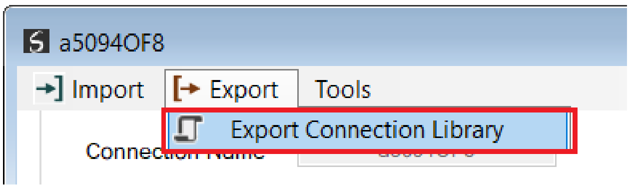


Figure 3.119 – Export Connection Library File

3.6.5.2. ETHERNET/IP EXPLICIT MESSAGE DEVICE CONNECTIONS

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

A. ETHERNET/IP DEVICES

This tab is enabled when the Primary Interface is set to *EtherNet/IP Originator*.

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

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

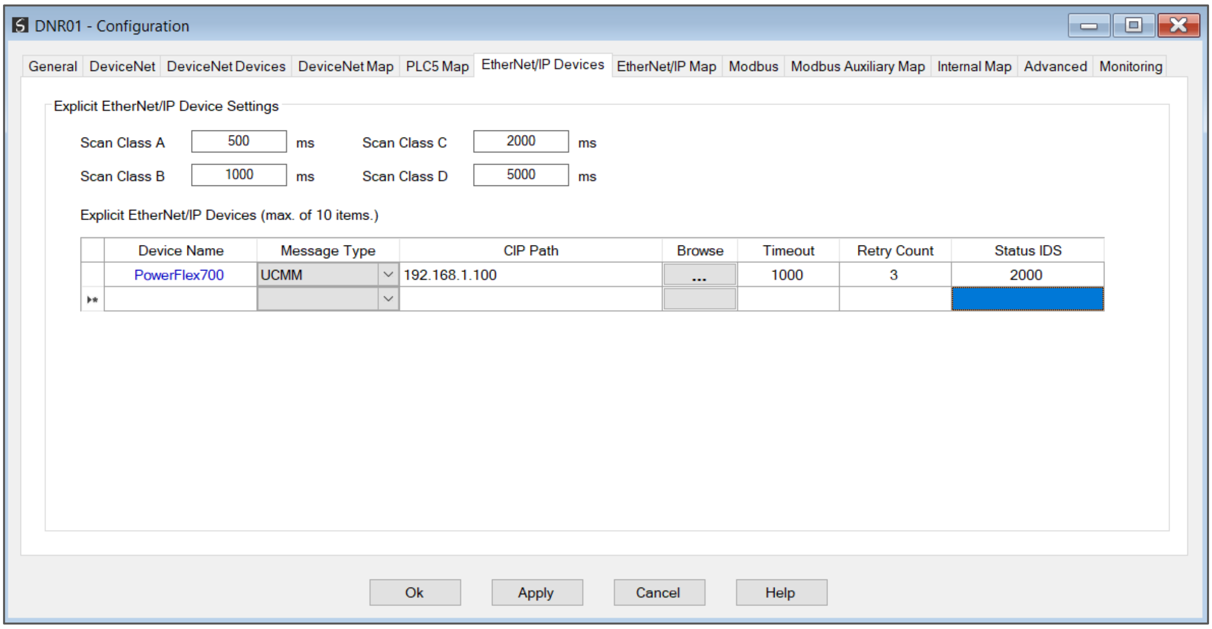


Figure 3.120 – EtherNet/IP Devices - Configuration

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

Table 3.14 – EtherNet/IP Devices configuration parameters

B. ETHERNET/IP MAP

This tab is enabled when the Primary Interface is set to *EtherNet/IP Originator*.

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

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

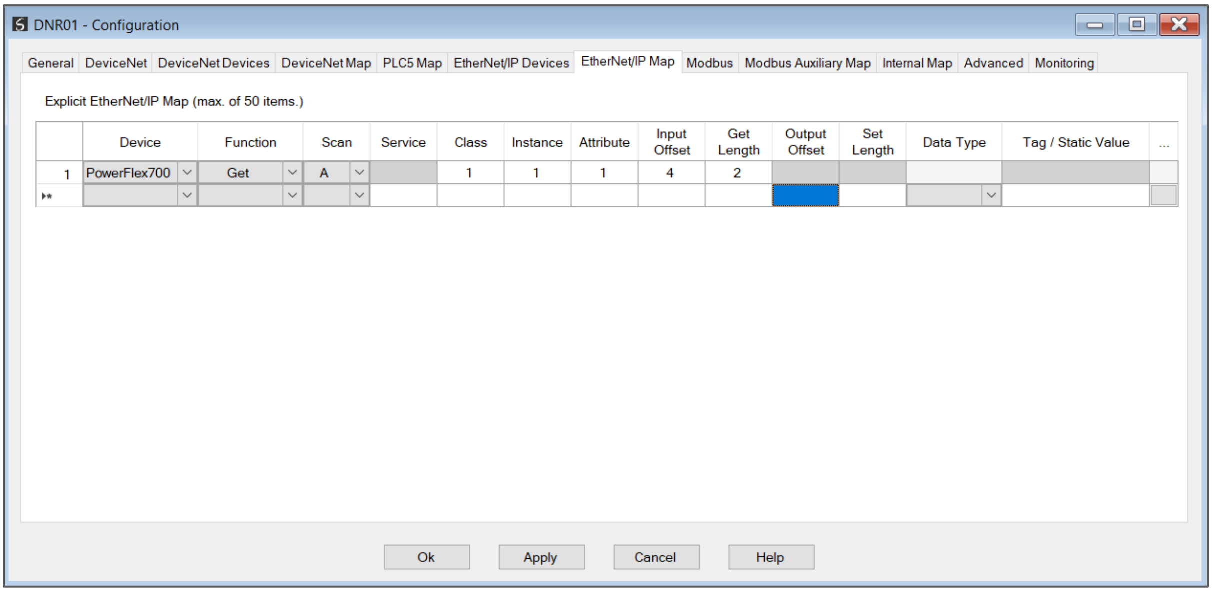


Figure 3.121 – EtherNet/IP Map configuration

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 Internal Data Space at the ***Input Offset*** location configured in this tab. **Set** The module will write data to the target EtherNet/IP device by using the Set Single Attribute CIP function. The data to be written will be retrieved from the Internal Data Space at the ***Output Offset*** location configured in this tab. **Set Static** Similar to the Set function above, but the data to be written will be fixed (equal to the *Static Value*) parameter in this configuration window. This function will typically be used with the single (**S**) Scan class which means the DeviceNet Router/B module can be setup to write the fixed value only once when the target device communication has been established. **Custom**

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

	This will only be available for ***Set*** and ***Custom*** functions.
Data Type	The data type of the Static Value. This will only be available for ***Set Static*** function.
Tag / Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.15 – EtherNet/IP Map configuration parameters

3.6.5.3. INTERNAL DATA SPACE MAPPING

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

A. IDS COPY – ETHERNET/IP ORIGINATOR SOURCE

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

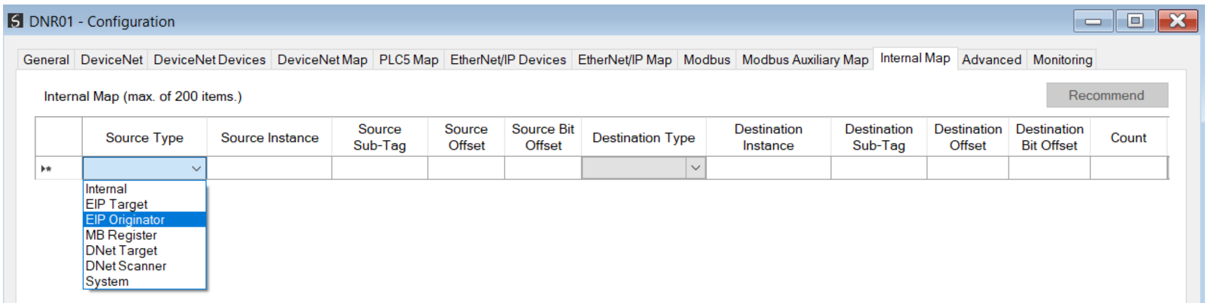


Figure 3.122 – IDS Copy – EtherNet/IP Originator Source Type

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

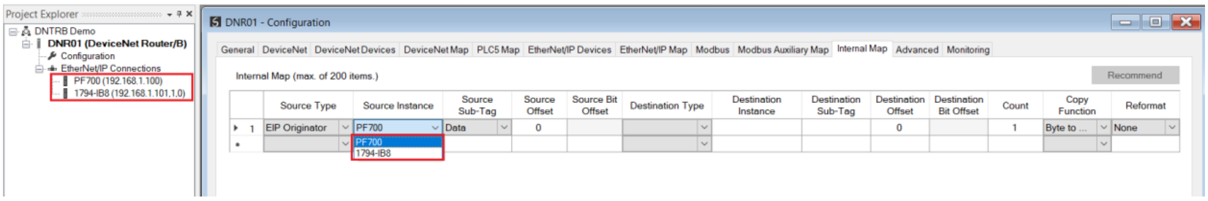


Figure 3.123 – IDS Copy – EtherNet/IP Originator Source Instance

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

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

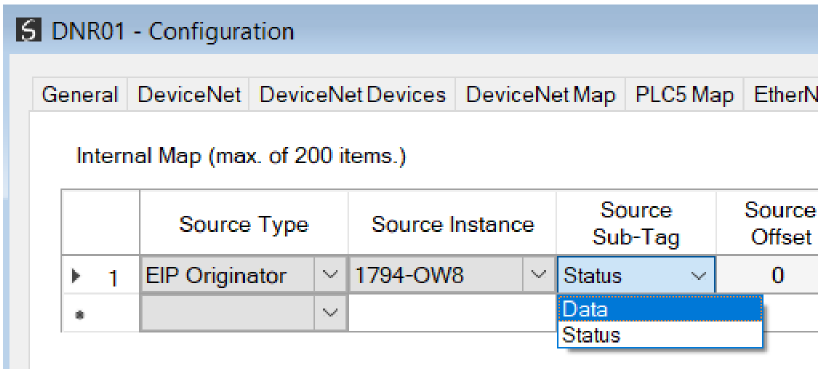


Figure 3.124 – IDS Copy – EtherNet/IP Originator Status

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

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

Table 3.16 – EtherNet/IP Originator Connection Status

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

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

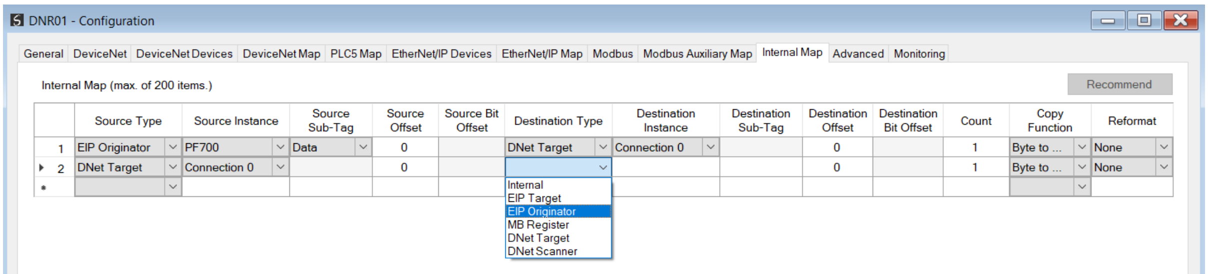


Figure 3.125 – IDS Copy – EtherNet/IP Originator Destination Type

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

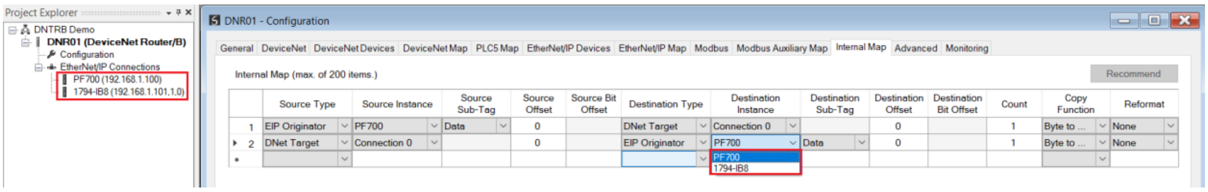


Figure 3.126 – IDS Copy – EtherNet/IP Originator Destination Instance

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

3.7. INTERNAL DATA SPACE MAP

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

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 3.17 – Internal Map Copy functions

The data in the destination source can also be reformatted. The reformat option provides five different reformat options.

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

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

Table 3.18 – Internal Map Reformat Options

3.7.1. COPY FROM

One of seven sources can be selected to copy from:
Internal, EIP Target, EIP Originator, DNet Target, DNet Scanner, and System.

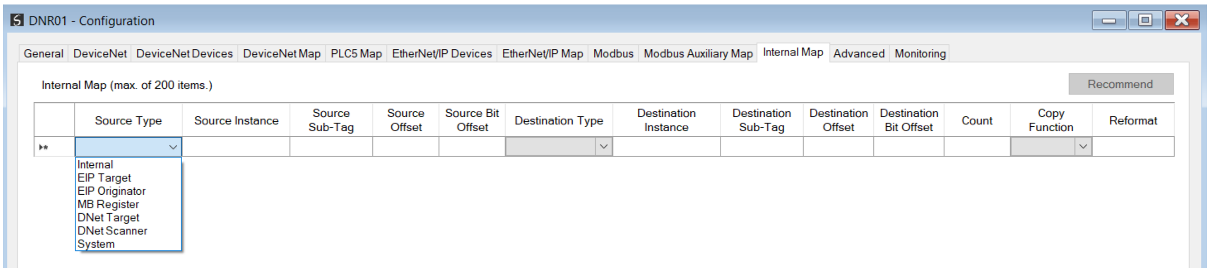


Figure 3.127 – Internal Map – Source Type

3.7.1.1. INTERNAL

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

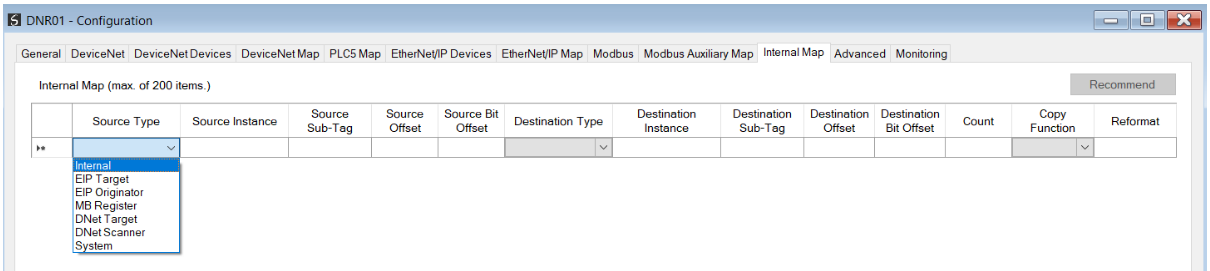


Figure 3.128 – IDS Copy – Internal Source Type

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

3.7.1.2. EIP TARGET

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

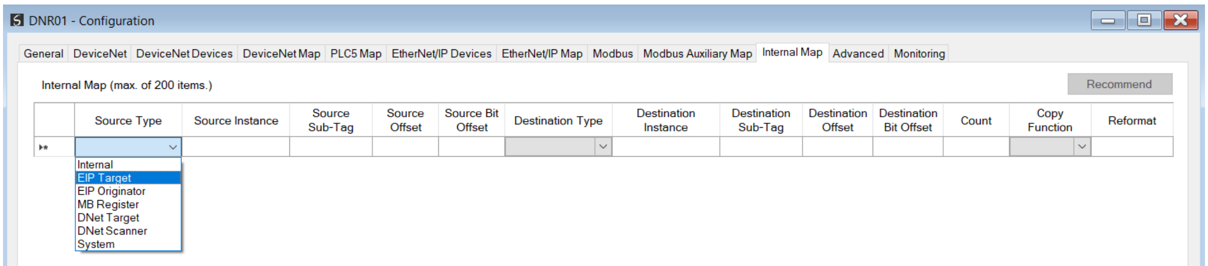


Figure 3.129 – IDS Copy – EtherNet/IP Target Source Type

The source instance will be the connection number, which can be connection 0 to 3, based on the number of connections configured. The Source Offset is the offset in the *Mapped Data* section of the EtherNet/IP output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.1.3. EIP ORIGINATOR

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

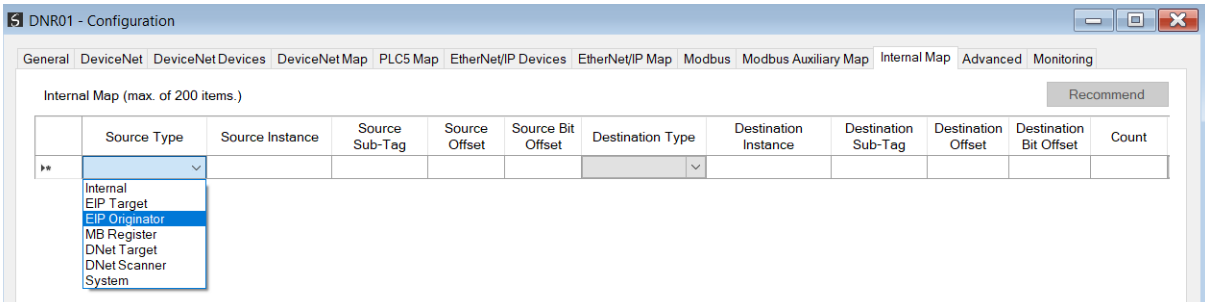


Figure 3.130 – IDS Copy – EtherNet/IP Originator Source Type

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

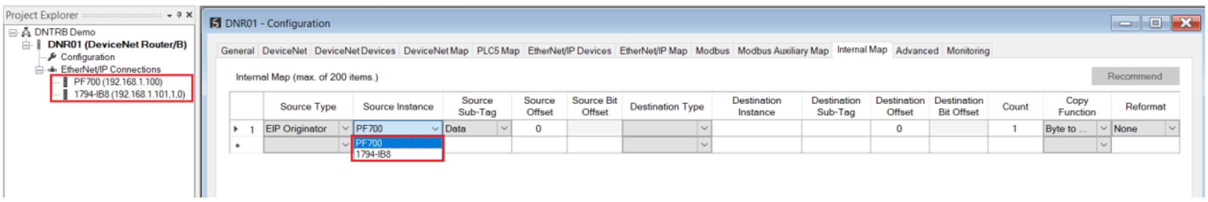


Figure 3.131 – IDS Copy – EtherNet/IP Originator Source Instance

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

The user can select to copy the data from the EtherNet/IP connection or the status.

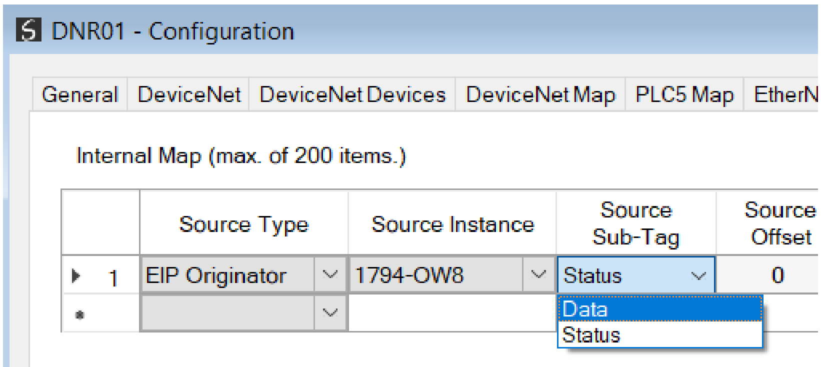


Figure 3.132 – IDS Copy – EtherNet/IP Originator Status

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

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

Table 3.19 – EtherNet/IP Originator Connection Status

3.7.1.4. MODBUS REGISTER

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

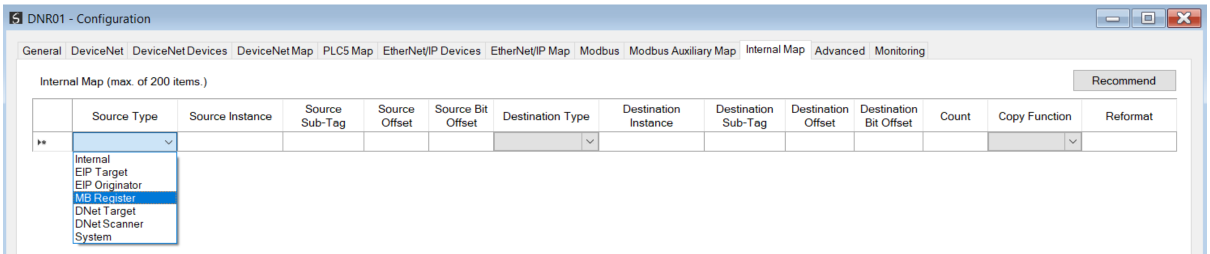


Figure 3.133 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

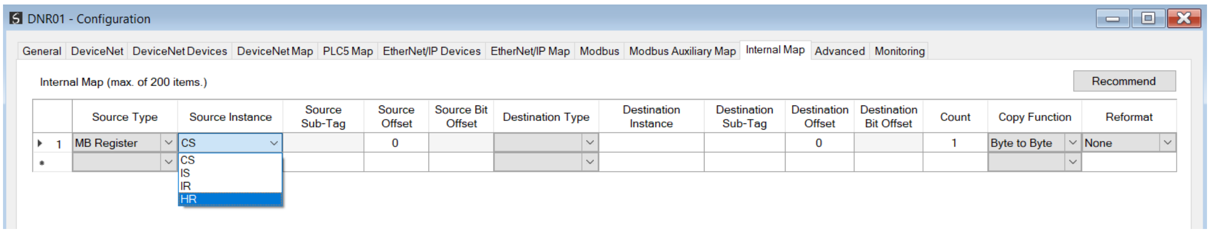


Figure 3.134 – IDS Copy - Modbus Source Instance

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

3.7.1.5. DNET TARGET

When copying data from a connection originator (e.g. the output assembly of the 1756-DNB) to the Ethernet interface, the source type needs to be DNet Target.

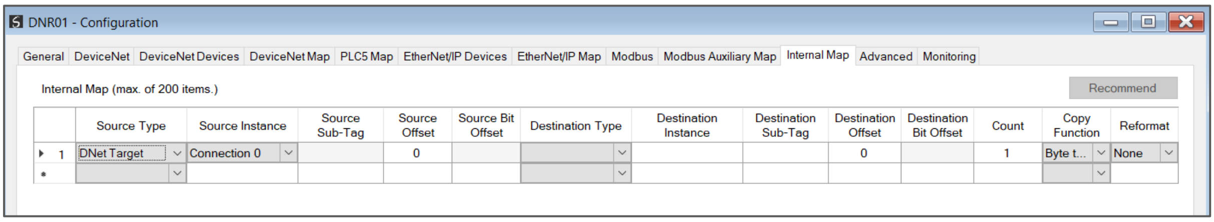


Figure 3.135 – IDS Copy – DeviceNet Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset of the consumed data from the DeviceNet originator (e.g., 1756-DNB) from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.1.6. DNET SCANNER

When copying data from a DeviceNet IO to the Ethernet interface, the source type needs to be DNet Scanner.

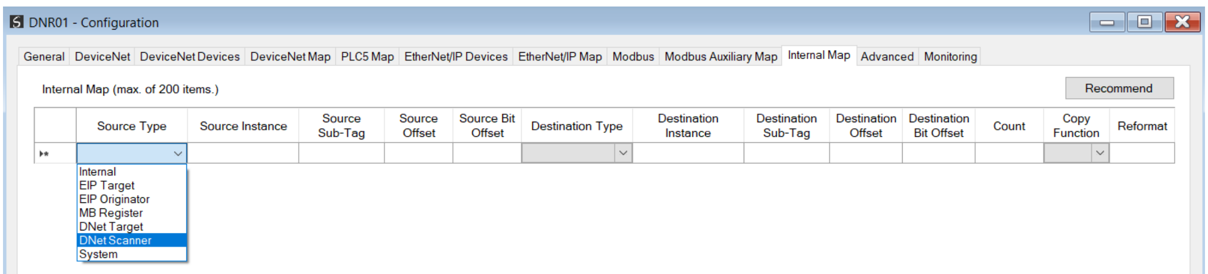


Figure 3.136 – IDS Copy – DeviceNet Scanner Source Type

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

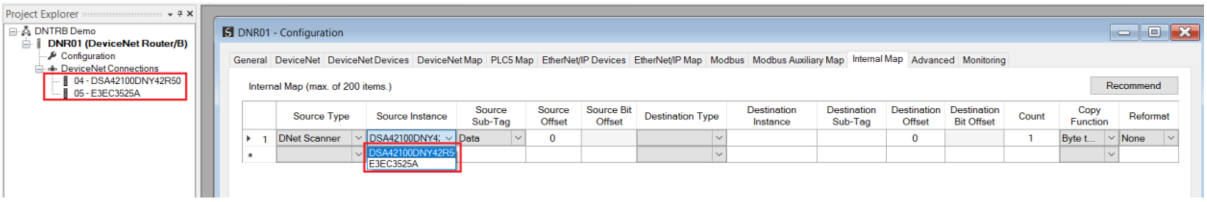


Figure 3.137 – IDS Copy – DeviceNet Scanner Source Instance

The Source Offset is the offset in the selected DeviceNet device Cyclic **Input** Assembly (data being produced by the DeviceNet IO device). The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

The user can select to copy the data from the EtherNet/IP connection or the status.

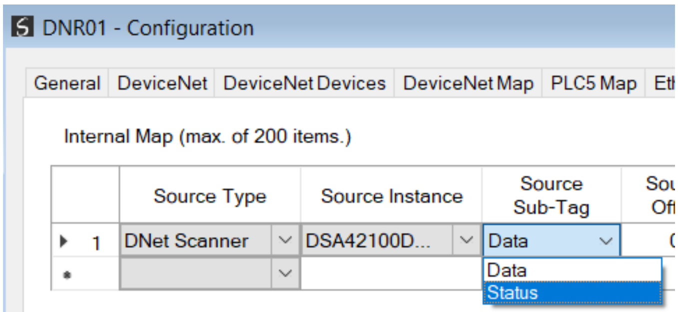


Figure 3.138 – IDS Copy – EtherNet/IP Originator Status

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

Parameter	Data Type	Description
DeviceNet Scanner Connection Status	DINT	Bit 0 – Online Bit 1 – Cyclic Data Exchange Ok Bit 2 – Device Mismatch
Node	SINT	The target DeviceNet device Node number.
Reserved	SINT	Reserved for future use.
Cyclic Communication Timeout Count	DINT	Number of times the cyclic connection has gone offline.
Cyclic Communication Tx Count	DINT	Number of cyclic connection bytes sent.
Cyclic Communication Rx Count	DINT	Number of cyclic connection byte received.

Table 3.20 – DeviceNet Scanner Connection Status

3.7.1.7. SYSTEM

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

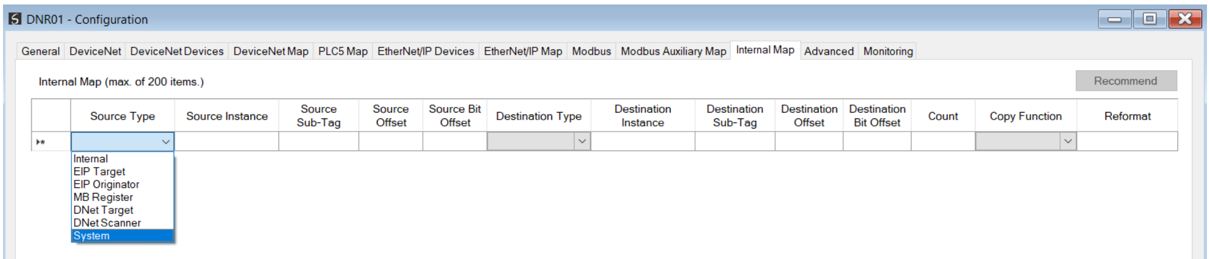


Figure 3.139 – IDS Copy – System Information

The module's System information has the following format.

Parameter	Data Type	Description
Status	INT	Module Status. Bit 0 – Module Config Valid Bit 1 – EtherNet/IP Originator Comms Ok Bit 2 – Modbus Comms Ok Bit 3 – DeviceNet Scanner Comms Ok Bit 4 – EtherNet/IP Target Comms Ok Bit 5 – DeviceNet Target Comms Ok Bit 6 – Power is connected to the bottom connector Bit 7 – Power is connected to the front connector. Bit 8 – Duplicate Node Number Bit 9 – NTP Ok
ConfigCRC	INT	The module configuration signature.
Actual BAUD	SINT	Current BAUD rate. 0 – 125K 1 – 250K 2 – 500K
Actual Node	SINT	Current DeviceNet node number
DeviceNet Target Status	INT	Module status when operating as a DeviceNet Target. Bit 0 – DeviceNet Poll Connection Active Bit 1 – DeviceNet NULL Connection Active Bit 2 – DeviceNet Change-Of-State (COS) Connection Active
Reserved	SINT[6]	Reserved for future use.

Table 3.21 – System Information Format

3.7.2. Copy To

One of six destinations can be selected to copy to:
Internal, EIP Target, EIP Originator, DNet Target, and DNet Scanner.

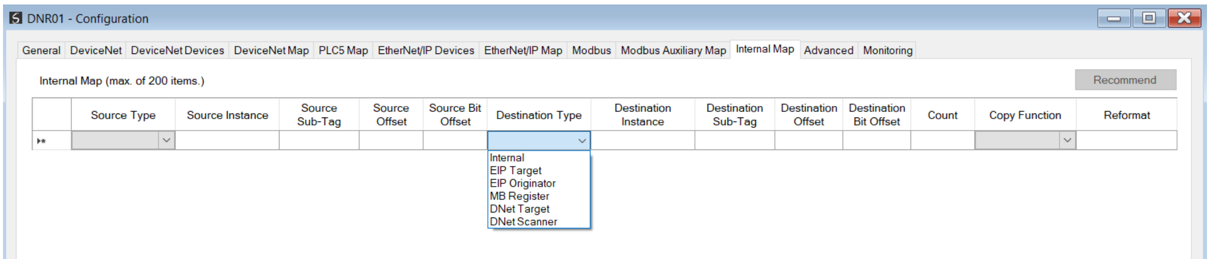


Figure 3.140 – Internal Map – Destination Type

3.7.2.1. INTERNAL

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

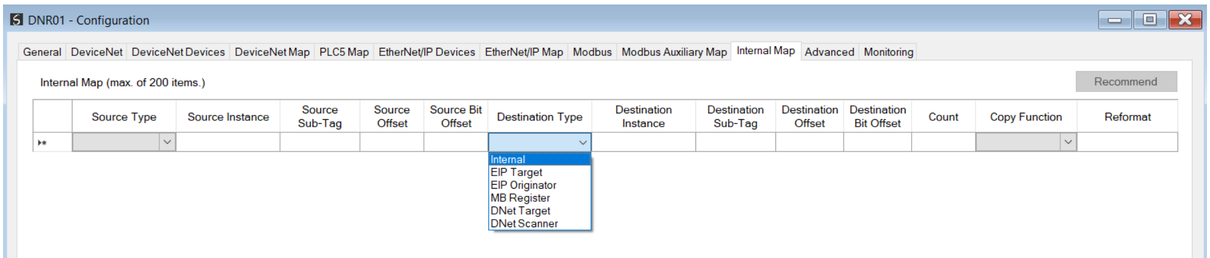


Figure 3.141 – IDS Copy – Internal Source Type

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

3.7.2.2. EIP TARGET

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

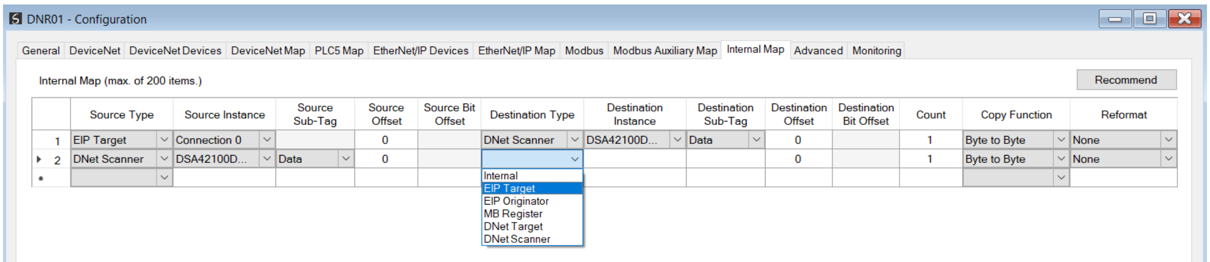


Figure 3.142 – IDS Copy – EtherNet/IP Target Destination Type

The destination instance will be the connection number, which can be connection 0 to 3, based on the number of connections configured. The Destination Offset is the offset of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.2.3. EIP ORIGINATOR

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

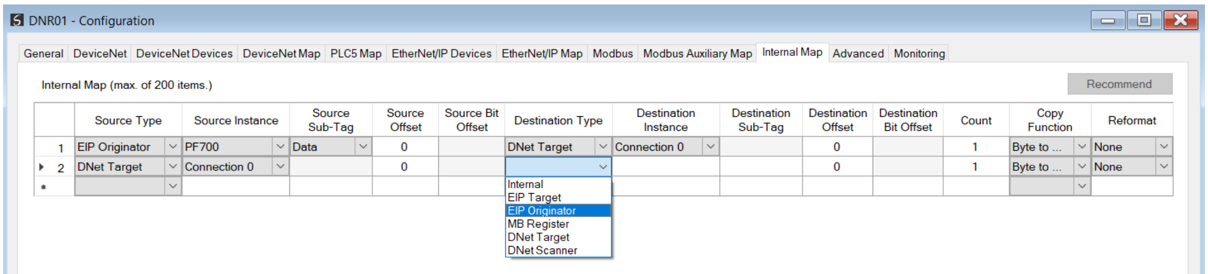


Figure 3.143 – IDS Copy – EtherNet/IP Originator Destination Type

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

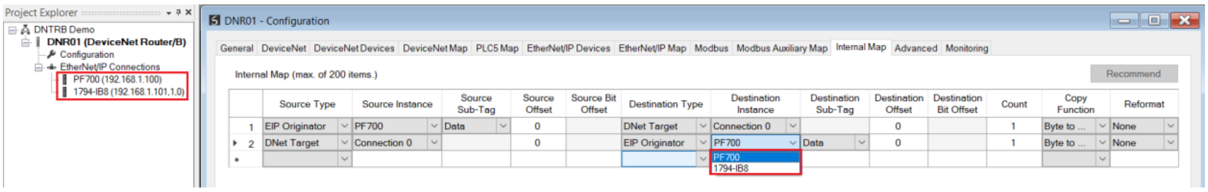


Figure 3.144 – IDS Copy – EtherNet/IP Originator Destination Instance

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

3.7.2.4. MODBUS REGISTER

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

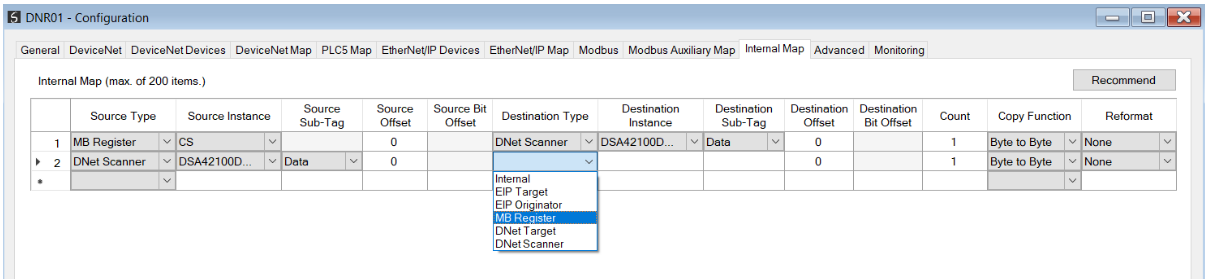


Figure 3.145 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

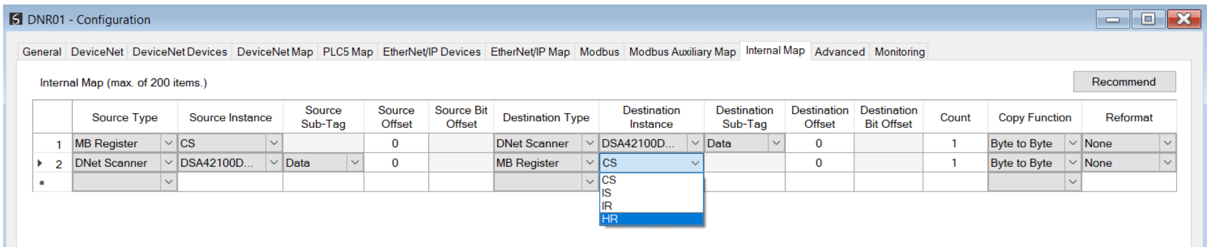


Figure 3.146 – IDS Copy - Modbus Destination Instance

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

3.7.2.5. DNET TARGET

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

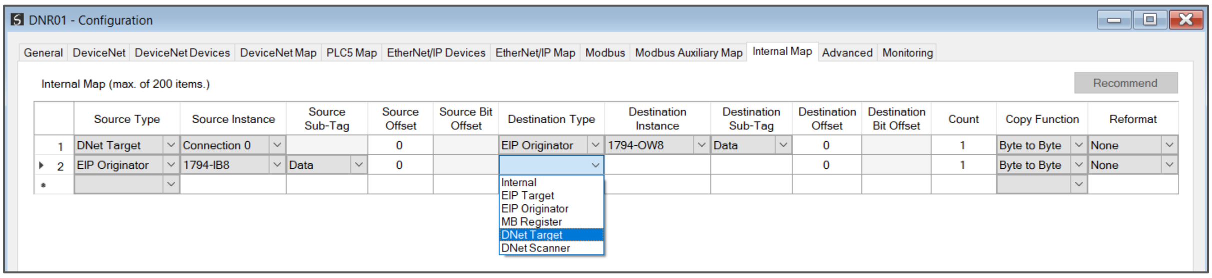


Figure 3.147 – IDS Copy – DeviceNet Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset of the produced data to the DeviceNet originator (e.g., 1756-DNB) from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.2.6. DNET SCANNER

When copying data from the Ethernet interface to a DeviceNet IO device **Output** Assembly, the destination type needs to be DNet Scanner.

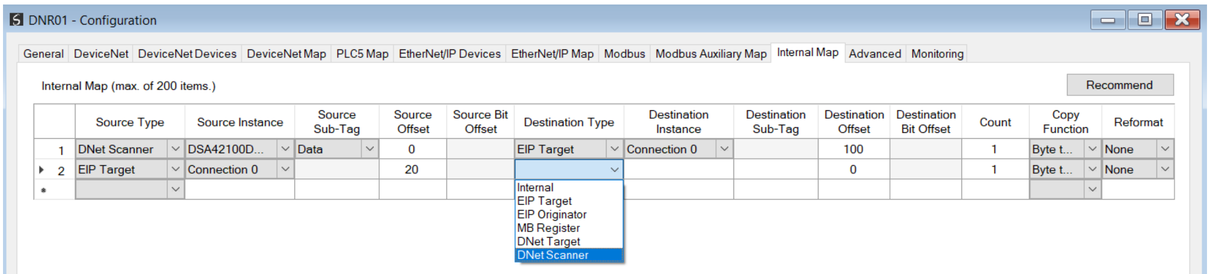


Figure 3.148 – IDS Copy – DeviceNet Scanner Destination Type

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

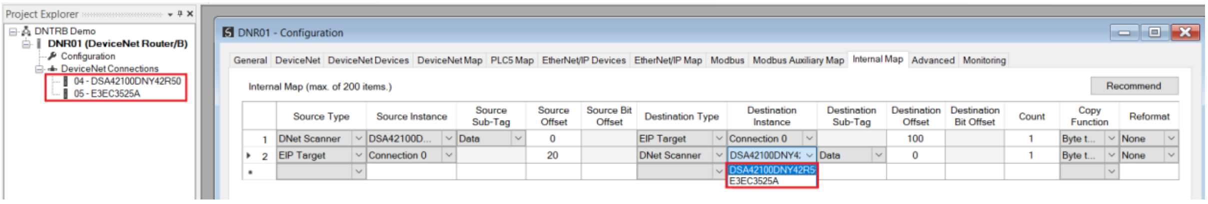


Figure 3.149 – IDS Copy – DeviceNet Scanner Destination Instance

The Destination Offset is the offset in the selected DeviceNet device Cyclic **Output** Assembly (data being consumed by the DeviceNet IO device). The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

NOTE: For more information regarding the specific Internal Map operation for specific interfaces, see the setup and configuration sections for the various DeviceNet and Primary Interfaces.

3.8. ADVANCED

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

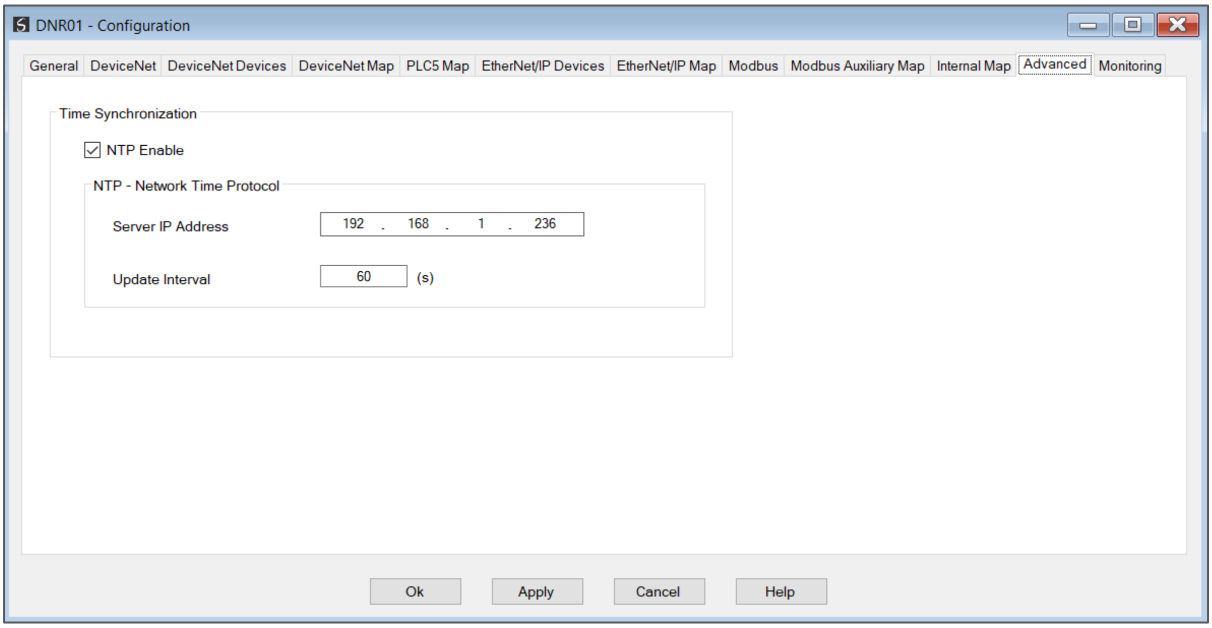


Figure 3.150 – Advanced configuration

The Advanced configuration consists of the following parameters:

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

Table 3.22 – Advanced configuration parameters

3.9. MONITORING

The Monitoring configuration will allow a user to send DeviceNet packet captures and statistics to a remote target (server) using Ethernet UDP communication to a configurable UDP port.

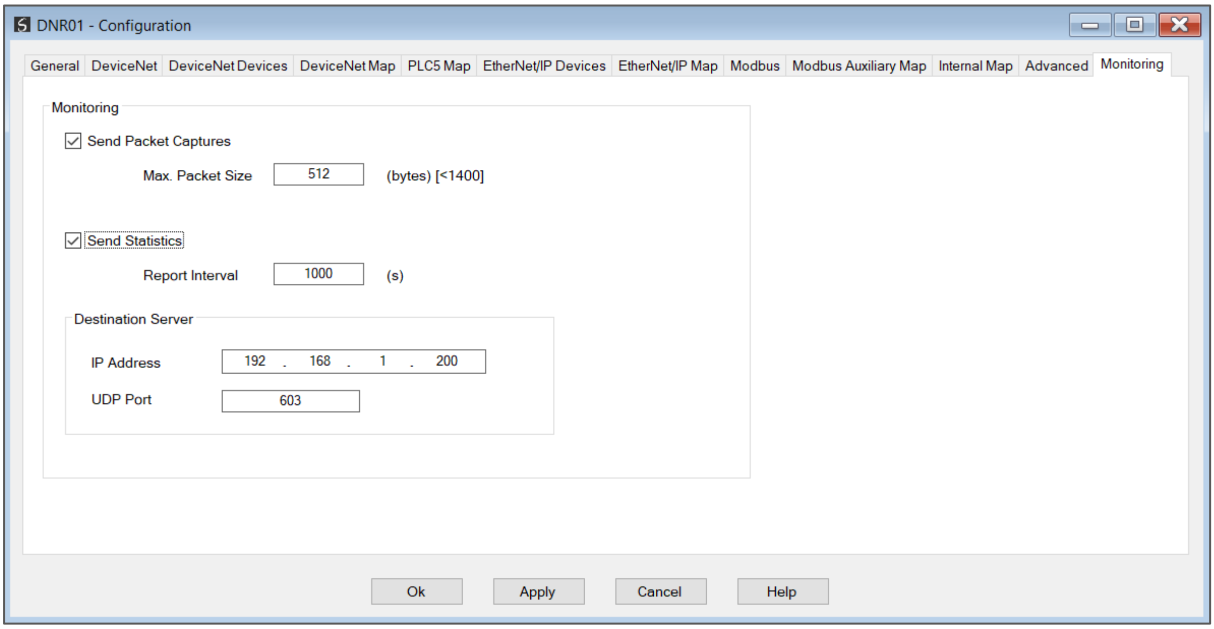


Figure 3.151 – Monitoring configuration

The Monitoring configuration consists of the following parameters:

Parameter	Description
Send Packet Captures	When enabled, the DeviceNet Router/B module will send the DeviceNet packet capture data to the Destination Server once the *Max Packet Size* has been reached.
Max Packet Size	When *Send Packet Captures* is enabled, then this parameter will determine what is the send trigger size. Once the DeviceNet packet capture buffer size reaches this parameter, then the UDP packet will be sent.
Send Statistics	When enabled, the DeviceNet Router/B module will send DeviceNet Statistics to the Destination Server at the *Report Interval*.
Report Interval	This is the rate at which the DeviceNet statistics will be sent to the Destination Server.
Destination Server	
IP Address	Destination Server IP address.
UDP Port	Destination Server UDP port.

Table 3.23 – Monitoring configuration parameters

The format of the DeviceNet Packet Capture that will be sent is shown below.

Parameter	Data Type	Length	Description
Start Flag	DINT	4	0xAABBCCDD

Packet Type	SINT	1	Fixed to 1 – Packet Capture
Packet Count	INT	2	Number of packets in the payload
Timestamp Date/Time			
Year	INT	2	Year when this UDP packet was sent
Month	SINT	1	Month when this UDP packet was sent
Day	SINT	1	Day when this UDP packet was sent
Hour	SINT	1	Hour when this UDP packet was sent
Minute	SINT	1	Minute when this UDP packet was sent
Seconds	SINT	1	Seconds when this UDP packet was sent
Milliseconds	INT	2	Milliseconds when this UDP packet was sent
Packets x Packet Count			
Packet Data	SINT[10]	10	The raw DeviceNet Packet data

Table 3.24 – DeviceNet Packet Capture format

The format of the DeviceNet Statistics Packet that will be sent is shown below.

Parameter	Data Type	Length	Description
Start Flag	DINT	4	0xAABBCCDD
Packet Type	SINT	1	Fixed to 2 – Real Time Statistics
Report ID	INT	2	Increased each time a report is sent
General (CAN Bus) Statistics			
Rx CAN Packet Count	INT	2	Received CAN message count.
Tx CAN Packet Count	INT	2	Transmitted CAN message count.
CAN CRC Errors	INT	2	CAN CRC failed message count.
CAN Bit Errors	INT	2	CAN Bit error count.
Can Stuff Errors	INT	2	CAN Stuff error count.
Bus Off	INT	2	The number of times the CAN receiver has detected the Bus Off state.
Ack Error	INT	2	The number of times the CAN message was no acknowledged.
Format Error	INT	2	The number of times a fixed format part of the received frame has the wrong format.
DeviceNet (Node Specific) Statistics			

Node 0 - Statistics	[NodeStats]	16	DeviceNet Node 0 Statistics
PacketTx	*INT*	*2*	*Transmitted CAN message count.*
PacketRx	*INT*	*2*	*Received CAN message count.*
IOPollRequests	*INT*	*2*	*Number IO Polls that have been sent to the node*
IOPollResponses	*INT*	*2*	*Number IO Polls that the node has sent*
ConnectionEstablishRequest	*INT*	*2*	*Number of Connection Establishments sent to the node*
ConnectionEstablishResponse	*INT*	*2*	*Number of Connection Responses sent by the node*
ExplicitRequests	*INT*	*2*	*Number Explicit Message Requests that have been sent to the node*
ExplicitRexponses	*INT*	*2*	*Number Explicit Message Requests that the node has sent*
Node 1 - Statistics	[NodeStats]	16	DeviceNet Node 1 Statistics
Node 2 - Statistics	[NodeStats]	16	DeviceNet Node 2 Statistics
Node 3 - Statistics	[NodeStats]	16	DeviceNet Node 3 Statistics
…	…	…	…
Node 62 - Statistics	[NodeStats]	16	DeviceNet Node 62 Statistics
Node 63 - Statistics	[NodeStats]	16	DeviceNet Node 63 Statistics

Table 3.25 – DeviceNet Statistics format

3.10. MODULE DOWNLOAD

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

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

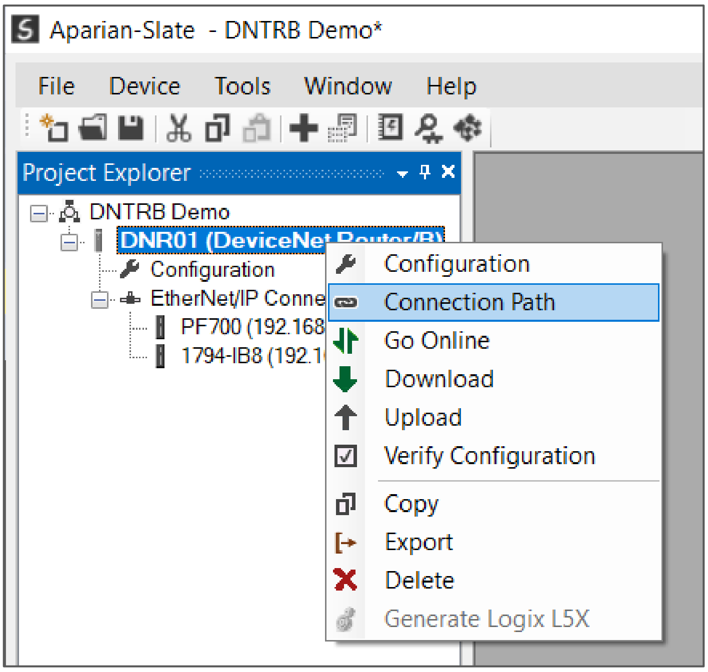


Figure 3.152 - Selecting Connection Path

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

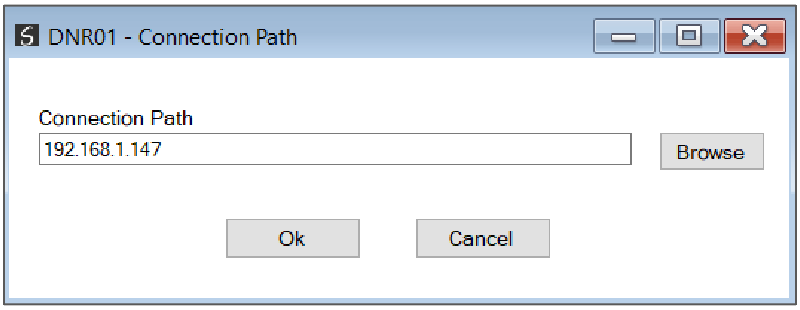


Figure 3.153 - Connection Path

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

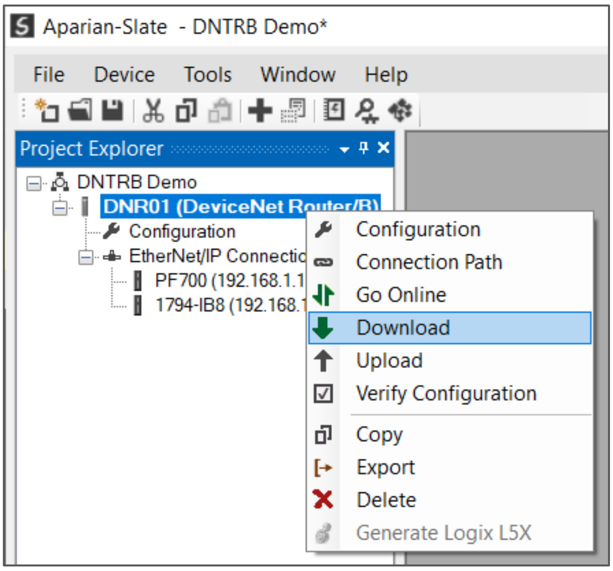


Figure 3.154 - Selecting Download

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

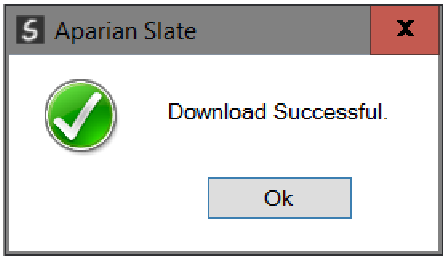


Figure 3.155 - Successful download

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.

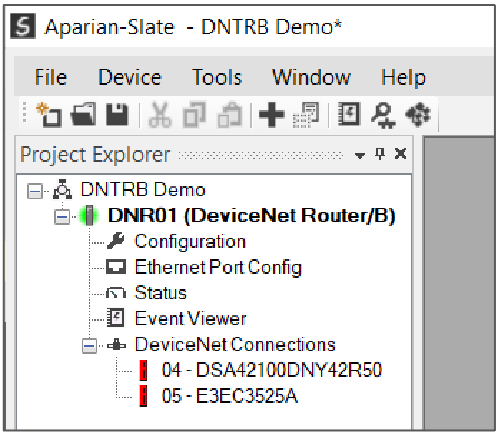


Figure 3.156 - Module online

4. DEVICE FIRMWARE UPDATE

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

To firmware upgrade the module, follow the steps below:

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

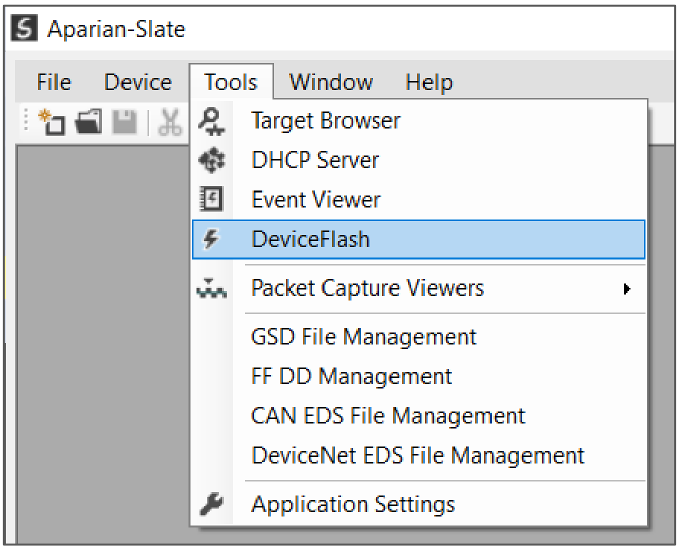


Figure 4.1 – Select DeviceFlash utility from Slate

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

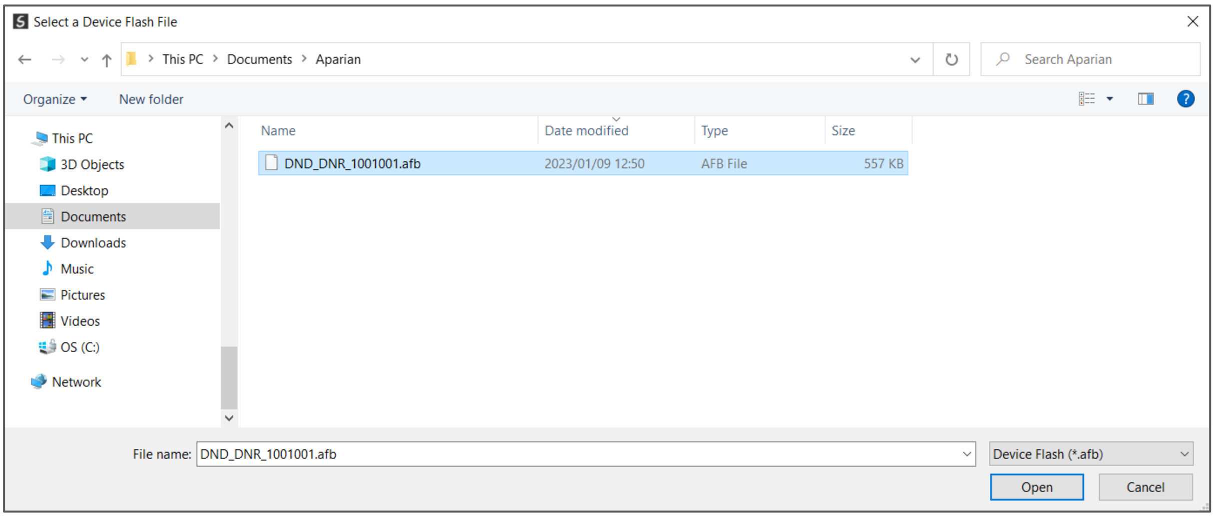


Figure 4.2 – Select the binary file

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

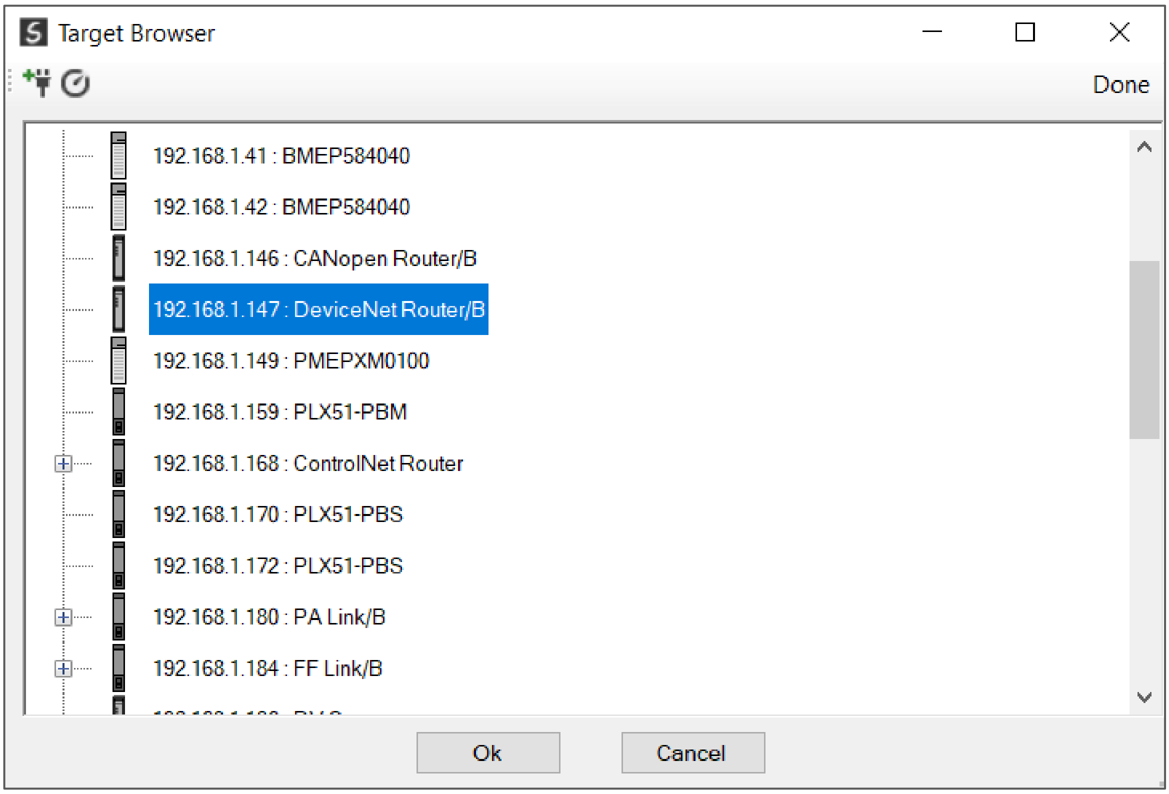


Figure 4.3 – Select the device to be updated

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

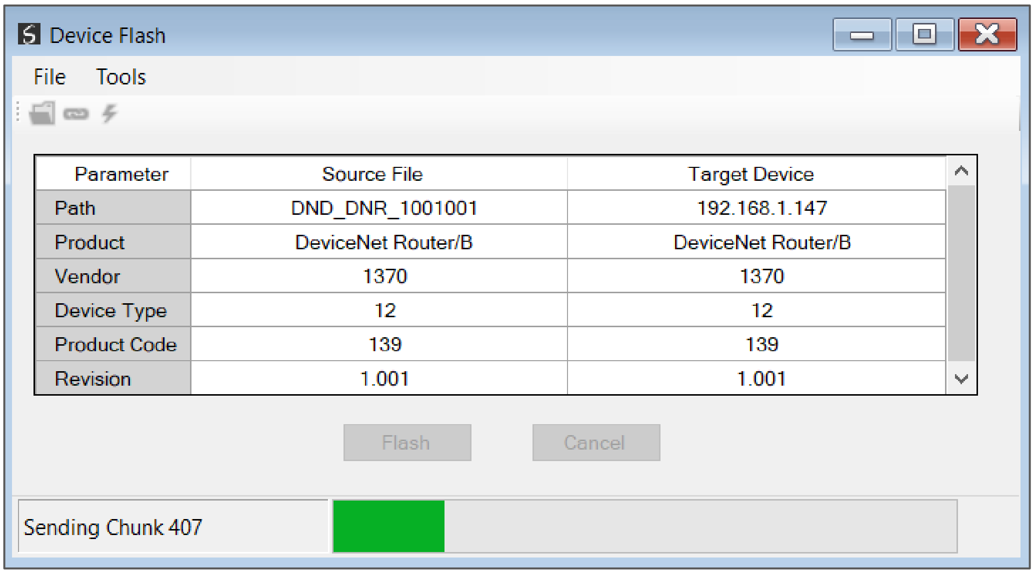


Figure 4.4 – Firmware update busy

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

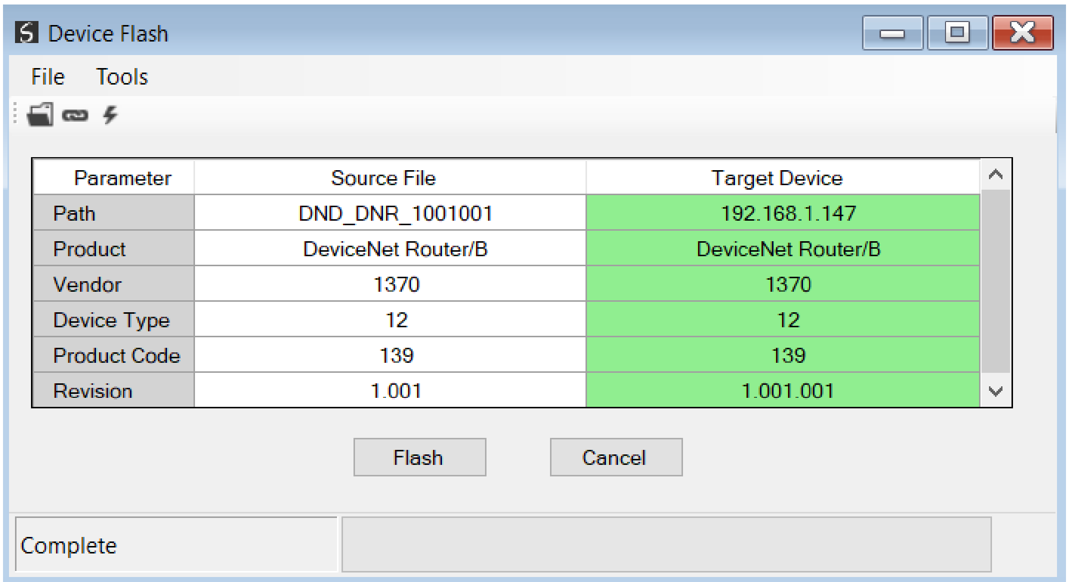


Figure 4.5 – Firmware update successfully completed.

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

5. OPERATION

5.1. DEVICENET TARGET

A Logix controller can own the DeviceNet Router/B over DeviceNet using a cyclic DeviceNet connection when the module is operating as a DeviceNet target. This will allow the DeviceNet Router/B to exchange data with the Logix controller using the input and output assembly of the DeviceNet Scanner module (e.g., 1756-DNB) to which has mapped the produced and consumed data of the DeviceNet Router/B module.

The module will produce the number of bytes configured in the DeviceNet configuration (in Slate) and consume the number of bytes configured.

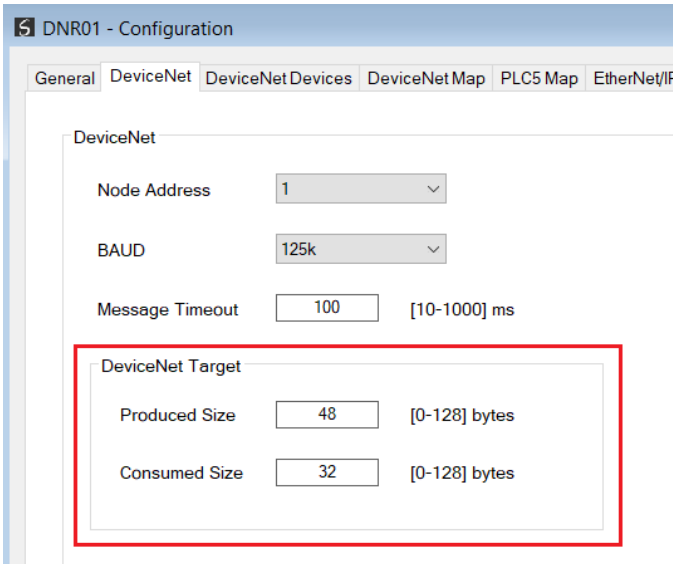


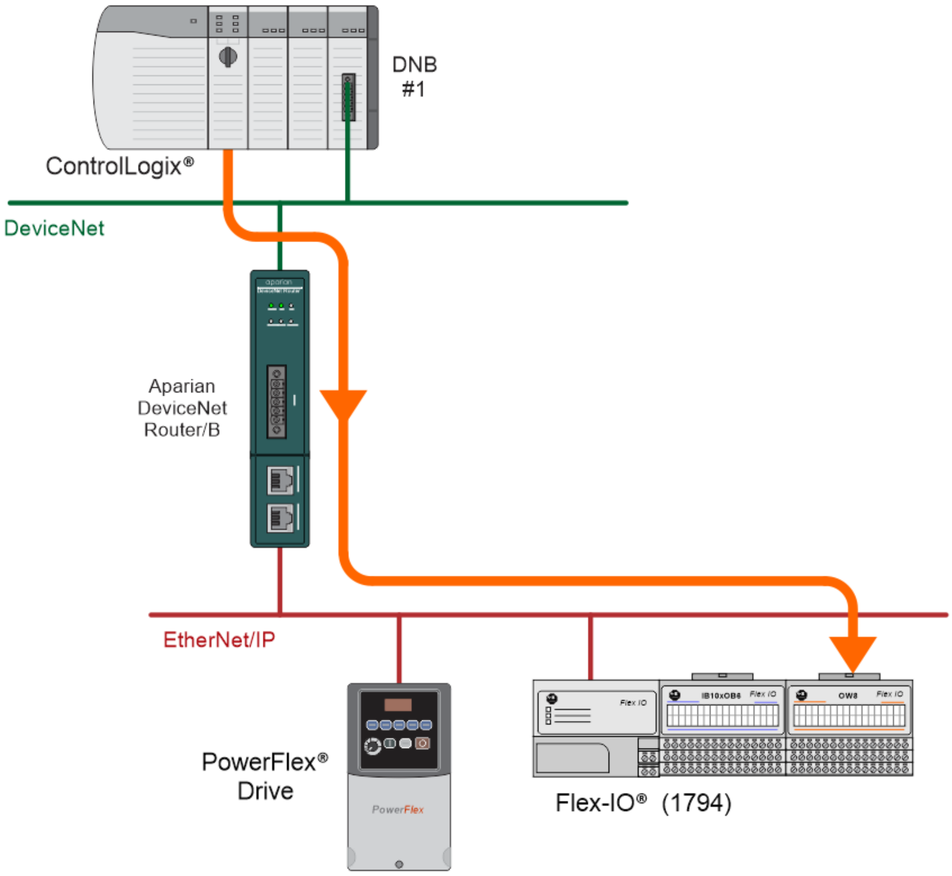
Figure 5.1 – DeviceNet Target Produce and Consume size

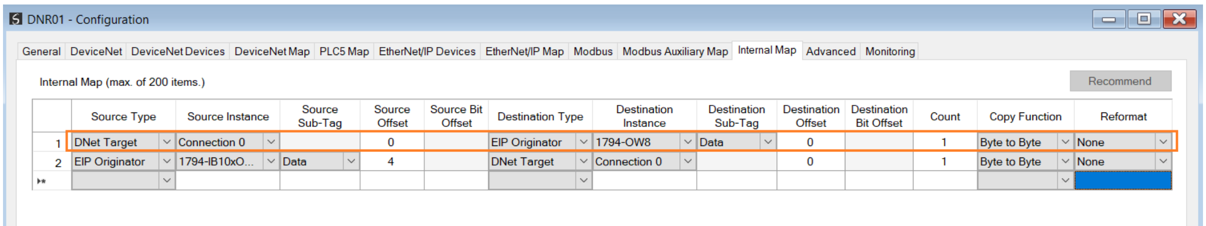
Using the module's internal mapping, the user can copy data to the DeviceNet Target produce array or copy data from the DeviceNet Target consume array. This will allow the exchange of devices on the Ethernet or Serial interface (e.g., EtherNet/IP, Modbus, etc) to exchange data with the DeviceNet Bridge which in turn can be accessed by a Logix or PLC5 controller.

In the example below, the DeviceNet Router/B module is connected to two 1794 Flex modules via an EtherNet/IP adapter. The module is also connected to a 1756-DNB Scanner which will map 48 bytes of the module DeviceNet produced data to the 1756-DNB input assembly and 32 bytes of the module DeviceNet consume data to the 1756-DNB output assembly.

The data from the Logix controller that has been copied to the output assembly of the 1756-DNB (where the DeviceNet Router/B has been mapped) will be sent to the DeviceNet

Router/B via the cyclic consumed DeviceNet connection. This data will then be copied to the output assembly of the Flex 1794-OW8 using the internal mapping which is communicated to the 1794-OW8 using the Class 1 EtherNet/IP connection.

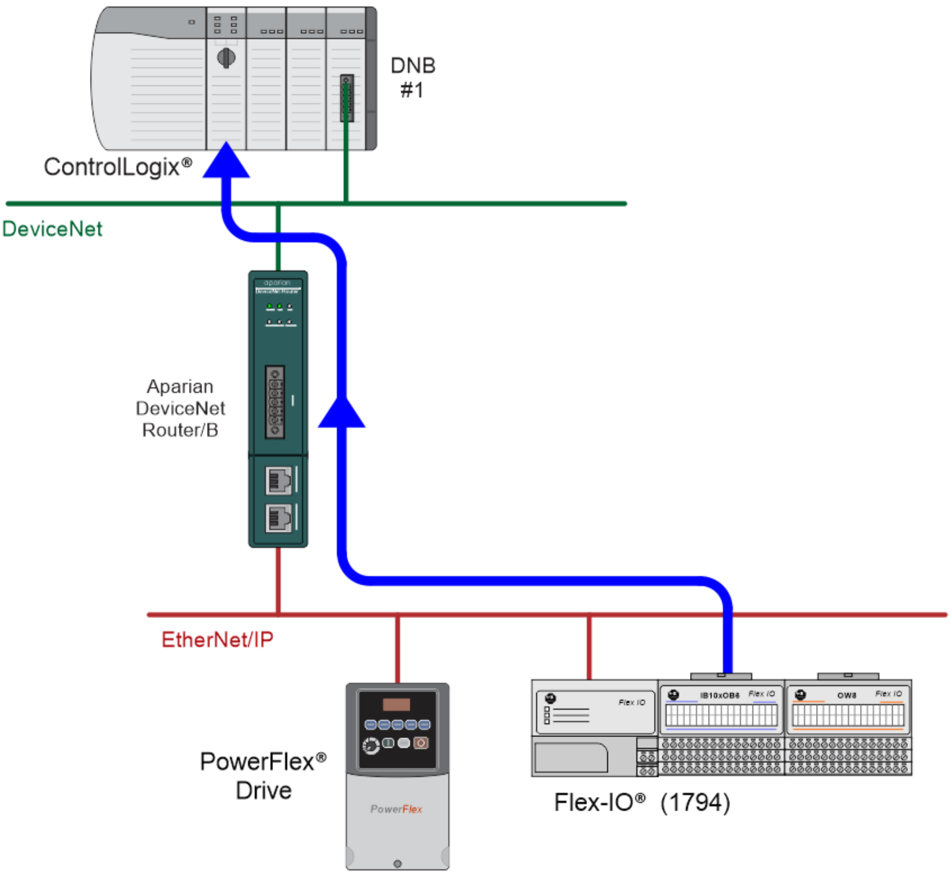


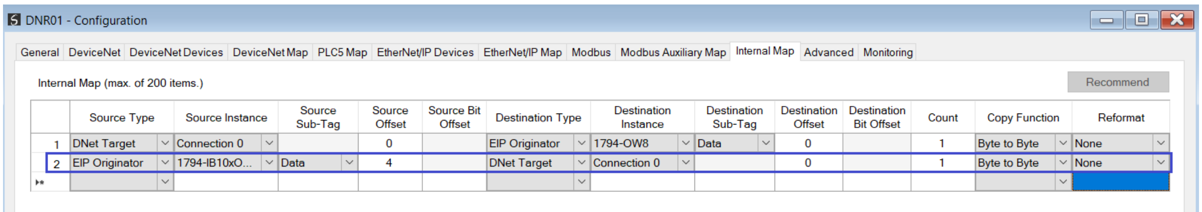


	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	DNet Target	Connection 0		0		EIP Originator	1794-OW8	Data	0		1	Byte to Byte	None
2	EIP Originator	1794-IB10xO...	Data	4		DNet Target	Connection 0		0		1	Byte to Byte	None

Figure 5.2 – Internal Mapping from DeviceNet Scanner to EtherNet/IP Originator

The data from the 1794-IB10xOB6 is exchanged with the DeviceNet Router/B using the input assembly of its class 1 EtherNet/IP connection. This data is then copied to the DeviceNet Router/B DeviceNet produce buffer which is exchanged with the DeviceNet Scanner (1756-DNB). The Logix controller can then access this data from the 1756-DNB input assembly where the data from the DeviceNet Router/B has been mapped.





	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	DNet Target	Connection 0		0		EIP Originator	1794-OW8	Data	0		1	Byte to Byte	None
2	EIP Originator	1794-IB10xO...	Data	4		DNet Target	Connection 0		0		1	Byte to Byte	None

Figure 5.3 – Internal Mapping from EtherNet/IP Originator to DeviceNet Scanner

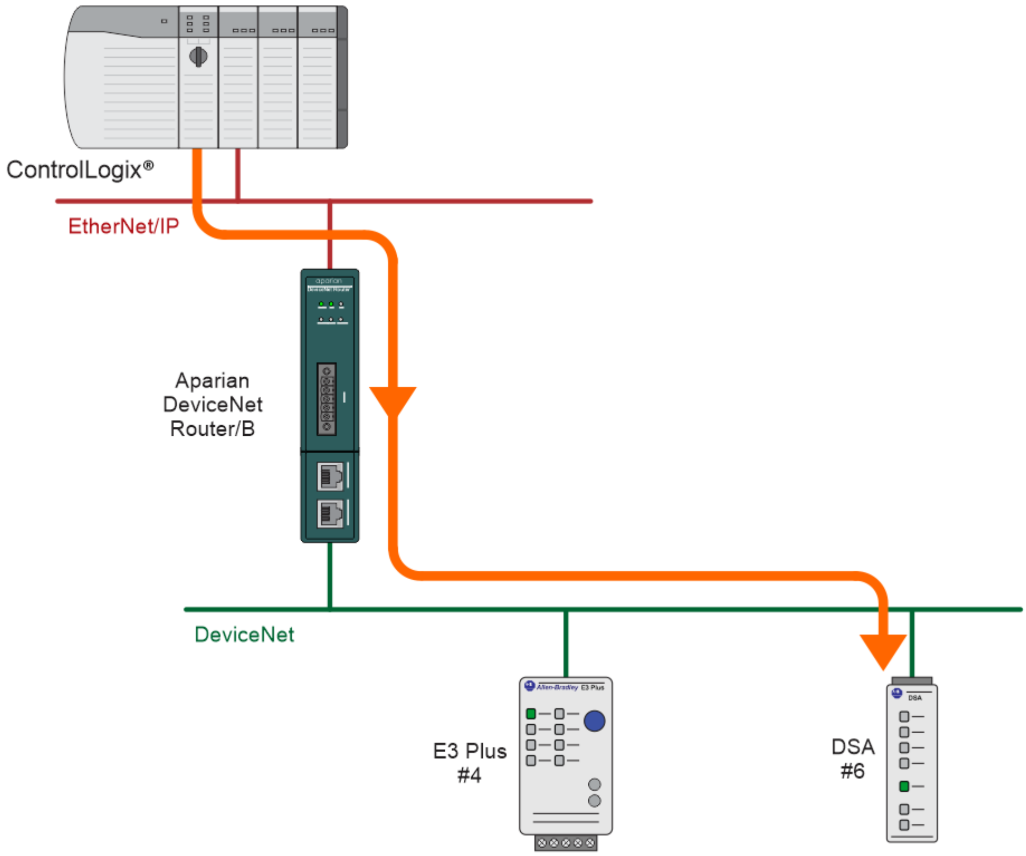
5.2. DEVICENET SCANNER

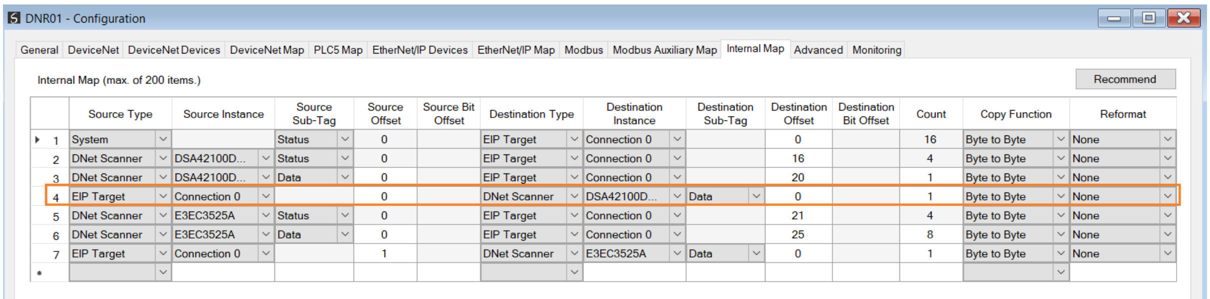
The DeviceNet Router/B module can operate as a DeviceNet connection originator. In this mode the module can exchange data from the Primary Interface with DeviceNet IO devices using either the produced and consumed images of the cyclic DeviceNet IO connection or using an explicit (unscheduled UCMM) DeviceNet message to read or write data.

5.2.1. CYCLIC DEVICENET CONNECTIONS

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

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

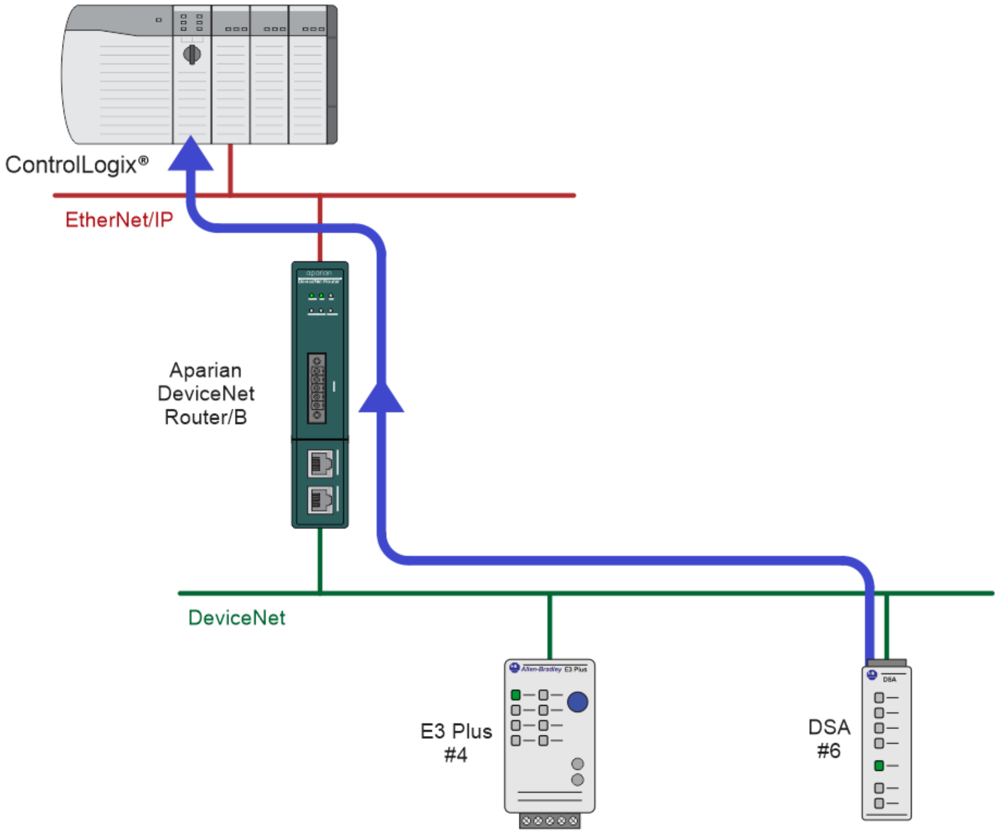


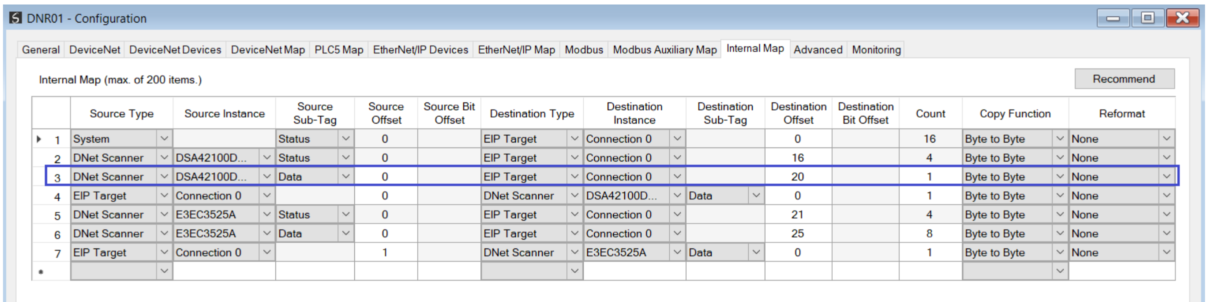


	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0		EIP Target	Connection 0		0		16	Byte to Byte	None
2	DNet Scanner	DSA42100D...	Status	0		EIP Target	Connection 0		16		4	Byte to Byte	None
3	DNet Scanner	DSA42100D...	Data	0		EIP Target	Connection 0		20		1	Byte to Byte	None
4	EIP Target	Connection 0		0		DNet Scanner	DSA42100D...	Data	0		1	Byte to Byte	None
5	DNet Scanner	E3EC3525A	Status	0		EIP Target	Connection 0		21		4	Byte to Byte	None
6	DNet Scanner	E3EC3525A	Data	0		EIP Target	Connection 0		25		8	Byte to Byte	None
7	EIP Target	Connection 0		1		DNet Scanner	E3EC3525A	Data	0		1	Byte to Byte	None

Figure 5.4 – Internal Mapping from EtherNet/IP to DeviceNet Originator

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





DNR01 - Configuration

General | DeviceNet | DeviceNet Devices | DeviceNet Map | PLC5 Map | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Advanced | Monitoring

Internal Map (max. of 200 items.)

Recommend

	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0		EIP Target	Connection 0		0		16	Byte to Byte	None
2	DNet Scanner	DSA42100D...	Status	0		EIP Target	Connection 0		16		4	Byte to Byte	None
3	DNet Scanner	DSA42100D...	Data	0		EIP Target	Connection 0		20		1	Byte to Byte	None
4	EIP Target	Connection 0		0		DNet Scanner	DSA42100D...	Data	0		1	Byte to Byte	None
5	DNet Scanner	E3EC3525A	Status	0		EIP Target	Connection 0		21		4	Byte to Byte	None
6	DNet Scanner	E3EC3525A	Data	0		EIP Target	Connection 0		25		8	Byte to Byte	None
7	EIP Target	Connection 0		1		DNet Scanner	E3EC3525A	Data	0		1	Byte to Byte	None
*													

Figure 5.5 – Internal Mapping from DeviceNet Originator to EtherNet/IP

5.2.1.1. CONNECTION STATUS

The user can select to copy the data from the EtherNet/IP connection or the status.

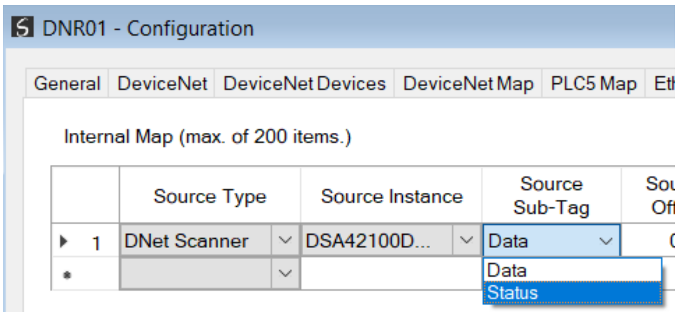


Figure 5.6 – IDS Copy – EtherNet/IP Originator Status

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

Parameter	Data Type	Description
DeviceNet Scanner Connection Status	DINT	Bit 0 – Online Bit 1 – Cyclic Data Exchange Ok Bit 2 – Device Mismatch
Node	SINT	The target DeviceNet device Node number.
Reserved	SINT	Reserved for future use.
Cyclic Communication Timeout Count	DINT	Number of times the cyclic connection has gone offline.
Cyclic Communication Tx Count	DINT	Number of cyclic connection bytes sent.
Cyclic Communication Rx Count	DINT	Number of cyclic connection byte received.

Table 5.1 – DeviceNet Scanner Connection Status

5.2.1.2. CONFIGURING AND MONITORING PARAMETERS

The device's parameters can be monitored and configured using the ***Parameters*** tab of the DeviceNet Cyclic Connection window.

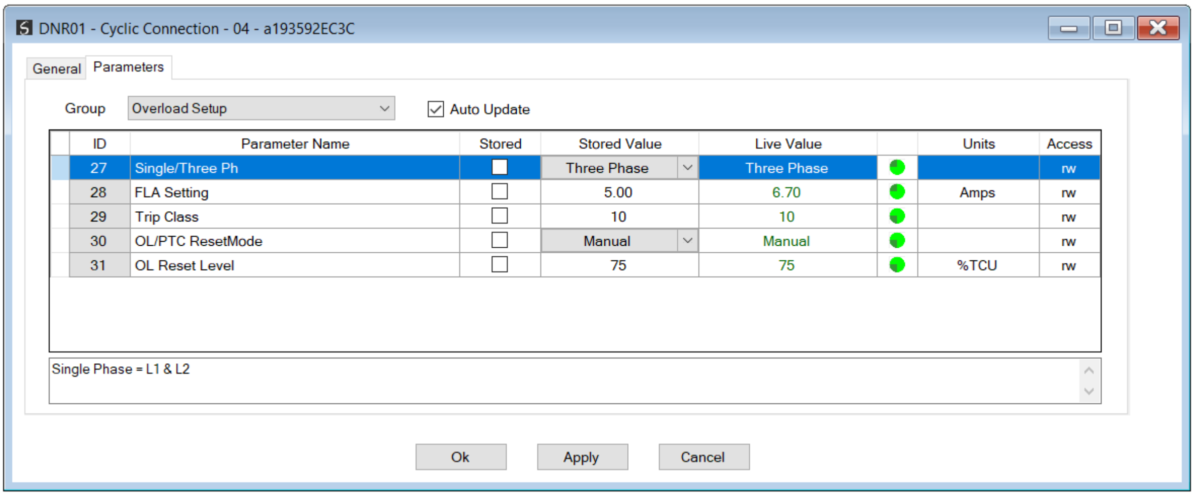


Figure 5.7 – Parameters

A number of functions can be performed on a parameter by right-clicking on it, which exposes the parameter context menu.

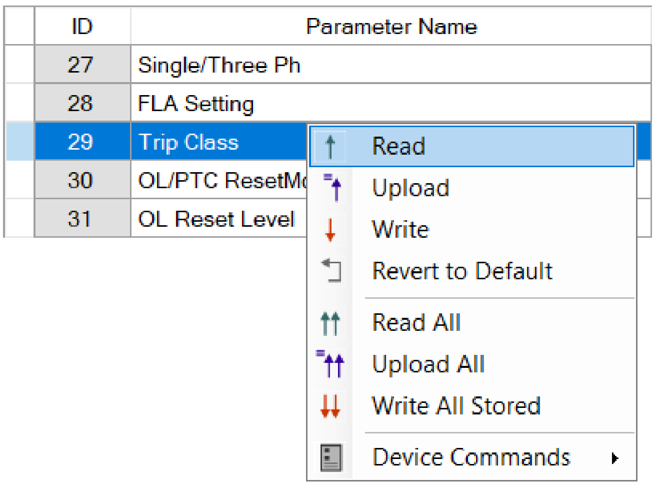


Figure 5.8 – Parameter Functions Menu

The parameter functions are follows:

Function	Description
Read	Reads the selected parameter from the device and displays it in the ***Live Value*** field. Note: This has no effect on the configuration.
Upload	Reads the selected parameter from the device and copies the value to the ***Stored Value*** field. Note: This has no effect on the configuration unless the ***Stored*** option is checked.
Write	Writes the value from the ***Stored Value*** field to the device.
Revert to Default	Reverts the value in the ***Stored Value*** field to the default value contained in the EDS file.
Read All	Reads all the parameters from the device and displays them in the corresponding ***Live Value*** field. Note: This has no effect on the configuration.
Upload All	Uploads all the parameters. That is, reads all the parameters from the device and copies them to the corresponding ***Stored Value*** field. Note: This has no effect on the configuration unless the ***Stored*** option is checked.
Write All Stored	Writes the value from the ***Stored Value*** field to the device for all parameters that have the ***Stored*** option checked.
Device Commands	The following Device Commands can be sent to the device. ***Note***: Many devices do not support these functions.
Save All to NV	Commands the device to transfer all its current parameter values to non-volatile storage.

Restore All from NV	Commands the device to overwrite all its current parameter values with those in non-volatile storage.
Reset All Values from Default	Commands the device to overwrite all its current parameter values with the system default values.

Table 5.2 – DeviceNet Parameter Functions

The column adjacent to the ***Live Value*** displays an icon which represents the status of the last action performed. The status icons are as follows:

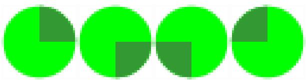
Icon/s	**Description**
	Read Successful. Each time a read is successful the icon is updated to the next in the series.
	Read Failed.
	Write Successful.
	Write Failed.

Table 5.3 – DeviceNet Parameter Status Icons

By selecting the ***Auto Update*** option, all the visible Parameters will be read up from the device and displayed in the ***Live Value*** column. Note that this will not affect the ***Stored Value*** of each parameter.

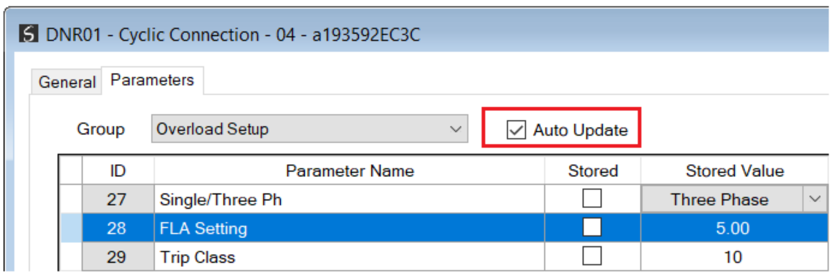


Figure 5.9 – Parameter Auto Update

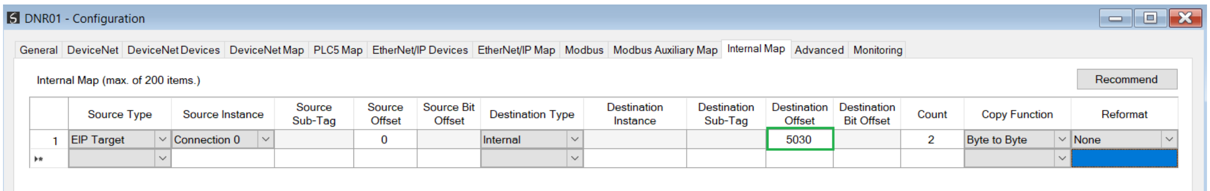
5.2.2. Explicit Messaging

When using the DeviceNet Explicit Messaging, the user can configure up to 63 DeviceNet devices which will be used for the Explicit Messaging. This configuration is located in the

DeviceNet Devices tab. Following this, the DeviceNet Map of explicit messages needs to be configured. The Explicit Messaging uses the internal data space (IDS) which is size where data can be stored for exchanges between the explicit DeviceNet devices and the Ethernet network.

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

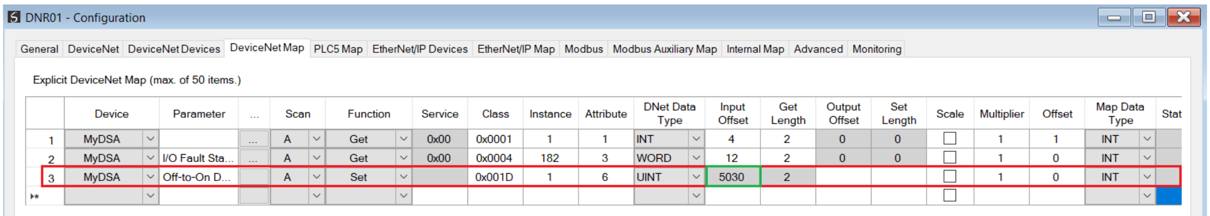
In the example below, the DeviceNet Router/B is receiving data from a Logix controller on EtherNet/IP, copying 2 bytes from the Class 1 EtherNet/IP Output assembly to the Internal Data Space at address 5030.



	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	EIP Target	Connection 0		0		Internal			5030		2	Byte to Byte	None

Figure 5.10 – Internal Mapping from EtherNet/IP to IDS

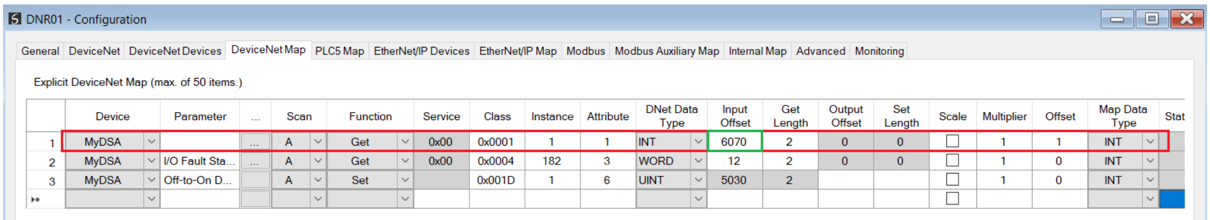
The DeviceNet Router/B then uses the data from the EtherNet/IP Output assembly (which was copied to IDS offset 5030) to execute a Set Single Attribute to a target DeviceNet device using explicit messaging.



	Device	Parameter	...	Scan	Function	Service	Class	Instance	Attribute	DNet Data Type	Input Offset	Get Length	Output Offset	Set Length	Scale	Multiplier	Offset	Map Data Type	Stat
1	MyDSA		...	A	Get	0x00	0x0001	1	1	INT	4	2	0	0	☐	1	1	INT	
2	MyDSA	I/O Fault Sta...	...	A	Get	0x00	0x0004	182	3	WORD	12	2	0	0	☐	1	0	INT	
3	MyDSA	Off-to-On D...		A	Set		0x001D	1	6	UINT	5030	2			☐	1	0	INT	

Figure 5.11 – Explicit Messaging – Set Function from IDS to DeviceNet device

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



	Device	Parameter	...	Scan	Function	Service	Class	Instance	Attribute	DNet Data Type	Input Offset	Get Length	Output Offset	Set Length	Scale	Multiplier	Offset	Map Data Type	Stat
1	MyDSA		...	A	Get	0x00	0x0001	1	1	INT	6070	2	0	0	☐	1	1	INT	
2	MyDSA	I/O Fault Sta...	...	A	Get	0x00	0x0004	182	3	WORD	12	2	0	0	☐	1	0	INT	
3	MyDSA	Off-to-On D...		A	Set		0x001D	1	6	UINT	5030	2			☐	1	0	INT	

Figure 5.12 – Explicit Messaging – Get Function from DeviceNet device to IDS

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

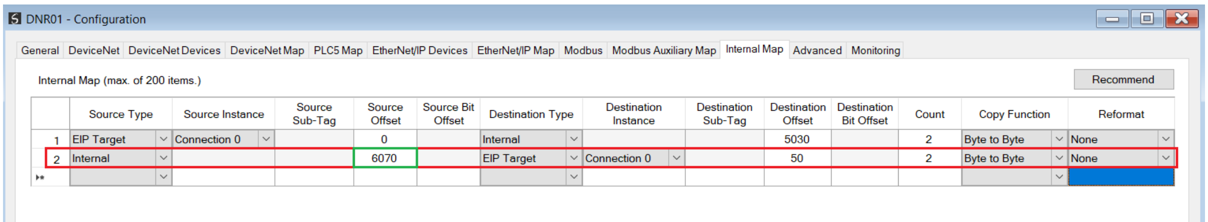


Figure 5.13 – Explicit Messaging – Internal Mapping from IDS to EtherNet/IP

5.2.3. Device Node Address and Baud Rate Assignment

The DeviceNet Router/B can scan the DeviceNet network to discover DeviceNet devices. This is done by going online with the module in Slate, and selecting the ***DeviceNet Discovery*** tab.

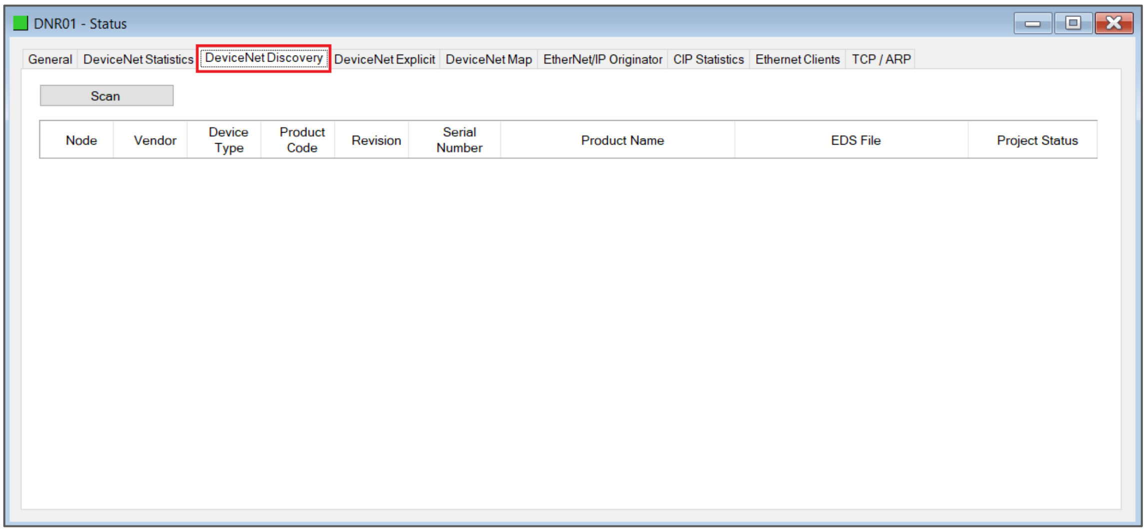


Figure 5.14 – DeviceNet Discovery

Once the ***Scan*** button is pressed, the module will start scanning the DeviceNet network for devices. If a device has been found it will be listed in the window and indicate the status of the device.

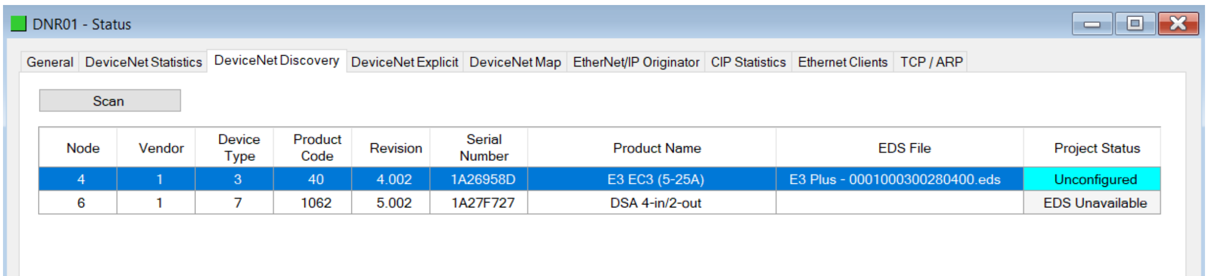


Figure 5.15 – DeviceNet Discovery – Found devices

For devices supporting this feature, the DeviceNet node number and/or DeviceNet Baud rate can be change for the selected DeviceNet device.

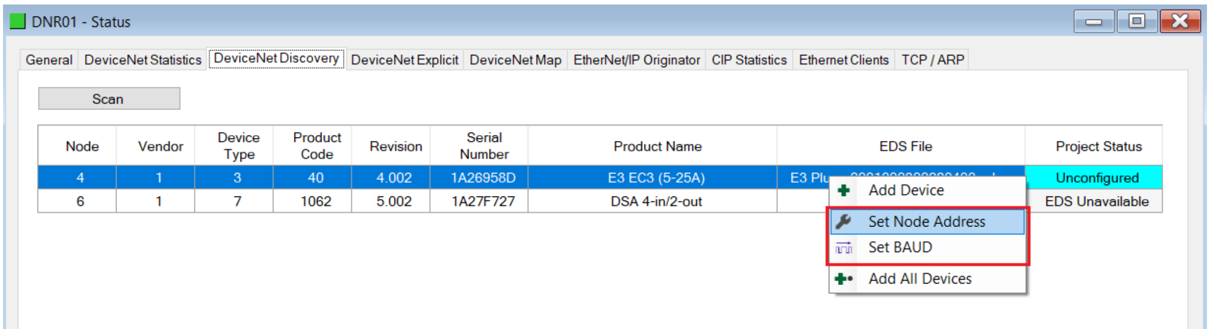


Figure 5.16 – DeviceNet Discovery – Set Node Address and BAUD

5.3. ETHERNET/IP TARGET

A controller (e.g. Logix controller) can own the DeviceNet Router/B over EtherNet/IP using up to 4 Class 1 EtherNet/IP connections when the DeviceNet Router/B is operating as an EtherNet/IP target. This will allow the DeviceNet Router/B to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connection.

NOTE: When using EtherNet/IP Target, it is recommended to use the ***Recommend*** button in the Internal Map configuration. This will automatically map and reformat all the required data in the Internal Map.

5.3.1. 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 DeviceNet Router/B. 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 bytes. 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 DeviceNet Router/B. The data of the various tags (System Status, Device Status, etc.) will be in the format as shown in section 5.8.

◢ DNR01In	{...}	{...}		DNR01In635A
◢ DNR01In.SystemStatus	{...}	{...}		DNetSystemStatus
DNR01In.SystemStatus.ConfigValid	0		Decimal	BOOL
DNR01In.SystemStatus.EIPOriginatorCommsOk	0		Decimal	BOOL
DNR01In.SystemStatus.ModbusOnline	0		Decimal	BOOL
DNR01In.SystemStatus.DnetOriginatorCommsOk	0		Decimal	BOOL
DNR01In.SystemStatus.EIPOwned	0		Decimal	BOOL
DNR01In.SystemStatus.DNetOwned	0		Decimal	BOOL
DNR01In.SystemStatus.PowerMainConnector	0		Decimal	BOOL
DNR01In.SystemStatus.PowerCANConnector	0		Decimal	BOOL
DNR01In.SystemStatus.DuplicateNodeNumber	0		Decimal	BOOL
DNR01In.SystemStatus.NTPOk	0		Decimal	BOOL
▸ DNR01In.SystemStatus.ConfigCRC	16#0000		Hex	INT
▸ DNR01In.SystemStatus.ActualBAUD	0		Decimal	SINT
▸ DNR01In.SystemStatus.ActualNode	0		Decimal	SINT
DNR01In.SystemStatus.DNetTargetPollActive	0		Decimal	BOOL
DNR01In.SystemStatus.DNetTargetNullPoll	0		Decimal	BOOL
DNR01In.SystemStatus.DNetTargetCOSActive	0		Decimal	BOOL

Figure 5.17 – Logix System Status Tag

◢ DNR01In.DSA42100DNY42R50Status	{...}	{...}		DNetDeviceStatus
DNR01In.DSA42100DNY42R50Status.Online	0		Decimal	BOOL
DNR01In.DSA42100DNY42R50Status.DataExchangeActive	0		Decimal	BOOL
DNR01In.DSA42100DNY42R50Status.DeviceMismatch	0		Decimal	BOOL
▸ DNR01In.DSA42100DNY42R50Status.Node	0		Decimal	SINT

Figure 5.18 – Logix Device Status Tag

5.3.2. Explicit Messaging

The DeviceNet Router/B allows the user to read or write data from and to the DeviceNet IO devices using explicit EtherNet/IP CIP messages. The required parameters for DeviceNet data extraction from a DeviceNet IO device are listed below.

5.3.2.1. DEVICENET PASSTHROUGH

A. CIP MESSAGE

Parameter	Description
Service Code	0x6B (Hex)
Class	0x436 (Hex)
Instance	1
Attribute	N/A

Request Data Length	16 - 512

Table 5.4 – DeviceNet Passthrough Message

B. REQUEST DATA

Parameter	Data Type	Description
Node	SINT	The Node Address of the target DeviceNet device
Reserved	SINT	Reserved for future use
Reserved	INT	Reserved for future use
Reserved	INT	Reserved for future use
CIP Service	INT	CIP service for the target DeviceNet device request.
CIP Class	INT	CIP Class for the target DeviceNet device request.
CIP Instance	INT	CIP Instance for the target DeviceNet device request.
CIP Attribute	INT	CIP Attribute for the target DeviceNet device request.
CIP Data Size	INT	The length of the CIP data payload to follow below.
CIP Data	SINT[]	The CIP data payload. Max 256 bytes.

Table 5.5 – DeviceNet Passthrough Request

C. RESPONSE DATA

Parameter	Data Type	Description
Response Status	INT	This is the status of the request. 0 – Success 1 – Timeout
CIP Service Response	SINT	CIP response service from the target DeviceNet device. If a CIP response service of 0x94 is received, then the target DeviceNet device returned an error. The error information will be in the DeviceNet Message Status.
DeviceNet Message Status	SINT	DeviceNet Message status. This value will be zero if the message was successful or provide the error information if the DeviceNet device returned an error. See the CIP Response Status Codes in the appendix
CIP Data Response Size	INT	The size of the CIP response data to follow.

Reserved	INT	Reserved for future use.
CIP Response Data	SINT[]	The CIP response data from the target DeviceNet device.

Table 5.6 – DeviceNet Passthrough Response

5.4. ETHERNET/IP ORIGINATOR

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

5.4.1. ETHERNET/IP CLASS 1 CONNECTIONS

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

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

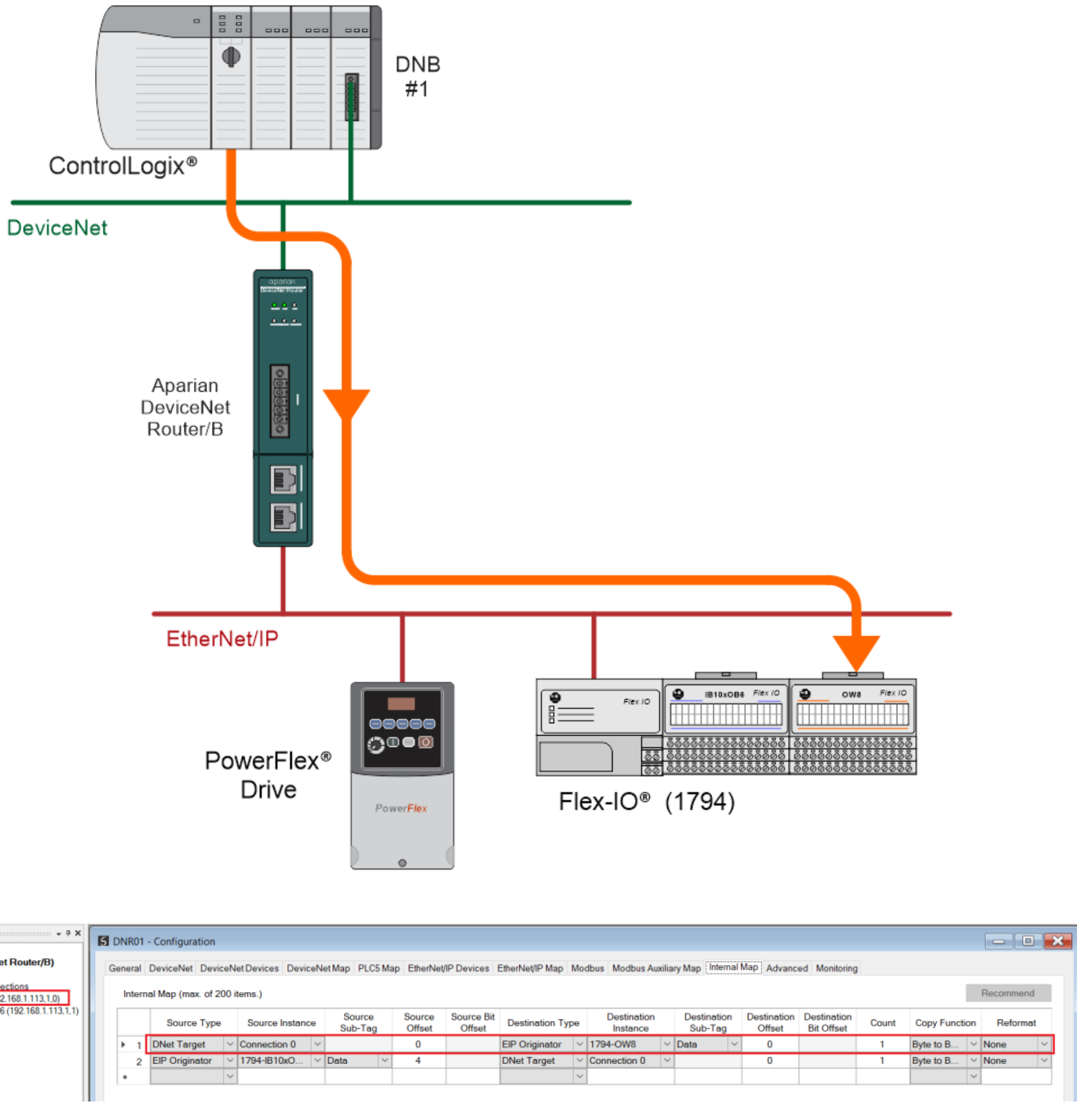


Figure 5.19 – Internal Mapping from DeviceNet to EtherNet/IP Originator

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

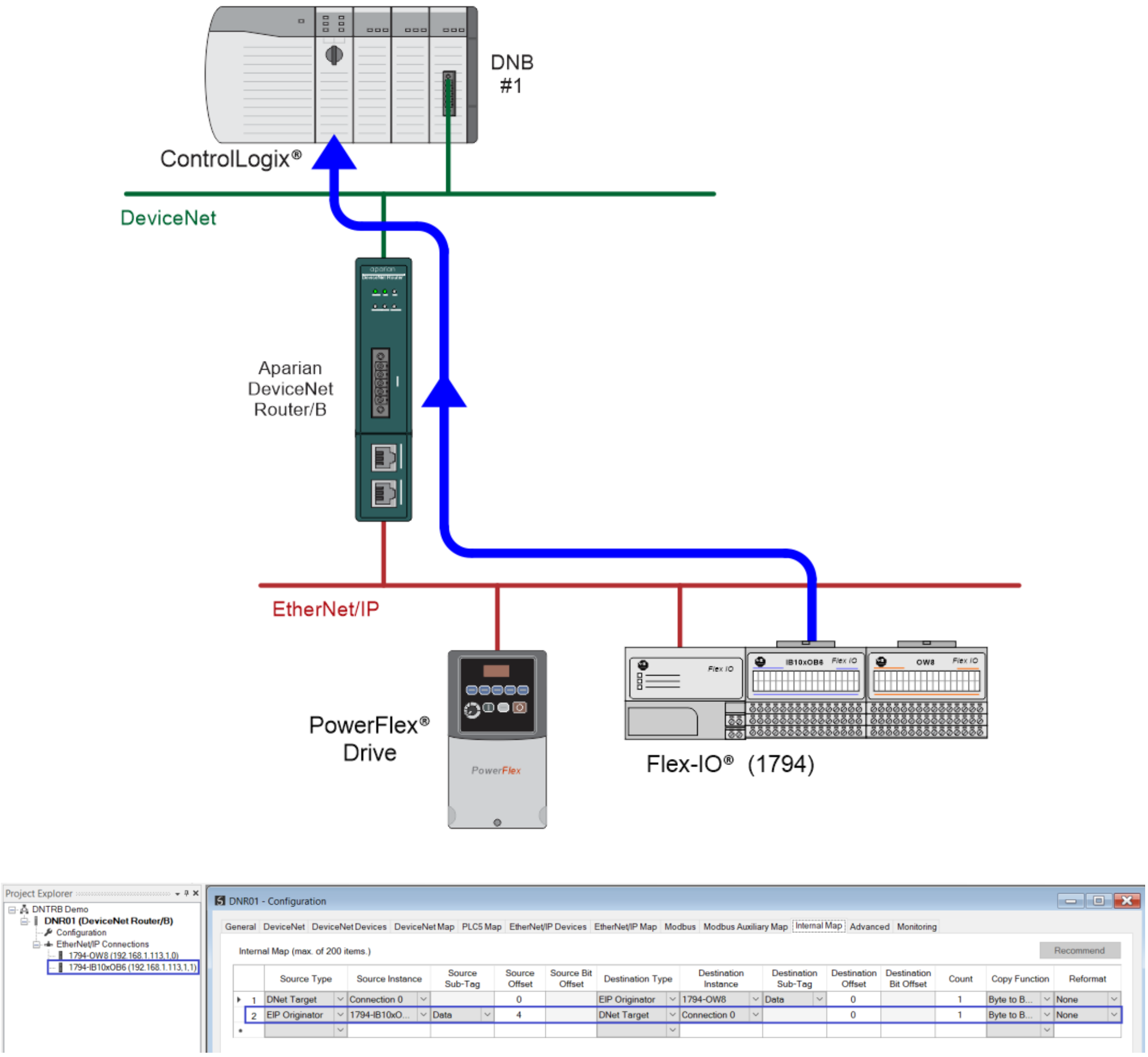


Figure 5.20 – Internal Mapping from EtherNet/IP Originator to DeviceNet

5.4.1.1. CONNECTION STATUS

The user can select to copy the data from the EtherNet/IP connection or the status.

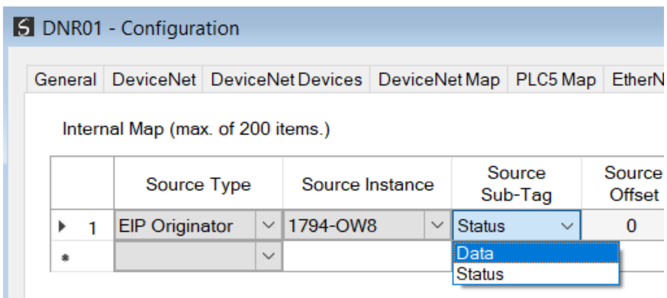


Figure 5.21 – IDS Copy – EtherNet/IP Originator Status

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

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

Table 5.7 – EtherNet/IP Originator Connection Status

5.4.2. Explicit Messaging

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

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

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

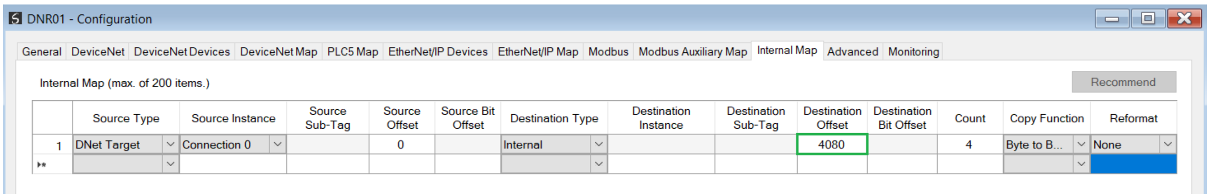


Figure 5.22 – Internal Mapping from DeviceNet to IDS

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

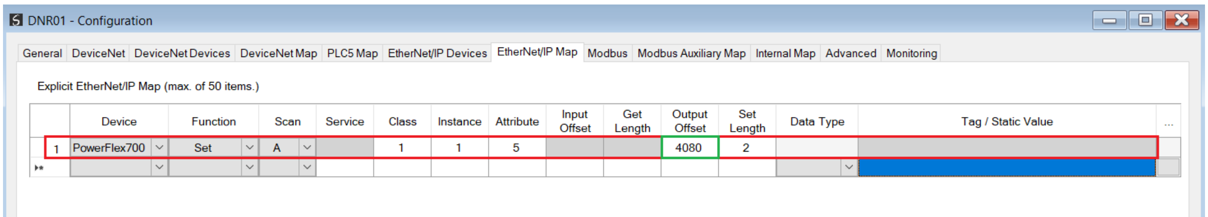


Figure 5.23 – Explicit Messaging – Set Function from IDS to EtherNet/IP device

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

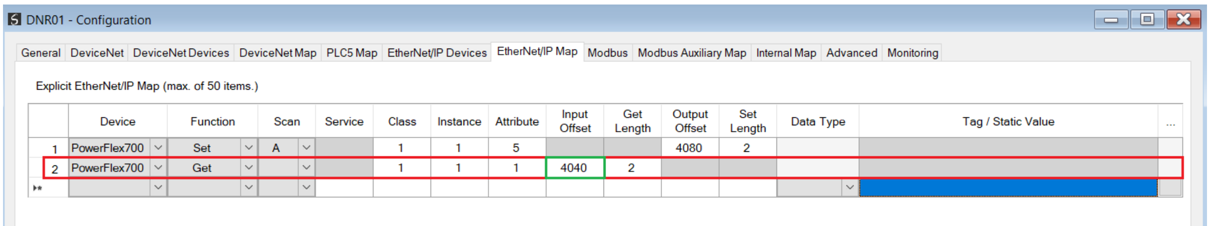


Figure 5.24 – Explicit Messaging – Get Function from EtherNet/IP device to IDS

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

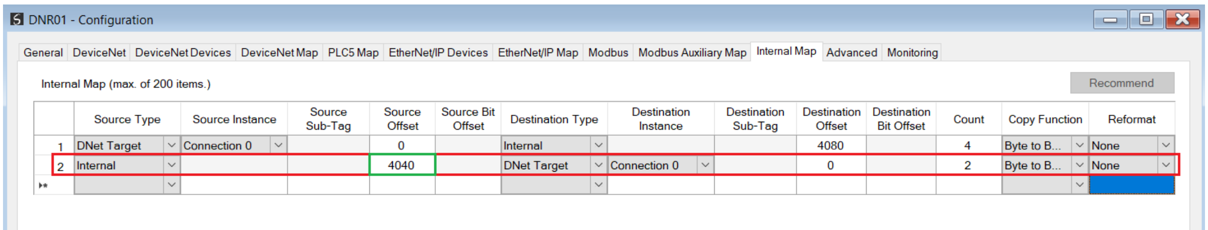


Figure 5.25 – Explicit Messaging – Internal Mapping from IDS to DeviceNet

5.5. MODBUS CLIENT

When the DeviceNet Router/B has the Primary Interface set to Modbus Client, then the DeviceNet data can be mapped to and from configurable internal Modbus Registers using the Internal Map.

The internal Modbus Registers are then asynchronously exchanged with Modbus devices as configured in the Modbus Auxiliary Map. In this mapping the user can exchange (read or write) data between the internal Modbus Registers and a remote Modbus device on Modbus TCP, RTU232, or RTU485.

In the example below the DeviceNet Router/B with the Primary Interface set to Modbus Client will read multiple Modbus Holding Registers from a Modbus Server device and then map the received data to a DeviceNet Input assembly where the module DeviceNet Mode is set to Target.

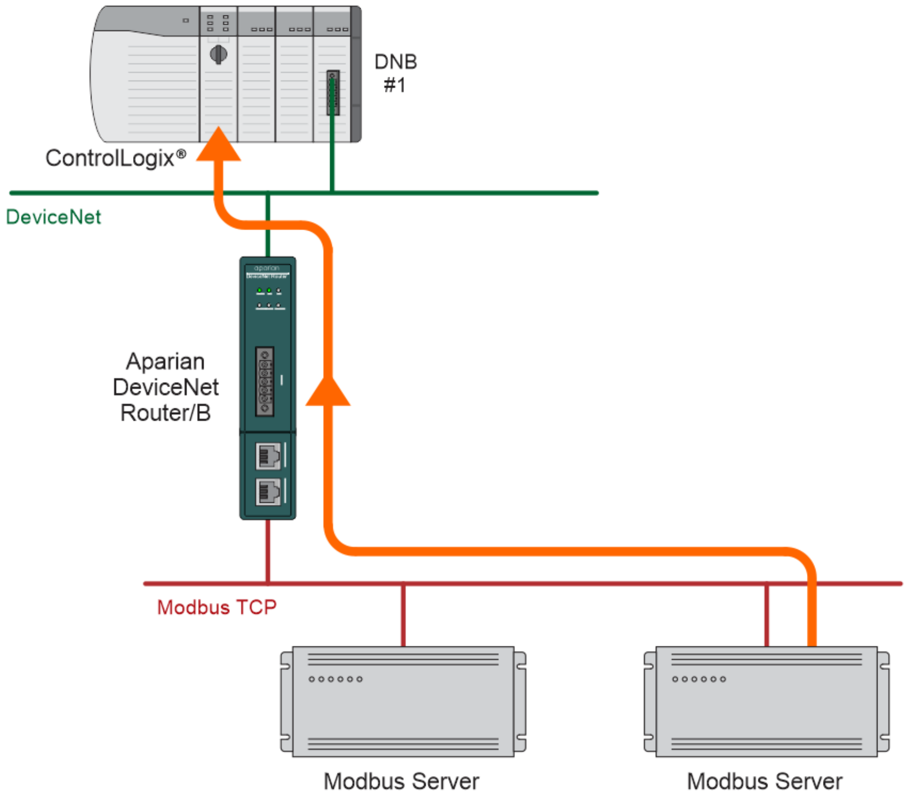
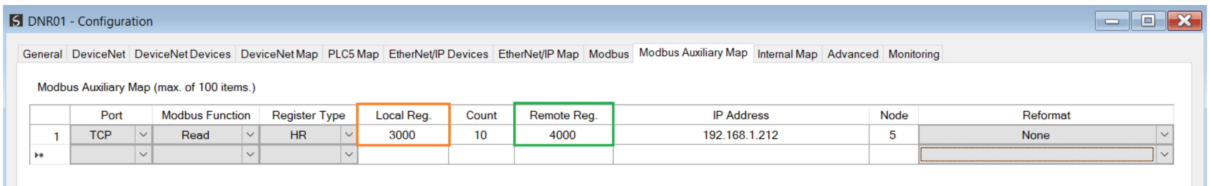


Figure 5.26 – Modbus Client to DeviceNet operation

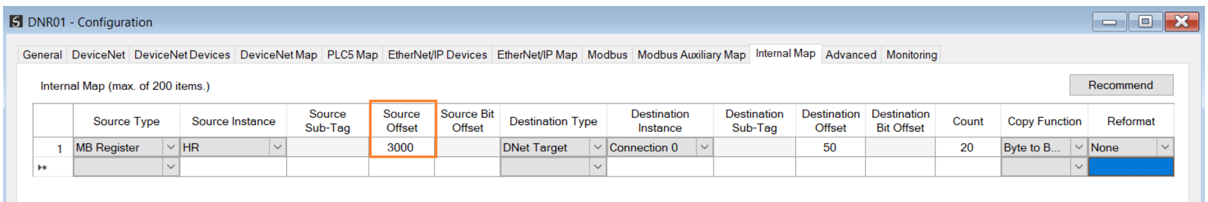
For this example the user will configure the Modbus Auxiliary Map to read data from a Modbus Server device on Ethernet. The DeviceNet Router/B will request data from Modbus Holding Register 4000 (from the external Modbus Server) and write it to the module's internal Modbus Holding Register 3000.



	Port	Modbus Function	Register Type	Local Reg.	Count	Remote Reg.	IP Address	Node	Reformat
1	TCP	Read	HR	3000	10	4000	192.168.1.212	5	None

Figure 5.27 – Modbus Client Aux Mapping

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



	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	MB Register	HR		3000		DNet Target	Connection 0		50		20	Byte to B...	None

Figure 5.28 – Internal Mapping from Modbus Register to DeviceNet

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

5.6. MODBUS SERVER

When the DeviceNet Router/B has the Primary Interface set to Modbus Server, then the DeviceNet data can be mapped to and from configurable internal Modbus Registers and offsets using the Internal Map.

The internal Modbus Registers can then be asynchronously exchanged with a remote Modbus Client on Modbus TCP, RTU232, or RTU485. The remote Modbus Client can read or write to the configured Modbus addresses to access the DeviceNet data that has been mapped to the Modbus Registers.

In the example below the DeviceNet Router/B, with the Primary Interface set to Modbus Server, will have multiple Modbus Holding Registers written from a Modbus TCP Client and then map the received data to DeviceNet IO that is owned with a cyclic DeviceNet connection (when DeviceNet mode is Originator).

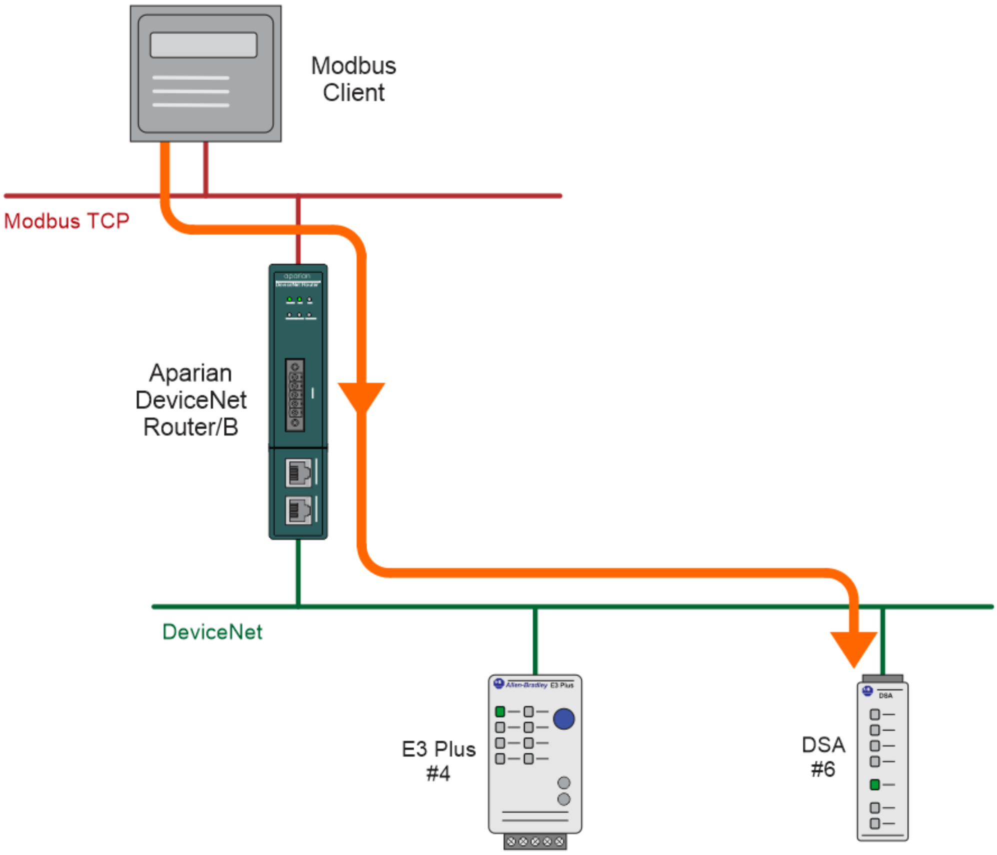


Figure 5.29 – Modbus TCP Client to DeviceNet IO operation

For this example the remote Modbus TCP Client will write data to Modbus Holding Register 4000 in the DeviceNet Router/B. The DeviceNet Router/B will map the received Modbus data (at Holding Register 4000) to the output assembly of DeviceNet IO (DSA) being owned (using cyclic DeviceNet) by the DeviceNet Router/B.

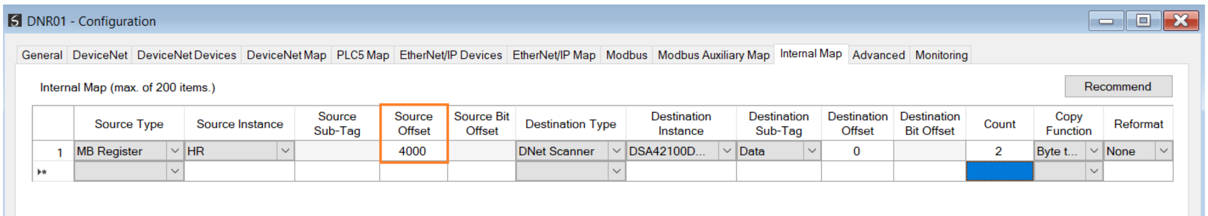


Figure 5.30 – Internal Mapping from Modbus Register to DeviceNet IO

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

5.7. FTVIEW / PANELVIEW INTERFACING

The DeviceNet Router/B can interface an Ethernet-only PanelView with a Logix controller over DeviceNet. The data being exchanged can then be accessed from a PanelView HMI by using PLC5 emulation.

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

The PLC5 Produced and Consumed File numbers are configured in the *PLC5Map* tab in the Slate configuration.

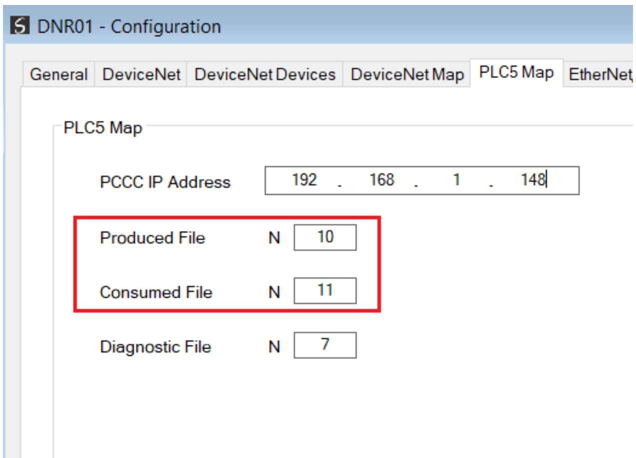


Figure 5.31 – PLC File configuration

In the configuration example above, the data being read from PLC File N11 will be the data received from the DeviceNet Scanner module (e.g., 1756-DNB output assembly) while the data being written to PLC File N10 is the data that will be sent to the DeviceNet Scanner module (e.g., 1756-DNB input assembly).

5.7.1. PanelView Reading Data From Logix

Using the example configuration in the image above, the PanelView will read N11. The data being written from the Logix Controller to the Output Assembly of the 1756-DNB (which has the DeviceNet Router/B mapped) will be copied to file N11. Below is a diagram of the PanelView reading data from the Logix Controller over cyclic DeviceNet using the PLC5 emulation.

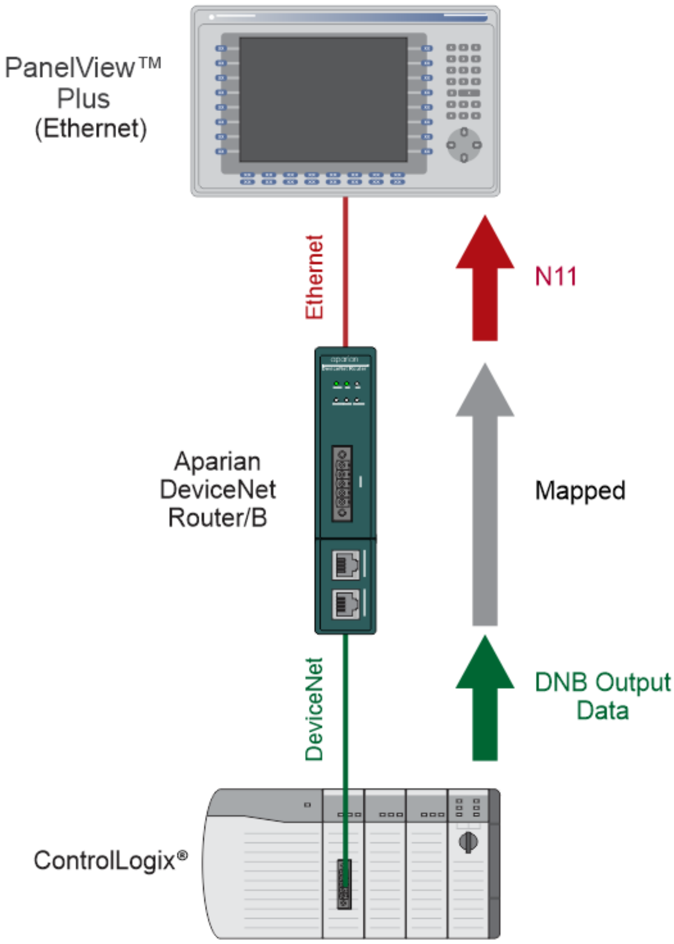


Figure 5.32 – PanelView reading Logix data over Cyclic DeviceNet

5.7.2. PanelView Writing Data To Logix

The PanelView will write to N10 when sending values to Logix. The data being read by the Logix Controller from the Input Assembly of the 1756-DNB which will be copied from file N10. Below is a diagram of the PanelView writing data to the Logix Controller over Cyclic DeviceNet using the PLC5 emulation.

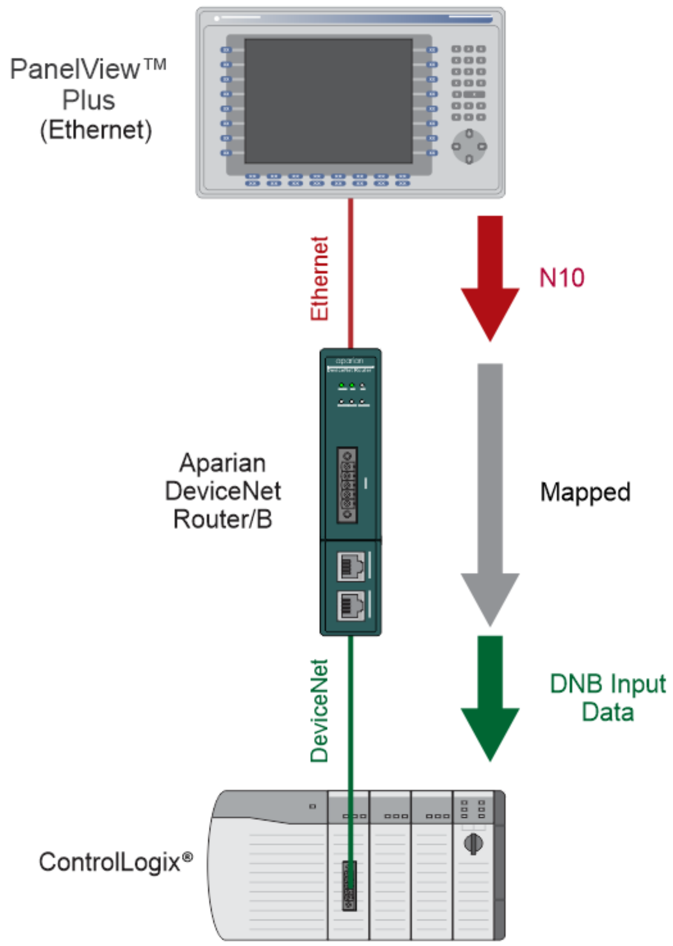


Figure 5.33 – PanelView writing Logix data over Cyclic DeviceNet

5.7.3. PanelView Reading Diagnostic Data from DeviceNet Router

The PanelView will read N7 when accessing diagnostics information from the DeviceNet Router/B. The data will be read similar to the above section where the PanelView will read the Logix data using the PLC5 driver emulation.

Offset	Group	Description
0	DeviceNet	DeviceNet Polling Status Bit 0 – Connection Active - Poll Bit 1 – Connection Standby Bit 2 – Connection Active - COS

1		Rx Can Packet Count
2		Tx Can Packet Count
3		CAN CRC Errors
4		CAN Bit Errors
5		Can Stuff Errors
6		UCCM Connection Open
7		UCCM Connection Close
8		IO Connections
9		Poll Commands
10		Fragment Ack Errors
11		Explicit Fragment Error
12		Poll Fragment Error
13		Explicit Client Not Found
14		Duplicate Node Detected
15	PCCC - Ethernet	PCCC Connection Requests
16		PCCC Read Requests
17		PCCC Write Requests
18		PCCC Unsupported Command
19		PCCC Unsupported FNC Code
20		PCCC Client Not Found
21		PCCC Client Max Reached
22		PCCC File Not Found
23		Current Connections
24	Module	DeviceNet Router/B Internal Temperature

Table 5.8 - Diagnostic File

5.8. INTERNAL MAP DATA FORMATS

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

NOTE: When using EtherNet/IP Target, Modbus Server, or Modbus Client, it is recommended to use the Recommended Mapping feature in the internal mapping. This will automatically map all the required data in the Internal Map.

5.8.1. SYSTEM STATUS

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

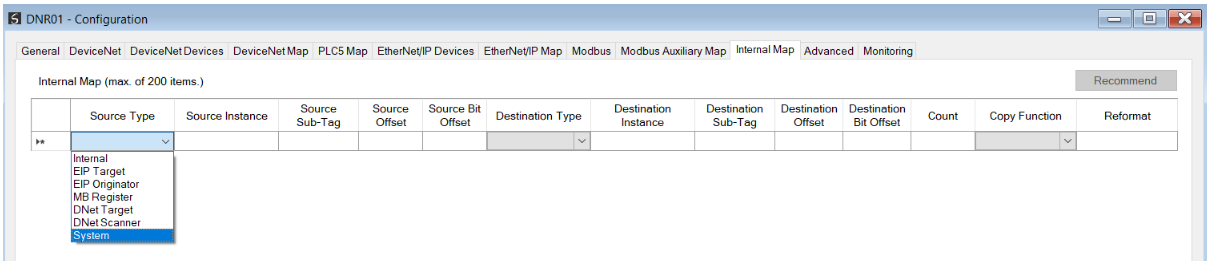


Figure 5.34 – IDS Copy – System Information

The module's system information has the following format.

Parameter	Data Type	Description
Status	INT	Module Status. Bit 0 – Module Config Valid Bit 1 – EtherNet/IP Originator Comms Ok Bit 2 – Modbus Comms Ok Bit 3 – DeviceNet Scanner Comms Ok Bit 4 – EtherNet/IP Target Comms Ok Bit 5 – DeviceNet Target Comms Ok Bit 6 – Power is connected to the bottom connector Bit 7 – Power is connected to the front connector. Bit 8 – Duplicate Node Number Bit 9 – NTP Ok
ConfigCRC	INT	The module configuration signature.
Actual BAUD	SINT	Current BAUD rate. 0 – 125K 1 – 250K 2 – 500K
Actual Node	SINT	Current DeviceNet node number
DeviceNet Target Status	INT	Module status when operating as a DeviceNet Target. Bit 0 – DeviceNet Poll Connection Active Bit 1 – DeviceNet NULL Connection Active Bit 2 – DeviceNet Change-Of-State (COS) Connection Active
Reserved	SINT[6]	Reserved for future use.

Table 5.9 – System Information Format

5.8.2. DeviceNet IO Device Status

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

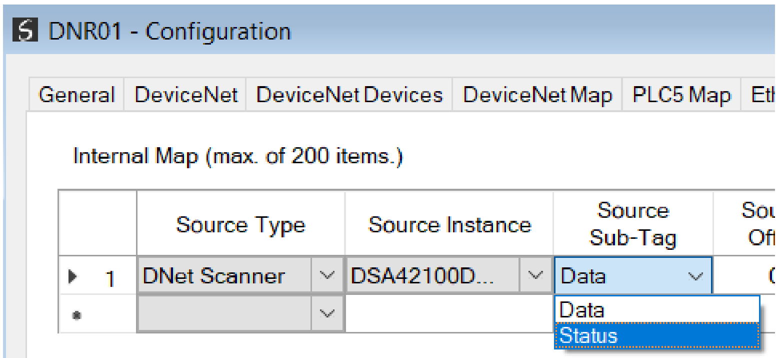


Figure 5.35 – IDS Copy – EtherNet/IP Originator Status

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

Parameter	Data Type	Description
DeviceNet Scanner Connection Status	DINT	Bit 0 – Online Bit 1 – Cyclic Data Exchange Ok Bit 2 – Device Mismatch
Node	SINT	The target DeviceNet device Node number.
Reserved	SINT	Reserved for future use.
Cyclic Communication Timeout Count	DINT	Number of times the cyclic connection has gone offline.
Cyclic Communication Tx Count	DINT	Number of cyclic connection bytes sent.
Cyclic Communication Rx Count	DINT	Number of cyclic connection byte received.

Table 5.10 – DeviceNet Scanner Connection Status

5.8.1. EtherNet/IP IO Device Status

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

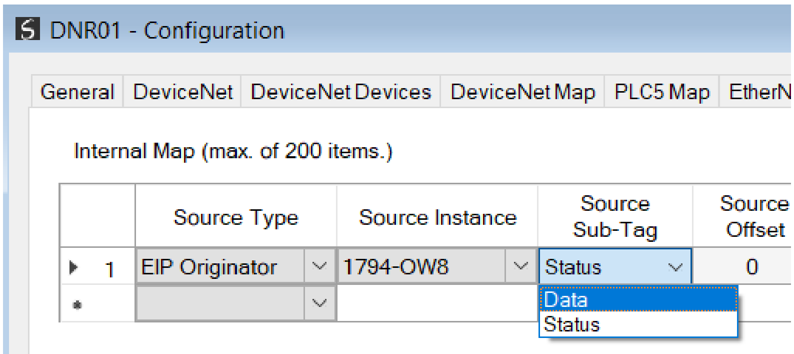


Figure 5.36 – IDS Copy – EtherNet/IP Originator Status

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

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

Table 5.11 – EtherNet/IP Originator Connection Status

6. DIAGNOSTICS

6.1. LEDS

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

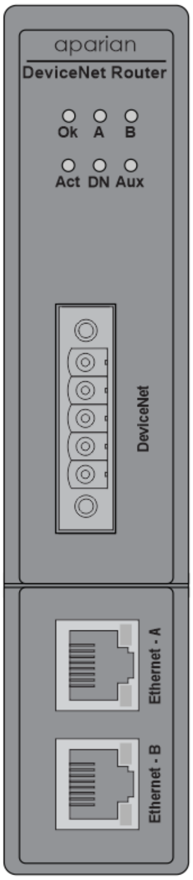


Figure 6.1 – DeviceNet Router/B front view

LED	Description
Ok	The module LED will provide information regarding the system-level operation of the module. If the LED is **red**, then the module is not operating correctly. For example, if the module application firmware has been corrupted or there is a hardware fault the module will have a red Module LED. If the LED is briefly **flashing red**, and then returning to either flashing green or solid green, then there is a duplicate IP address on the Ethernet network similar to that of the local module.

	If the LED is **green (flashing)**, then the module has booted and is running correctly **without** any application configuration loaded. If the LED is **green (solid)**, then the module has booted and is running correctly **with** application configuration loaded.
A / B	The Ethernet LED will light up when an Ethernet link has been detected (by plugging in a connected Ethernet cable). The LED will flash every time traffic was detected. This module has two Ethernet ports A and B. Each LEDs represents each specific port.
Act	The Act LED indicates if the module is operating as a DeviceNet target or DeviceNet Scanner. Solid Green – The local DeviceNet Router/B is operating as a DeviceNet Scanner. Off - The local DeviceNet Router/B is operating as a DeviceNet Target.
DN	The DeviceNet LED indicates the activity on the DeviceNet network. Flashing Red – A corrupted or incorrect DeviceNet packet was received. Flashing Green – A valid DeviceNet packet was received. Off – No DeviceNet packets are being received.
Aux	The Aux LED will flash each time there was activity on any of the primary interfaces. Flashing Red – A corrupted or incorrect packet was received on one of the Primary Interfaces (EtherNet/IP, Modbus TCP/RTU232/RTU485). Flashing Green – A valid packet was received on one of the Primary Interfaces (EtherNet/IP, Modbus TCP/RTU232/RTU485). Off – No activity.

Table 6.1 - Module LED operation

6.2. MODULE STATUS MONITORING IN SLATE

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

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

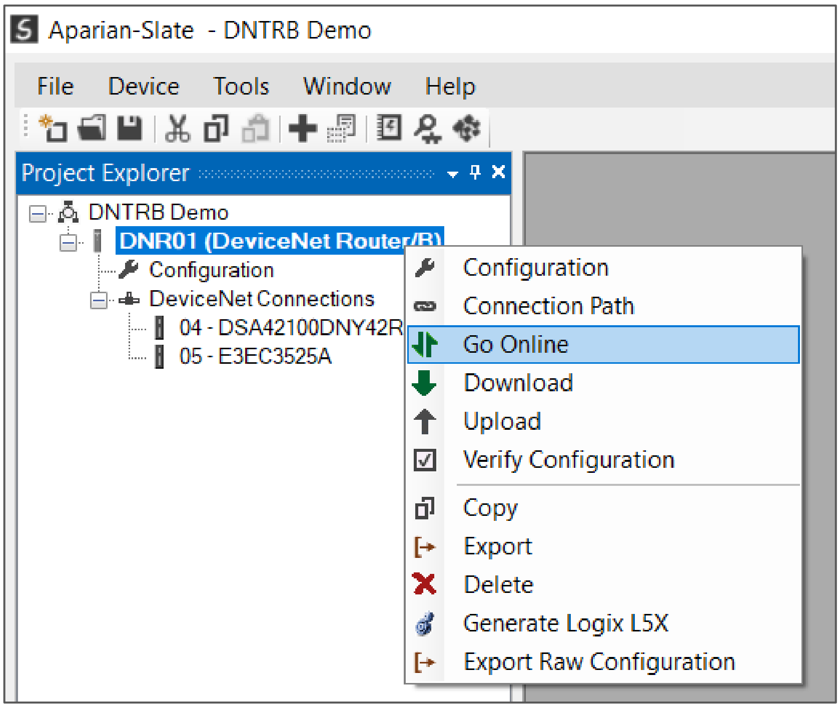


Figure 6.2 - Selecting to Go Online

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

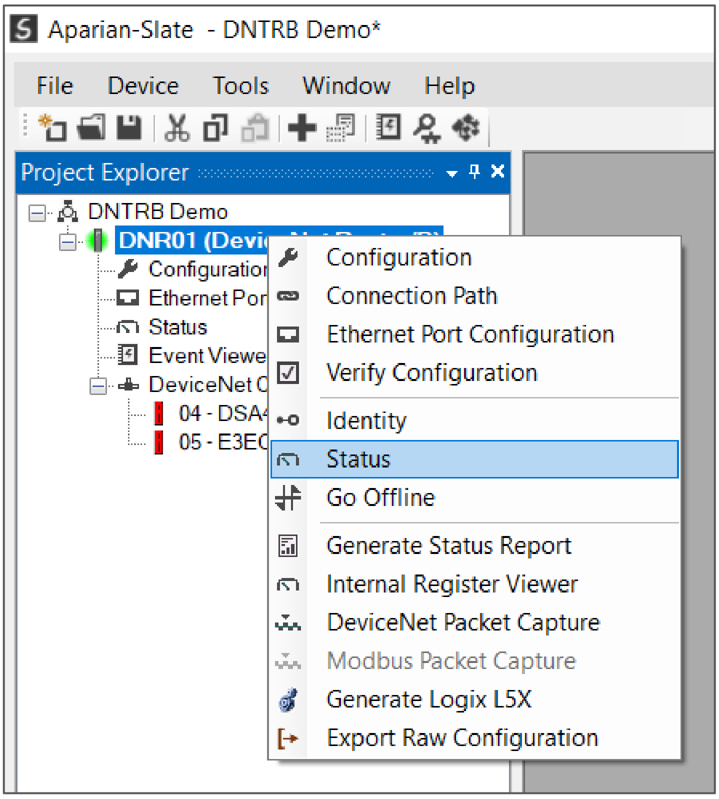


Figure 6.3 - Selecting online Status

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

6.2.1. General

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

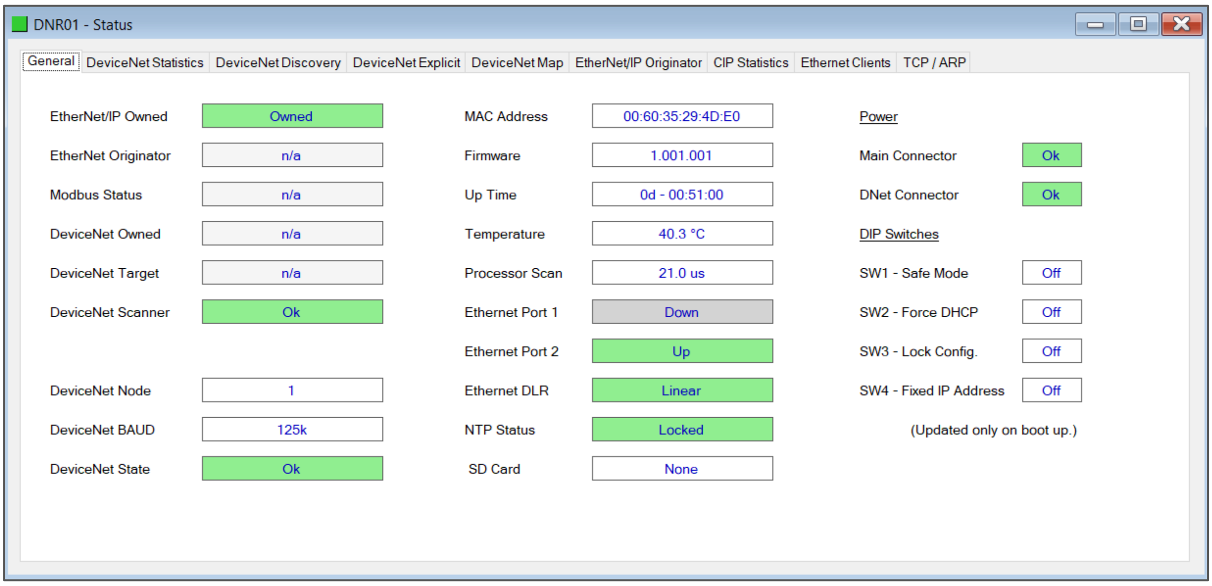


Figure 6.4 - Status monitoring – General

The General tab displays the following general parameters:

Parameter	**Description**
EtherNet/IP Owned	When the module is configured as an EtherNet/IP Target, this will indicate if the module is owned by an EtherNet/IP connection originator.
EtherNet/IP Originator	When the module is configured as an EtherNet/IP Originator, this will show if all the Class 1 and Explicit Message connections to EtherNet/IP target devices are established and returning valid data.
Modbus Status	When the module is operating as a **Modbus Server**, this parameter will indicate that the module has received a valid Modbus request within the Modbus inactivity time. When the module is operating as a **Modbus Client**, this parameter will indicate that all the mapping items in the Modbus Auxiliary Map are executing correctly.

DeviceNet Owned	When the module is configured as a DeviceNet Target, this will indicate if the module is owned by a DeviceNet connection originator.
DeviceNet Target	When the module is configured as a DeviceNet Target, this will show the remote scanner's connection state: **Not Connected** **Poll (Null)**: Connected (e.g. remote in Program mode) **Poll Active**: Connected - Active Polling **COS Active**: Connected – Change of State
DeviceNet Scanner	When the module is configured as a DeviceNet Scanner, this will show if all the Cyclic and Unscheduled Explicit Message connections to DeviceNet target devices are established and returning valid data.
DeviceNet Node	The current DeviceNet node number.
DeviceNet BAUD	The BAUD rate of the DeviceNet network. **125k** **250k** **500k**
DeviceNet State	Indicates whether or not a Duplicate DeviceNet node has been detected.
MAC Address	Displays the module's unique Ethernet MAC address.
Firmware	The version of the module's firmware.
Up Time	Indicates the elapsed time since the module was powered-up.
Temperature	The internal temperature of the module.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
Ethernet Port 1/2	This is the status of each Ethernet port. **Down** The Ethernet connector has **not been** successfully connected to an Ethernet network. **Up** The Ethernet connector has successfully connected to an Ethernet network. **Mirror Enabled** The Ethernet port is mirroring the traffic on the other Ethernet port.
Ethernet DLR (Device Level Ring)	The status of the Ethernet DLR. **Disabled** Device Level Ring functionality has been disabled.

	Linear The DLR functionality has been enabled and the Ethernet network architecture is linear. **Ring – Fault** The DLR functionality has been enabled and the Ethernet network architecture is ring, but there is a fault with the network. **Ring – Ok** The DLR functionality has been enabled and the Ethernet network architecture is ring and is operating as expected.
NTP Status	The status of the local NTP Client. **Disabled** The NTP time synchronization has been disabled. **Locked** NTP time synchronization has been enabled and the module has locked onto the target time server. **Not Locked** NTP time synchronization has been enabled and the module has not locked onto the target time server.
SD Card	Indicates if a SD Card is present or not.
Power	Indication from which port the module is receiving power. **Main Connector** The power is present at the bottom connector. **CAN Connector** The power is present at the CAN connector.
DIP Switch Position	The status of the DIP switches when the module booted. **NOTE**: This status will not change if the DIP switches are altered when the module is running.

Table 6.2 - Parameters displayed in the Status Monitoring – General Tab

6.2.2. DeviceNet Statistics

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

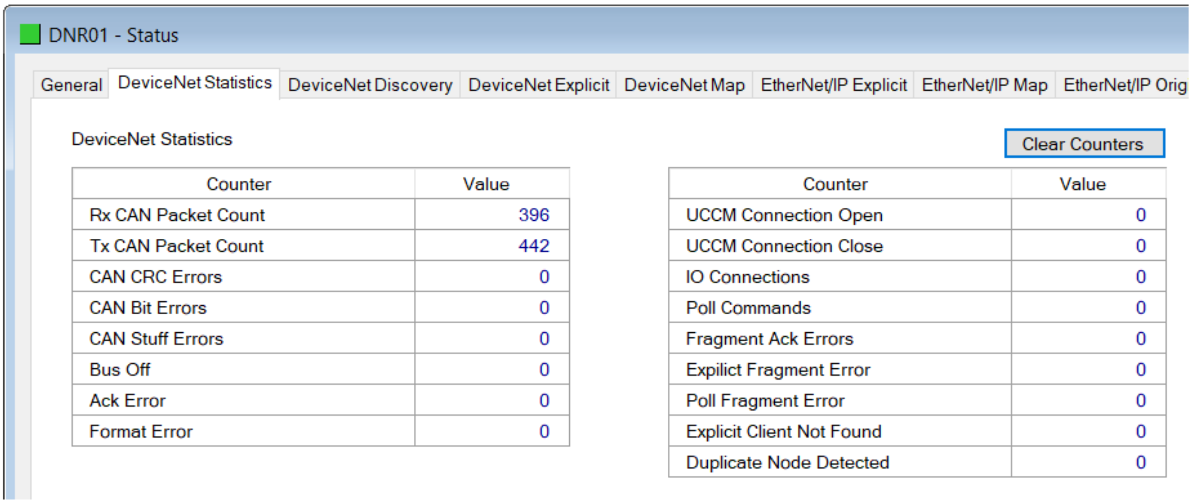


Figure 6.5 - Status monitoring – DeviceNet Statistics

Statistic	Description
Rx CAN Packet Count	The number of DeviceNet packets received.
Tx CAN Packet Count	The number of DeviceNet packets sent.
CAN CRC Errors	The number of received packets where the packet checksum does not match the calculated packet checksum. This implies one or more bits in the frame have been corrupted. May indicate an intermittent CAN cable connection or induced electrical noise.
CAN Bit Errors	The number of transmitted bits where the transmitted bit state does not match the instantaneous read-back state. This may indicate that another device is transmitting at the same time, or one of the CAN lines shorted to power, shorted together, or incorrectly termination.
CAN Stuff Errors	The number of frames received where the required inserted (opposite) bit was not received after 5 identical bits. This may be an indication of bus noise, bad physical cable connection, or a faulty device.
Bus Off	The number of Bus-Off Events. A node will enter the Bus-Off state when the transmit Error Count exceeds a certain threshold (typically 125). This may indicate a cable break or loss of power causing the Scanner to enter the Bus-Off state.

Ack Error	The number of transmitted bits that are not read-back and acknowledged by at least one other node. Typically seen when the device is alone on the bus, and there is no other node to acknowledge the frame.
Format Error	The number of received frames where the fixed format part of a received frame is invalid, or the Frame structure is non-standard. (Frame size/type start delimiter etc.) May indicate an intermittent CAN cable connection, induced electrical noise or a node present with an incorrect BAUD rate.
UCCM Connection Open	The number of Unconnected Connection allocations received.
UCCM Connection Close	The number of Unconnected Connection releases received.
IO Connections	The number of concurrent IO connections.
Poll Commands	The number of Poll commands sent / received.
Fragment Ack Errors	The number of fragmented multi-packet acknowledge errors.
Explicit Fragment Error	The number of fragmented multi-packet count errors for explicit messages.
Poll Fragment Error	The number of fragmented multi-packet count errors for poll messages.
Explicit Client Not Found	The number of times a request has been received with no connection allocated.
Duplicate Node Detected	The number of times a duplicate node was detected.

Table 6.3 – DeviceNet statistics

6.2.3. DeviceNet Explicit

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

NOTE: This tab is only applicable when the module has the DeviceNet mode set to Scanner.

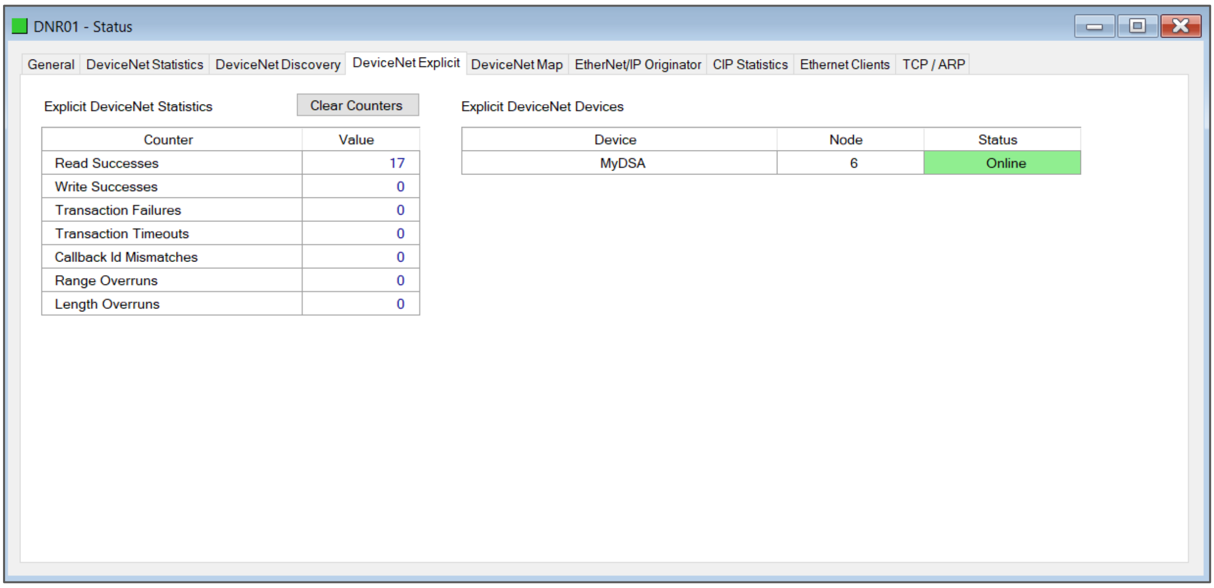


Figure 6.6 - Status monitoring – DeviceNet Explicit Statistics

Statistic	Description
Read Successes	The number of successful reads from the target DeviceNet device.
Write Successes	The number of successful writes to the target DeviceNet device.
Transaction Failures	The number of failed reads/writes to the target DeviceNet device (e.g. error response).
Transaction Timeouts	The number of times the target DeviceNet device failed to respond.
Callback Id Mismatches	The DeviceNet Unscheduled UCMM response does not match the request.
Range Overruns	The number of times the returned data amount runs over the max Internal Data Space.
Length Overruns	The number of times the returned data is greater than the configured Get Length.

Table 6.4 – DeviceNet Explicit Statistics

6.2.4. DeviceNet Map

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

NOTE: This tab is only relevant when the module has the DeviceNet mode set to Scanner.

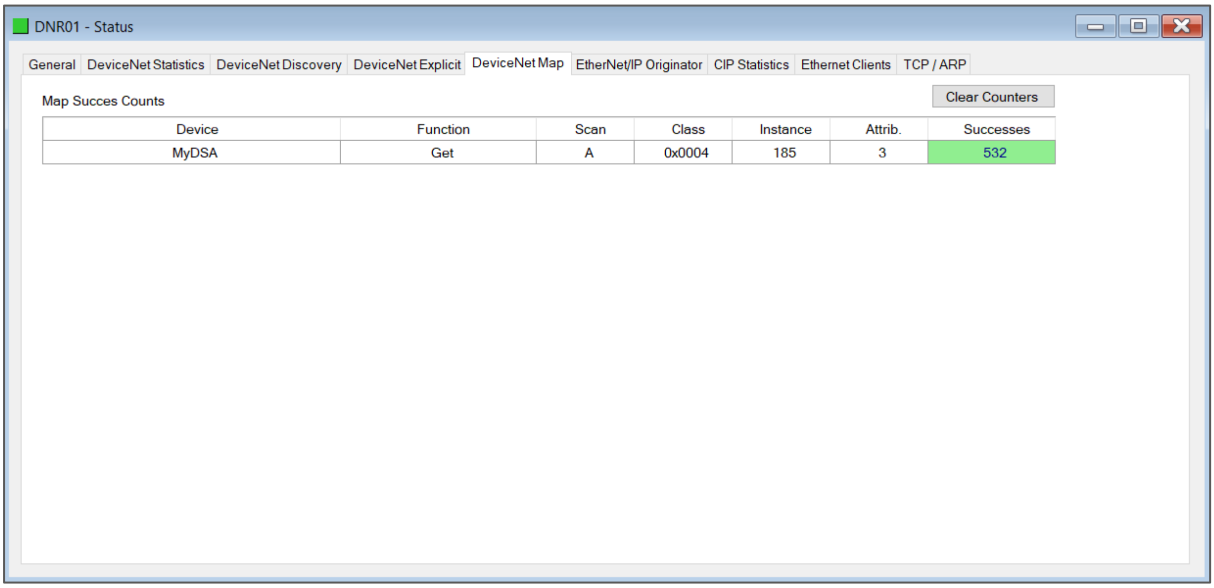


Figure 6.7 - Status monitoring – DeviceNet Map

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.

6.2.5. PCCC Statistics

The PCCC tab displays the Ethernet PCCC statistics.

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

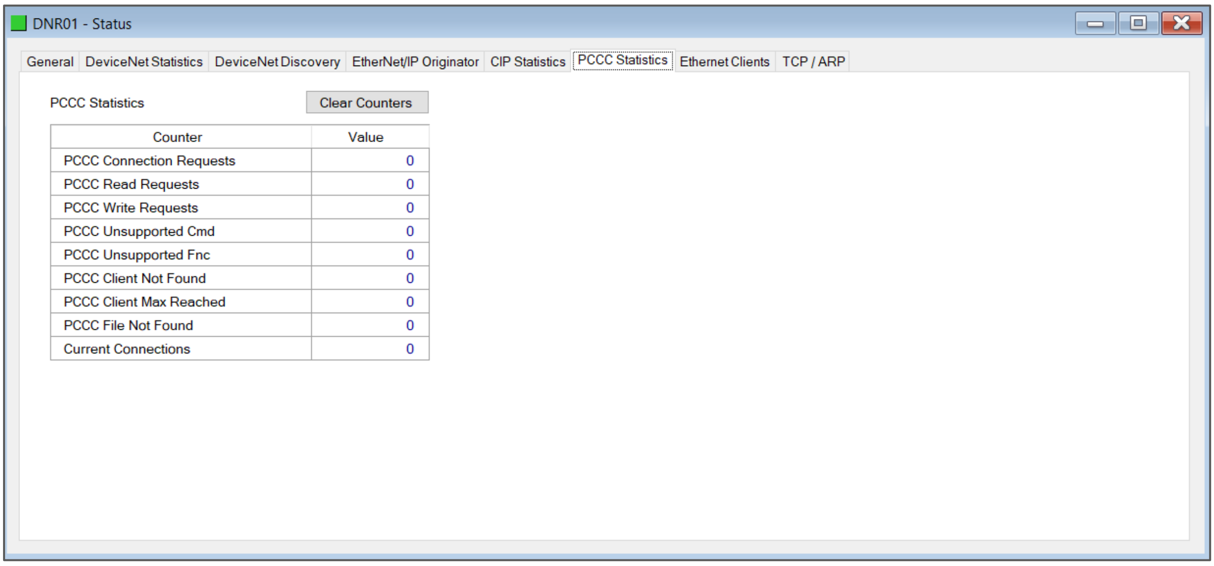


Figure 6.8 - Status monitoring – PCCC Statistics

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

Table 6.5 – PCCC statistics

6.2.6. EtherNet/IP Explicit

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

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

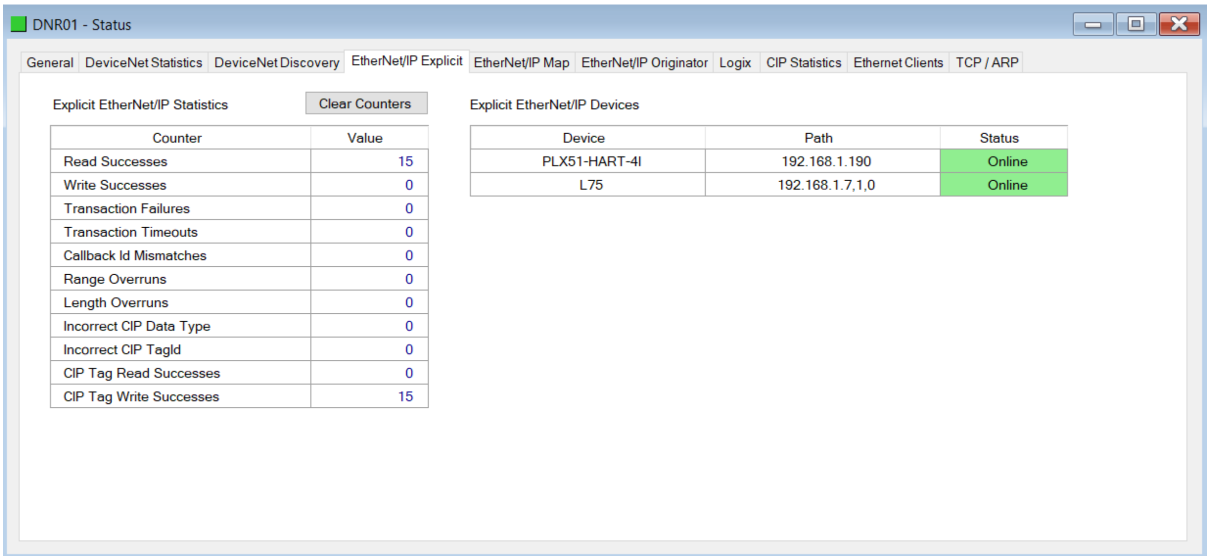


Figure 6.9 - Status monitoring – EtherNet/IP Explicit

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

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

Table 6.6 – EtherNet/IP Explicit Statistics

6.2.7. 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.

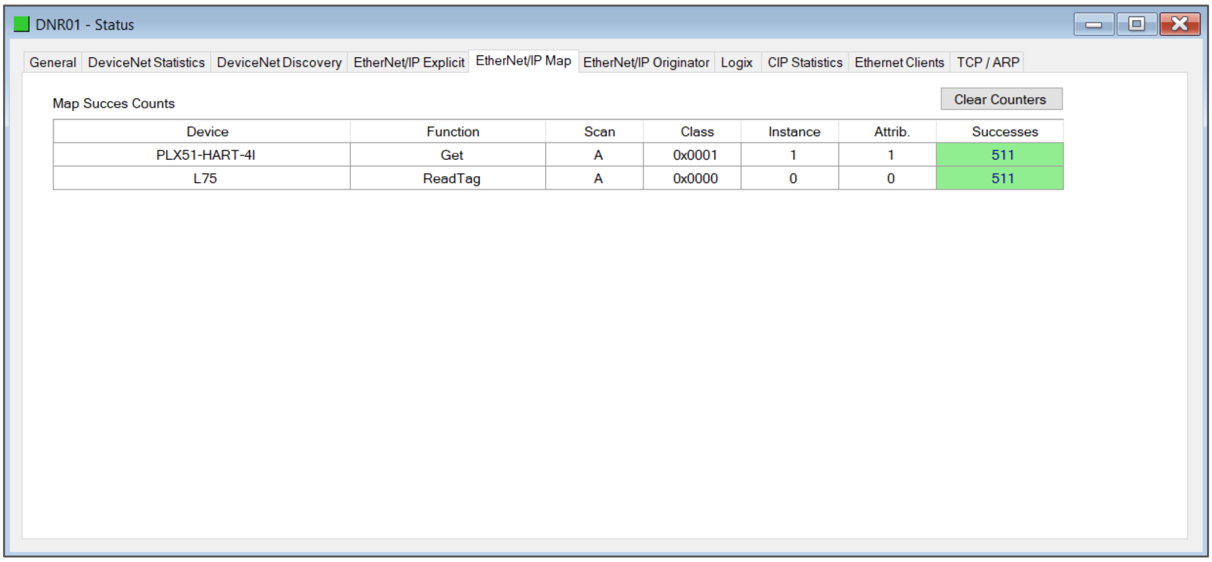


Figure 6.10 - Status monitoring – EtherNet/IP Map

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.

6.2.8. 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.

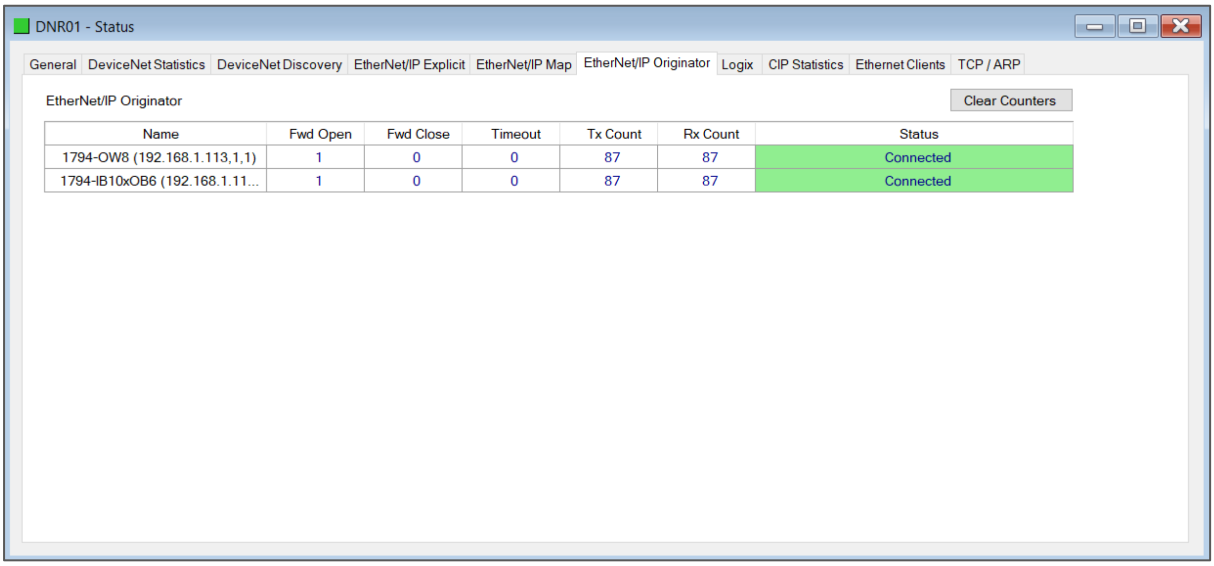


Figure 6.11 - Status monitoring – EtherNet/IP Originator

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

Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.

Table 6.7 – EtherNet/IP Class 1 status and statistics

6.2.9. 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.

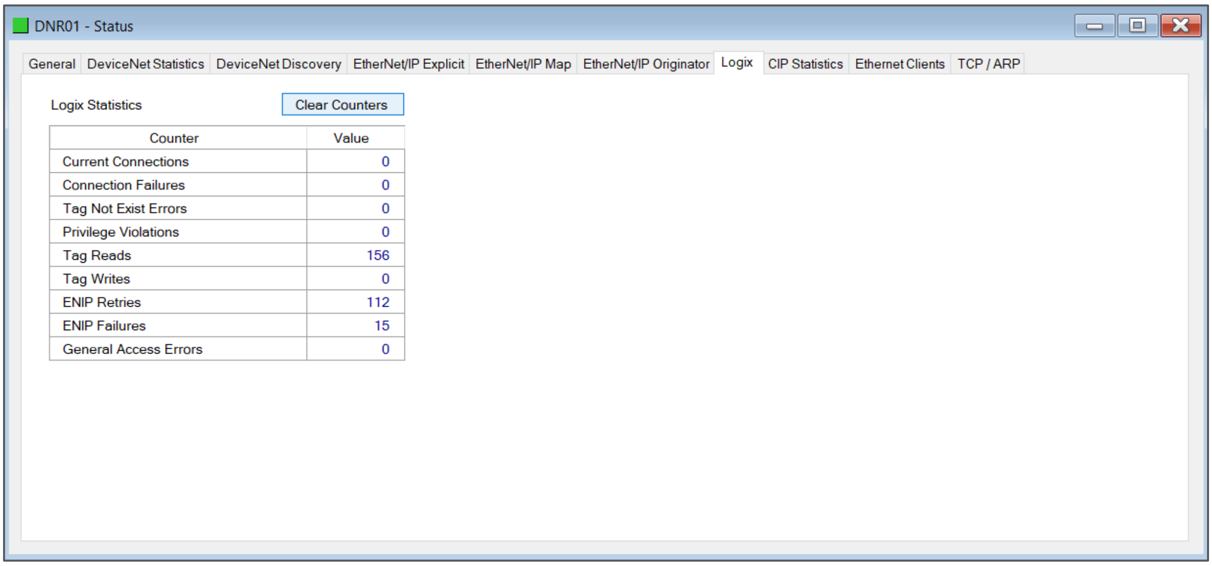


Figure 6.12 - Status monitoring – Logix Statistics

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 DeviceNet Router/B module.
Tag Writes	The number of tag write transactions executed by the DeviceNet Router/B 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 and no response has been received from the Logix Controller.
Tag Access General Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 6.8 – Logix Statistics Tab

6.2.10. MODBUS

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

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

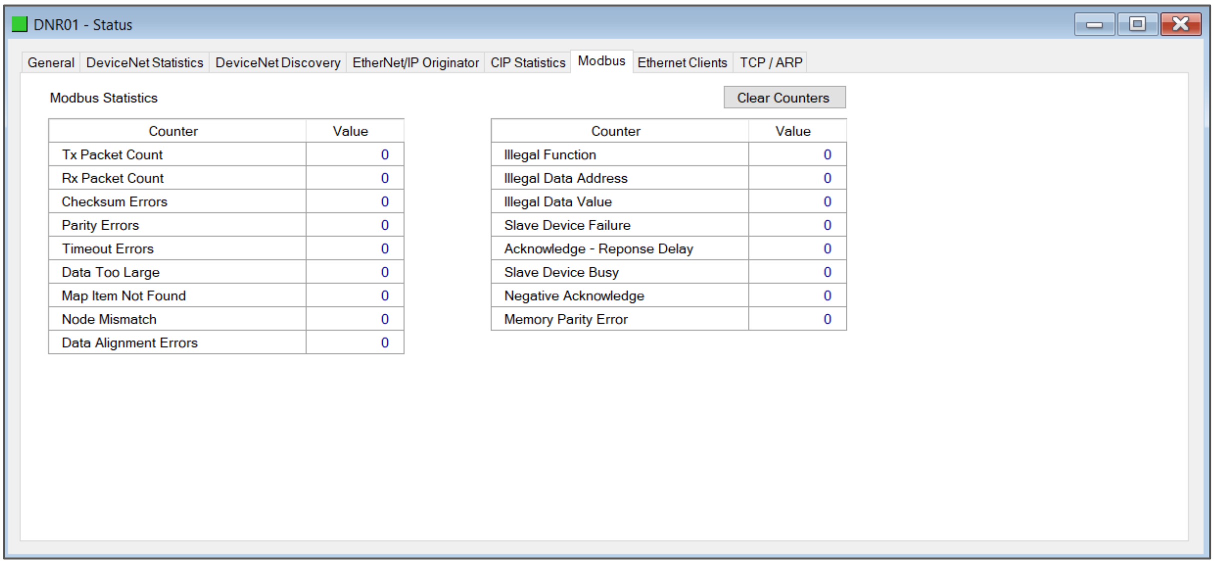


Figure 6.13. - Status monitoring – Modbus Statistics

The Modbus tab displays the following parameters:

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

Table 6.9 - Modbus Statistics Tab

6.2.11. CIP STATISTICS

The CIP tab displays the Ethernet CIP statistics.

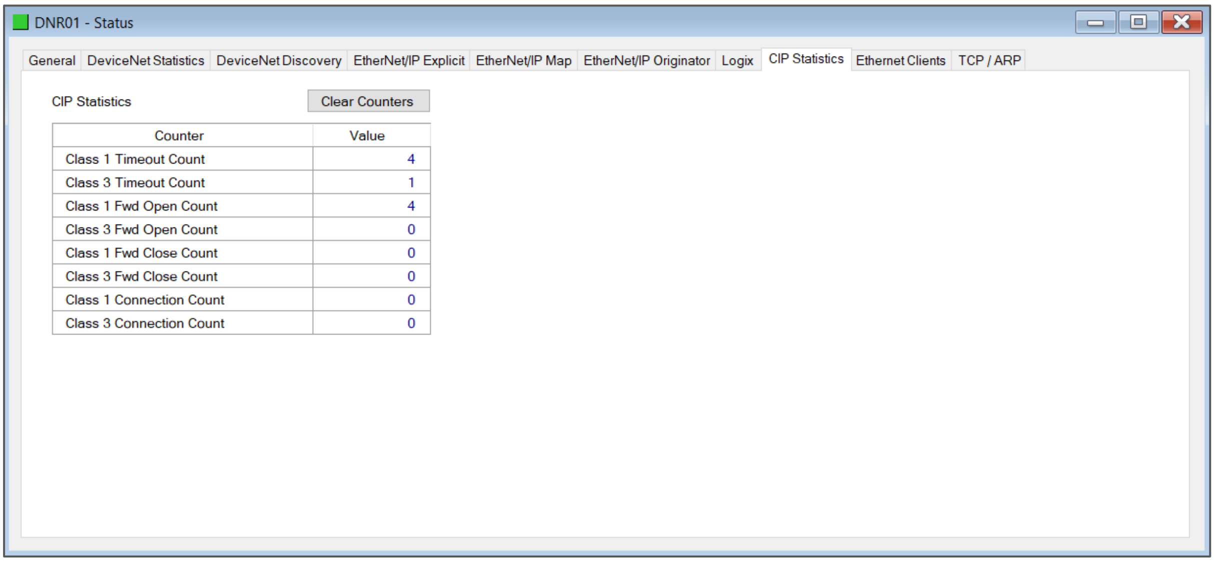


Figure 6.14 - Status monitoring – CIP Statistics

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 6.10 – Mapped Item statistics

6.2.12. Ethernet Clients

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

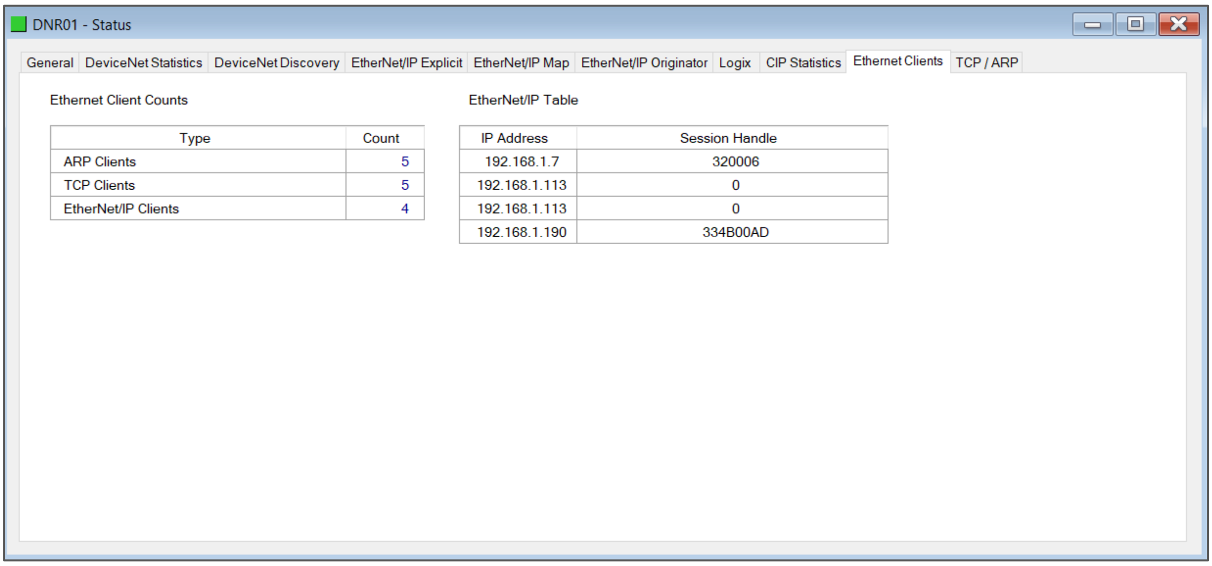


Figure 6.15 –Status monitoring – Ethernet Client Statistics

6.2.13. TCP/ARP

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

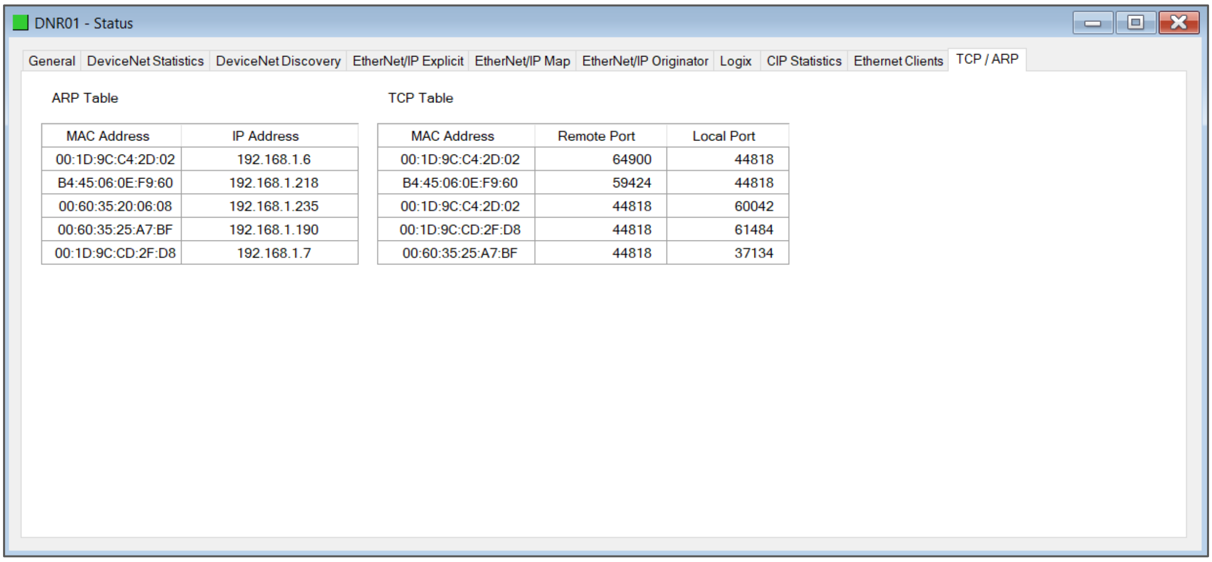


Figure 6.16 – Status monitoring – Ethernet TCP / ARP Statistics

6.3. TARGET DEVICE STATUS MONITORING IN SLATE

The DeviceNet Router/B can also provide individual statistics and status for each for each of the EtherNet/IP Class 1 or DeviceNet Cyclic IO devices when the Primary Interface is *EtherNet/IP Originator* or the DeviceNet Mode is *DeviceNet Originator.*

6.3.1. ETHERNET/IP

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

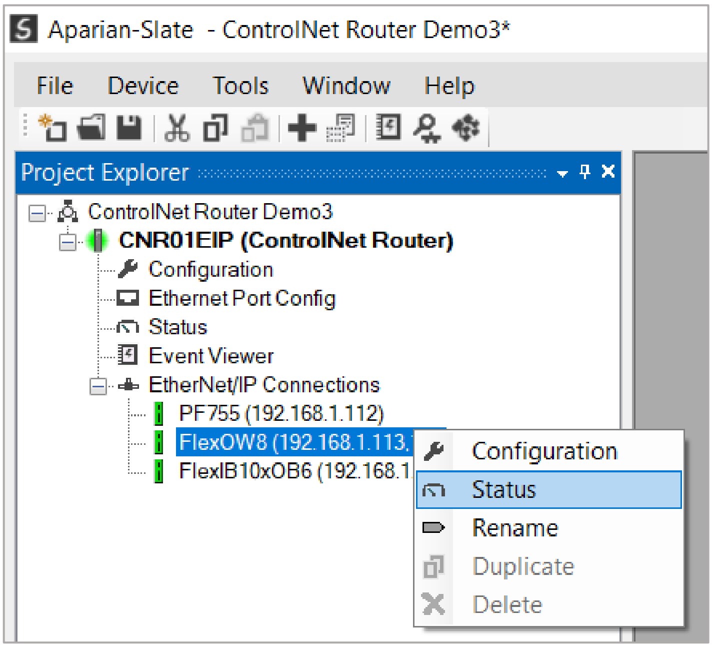


Figure 6.17 – EtherNet/IP Device Status – Status selection

6.3.1.1. GENERAL

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

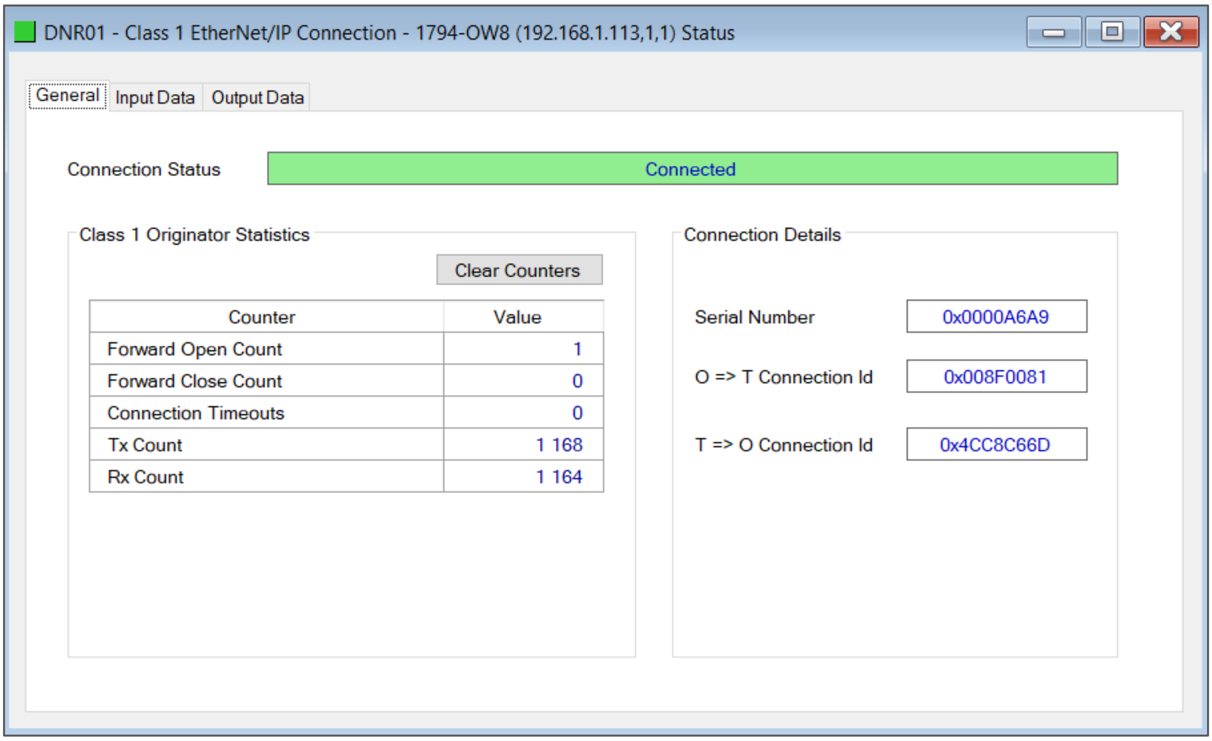


Figure 6.18 – EtherNet/IP Device Status – General Status

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

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

Table 6.11 – EtherNet/IP Class 1 Device status and statistics

6.3.1.2. INPUT DATA

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

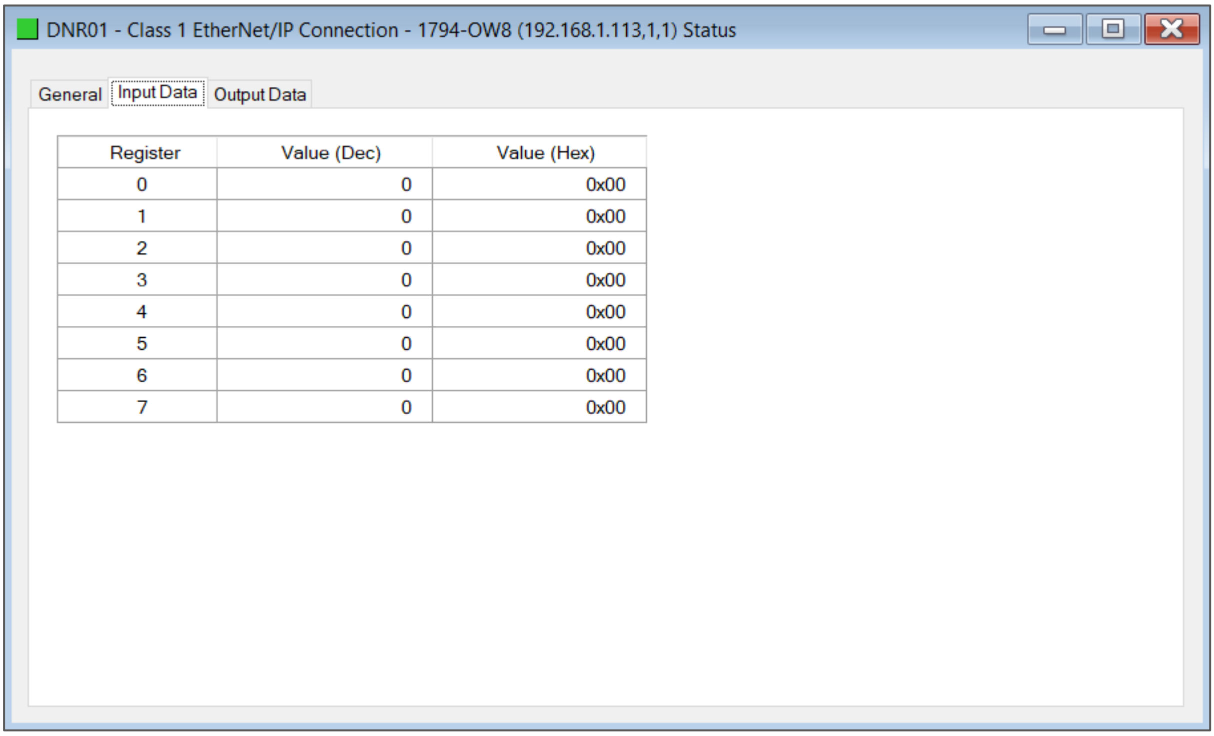
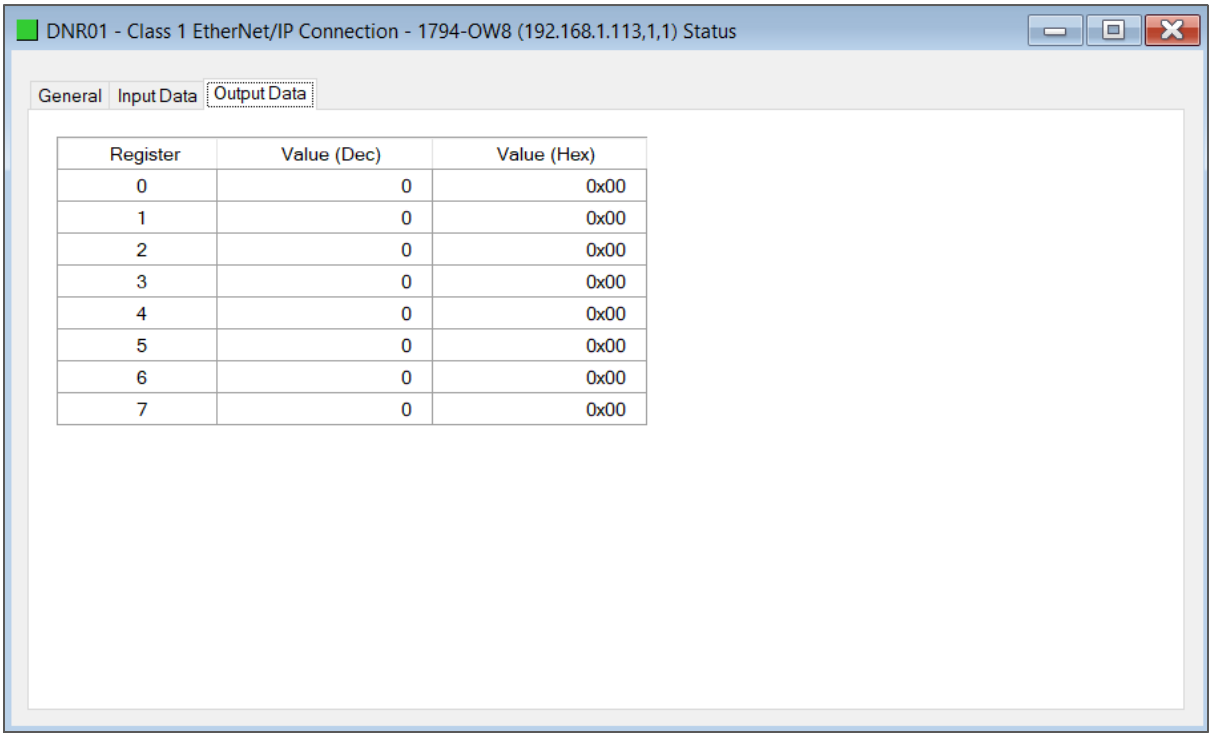


Figure 6.19 – EtherNet/IP Device Status – Input Data

6.3.1.3. OUTPUT DATA

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



Register	Value (Dec)	Value (Hex)
0	0	0x00
1	0	0x00
2	0	0x00
3	0	0x00
4	0	0x00
5	0	0x00
6	0	0x00
7	0	0x00

Figure 6.20 – EtherNet/IP Device Status – Output Data

6.3.2. DeviceNet

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

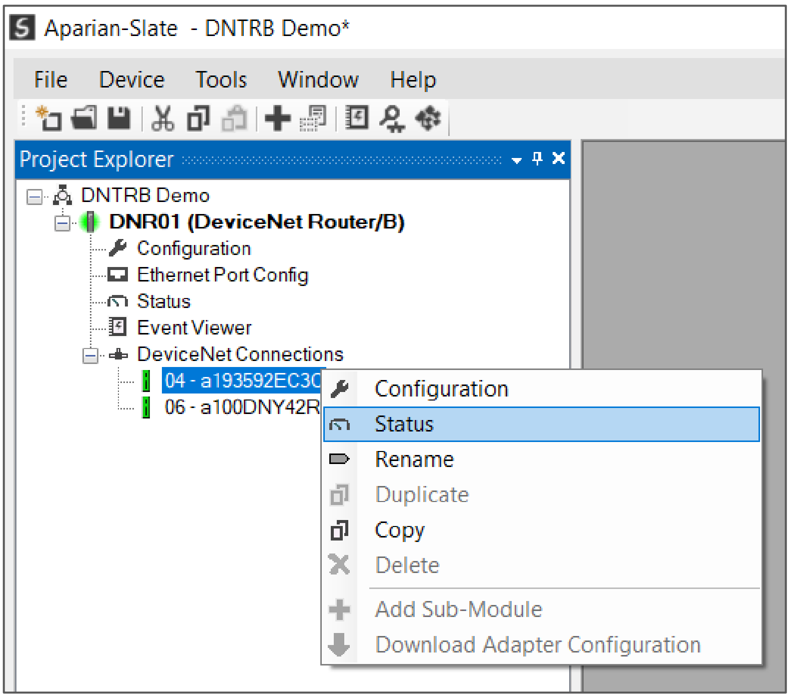


Figure 6.21 – DeviceNet Device Status – Status selection

6.3.2.1. GENERAL

The General Status for the DeviceNet device shows the connection statistics and parameters associated with the DeviceNet Cyclic connection.

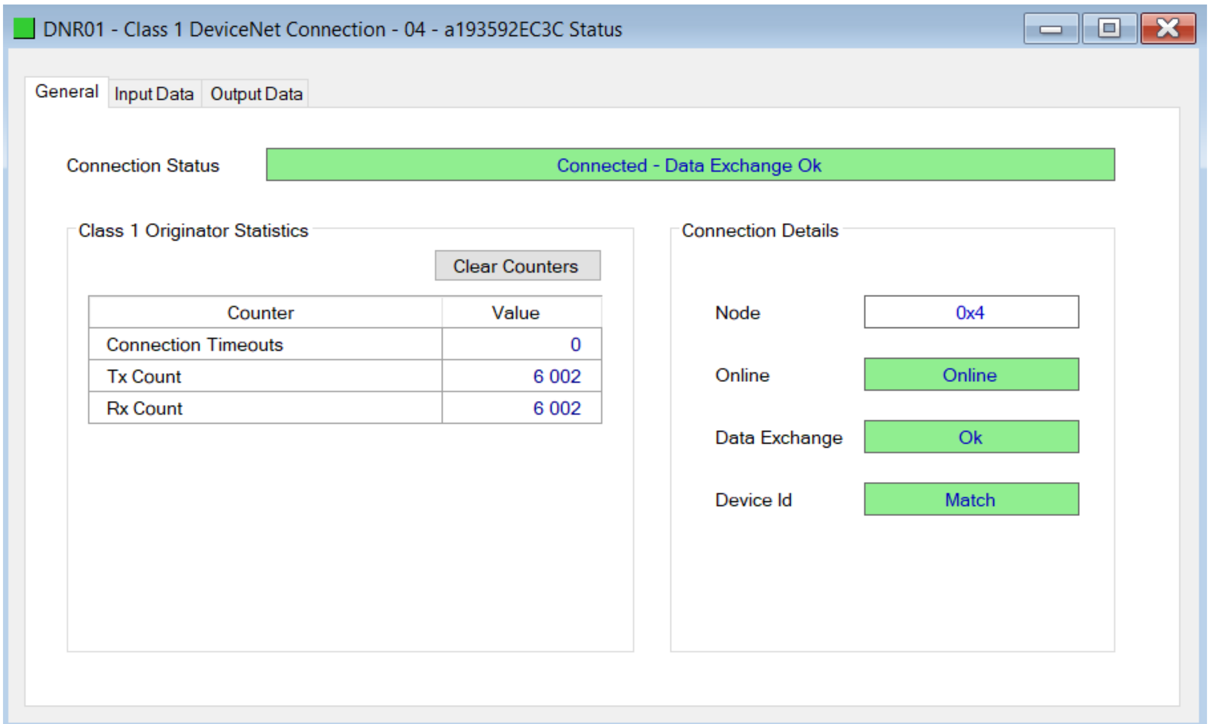


Figure 6.22 – DeviceNet Device Status – General Status

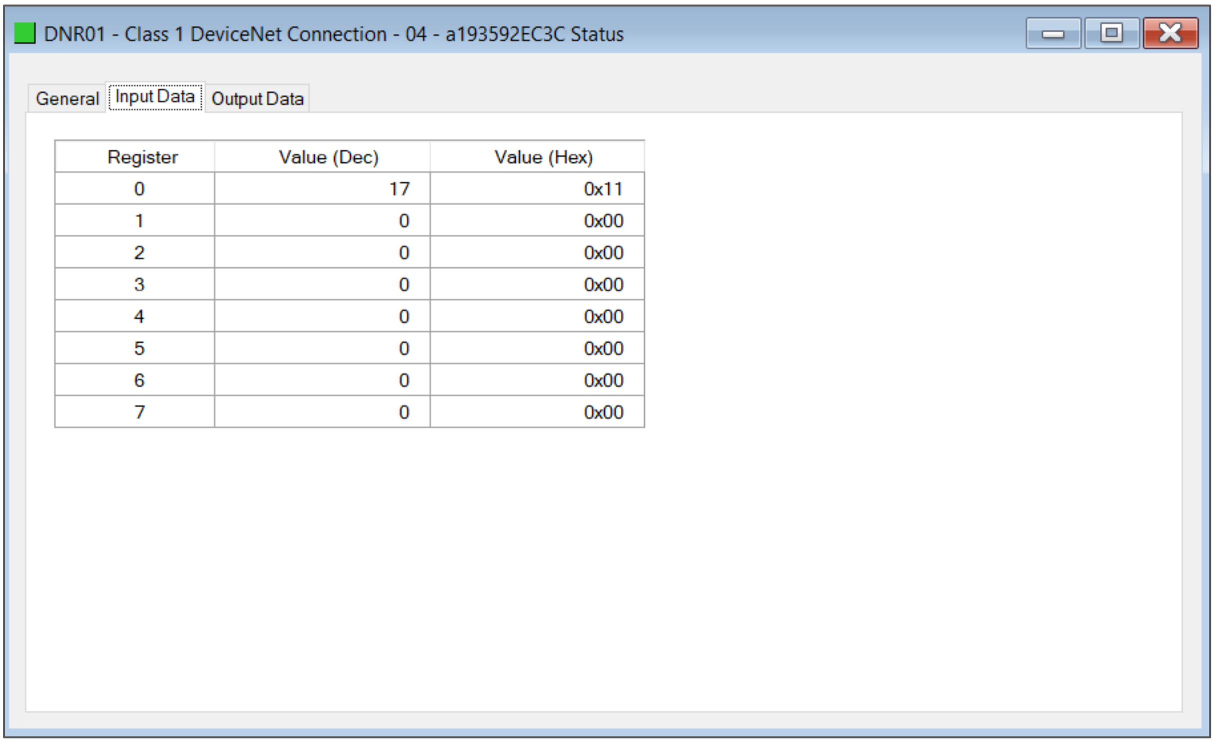
Statistic	Description
Connection Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Cyclic communication. **Offline** The device it offline and not connected **Various response faults** If the connection parameters entered are not correct,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

Connection Details	
Node	The node number of the DeviceNet cyclic device.
Online	Indicates if the DeviceNet device is online.
Data Exchange	Indicates if the DeviceNet device is exchanging data with the DeviceNet Router/B.
Device Id	Indicates if the actual DeviceNet device matches configured device.

Table 6.12 – DeviceNet Cyclic Device status and statistics

6.3.2.2. INPUT DATA

The Input Data for the DeviceNet device shows the Input Assembly associated with the DeviceNet Cyclic connection.



Register	Value (Dec)	Value (Hex)
0	17	0x11
1	0	0x00
2	0	0x00
3	0	0x00
4	0	0x00
5	0	0x00
6	0	0x00
7	0	0x00

Figure 6.23 – DeviceNet Device Status – Input Data

6.3.2.3. OUTPUT DATA

The Output Data for the DeviceNet device shows the Output Assembly associated with the DeviceNet Cyclic connection.

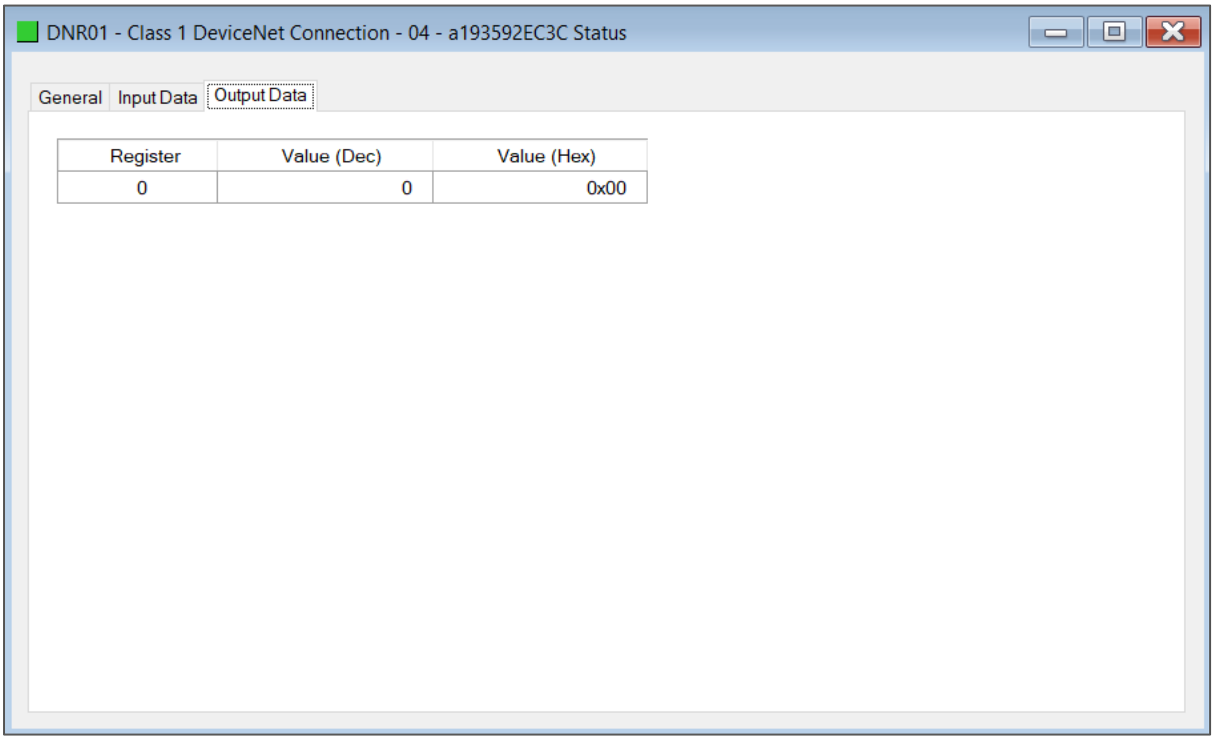


Figure 6.24 – DeviceNet Device Status – Output Data

6.4. MODULE EVENT LOG

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

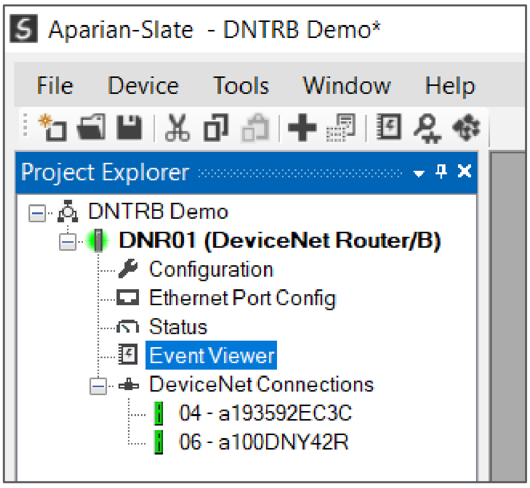
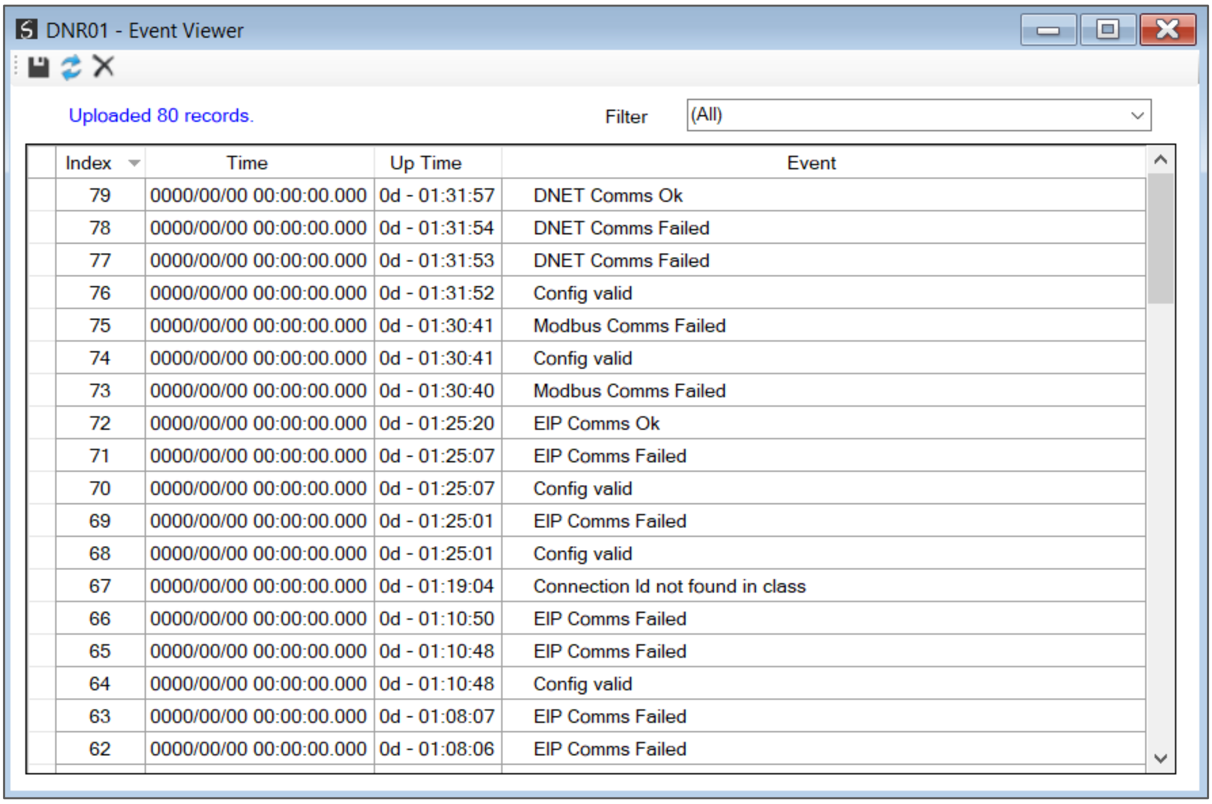


Figure 6.25 - Selecting the module Event Log

The Event Log window will open and automatically read all the events from the module. The log entries are sorted 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
79	0000/00/00 00:00:00.000	0d - 01:31:57	DNET Comms Ok
78	0000/00/00 00:00:00.000	0d - 01:31:54	DNET Comms Failed
77	0000/00/00 00:00:00.000	0d - 01:31:53	DNET Comms Failed
76	0000/00/00 00:00:00.000	0d - 01:31:52	Config valid
75	0000/00/00 00:00:00.000	0d - 01:30:41	Modbus Comms Failed
74	0000/00/00 00:00:00.000	0d - 01:30:41	Config valid
73	0000/00/00 00:00:00.000	0d - 01:30:40	Modbus Comms Failed
72	0000/00/00 00:00:00.000	0d - 01:25:20	EIP Comms Ok
71	0000/00/00 00:00:00.000	0d - 01:25:07	EIP Comms Failed
70	0000/00/00 00:00:00.000	0d - 01:25:07	Config valid
69	0000/00/00 00:00:00.000	0d - 01:25:01	EIP Comms Failed
68	0000/00/00 00:00:00.000	0d - 01:25:01	Config valid
67	0000/00/00 00:00:00.000	0d - 01:19:04	Connection Id not found in class
66	0000/00/00 00:00:00.000	0d - 01:10:50	EIP Comms Failed
65	0000/00/00 00:00:00.000	0d - 01:10:48	EIP Comms Failed
64	0000/00/00 00:00:00.000	0d - 01:10:48	Config valid
63	0000/00/00 00:00:00.000	0d - 01:08:07	EIP Comms Failed
62	0000/00/00 00:00:00.000	0d - 01:08:06	EIP Comms Failed

Figure 6.26 - Module Event Log

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

6.5. WEB SERVER

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

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

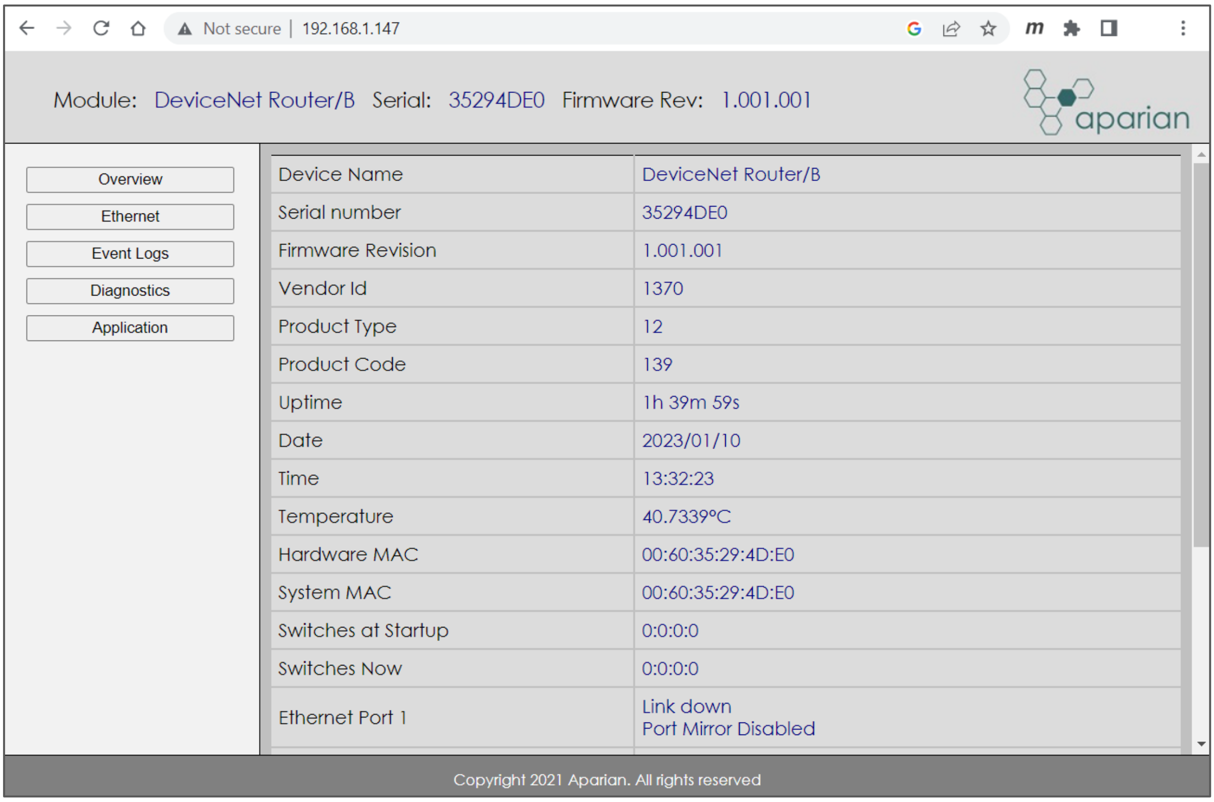


Figure 6.27 – Web interface

6.6. DEVICENET PACKET CAPTURE

The module provides the capability to capture the DeviceNet 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, double-click on the DeviceNet Packet Capture item in the Project Explorer tree.

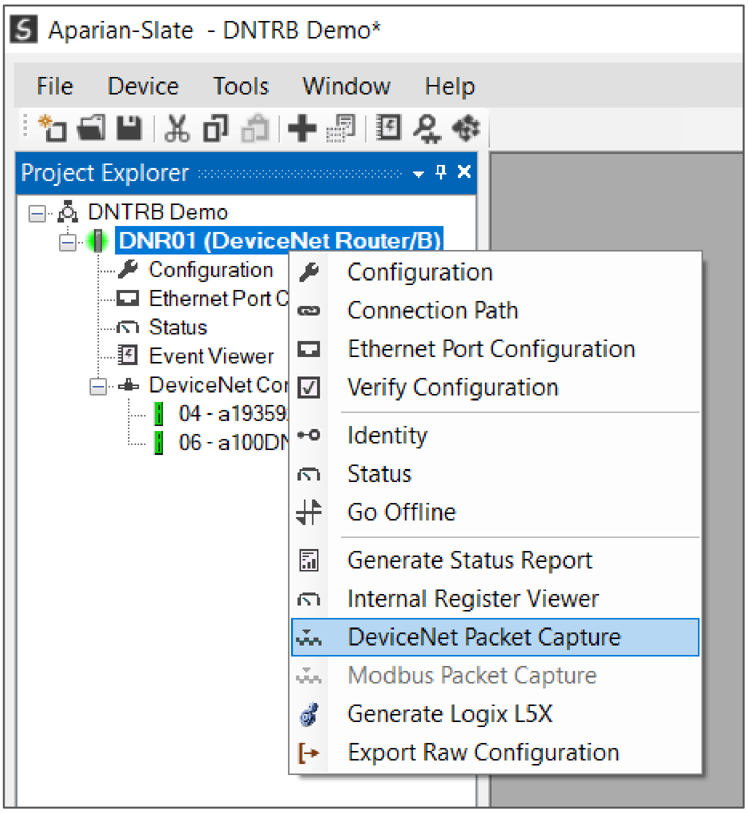


Figure 6.28 - Selecting Modbus Packet Capture

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

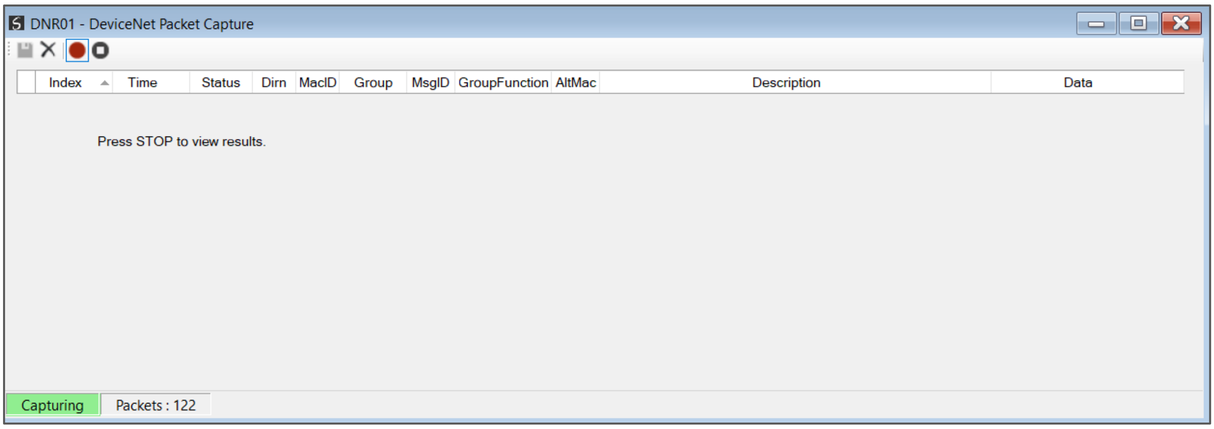
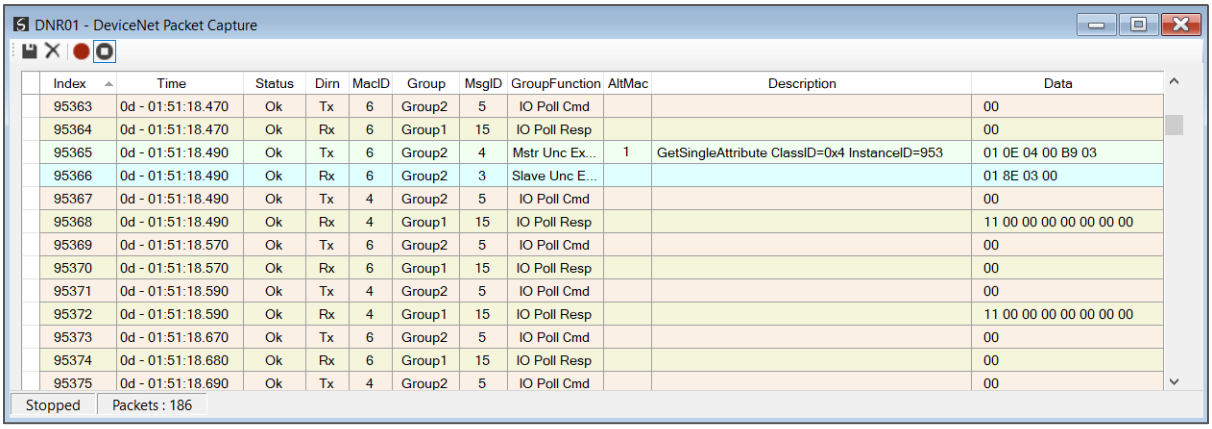


Figure 6.29 – Modbus packet capture

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



DNR01 - DeviceNet Packet Capture

Index	Time	Status	Dirn	MacID	Group	MsgID	GroupFunction	AltMac	Description	Data
95363	0d - 01:51:18.470	Ok	Tx	6	Group2	5	IO Poll Cmd			00
95364	0d - 01:51:18.470	Ok	Rx	6	Group1	15	IO Poll Resp			00
95365	0d - 01:51:18.490	Ok	Tx	6	Group2	4	Mstr Unc Ex...	1	GetSingleAttribute ClassID=0x4 InstanceID=953	01 0E 04 00 B9 03
95366	0d - 01:51:18.490	Ok	Rx	6	Group2	3	Slave Unc E...			01 8E 03 00
95367	0d - 01:51:18.490	Ok	Tx	4	Group2	5	IO Poll Cmd			00
95368	0d - 01:51:18.490	Ok	Rx	4	Group1	15	IO Poll Resp			11 00 00 00 00 00 00 00
95369	0d - 01:51:18.570	Ok	Tx	6	Group2	5	IO Poll Cmd			00
95370	0d - 01:51:18.570	Ok	Rx	6	Group1	15	IO Poll Resp			00
95371	0d - 01:51:18.590	Ok	Tx	4	Group2	5	IO Poll Cmd			00
95372	0d - 01:51:18.590	Ok	Rx	4	Group1	15	IO Poll Resp			11 00 00 00 00 00 00 00
95373	0d - 01:51:18.670	Ok	Tx	6	Group2	5	IO Poll Cmd			00
95374	0d - 01:51:18.680	Ok	Rx	6	Group1	15	IO Poll Resp			00
95375	0d - 01:51:18.690	Ok	Tx	4	Group2	5	IO Poll Cmd			00

Stopped | Packets : 186

Figure 6.30 – DeviceNet Packet Capture complete

The captured DeviceNet 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 DeviceNet constructs and valid checksums.
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
MacID	The DeviceNet MAC ID (0-63) of the packet. This is usually the source MAC, but with IO connection data can be the destination MAC.
Group	The message group number. Either Group 1,2,3 or 4.
MsgID	The Message ID is used to identify a message within a particular group. Can be used to indicate specific types of messages, or specific (previously established) connections.
Group Function	The Group Function. Certain Group and Message ID combination have specific meanings which are displayed here.
Alt MAC	The alternate MAC, depending on the type of message. This is usually the destination MAC ID.
Description	A brief description of the packet.
Data	The raw packet data.

Table 6.13 – DeviceNet 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 DeviceNet Packet Capture files can be viewed by selecting the ***DeviceNet Packet Capture Viewer*** option in the ***Tools*** menu.

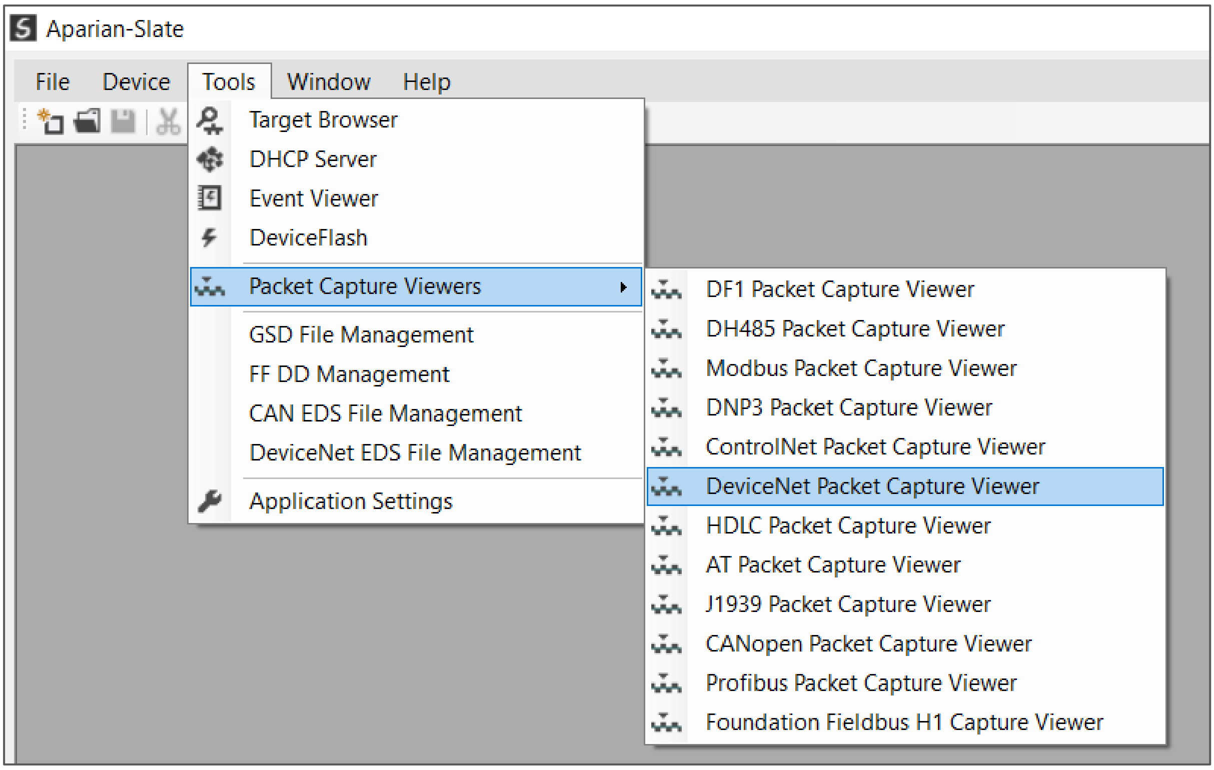


Figure 6.31 - Selecting the DeviceNet Packet Capture Viewer

6.7. 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, double-click on the Modbus Packet Capture item in the Project Explorer tree.

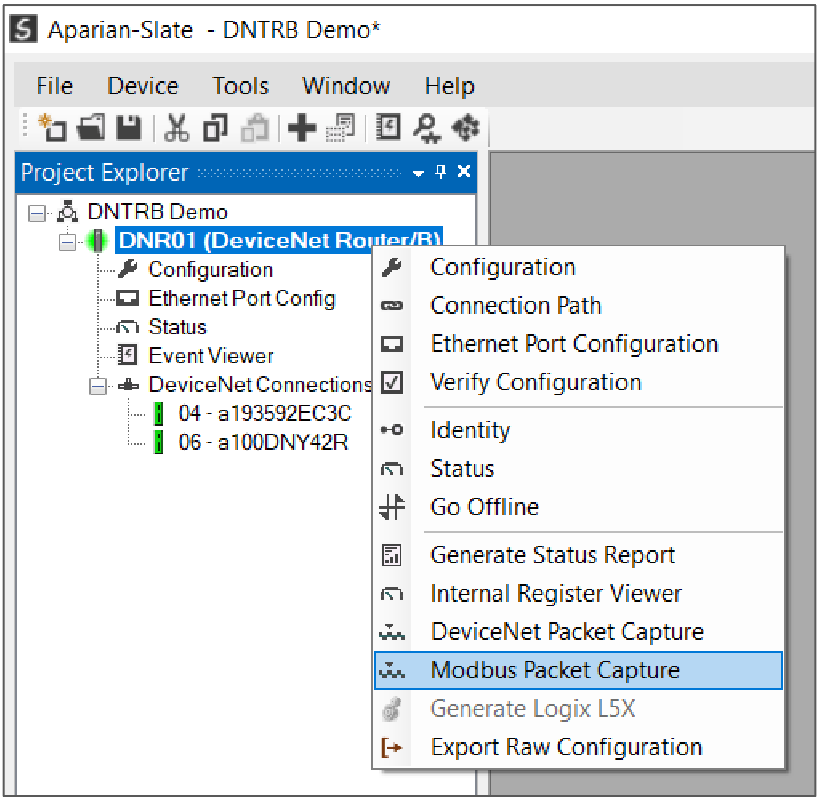


Figure 6.32 - Selecting Modbus Packet Capture

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

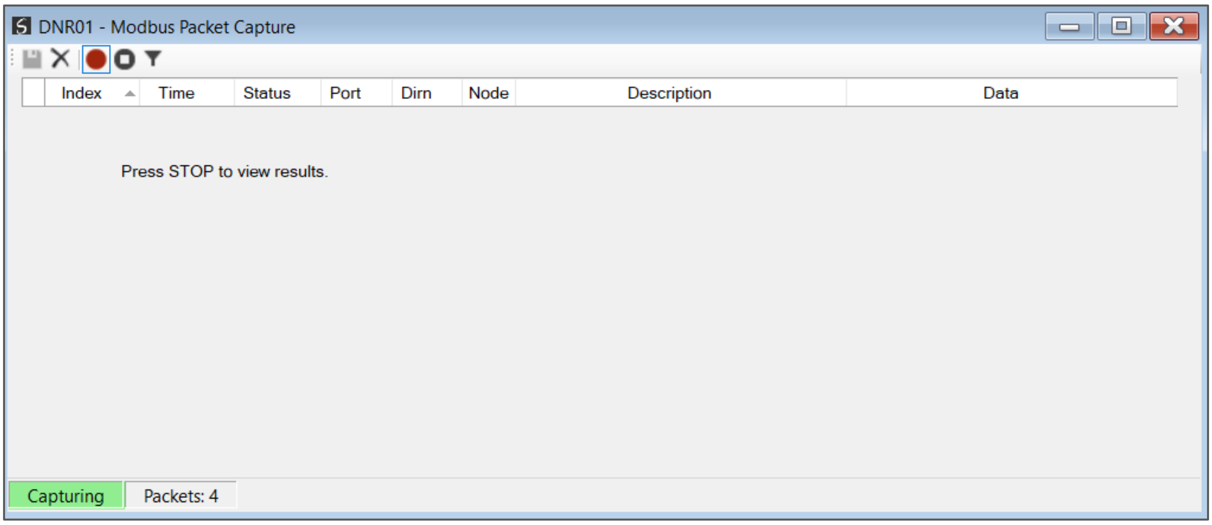


Figure 6.33 – Modbus packet capture

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

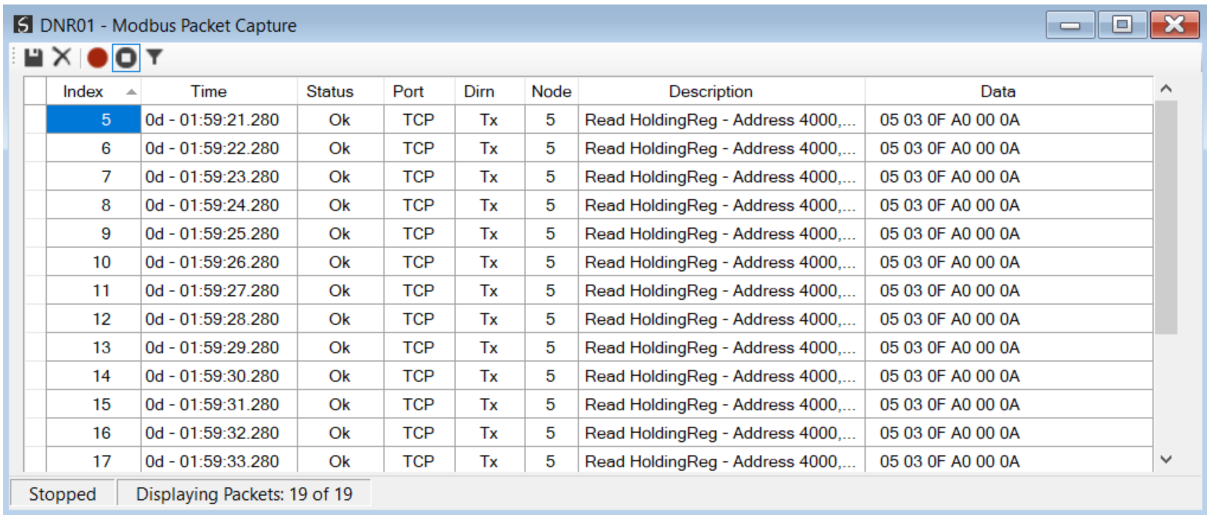


Figure 6.34 – Modbus Packet Capture complete

The captured Modbus packets are tabulated as follows:

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

Table 6.14 – Modbus Packet Capture fields

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

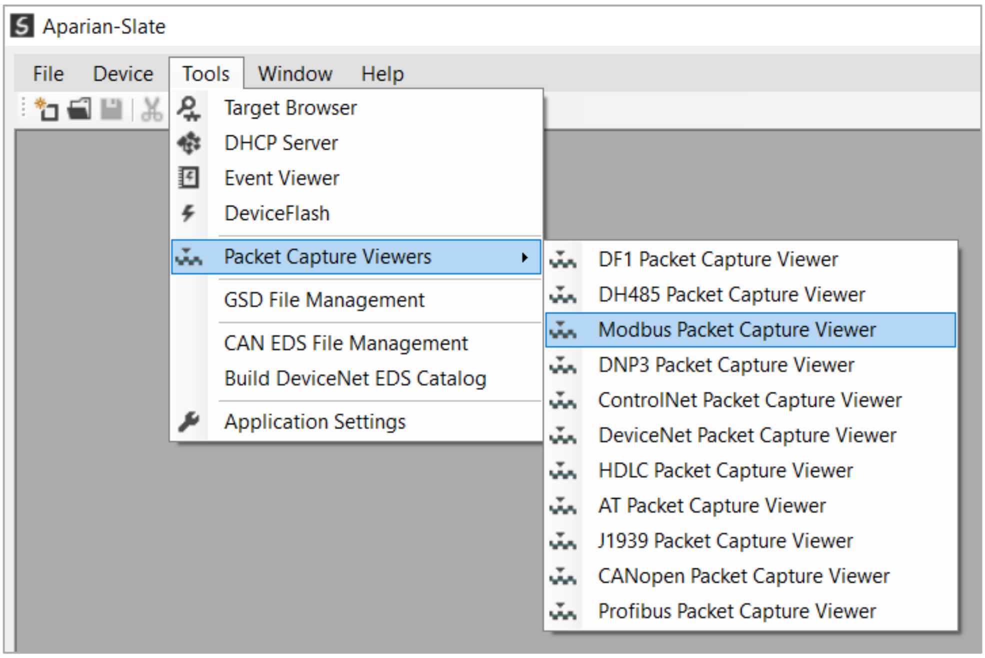


Figure 6.35 - Selecting the Modbus Packet Capture Viewer

6.8. MODULE STATUS REPORT

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

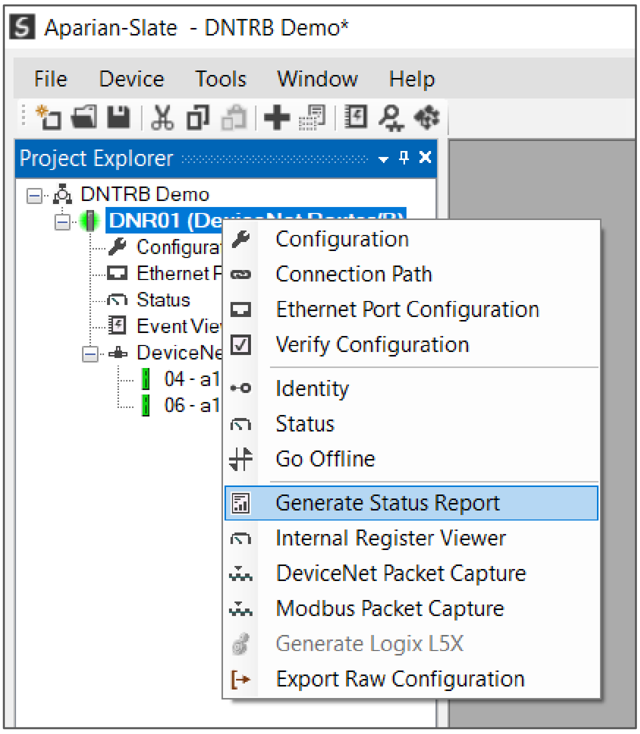


Figure 6.36 – Module Status Report

7. TECHNICAL SPECIFICATIONS

7.1. DIMENSIONS

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

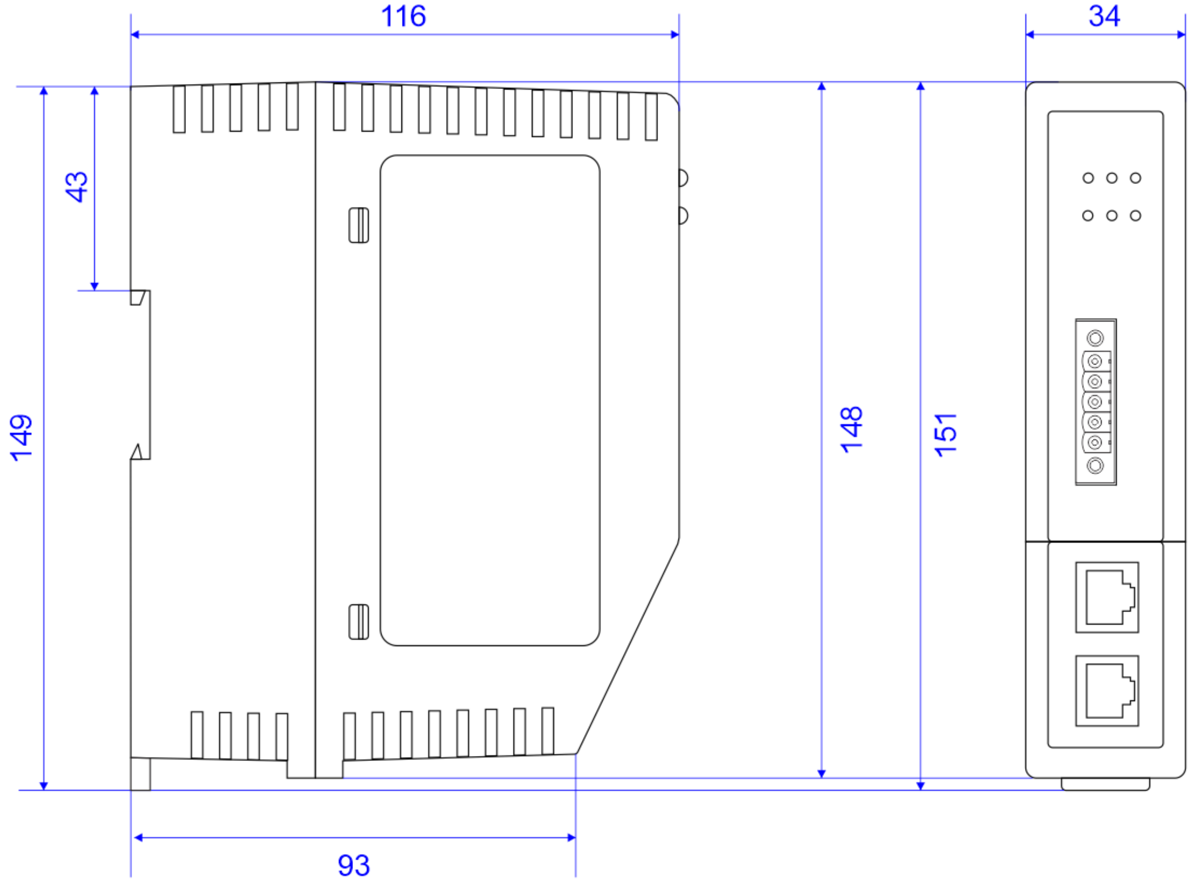


Figure 7.1 – DeviceNet Router/B enclosure dimensions

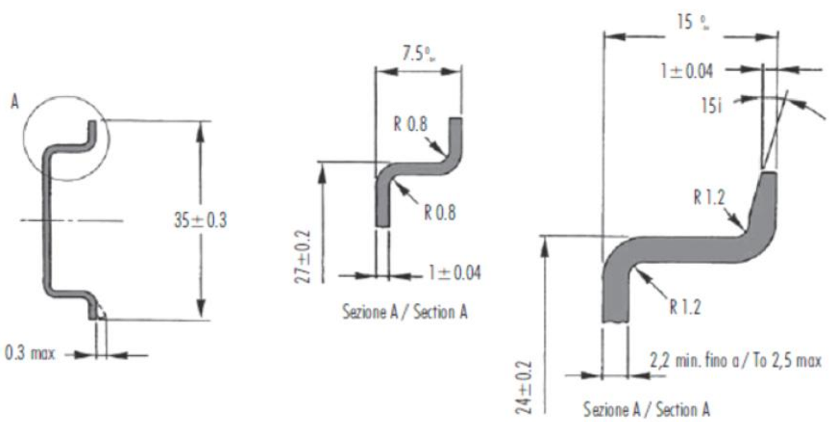


Figure 7.2 - Required DIN dimensions

7.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 32V DC
Power consumption	2.2 W (Max.) Current: 180 mA @ 10 V Current: 85 mA @ 24 V
Connector	3-way terminal
Conductors	24 – 18 AWG
Enclosure rating	IP20, NEMA/UL Open Type
Temperature	-20 – 70 °C
Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 7.1 - Electrical specification

7.3. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 100
TCP connections	Max 100
CIP connections	Max 15
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 7.2 - Ethernet specification

7.4. SERIAL PORT (RS232)

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

Table 7.3 – RS232 Serial Port specification

7.5. SERIAL PORT (RS485)

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

Table 7.4 – RS485 Serial Port specification

7.6. DEVICENET

Specification	Rating
Connector	5-way terminal, 5.08mm pitch.
Modes	DeviceNet Scanner DeviceNet Target
Passthrough Messaging	Supported
Supported Baud Rates	125k 250k 500k
DeviceNet Terminator	120 Ω - Software Enabled

Table 7.5 – DeviceNet specification

7.7. DEVICENET SCANNER

Specification	Rating
DeviceNet Device Count	63
Set Target Device Node	Supported
Set Target Device BAUD	Supported
Device Discovery	Supported
Explicit Messaging	Supported
Connections Supported	Polled Change of State (COS)

Table 7.6 – DeviceNet Scanner specification

7.8. DEVICENET TARGET

Specification	Rating
Connections Supported	Polled Change of State (COS)
Input / Output Data Max	128 bytes input 128 bytes output

Table 7.7 – DeviceNet Device specification

7.9. PCCC

Specification	Rating
Max PCCC Connections	20
Max PCCC Payload	1000 bytes

Table 7.8 – PCCC specification

7.10. ETHERNET/IP TARGET

Specification	Rating
Class 1 Cyclic connection count	4
Logix Direct-to-Tag Supported	Yes

Table 7.9 – EtherNet/IP Target specification

7.11. ETHERNET/IP ORIGINATOR

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

Table 7.10 – EtherNet/IP Originator specification

7.12. MODBUS CLIENT

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

Table 7.11 – Modbus Client specification

7.13. MODBUS SERVER

Specification	**Rating**
Modes Supported	Modbus TCP, Modbus RTU232, Modbus RTU485 (simultaneous)
Modbus RTU485 Termination	Software set
Mapping Ranges	Holding Register 0 – 65535 Input Register 0 – 65535 Input Status 0 – 65535 Coil Status 0 – 65535
Base Offset	Modbus (Base 0) PLC (Base 1)
Configurable Modbus TCP Port	Yes

Table 7.12 – Modbus Server specification

7.14. CERTIFICATIONS

Certification	**Mark**
CE Mark	

RoHS2 Compliant	
UL Mark File: E494895	CLASS 1, DIV 2, GROUPS A, B, C, D
UKCA	
ATEX	II 3 G Ex ec IIC T5 -25°C ≤ Ta ≤ 70 °C

Table 7.13 – Certifications

8. CIP RESPONSE STATUS CODES

General Status Code	Name
0x00	Sucess
0x01	Communications Related Problem
0x02	Resource unavailable
0x03	Invalid parameter value
0x04	Path segment error
0x05	Path destination unknown
0x06	Partial transfer
0x07	Connection lost
0x08	Service not supported
0x09	Invalid attribute value
0x0A	Attribute list error
0x0B	Already in requested mode/state
0x0C	Object state conflict
0x0D	Object already exists
0x0E	Attribute not settable
0x0F	Privilege violation
0x10	Device state conflict
0x11	Reply data too large
0x12	Fragmentation of a primitive value
0x13	Not enough data
0x14	Attribute not supported
0x15	Too much data
0x16	Object instance does not exist
0x17	Service fragmentation out of sequence
0x18	No stored attribute data
0x19	Store operation failure
0x1A	Routing failure, request packet too large
0x1B	Routing failure, response packet too large

0x1C	Missing attribute list entry Data
0x1D	Invalid attribute value list
0x1E	Embedded service error
0x1F	Vendor specific error
0x20	Invalid parameter
0x21	Write-once value or medium already written
0x22	Invalid Reply Received
0x23	Buffer Overflow
0x24	Message Format Error
0x25	Key Failure in path
0x26	Path Size Invalid
0x27	Unexpected attribute in list
0x28	Invalid Member ID
0x29	Member not settable
0x2A	Group 2 only server general failure
0x2B	Unknown Modbus Error
0x2C	Attribute not gettable
0x2D	Instance Not Deletable
0x2E	Service Not Supported for Specified Path
0x2F	Fragmentation Needed

Table 8.1 – CIP Response Codes

9. INDEX