ProSoft TECHNOLOGY

Where Automation Connects.

MVI56-DNP

ControlLogix Platform

DNP 3.0 Master/Slave
Communication Module

July 28, 2022

USER MANUAL

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology, Inc.
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

MVI56-DNP User Manual
For Public Use.

July 28, 2022

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;
WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES
WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.
THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

Warnings

North America Warnings

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
C Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

Avertissement - Risque d'explosion - Avant de déconnecter l'équipement, couper le courant ou s'assurer que l'emplacement est désigné non dangereux.

D Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

MVI (Multi Vendor Interface) Modules

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'ÉQUIPEMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

Battery Life Advisory

The MVI46, MVI56, MVI56E, MVI69, and MVI71 modules use a rechargeable Lithium Vanadium Pentoxide battery to backup the real-time clock and CMOS. The battery should last for the life of the module. The module must be powered for approximately twenty hours before the battery becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and the real-time clock for approximately 21 days. When the battery is fully discharged, the module will revert to the default BIOS and clock settings.

Note: The battery is not user replaceable.

Markings

Electrical Ratings

- Backplane Current Load: 800 mA @ 5.1 Vdc; 3 mA @ 24 Vdc
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Shock: 30 g, operational; 50 g, non-operational; Vibration: 5 g from 10 Hz to 150 Hz
- Relative Humidity: 5% to 95% with no condensation
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least $4mm^2$.

Agency Approvals and Certifications

Please visit our website: www.prosoft-technology.com

Contents

Guide to the MVI56-DNP User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 11)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 69)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications	→	Reference (page 99) Product Specifications (page 99)	These sections contain general references associated with this product and its Specifications..
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 161) Index	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

To get the most benefit from this User Manual, you should have the following skills:

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect Distributed Network Protocol and ControlLogix devices to a power source and to the MVI56-DNP module's application port(s)

1.1 System Requirements

The MVI56-DNP module requires the following minimum hardware and software components:

- Rockwell Automation ControlLogix™ processor, with compatible power supply and one free slot in the rack, for the MVI56-DNP module. The module requires 800 mA of available power.
- Rockwell Automation RSLogix 5000 programming software version 2.51 or higher
- Rockwell Automation RSLinx communication software
- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP Professional with Service Pack 1 or 2
  - Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3
  - Microsoft Windows Server 2003
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- ProSoft Configuration Builder, HyperTerminal or other terminal emulator program.

Note: You can install the module in a local or remote rack. For remote rack installation, the module requires EtherNet/IP or ControlNet communication with the processor.

1.2 Package Contents

The following components are included with your MVI56-DNP module, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI56-DNP Module	MVI56-DNP	DNP 3.0 Master/Slave Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module

If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the module. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

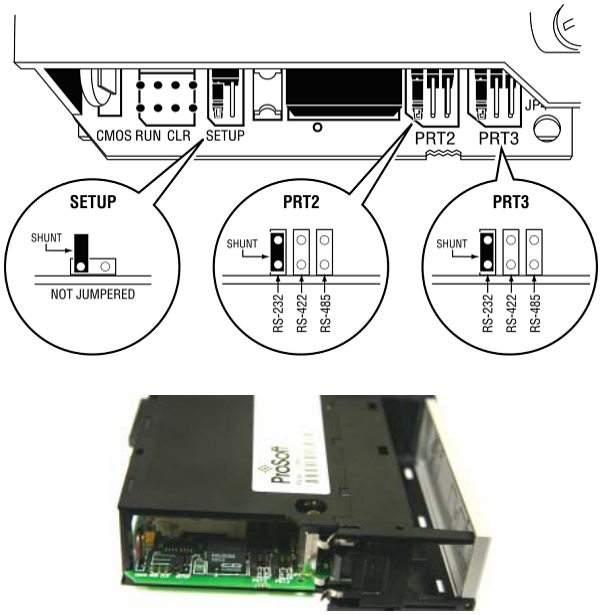
To install ProSoft Configuration Builder from the ProSoft Technology website

1. Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
2. Click the **DOWNLOAD HERE** link to download the latest version of *ProSoft Configuration Builder*.
3. Choose **SAVE** or **SAVE FILE** when prompted.
4. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

1.4 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to select the interface you wish to use. There are three jumpers located at the bottom of the module.

The following illustration shows the MVI56-DNP jumper configuration:



1. Set the PRT 2 (for application port 1) and PRT 3 (for application port 2) jumpers select RS232, RS422, or RS485 to match the wiring needed for your application. The default jumper setting for both application ports is RS-232.
2. The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

1.5 Installing the Module in the Rack

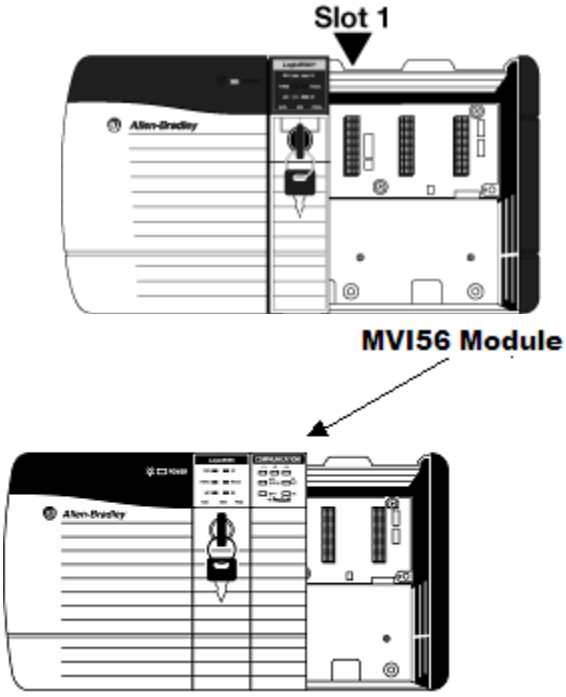
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-DNP module. Refer to your Rockwell Automation product documentation for installation instructions.

Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify that suitable safety procedures are in place before installing or servicing the device.

After you have checked the placement of the jumpers, insert MVI56-DNP into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

Warning: When you insert or remove the module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Verify that power is removed or the area is non-hazardous before proceeding. Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

1. Turn power OFF.
2. Align the module with the top and bottom guides, and slide it into the rack until the module is firmly against the backplane connector.



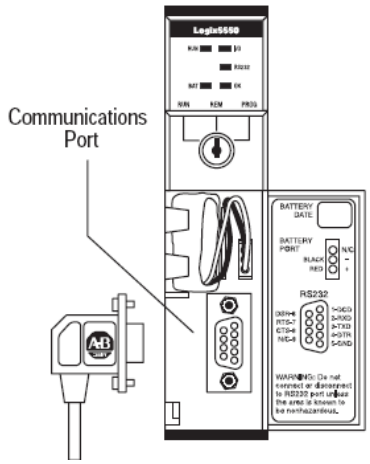
3 With a firm but steady push, snap the module into place.
4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5 Make a note of the slot location. You must identify the slot in which the module is installed in order for the sample program to work correctly. Slot numbers are identified on the green circuit board (backplane) of the ControlLogix rack.
6 Turn power ON.

Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

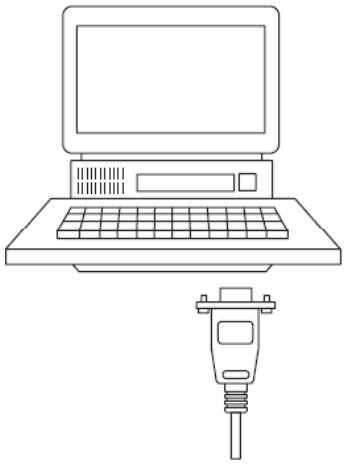
1.6 Connecting Your PC to the ControlLogix Processor

There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface. It is not mandatory that you use the processor's serial interface. You may access the processor through whatever network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

1 Connect the right-angle connector end of the cable to your controller at the communications port.



2 Connect the straight connector end of the cable to the serial port on your computer.



1.7 Using the Sample Ladder Logic

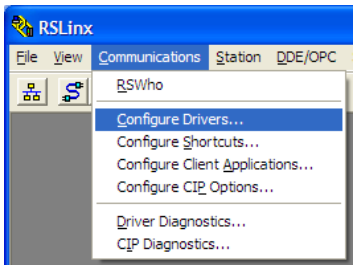
The sample program for your MVI56-DNP module includes custom tags, data types, and ladder logic for data I/O and status monitoring. For most applications, you can run the sample ladder program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

The ProSoft Technology website provides multiple versions of the sample ladder logic. The version number appended to the file name corresponds with the firmware version number of your ControlLogix processor. The firmware version and sample program version must match.

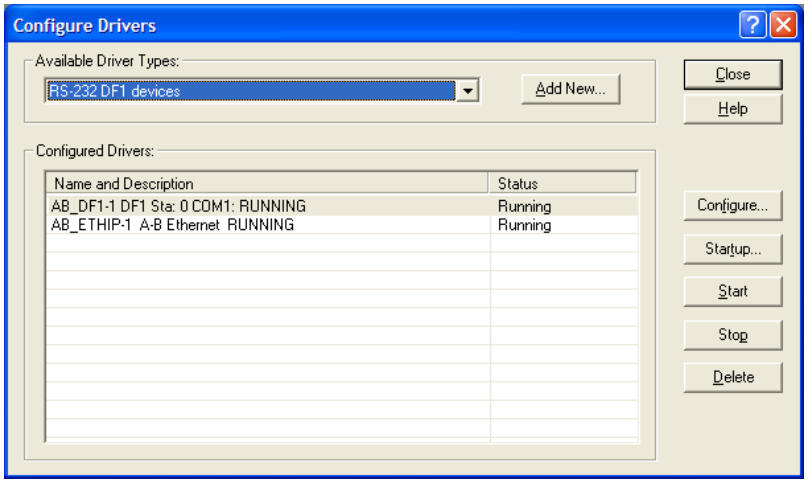
1.7.1 Configuring the RSLinx Driver for the PC COM Port

If RSLogix is unable to establish communication with the processor, follow these steps.

1. Open *RSLinx*.
2. Open the **COMMUNICATIONS** menu, and choose **CONFIGURE DRIVERS**.

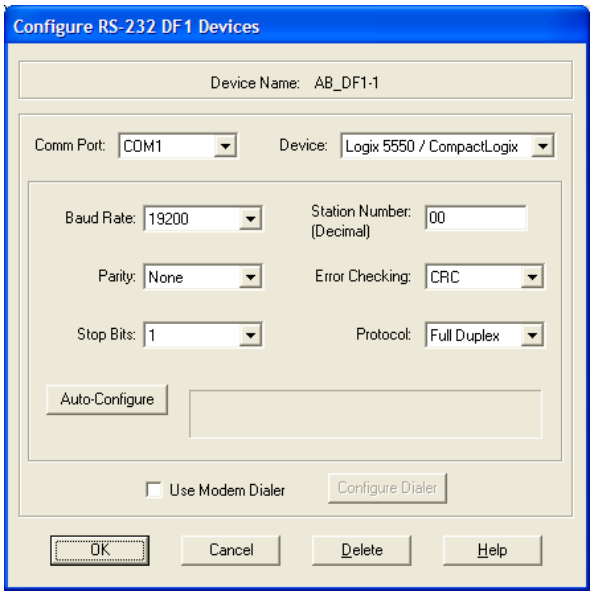


This action opens the *Configure Drivers* dialog box.



Note: If the list of configured drivers is blank, you must first choose and configure a driver from the *Available Driver Types* list. The recommended driver type to choose for serial communication with the processor is *RS-232 DF1 Devices*.

3 Click to select the driver, and then click **CONFIGURE**. This action opens the *Configure RS-232 DF1 Devices* dialog box.



4 Click the **AUTO-CONFIGURE** button. RSLinx will attempt to configure your serial port to work with the selected driver.

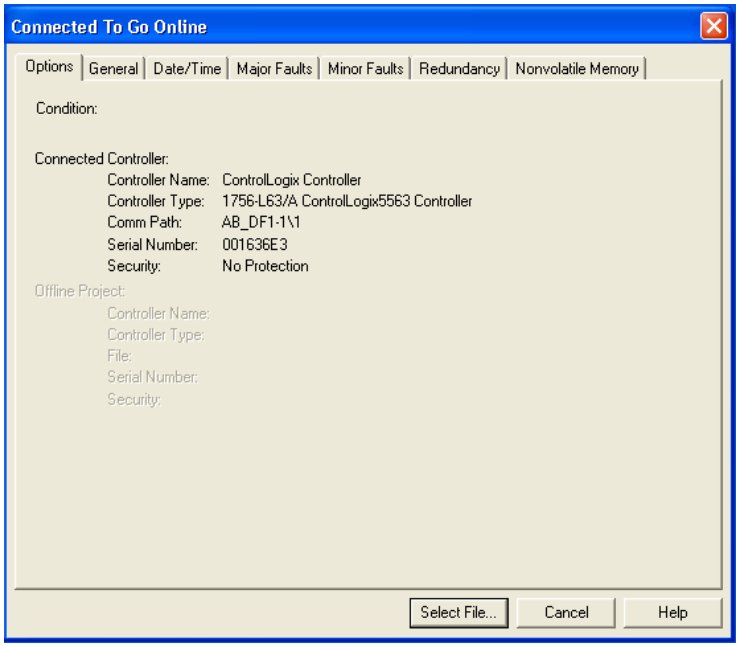
5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

Note: If the auto-configuration procedure fails, verify that the cables are connected correctly between the processor and the serial port on your computer, and then try again. If you are still unable to auto-configure the port, refer to your RSLinx documentation for further troubleshooting steps.

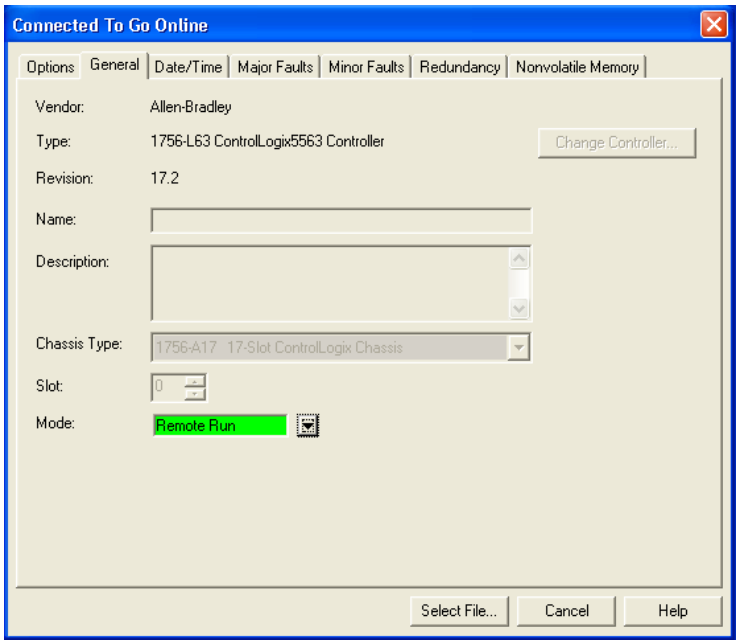
1.7.2 Determining the Firmware Version of Your Processor

Important: The RSLinx service must be installed and running on your computer in order for RSLogix to communicate with the processor. Refer to your RSLinx and RSLogix documentation for help configuring and troubleshooting these applications.

1. Connect an RS-232 serial cable from the COM (serial) port on your PC to the communication port on the front of the processor.
2. Start RSLogix 5000 and close any existing project that may be loaded.
3. Open the **COMMUNICATIONS** menu and choose **GO ONLINE**. RSLogix will establish communication with the processor. This may take a few moments.
4. When RSLogix has established communication with the processor, the *Connected To Go Online* dialog box will open.



5 In the *Connected To Go Online* dialog box, click the **GENERAL** tab. This tab shows information about the processor, including the *Revision* (firmware) version. In the following illustration, the firmware version is 17.2.



1.7.3 Adding the Module in Your Project

This topic describes how to add the module to your RSLogix 5000 project.

Note: The RSLogix 5000 software should be in "off-line" mode to add the module to a project. Although some newer versions of RSLogix 5000 may allow new modules to be added while in "online" mode, it is always considered safer to add new modules off-line and test the new configuration in a test system before putting the modified program online.

This process consists of the following general steps.

1. Add the module to the project I/O configuration.
2. Select the sample ladder logic version that matches your processor firmware version number and use it as a starting point for a new project or for copying components into an existing project.

   (Example ladder logic files are provided at www.prosoft-technolgoy.com)
3. Modify the example ladder logic to meet the needs of your application, if necessary.
4. Download the ladder logic to the processor.

Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

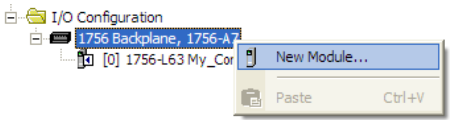
The ladder logic samples show how to process one data block for each supported data type, as well as showing how to use all of the special functions and control variables used by the module. Depending on the point counts you configure for each data type in the DNP.CFG file, the sample ladder logic may or may not have sufficient data transfer capacity.

If your application has higher point counts than what is supported in the sample ladder, you can add additional rungs with logic similar to that shown to process data for up to two additional data blocks for each supported data type.

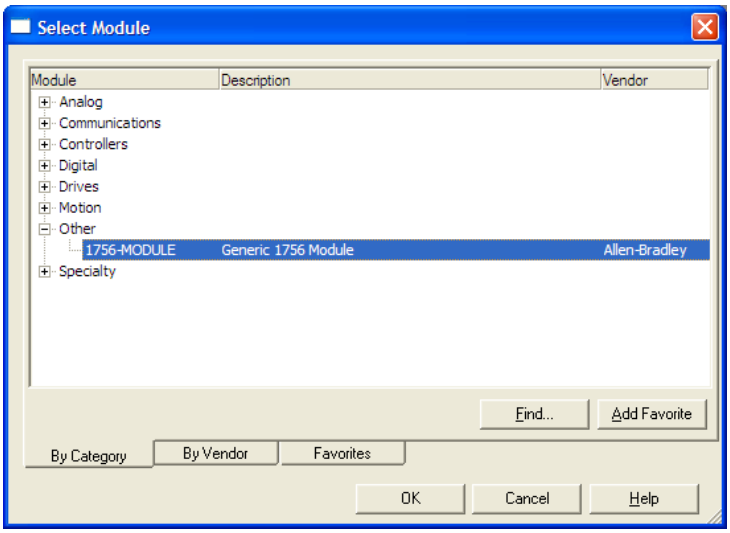
To Copy the Sample Ladder Logic into a New or an Existing Project

1. Add the MVI56-DNP module to the project.

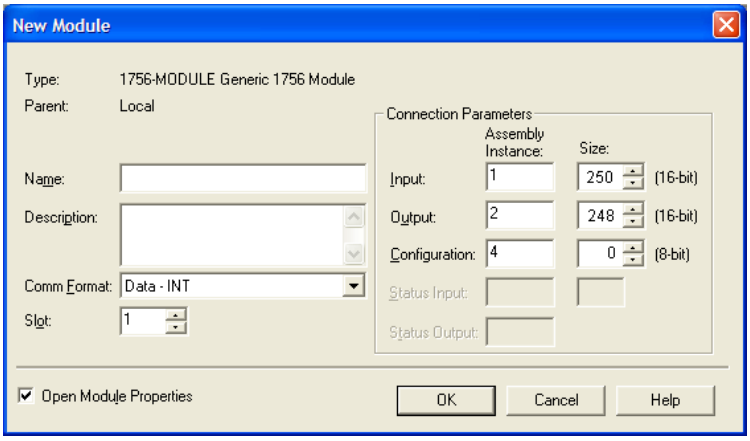
   In the **CONTROLLER ORGANIZATION** window, select **I/O CONFIGURATION** and click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE...**



This action opens the **SELECT MODULE** dialog box.



2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK.** This action opens the **NEW MODULE** dialog box.

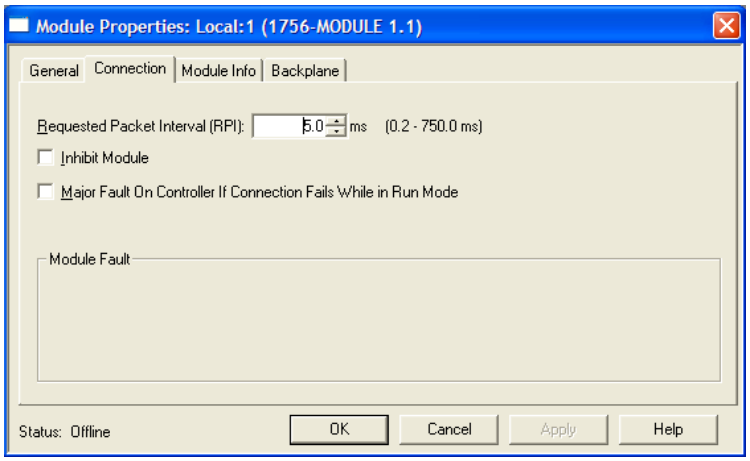


3 Set the Module Properties:

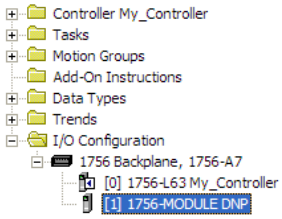
Parameter	Value
Name	Enter a module identification string. Example: DNP.
Description	Enter a description for the module. Example: DNP 3.0 Master/Slave Communication Module
Comm Format	Select **DATA-INT** (no other option will work)
Slot	Enter the slot number in the rack where the MVI56-DNP module is located.
Input Assembly Instance	1 (must use this value)
Input Size	250 (must use this value)
Output Assembly Instance	2 (must use this value)
Output Size	248 (must use this value)
Configuration Assembly Instance	4 (must use this value)
Configuration Size	0 (must use this value)

4 Select the Requested Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications.

On the Connection tab, set the RPI value for your project. Click OK to confirm.

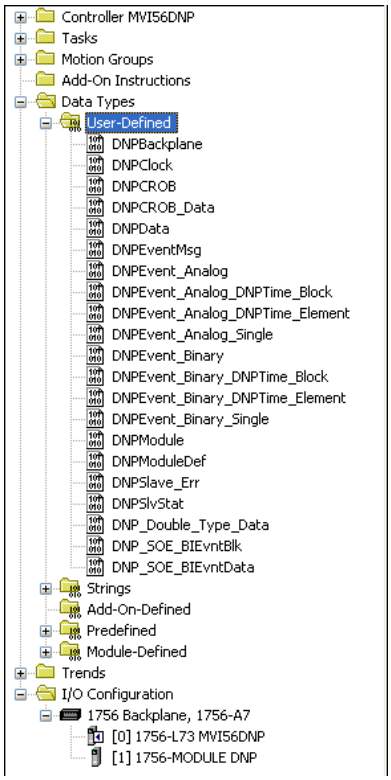


After completing the module setup, the Controller Organization list will display the module's presence. The data required for the module will be defined to the application, and objects will be allocated in the Controller Tags data area. The following is an example of the Controller Organization list:

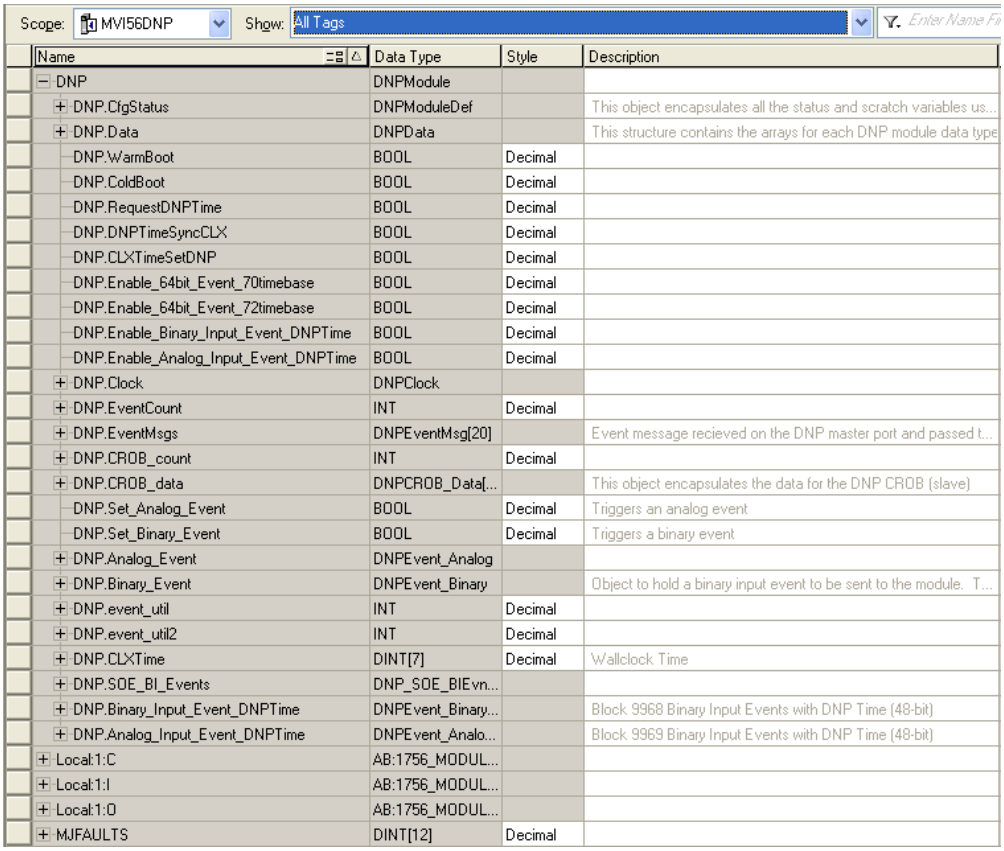


5 In a separate instance of RSLogix 5000, open the version of the sample ladder logic project that matches the firmware revision number of your ControlLogix processor. This is so you may copy and paste from the sample into your existing project. If you are starting a new project, simply open the appropriate sample version, pick your controller model and rack size, and use this as the starting point for your new project.

6 Add the User Defined Data Types for the module. Copy these data types from the sample ladder logic into your project. The Controller Organization list should display the User Defined Data Types shown in the following example:



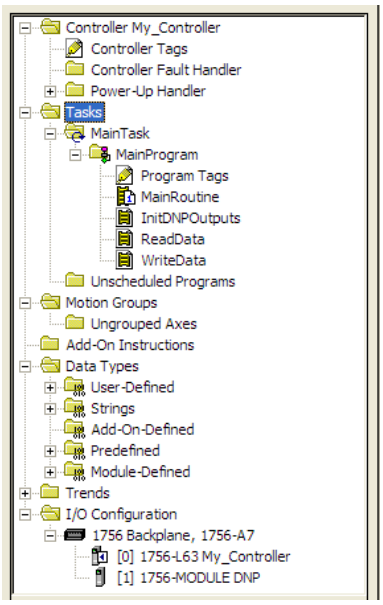
7 Add all controller tags. The MVI56-DNP module *DNPCfgStatus* tag array holds the module status data. The *DNPData* tag array holds all the DNP and IED data for each data type. Other more specialized tags and tag arrays hold data to be sent or received by the Special Functions supported by the module as well as bits and words used for sample ladder logic flow control and processing (control bits and words).



Scope: MVI56DNP Show: All Tags Enter Name Fi

Name	Data Type	Style	Description
DNP	DNPModule		
DNP.CfgStatus	DNPModuleDef		This object encapsulates all the status and scratch variables us...
DNP.Data	DNPData		This structure contains the arrays for each DNP module data type
DNP.WarmBoot	BOOL	Decimal	
DNP.ColdBoot	BOOL	Decimal	
DNP.RequestDNPTime	BOOL	Decimal	
DNP.DNPTimeSyncCLX	BOOL	Decimal	
DNP.CLXTimeSetDNP	BOOL	Decimal	
DNP.Enable_64bit_Event_70timebase	BOOL	Decimal	
DNP.Enable_64bit_Event_72timebase	BOOL	Decimal	
DNP.Enable_Binary_Input_Event_DNPTime	BOOL	Decimal	
DNP.Enable_Analog_Input_Event_DNPTime	BOOL	Decimal	
DNP.Clock	DNPClock		
DNP.EventCount	INT	Decimal	
DNP.EventMsgs	DNPEventMsg[20]		Event message recieved on the DNP master port and passed t...
DNP.CROB_count	INT	Decimal	
DNP.CROB_data	DNPCROB_Data[...		This object encapsulates the data for the DNP CROB (slave)
DNP.Set_Analog_Event	BOOL	Decimal	Triggers an analog event
DNP.Set_Binary_Event	BOOL	Decimal	Triggers a binary event
DNP.Analog_Event	DNPEvent_Analog		
DNP.Binary_Event	DNPEvent_Binary		Object to hold a binary input event to be sent to the module. T...
DNP.event_util	INT	Decimal	
DNP.event_util2	INT	Decimal	
DNP.CLXTime	DINT[7]	Decimal	Wallclock Time
DNP.SOE_BI_Events	DNP_SOE_BIEvn...		
DNP.Binary_Input_Event_DNPTime	DNPEvent_Binary...		Block 9968 Binary Input Events with DNP Time (48-bit)
DNP.Analog_Input_Event_DNPTime	DNPEvent_Analo...		Block 9969 Binary Input Events with DNP Time (48-bit)
Local:1:C	AB:1756_MODUL...		
Local:1:I	AB:1756_MODUL...		
Local:1:O	AB:1756_MODUL...		
MJFAULTS	DINT[12]	Decimal	

8 The last step is to add the ladder logic. If you are using the sample ladder logic, you may need to adjust it to fit your application. If you are not using the ladder example, copy the ladder logic from the sample into your application and make any modifications which may be needed for your application.

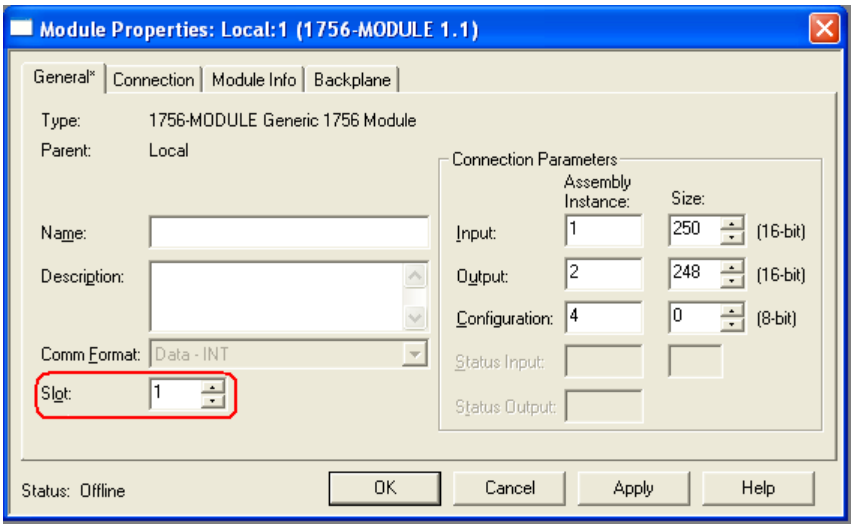


1.7.4 Selecting the Slot Number for the Module

This sample application is for a module installed in Slot 1 in a ControlLogix rack. The ladder logic uses the slot number to identify the module. If you are installing the module in a different slot, you must update the ladder logic so that program tags and variables are correct, and do not conflict with other modules in the rack.

To change the slot number

1. In the *Controller Organization* list, select the module and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **PROPERTIES**. This action opens the *Module Properties* dialog box.



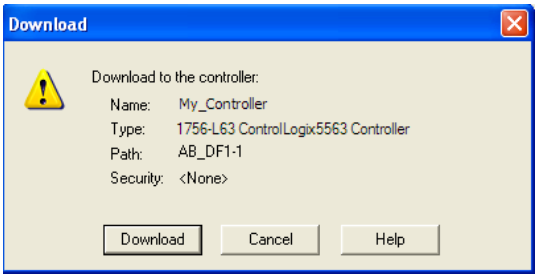
3. In the *Slot* field, use the spinners on the right side of the field to select the slot number where the module will reside in the rack, and then click **OK**.

RSLogix will automatically apply the slot number change to all tags, variables and ladder logic rungs that use the MVI56-DNP slot number for computation.

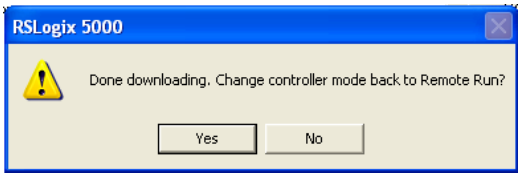
1.8 Downloading the Sample Program to the Processor

Note: The key switch on the front of the ControlLogix processor must be in the REM or PROG position.

1. If you are not already online with the processor, open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.
2. When communication is established, RSLogix 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.

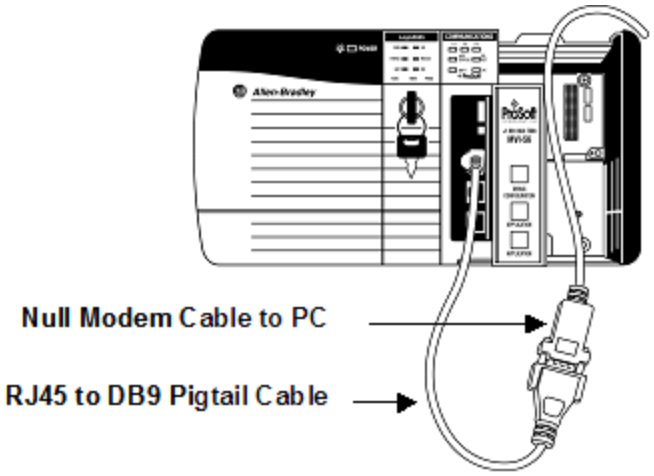


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.9 Connecting Your PC to the Module

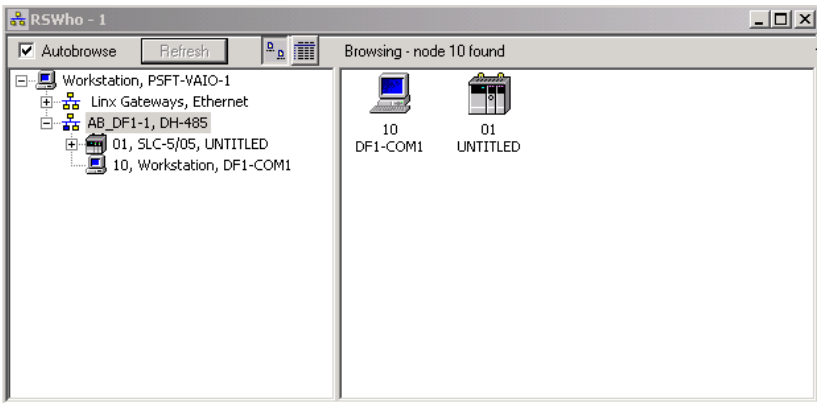
With the module securely mounted, connect your PC to the **Configuration/Debug** port using an RJ45-DB-9 Serial Adapter Cable and a Null Modem Cable.

1. Attach both cables as shown.
2. Insert the RJ45 cable connector into the *Config/Debug* port of the module.
3. Attach the other end to the serial port on your PC.

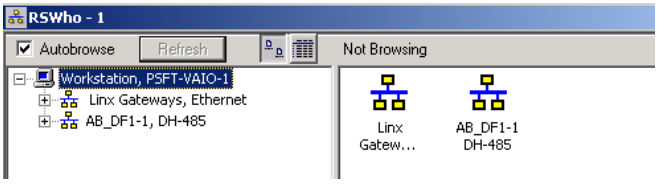


The communication port driver in *RSLinx* can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using *ProSoft Configuration Builder (PCB)*, *HyperTerminal* or another terminal emulator, follow these steps to disable the *RSLinx* driver.

1. Open *RSLinx* and go to **COMMUNICATIONS** > **RSWHO**.
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network.



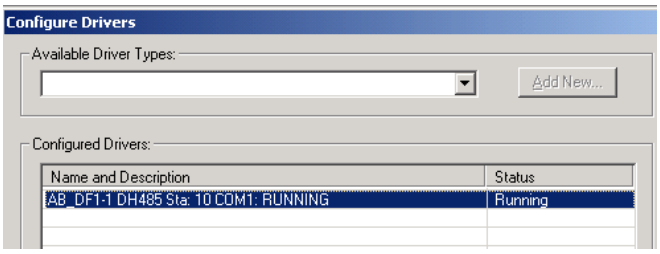
3 Notice how the DF1 driver is opened, and the driver is looking for a processor on Node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your *RSWho* screen should look like this:



Branches are displayed or hidden by clicking on the  or the  icons.

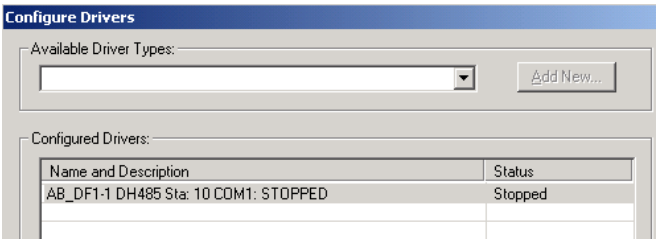
4 When you have verified that the driver is not being browsed, go to **COMMUNICATIONS > CONFIGURE DRIVERS**.

You may see something like this:



5 If you see the status as running, you will not be able to use this COM port for anything other than communication to the processor. To stop the driver press the **STOP** button on the side of the window:

6 After you have stopped the driver you will see the following.



7 You may now use the COM port to connect to the *Config/Debug* port of the module.

Note: You may need to shut down and restart your PC before it will allow you to stop the driver (usually only on *Windows NT* machines). If you have followed all of the above steps, and it will not stop the driver, then make sure you do not have *RSLogix* open. If *RSLogix* is open, you will not be able to stop the DF1 driver. If *RSLogix* is not open, and you still cannot stop the driver, then reboot your PC.

2 Configuring the MVI56-DNP Module

In This Chapter

2.1 Module Configuration File, DNP.CFG

In order for the module to operate, a configuration file (DNP.CFG) is required. This configuration file contains all the information required to configure the module's Master drivers, set up the databases for the controlled devices and establish a command list. Each parameter in the file must be set carefully in order for the application to be implemented successfully.

The configuration file is separated into sections with topic header names enclosed in the **[ ]** characters. Any record that begins with the "#" character is considered to be a comment record. Any text to the right of a # character is ignored by the program, and can be used to provide documentation within the configuration file. Liberal use of comments within the file can ease the use and interpretation of the data in the file.

The following topics describe each section of the configuration file.

Important: The configuration file must be named DNP.CFG, otherwise the configuration file will not be recognized by the module.
Important: This module supports a maximum configuration file size of 128 kilobytes (131072 bytes). If the configuration file is larger than this size, the module will not accept the download. You can reduce the size of the configuration file by opening the file in a text editor and removing comment lines (lines preceded with the # character).

2.1.1 MVI56-DNP Communication Module Configuration

[Section]/Item	Range	Description
[MODULE]		General module configuration section
Module Name:	0 to 80 characters	This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.
Module Type:		This parameter is fixed and not user-configurable. It identifies the module type when a DNP.CFG file is imported in ProSoft Configuration Builder (PCB).

[Section]/Item	Range	Description
[DNP Slave]		DNP Slave configuration information
Internal Slave ID:	0 to 65534	This is the DNP address for the module. All messages with this address from the master will be processed by the module.
Baud Rate:	Baud rate value	Primary DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400), 576 (57600), 115 (115200)
RTS On:	0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.
RTS Off:	0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.
Min Response Delay:	0 to 65535 milliseconds	Minimum time between receiving a request and transmitting a response. Allows Master time to disable transmitter on an RS-485 network.
Modem:	Yes or No	This parameter defines if a dial-up modem is used on the secondary DNP slave port. A modem cannot be used if the port is configured as a Master.
Connect Timeout:	0 to 65535	Defines the number of milliseconds to wait for the CD signal to be set high. The CD signal indicates a connection is made using a dial-up modem.
First Character Delay:	0 to 65535	Defines the number of milliseconds to wait before sending the first message after the connection is first made. This delay only applies to the first packet sent to the modem.
Redial Delay Time:	0 to 32000	Defines the minimum number of milliseconds to wait before a redial attempt is made by the slave.
Redial Random Delay:	0 to 32000	Defines a random millisecond time range to be added to the redial delay time before the modem is accessed.
Idle Timeout:	0 to 65535	Defines the number of milliseconds the modem is inactive before it will disconnect.
Phone Number:	ASCII String Data	This field contain a null-terminated, ASCII character string used by the dial-up modem. The string must contain all characters required by the modem. An example string is ATDT1800222333. Maximum length is 34 bytes including the terminating 0.
Collision Avoidance:	Yes or No	This parameter defines if collision avoidance will be utilized on the primary DNP slave port.
CD Idle Time:	0 to 32000	Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.
CD Random Time:	0 to 32000	Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.

[Section]/Item	Range	Description
CD Time Before Receive:	0 to 65535	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.
BI Class:	0 to 3	This parameter specifies the default class to be utilized for all the binary input points in the DNP database that are not defined in the override list section.
AI Class:	0 to 3	This parameter specifies the default class to be utilized for all the analog input points in the DNP database that are not defined in the override list section.
Float Class:	0 to 3	This parameter specifies the default class to be utilized for all the floating-point input points in the DNP database that are not defined in the override list section.
Double Class:	0 to 3	This parameter specifies the default class to be utilized for all the double floating-point input points in the DNP database that are not defined in the override list section.
AI Deadband:	0 to 32767	This parameter specifies the default deadband value assigned to all points not defined in the override list for the analog input point type in the DNP database.
Float Deadband:	0 to maximum float value	This parameter specifies the default deadband value assigned to all points not defined in the override list for the floating-point input point type in the DNP database.
Double Deadband:	0 to maximum double value	This parameter specifies the default deadband value assigned to all points not defined in the override list for the double floating-point input point type in the DNP database.
Select/Operate Arm Time:	1 to 65535 milliseconds	Time period after select command received in which operate command will be performed. After the select command is received, the operate command will only be honored if it arrives within this period of time.
Write Time Interval:	0 to 1440 minutes	Time interval to set the need time IIN bit (0=never), which will cause the Master to write the time. Stored in milliseconds in the module memory.
Data Link Confirm Mode:	Coded Value (N=Never, S=Sometimes, A=Always)	IED can request acknowledgement from Master station when sending data. The codes are as follows: 0=Never, 1=Sometimes, 2=Always
Data Link Confirm Tout: (Tout = Timeout)	1 to 65535 milliseconds	Time period to wait for Master Data Link confirmation of last frame sent. This time is in milliseconds. This parameter is only used if the frame is sent with confirmation requested.
Data Link Max Retry:	0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.
App Layer Confirm Tout:	1 to 65535 milliseconds	Event data contained in the last response may be sent again if not confirmed within the millisecond time period set. If application layer confirms are used with data link confirms, ensure that the application layer confirm timeout is set long enough.
Unsolicited Response:	Yes or No	Set if the slave unit will send unsolicited response messages. If set to No, the slave will not send unsolicited responses. If set to Yes, the slave will send unsolicited responses.
Class 1 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 1 required before an unsolicited response will be generated.
Class 2 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 2 required before an unsolicited response will be generated.

[Section]/Item	Range	Description
Class 3 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 3 required before an unsolicited response will be generated.
Unsol Resp Delay:	0 to 65535 milliseconds	Maximum number of 1 millisecond intervals to wait after an event occurs before sending an unsolicited response message. If set to 0, only use minimum number of events.
Uresp Master Address:	0 to 65534	DNP destination address where unsolicited response messages are sent.
Uresp Retry Count:	0 to 255 retries	Determines the number of unsolicited message retries sent on primary DNP port before changing to secondary port. If the value is 0, port switching will be disabled.
AI Events with time:	Yes or No	This parameter sets if the analog input events generated by the module will include the date and time of the event. If the parameter is set to No, the default is set to no time data. If the parameter is set to Yes, the default object will include the time of the event.
Time Sync Before Events:	Yes or No	This parameter determines if events are to be generated by the module before the time synchronization from the Master unit. If the parameter is set to No, no events will be generated until the module's time has been synchronized. If the parameter is set to Yes, events will always be generated.
Initialize DNP Database:	Y or N	This parameter determines if the module will request data from the processor to initialize the DNP database output data areas. If this option is utilized, ladder logic is required to send the requested block from the processor to the module.
Pass-Through CROB	Y or N	This parameter will pass CROB functions through to the Ladder Logic. Block 9910 wil be sent to the CLX processor with the data received for Trip/Close or Pulse CROB functions from an attached DNP Master.
Use Trip/Close Single Point	Y or N	Used for backward-compatibility with older MVI56-DNP modules. If Y (Yes), will cause Trip/Close operations to use a single point operation.

[Section]/Item	Range	Description
[DNP Slave Database]		DNP Slave Database definition
Binary Inputs:	0 to 8000 points	Number of digital input points to configure in the DNP slave device. Each point will be stored as a single bit in the module memory.
PLC Binary Inputs:	0 to 8000 points	Number of digital input points configured above that are to be obtained from the ControlLogix processor. All other binary input points must come from the attached IED units.
Analog Inputs:	0 to 500 points	Number of analog input points to configure in the DNP slave device. Each point will occupy a one word area in the module memory.
PLC Analog Inputs:	0 to 500 points	Number of analog input points configured above that are to be obtained from the ControlLogix processor. All other analog input points must come from the attached IED units.
Float Inputs:	0 to 250 points	Number of floating-point input points to configure in the DNP slave device. Each point will occupy a two-word area in the module memory.
PLC Float Inputs:	0 to 250 points	Number of floating-point input points configured above that are to be obtained from the PLC.
Double Inputs:	0 to 125 points	Number of double floating-point input points to configure in the DNP slave device. Each point will occupy a four-word area in the module memory.
PLC Double Inputs:	0 to 125 points	Number of double floating-point input points configured above that are to be obtained from the ControlLogix processor.

[Section]/Item	Range	Description
Counters:	0 to 250 points	Number of counter points to configure in the DNP slave device. Each point will occupy a two word area in the module memory. This number corresponds to the number of frozen counters. The application maps the counters to the frozen counters directly.
PLC Counters:	0 to 250 points	Number of counter points configured above that are to be obtained from the ControlLogix processor. All other counter points must come from the attached IED units.
Binary Outputs:	0 to 8000 points	Number of digital output points to configure in the DNP slave device. Each point will be stored as a single bit in the module memory.
PLC Binary Outputs:	0 to 8000 points	Number of digital output points configured above that are to be sent to the ControlLogix processor. All other binary output points will be sent to the attached IED units.
Analog Outputs:	0 to 500 points	Number of analog output points to configure in the DNP slave device. Each point will occupy a one word area in the module memory.
PLC Analog Outputs:	0 to 500 points	Number of analog output points configured above that are to be sent to the ControlLogix processor. All other analog output points will be sent to the attached IED units.
Float Outputs:	0 to 250 points	Number of floating-point output points to configure in the DNP slave device. Each point will occupy a two- word area in the module memory.
PLC Float Outputs:	0 to 250 points	Number of floating-point output points configured above that are to be sent to the ControlLogix.
Double Outputs:	0 to 125 points	Number of double floating-point output points to configure in the DNP slave device. Each point will occupy a four-word area in the module memory.
PLC Double Outputs:	0 to 125 points	Number of double floating-point output points configured above that are to be sent to the ControlLogix processor.
Enable Quality Flags	Y or N	Flag that enables Quality Flags to be reported along with DNP data. To use this functionality, load the sample ladder logic that supports the Quality Flag structures.

[Section]/Item	[redacted]	Description
[DNP Slave Binary Inputs]		DNP database binary input override values

```
# This area is to override the class assignment for binary input database
points. Enter list of points between the START and END labels.
#
# Point#   Class
START
END
```

[Section]/Item	[redacted]	Description
[DNP Slave Analog Inputs]		DNP database analog input override values

```
START
# This area is to override the class and deadband assignment for analog
input database points. Enter list of points between the START and END
labels.
#
# Point#   Class   Deadband
START
END
```

[Section]/Item		Description
[DNP Slave Float Inputs]		DNP database floating-point input override values

```
# This area is to override the class and deadband assignment for float
input database points. Enter list of points between the START and END
labels.
#
# Point#   Class   Deadband
START
END
```

[Section]/Item		Description
[DNP Slave Double Inputs]		DNP database double floating-point input override values

```
# This area is to override the class and deadband assignment for double
input database points. Enter list of points between the START and END
labels.
#
# Point#   Class   Deadband
START
END
```

[Section]/Item	Range	Description
[Secondary Port]		Definitions for secondary port on module
Type:	M or S or blank	This parameter defines the functionality of the secondary port on the module. M = emulate a DNP Master port S = back-up DNP slave port to the primary port. Any other value will disable the port.
Baud Rate:	Baud rate value	Secondary DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400), 576 (57600), 115 (115200)
RTS On:	0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.
RTS Off:	0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.
Min Response Delay:	0 to 65535 milliseconds	Minimum time between receiving a request and transmitting a response. Allows Master time to disable transmitter on an RS-485 network.
Collision Avoidance:	Yes or No	This parameter defines if collision avoidance will be utilized on the primary DNP slave port.
CD Idle Time:	0 to 32000	Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.
CD Random Time:	0 to 32000	Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.
CD Time Before Receive:	0 to 65535	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.

[Section]/Item	Range	Description
[DNP Master]		Definitions for DNP Master port if utilized.
Internal ID:	0 to 65534	This is the DNP address for the module. All messages with this address from the Master will be processed by the module.
Initialize IED Database:	Yes or No	This parameter determines if the module will request data from the processor to initialize the IED database input data areas. If this option is utilized, ladder logic is required to send the requested block from the processor to the module.
Event Messages to PLC:	Yes or No	This parameter determines if event messages received on the Master port will be sent to the processor. If this option is utilized, ladder logic must handle the 9903 blocks generated by the module.
Use IED BO Read Data	Yes or No	This parameter determines whether or not the IED BO Read data will be stored in the database (default = No).
Use IED AO Read Data	Yes or No	This parameter determines whether or not the IED AO Read data will be stored in the database (default = No).

[Section]/Item	Range	Description
[IED Database]		Database definition for DNP Master port if utilized
Binary Inputs:	0 to 8000 points	Number of binary input points contained in the IED database to transfer to the ControlLogix processor and obtained from the attached IED units..
Analog Inputs:	0 to 500 points	Number of analog input points contained in the IED database to transfer to the ControlLogix processor and obtained from the attached IED units..
Counters:	0 to 250 points	Number of counter points contained in the IED database to transfer to the ControlLogix processor and obtained from the attached IED units..
Binary Outputs:	0 to 8000 points	Number of binary output points contained in the IED database which are transferred from the ControlLogix processor and used by the attached IED units..
Analog Outputs:	0 to 500 points	Number of analog output points contained in the IED database which are transferred from the ControlLogix processor and used by the attached IED units..

[Section]/Item		Description
[DNP Master Slave List]		Definition of the IED units to communicate with the DNP Master port, if utilized

[Section]/Item		Description

This section stores information about each slave to be used by the Master port. There must be an entry in this table for each node to be used in the command list. Two of the parameters in this list are coded values:
Conf Mode ➔ 0=Never, 1=Sometimes and 2=Always (select 0)
Flags is bit coded as follows:
Bit 0 (decimal 1) ➔ Enable the slave
Bit 1 (decimal 2) ➔ Use Unsolicited messaging with this slave
Bit 2 (decimal 4) ➔ Use delay measurement with this slave
Bit 3 (decimal 8) ➔ Auto time synchronization enabled
#
Node DL Conf Conf Conf App Rsp
Address Mode Timeout Retry Timeout Flags
START
END

[Section]/Item		Description
[DNP Master Commands]		Definition of the commands to be issued to the IED units by the DNP Master port.

This section contains the list of commands to process on the Master port.
Node addresses present in the command list must have an entry in the
#DNP Slave List]. Commands with nodes not present in the list will not be
executed.
1 2 3 4 5 6 7 8 9 10
#Flags/ Node Data Data Cmd Device Point DNP DB IED DB Poll
#Enable Address Object Variation Func Address Count Address Address interval
START
END

2.1.2 Slave List

The slave list defines the IED units and their specific communication parameters for a DNP Master port. Up to 40 IED units can be defined in the module to be associated with the Master port. The structure of each row in the list is described in the following table.

Column	Variable Name	Data Range	Description	IF Error
1	DNP Slave Address	0 to 65534	This is the slave address for the unit to override the default values.	Ignore
2	Data Link Confirm Mode	Coded Value (0=Never, 1=Sometimes, 2=Always)	This value specifies if data link frames sent to the remote device require a data link confirm. This value should always be set to zero for almost all applications.	0
3	Data Link Confirm Time-out	1 to 65535 milliseconds	This parameter specifies the time to wait for a data link confirm from the remote device before a retry is attempted.	300
4	Maximum Retries for Data Link Confirm	0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.	3
5	Application Layer Response Time-out	1 to 65535 milliseconds	Time-out period the Master will wait for each response message fragment. If data link confirms are enabled, make sure the time-out period is set long enough to permit all data confirm retries.	5000

Column	Variable Name	Data Range	Description	IF Error
6	Slave Mode	Coded Value (Bit-mapped) Bit 0=Enable Bit 1=Unsol Msg Bit 2=Use DM Bit 3=Auto Time Sync	This word contains bits that define the slave mode. The slave mode defines the functionality of the slave device and can be combined in any combination. The fields have the following definitions: Bit 0 ENABLE: Determines if this slave will be used. UNSOL MSG: Causes an enabled unsolicited response message to be sent to the slave when its *RESTART IIN* bit is set. This parameter is also required for unsolicited message reporting by the IED unit. USE DM: Uses delay measurement. AUTO TIME SYNC: Time synchronization used when *NEED TIME IIN* bit set.	5

2.1.3 Command List

The command list stores the commands to be used by the DNP Master port. This list must be defined only if the DNP Master port is used. Up to 300 commands can be defined for the Master port. The structure of each row in the list is shown in the following table.

Word Offset	Definitions
0	Port/Flags
1	Slave Address
2	Object
3	Variation
4	Function
5	Address in Slave
6	Point Count
7	DNP DB Address
8	IED DB Address
9	Poll Interval

Port Flags

Bits in the Port/Flags parameter are dependent on the data type.

For Binary Input, Analog Input and Counter data points:

Port/Flags Bits	Description	Decimal Equivalent
0 to 1	Communication port (0=Internal, 2=Port 2 [Master])	0 or 2
2	Enable/Disable Command (1=Enable, 0=Disable)	4
3	RBE Flag(1=Events from IED, 0=Events by module)	8
4 to 7	Not Used	

For these data types, the qualifier used in the data request depends on the Point Count and Address in Slave fields in the command as follows:

If Point Count < 0, then use Qualifier 06h (All points, packed & -Point Count = # of points to consider)

If Address in Slave = 0 & Point Count > 0, then use Qualifier 00h or 01h (points 0 to Point Count -1)

If Address in Slave > 0 & Point Count > 0, then use Qualifier 00h or 01h (Address in Slave to Address in Slave + Point Count -1)

For Binary Output and Analog Output points:

Port/Flags Bits	Description	Decimal Equivalent
0 to 1	Communication port (0=Internal, 2=Port 2)	0 or 2
2	Enable/Disable Command (1=Enable, 0=Disable)	4
3	Poll Type (0=Poll, 1=Exception)	8
4	Data Source(0=DNP Database, 1=IED Database)	16
5 to 7	Not Used	

For these data types the qualifier used in the data request depends on the Point Count and Address in Slave fields in the command as follows:

If Address in Slave = 0 & Point Count > 0, then use Qualifier 17h or 28h (Point Count specified starting at point 0)

If Address in Slave > 0 & Point Count > 0, then use Qualifier 17h or28h (points from Address in Slave to Address in Slave + Point Count -1)

If Point Count <= 0, then ignore because this is illegal for outputs.

Slave Address

This parameter specifies the *Slave Address* of the IED device on the DNP network to which the command will be sent. The parameter has a range of 0 to 65535. The value of 65535 is reserved for broadcast messages. Verify that the slave configuration information is set up in the module for each slave defined in the command list.

Object

This parameter specifies the DNP *Object* type in the command. Valid *Objects* for the module are 1, 2, 10, 12, 20, 21, 30, 32, 40, 41, 50, 60 and 80. A value of 0 is permitted in this field for a set of special commands.

Variation

This parameter is specific to the object type selected.

Function

This parameter specifies the DNP *Function* for the command list *Object*. The *Object* type determines the value of the *Functions* permitted. For example, the only *Function* permitted for Binary Input data points is the **READ FUNCTION (FUNCTION CODE 1)**. For Counter and Output *Objects*, more functions are available.

Address In Slave

This value must be greater than or equal to zero. If it is set to a value less than zero, the command will be ignored. This parameter specifies the starting point address in the IED unit.

Point Count

This parameter defines the number of points in the IED unit that will be affected by the command. Refer to the discussion in the Command List topic, above, to interpret this parameter's meaning for the different *Object* types.

DNP DB Address

This parameter defines the starting point address in the DNP database for the command. If the parameter has a value of -1, the DNP database is not used with the point.

IED DB Address

This parameter defines the starting point address in the IED database for the command. If the parameter has a value of -1, the IED database is not used with the point.

Poll Interval

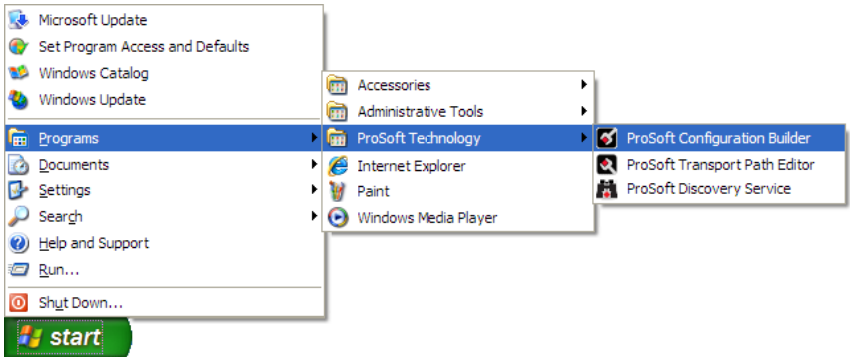
This parameter specifies the minimum frequency at which the module should execute the command. The value is entered in units of seconds. For example, to execute a command every 10 seconds, enter a value of 10 in this field. A value of 0 for the parameter implies that the command should be executed every scan of the list, as often as possible.

2.2 Using ProSoft Configuration Builder

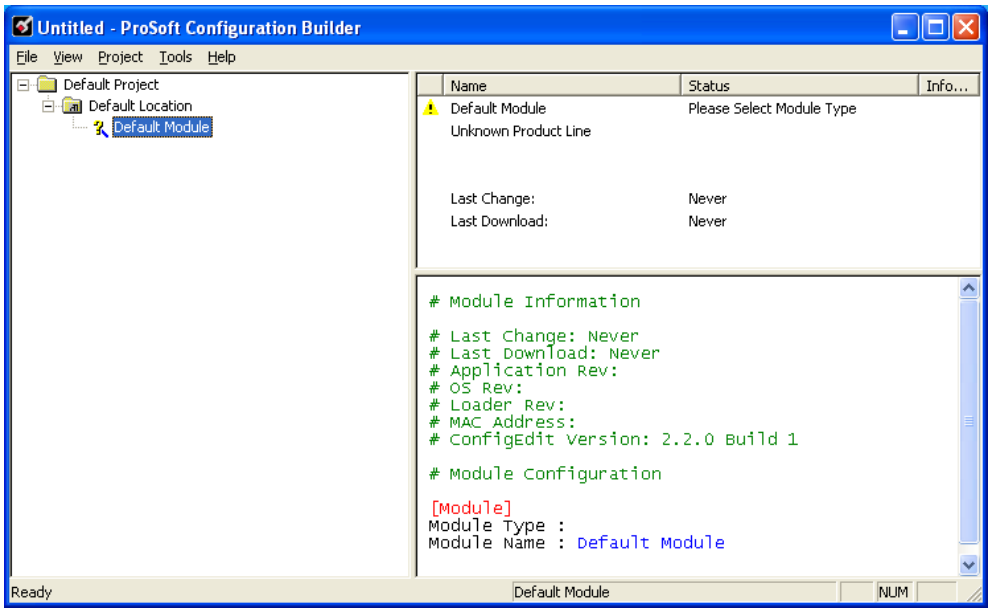
ProSoft Configuration Builder (PCB) provides a convenient way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

2.2.1 Setting Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).

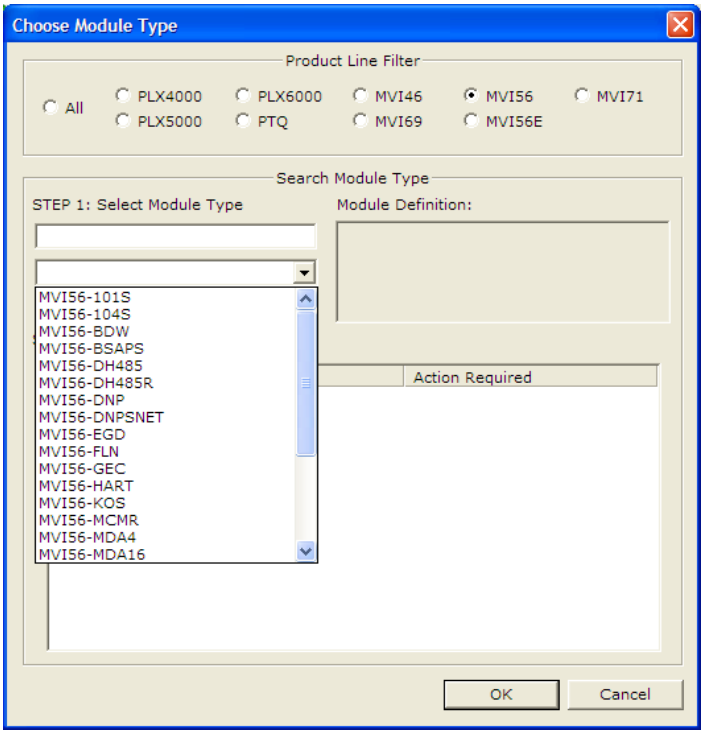


If you have used other Windows configuration tools before, you will find the screen layout familiar. *PCB*'s window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When you first start *PCB*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *PCB* window with a new project.



Adding the MVI56-DNP module to the project

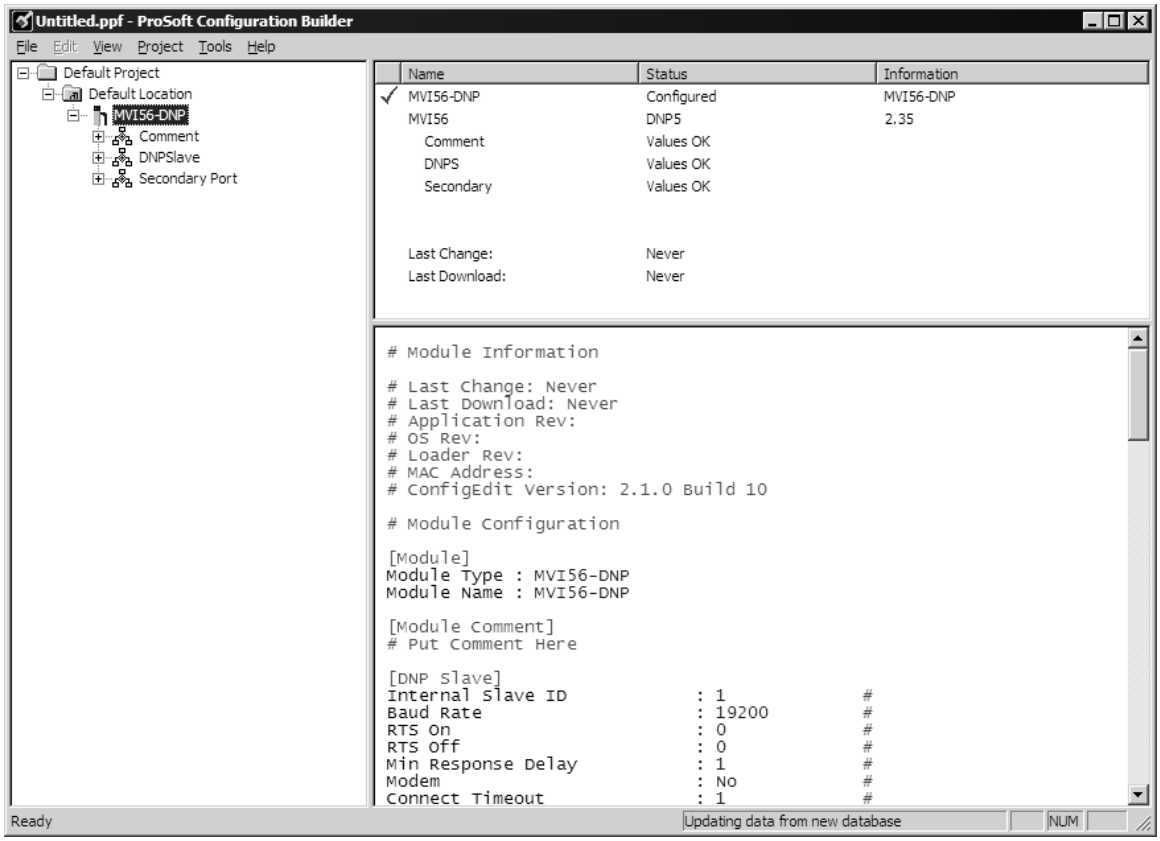
1. Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



3. In the *Product Line Filter* area of the dialog box, select **MVI56**. In the *Select Module Type* dropdown list, select **MVI56-DNP**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

2.2.2 Renaming PCB Objects

Notice that the contents of the information pane and the configuration pane changed when you added the module to the project.



At this time, you may wish to rename the *Default Project* and *Default Location* folders in the tree view.

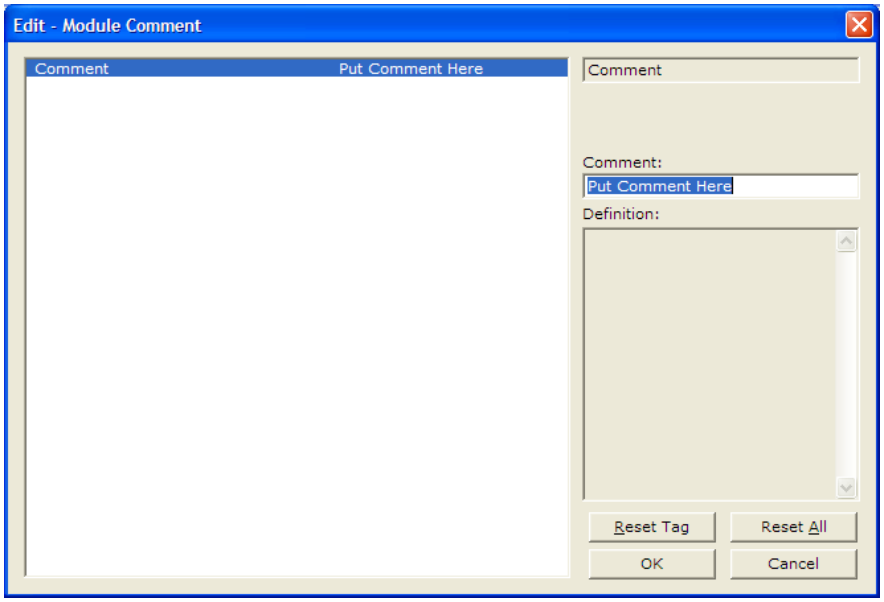
1. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2. Type the name to assign to the object.
3. Click *away* from the object to save the new name.

Configuring Module Parameters

1. Click the **[+]** sign next to the module icon to expand module information.
2. Click the **[+]** sign next to any icon to view module information and configuration options.
3. Double-click any icon to open an *Edit* dialog box.
4. To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
5. Click **OK** to save your changes.

Creating Optional Comment Entries

1. Click the **[+]** to the left of the Comment icon to expand the module comments.
2. Double-click the Module Comment icon. The *Edit - Module Comment* dialog box appears.



3. Enter your comment and click **OK** to save your changes.

Printing a Configuration File

1. Select the module icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.
3. In the *View Configuration* window, open the **FILE** menu, and choose **PRINT**. This action opens the *Print* dialog box.
4. In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK**.

3 Ladder Logic

In This Chapter

Ladder logic is required for the MVI56-DNP module to work. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

The sample ladder logic is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

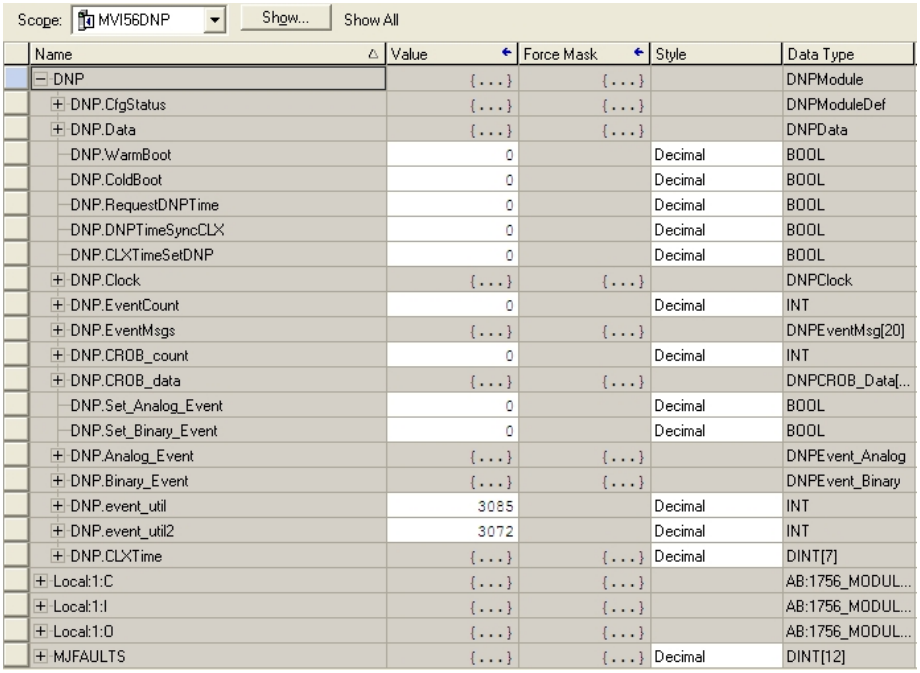
If the module's "Enable Quality Flags" parameter is set to **Yes**, additional ladder logic will be required. It can be found in the "**MVI56DNP_vxx+QualityFlags.ACD**" file.

3.1 DNP Module

The sample ladder logic relies heavily on the use of User-defined Data Types (UDTs) to help group and structure the wide variety and volume of data and control features the module offers. Lower-order UDT structures are often embedded in higher-order structures to help further organize data into more easily understood data collections.

All data and control parameters related to the MVI56-DNP are contained in twenty-one (21) User-defined Data Types (UDTs). The *DNPModule* UDT is the primary, top level data structure in which all other lower-order data types are grouped and organized. All groups branch down from this UDT.

To utilize all the features and functions of the module, an instance of each data type is required. This is accomplished by declaring controller tag variables using these data types in the Controller Tags Edit Tags dialog box. You can start with the sample ladder logic and build your application around the sample; or, you can duplicate the I/O Configuration, UDTs, Controller Tags, and Ladder Logic (Tasks) from the sample program into your existing application project.



Scope: MVI56DNP | Show... | Show All

Name	Value	Force Mask	Style	Data Type
DNP	{...}	{...}		DNPModule
DNP.CfgStatus	{...}	{...}		DNPModuleDef
DNP.Data	{...}	{...}		DNPData
DNP.WarmBoot	0		Decimal	BOOL
DNP.ColdBoot	0		Decimal	BOOL
DNP.RequestDNPTime	0		Decimal	BOOL
DNP.DNPTimeSyncCLX	0		Decimal	BOOL
DNP.CLXTimeSetDNP	0		Decimal	BOOL
DNP.Clock	{...}	{...}		DNPClock
DNP.EventCount	0		Decimal	INT
DNP.EventMsgs	{...}	{...}		DNPEventMsg[20]
DNP.CROB_count	0		Decimal	INT
DNP.CROB_data	{...}	{...}		DNPCROB_Data[...
DNP.Set_Analog_Event	0		Decimal	BOOL
DNP.Set_Binary_Event	0		Decimal	BOOL
DNP.Analog_Event	{...}	{...}		DNPEvent_Analog
DNP.Binary_Event	{...}	{...}		DNPEvent_Binary
DNP.event_util	3085		Decimal	INT
DNP.event_util2	3072		Decimal	INT
DNP.CLXTime	{...}	{...}	Decimal	DINT[7]
Local:1:C	{...}	{...}		AB:1756_MODUL...
Local:1:I	{...}	{...}		AB:1756_MODUL...
Local:1:O	{...}	{...}		AB:1756_MODUL...
MJFAULTS	{...}	{...}	Decimal	DINT[12]

Some UDTs hold process or status data *(Module Data Objects)*. This data can be monitored and manipulated by your application-specific ladder logic program. Other UDTs are used to store and organize the parameters needed for special functions and control features *(Special Data Objects)*. All of these data types will be discussed in more detail in succeeding topics.

3.2 Module Data Objects

These objects hold process and status data values. All supported DNP data types have their own UDTs and controller tags. This makes it much easier to identify and use the various data types.

3.2.1 DNPModuleDef Object

The *DNPModuleDef* object contains all the MVI56-DNP module status data.

Name	Style		Description
GenStat	DNPSlvStat		General status information
ErrList	INT[60]	Decimal	List of last 60 slave errors
IINSlaveBits	INT[40]		IIN Bits received from slaves
BP	DNPBackplane		Data to handle backplane logic

This object contains lower-order objects that organize and structure status data from the module. The *ErrList* member of this object stores a list of the 60 most recent slave errors generated by the module. This data is passed to the processor from the module by Read Block 100. Ladder logic transfers this information from the backplane transfer block into the *ErrList* controller tag array.

Read Block 100 is also used to move the slaves' Internal Indication (IIN) bits (status and error flag bits) from the module to the processor. These bits are used by the slaves to report status and error information to the module's DNP Master driver. Ladder logic transfers this data from backplane transfer Block 100 into the *IINSlaveBits* controller tag array. The relationship of which slave relates to which word in the table is determined by the order in which slaves are listed in the *DNP Master Slave List* in the configuration file, regardless of their slave ID number on the DNP network.

DNP Master Slave List Position	IINSlaveBits Controller Tag Array Element
First (1st) slave in the list	DNP.CfgStatus.IINSlaveBits[0]
Second (2nd) slave in the list	DNP.CfgStatus.IINSlaveBits[1]
Third (3rd) slave in the list	DNP.CfgStatus.IINSlaveBits[2]
(and so on)	(and so on)
Fortieth (40th) slave in the list (last slave)	DNP.CfgStatus.IINSlaveBits[39]

The IIN bits are stored in a word that follows the function code in all response messages from each slave. After ladder logic transfer, each word in the array is a bit-map holding IIN bits from one slave device. If a bit value is set to one (1), it indicates the slave is experiencing the condition or conditions as described in the following table, where [x] equals an array element number from 0 to 39.

BIT	DESCRIPTION
DNP.CfgStatus.IINSlaveBits[x].0	All stations message received. Set when a request is received with the destination address set to 0xffff. Cleared after next response. Used to let master station know broadcast received.
DNP.CfgStatus.IINSlaveBits[x].1	Class 1 data available. Set when class 1 data is ready to be sent from the slave to the master. Master should request class 1 data when this bit is set.
DNP.CfgStatus.IINSlaveBits[x].2	Class 2 data available. Set when class 2 data is ready to be sent from the slave to the master. Master should request class 2 data when this bit is set.
DNP.CfgStatus.IINSlaveBits[x].3	Class 3 data available. Set when class 3 data is ready to be sent from the slave to the master. Master should request class 3 data when this bit is set.
DNP.CfgStatus.IINSlaveBits[x].4	Time synchronization required from master. The master should write the date and time when this bit is set. After receiving the write command the bit will be cleared.
DNP.CfgStatus.IINSlaveBits[x].5	Slave digital outputs are in local control. This bit is not used in this application.
DNP.CfgStatus.IINSlaveBits[x].6	Device trouble. When this bit is set, the data reported by the module may not be that currently present in the PLC because the block transfer operation is not successful.
DNP.CfgStatus.IINSlaveBits[x].7	Device restart. This bit is set when the slave either warm or cold boots. It is cleared after a master writes a 0 to the bit.
DNP.CfgStatus.IINSlaveBits[x].8	Bad function code. The function code contained in the master request is not supported for the specified object/variation.
DNP.CfgStatus.IINSlaveBits[x].9	Requested object(s) unknown. Object requested by master is not supported by the application.
DNP.CfgStatus.IINSlaveBits[x].10	Parameters in the qualifier, range or data fields are not valid or out of range for the slave.
DNP.CfgStatus.IINSlaveBits[x].11	Event buffer(s) or other application buffers have overflowed. This bit is also set if the slave receives a multi-frame message from the master.
DNP.CfgStatus.IINSlaveBits[x].12	Request understood but requested operation is already executing. The slave will never set this bit.
DNP.CfgStatus.IINSlaveBits[x].13	Bad configuration. The slave configuration is invalid and should be re-configured. If the configuration is invalid, the slave will set the invalid parameters to default values and continue to run. Check error log using debug port.
DNP.CfgStatus.IINSlaveBits[x].14	Reserved, always 0.
DNP.CfgStatus.IINSlaveBits[x].15	Reserved, always 0.

3.2.2 DNPSlvStat Object

The *DNPSlvStat* object stores the status data passed from the module to the processor by Read Block 100. This block of data contains information that can be used to determine the "health" of the module and the tasks running.

Name	Style	Description
Cur_Port	INT	This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a slave. The possible values are detailed in the application documentation.
Last_Err	INT	This value represents the last error code transmitted to the master by this slave port.
Msg_Me	INT	This value represents the total number of message frames received by the slave, regardless of the slave address in the message.
Msg_Sent	INT	This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
Msg_Rec	INT	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.
Err_Sync	INT	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
Err_Overrun	INT	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.
Err_Length	INT	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
Err_CRC	INT	This value counts the number of times a bad CRC value is received in a message.
Err_Overflow	INT	This value counts the number of times the application layer receives a message fragment buffer which is too small.
Err_Seq	INT	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
Err_Address	INT	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.
BI_Events	INT	This value contains the total number of binary input events which have occurred.
BI_Buffer	INT	This value represents the number of binary input events which are waiting to be sent to the master.
AI_Events	INT	This value contains the total number of analog input events which have occurred.
AI_Buffer	INT	This value represents the number of analog input events which are waiting to be sent to the master.
Err_Func	INT	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
Err_Obj	INT	This value counts the number of times a request for an unsupported object is received by the slave device.

Name	Style	Description
Err_Range	INT	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
Err_MOverflow	INT	This value counts the number of times an application response message from the slave is too long to transmit.
Err_Frame	INT	This value counts the number of times the slave receives a multi-frame message from the master. **NOTE: The module currently does not support multi-frame master messages.**
Blk_Total	INT	Total BTR/BTW or side-connect interface transfers attempted by the module.
Blk_Good	INT	This value represents the total number of transfer operations between the PLC and module that are successful.
Blk_Err	INT	Total number of transfers that resulted in an error condition.
Blk_RErr	INT	Total number of BTR or write transfers that resulted in an error.
Blk_WErr	INT	Total number of BTW or read transfers that resulted in an error.
Blk_NErr	INT	Number of BTW requests that resulted in an incorrect BTW identification code.
Blk_ECntr	INT	Count of sequential data transfer errors. When this value exceeds that specified for the data transfer operation, the error flag below will be set.
Blk_EFlag	INT	This flag is used to indicate that data is not being successfully transferred between the PLC and the module. This flag corresponds to the Device Trouble IIN bit.
Cfg_Type	INT	This is a coded field that defines the configuration of the module. The codes are as follows: 0=Single Slave Configuration, 1=Dual Slave Configuration, 2=Slave/Master Configuration
Product	SINT[4]	These two words contain the product name of the module in ASCII format.
Rev	SINT[4]	These two words contain the product revision level of the firmware in ASCII format.
Op_Sys	SINT[4]	These two words contain the module's internal operating system revision level in ASCII format.
Run	SINT[4]	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
Slave_Count	INT	This is the total number of slaves configured for the DNP Master port. This may not represent the number of active slaves as it includes slaves that are not enabled.
Cmd_Count	INT	This is the total number of commands configured for the DNP Master port. This may not represent the number of active commands as it includes commands that are disabled.
Mem_Blk	INT	This value represents the number of memory allocation blocks for slave devices. This number should be one greater than the number of slave devices. The extra device is held for the broadcast device.
Mem_Frame	INT	This value represents the number of physical layer frame memory allocation blocks used by the program.
Mem_DLRec	INT	This value represents the number of receive data link layer memory blocks allocated.
Mem_DLTx	INT	This value represents the number of transmit data link layer memory blocks allocated.
Mem_AppRec	INT	This value represents the number of application layer receive memory blocks allocated.

Name	Style	Description
Mem_AppTx	INT	This value represents the number of application layer transmit memory blocks allocated.
Mem_DevErr	INT	This value represents the number of memory allocation errors for device blocks.
Mem_PhyErr	INT	This value represents the number of memory allocation errors for physical layer frame blocks.
Mem_DLRErr	INT	This value represents the number of memory allocation errors for data link layer receive blocks.
Mem_DLTErr	INT	This value represents the number of memory allocation errors for data link layer transmit blocks.
Mem_AppRErr	INT	This value represents the number of memory allocation errors for application layer receive blocks.
Mem_AppTErr	INT	This value represents the number of memory allocation errors for application layer transmit blocks.
Mstr_Sync	INT	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
Mstr_Length	INT	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
Mstr_CRC	INT	This value counts the number of times a bad CRC value is received in a message.
Scan_Count	DINT	Program scan counter
Mem_Free	DINT	Free memory in module
P1_TX_State	INT	Value of the DNP Slave state machine for transmit.
FloatEvents	INT	Total number of events generated for analog floating-point input data points.
DoubleEvents	INT	Total number of events generated for analog double, floating-point input data points.
EventQueue	INT	Number of event messages waiting to send to processor.
EvtQueueOF	INT	Flag to indicate if the event message queue has overflowed. If more than 200 event messages are received on the master port and they are not sent to the processor, this flag will be set (1). The flag will clear after the messages are sent to the processor.
Local Slave IIN Bits	INT	Local slave IIN bits being reported to controlling station when the module is acting as a DNP slave

Ladder logic is required to transfer the data sent from the module to the processor into this data object. If the ladder logic is present and the module is operating, this object can be viewed in the Controller Tags Monitor window to determine current module status.

3.2.3 DNPBackplane Object

The *DNPBackplane* object stores the variables required for backplane data transfer between the module and the processor.

Name	Style	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table
Last_Block_Write	INT	Variable for backplane block handler logic

3.2.4 DNPData Object

The *DNPData* object stores all the process-related data for a MVI56-DNP module. This includes data for the primary DNP slave port (DNP data set) and the data received from or sent to DNP slave devices (IED data set) by the secondary DNP port when configured as a DNP Master.

Contained within this data object is an array for each possible data type for each of the two data sets. The array sizes are set to match the maximum possible module configuration. If multiple MVI56-DNP modules are used within a rack, a copy of this structure will have to be made to permit each module to have its own databases.

Ladder logic is required to transfer data between the module and the processor controller tags. Each data type has its own set of unique block identification codes to distinguish the data contained in the read or write block.

Name	Style Per Block	Description	Block IDs Assigned
DNP_BI	INT[240]	DNP BI data words array	0, 1, 2
DNP_BI_QualityFlags	DNP_QualityFlag_BI[240]	DNP Binary Input Quality Flags	200-299
DNP_BO	INT[240]	DNP BO data words (Object 12 read) array	4, 5, 6
DNP_BO_Read	INT[240]	DNP BO data words (Object 10 read) array	56, 57, 58
DNP_BO_QualityFlags	DNP_QualityFlag_BO[240]	DNP Binary Output Quality Flags	300-399
DNP_Cntr	DINT[240]	DNP counter double-words array	8, 9, 10
DNP_AI	INT[240]	DNP AI data words array	12, 13, 14
DNP_AI_QualityFlags	DNP_QualityFlag_AI[240]	DNP Analog Input Quality Flags	400-499
DNP_FLTI	REAL[120]	DNP Floating-point data words array	40, 41, 42
DNP_FLTI_QualityFlags	DNP_QualityFlag_FLTI[240]	DNP Float Input Quality Flags	600-699
DNP_AO	INT[240]	DNP AO data words array	16, 17, 18
DNP_AO_QualityFlags	DNP_QualityFlag_AO[240]	DNP Analog Output Quality Flags	500-599
DNP_FLTO	REAL[120]	DNP AI data words array	48, 49, 50
DNP_FLTO_QualityFlags	DNP_QualityFlag_FLTO[240]	DNP Float Output Quality Flags	800-899
IED_BI	INT[240]	IED BI data words array	20, 21, 22
IED_BO	INT[240]	IED BO data words array	24, 25, 26
IED_Cntr	DINT[240]	IED counter double-words array	28, 29, 30
IED_AI	INT[240]	IED AI data words array	32, 33, 34
IED_AO	INT[240]	IED AO data words (Object 41 read) array	36, 37, 38
IED_AO_Read	INT[240]	IED AO data words (Object 40 read) array	60, 61, 62
DNP_DBLI	DNP_Double_ Type_Data[10]	DNP double-word input data array	44, 45, 46
DNP_DBLI_QualityFlags	DNP_QualityFlag_DBLI[240]	DNP Double Input Quality Flags	700-799
DNP_DBLO	DNP_Double_ Type_Data[10]	DNP double-word output data array	52, 53, 54
DNP_DBLO_QualityFlags	DNP_QualityFlag_DBLO[240]	DNP Double Output Quality Flags	900-999

*Quality Flags are available in the Sample Ladder "**MVI56DNP_vxxQualityFlags.ACD**".

3.2.5 DNP_Double_Type_Data

The *DNP_Double_Type_Data* object is an eight (8) element, single-byte integer (SINT) array, which can be used to create controller tags that hold 8-bytes-long data, such as 64-bit long-integer or 64-bit, high-precision, floating-point values. It is used to build the *DNP.Data.DNP_DBLI* and *DNP.Data.DNP_DBLO* input and output controller tag arrays.

3.3 Special Data Objects

These objects organize the parameters needed for some of the module's advanced features.

3.3.1 DNPClock

The *DNPClock* object is used by the controller tags that synchronize the ControlLogix clock with DNP network time.

Name	Style	Description
Year	DINT	Year returned from GSV to processor
Month	DINT	Month returned from GSV to processor
Day	DINT	Day returned from GSV to processor
Hours	DINT	Hours returned from GSV to processor
Minutes	DINT	Minutes returned from GSV to processor
Seconds	DINT	Seconds returned from GSV to processor
Microseconds	DINT	Microseconds returned from GSV to processor
Synchronized	INT	1 = time has been set by DNP master. 0 = waiting for time to be set.

3.3.2 DNPCROB

The *DNPCROB* object is used by the Special Function CROB Block 9901. This block sends a pulse output command to a single-point relay or a trip/close relay. All the parameters required for each command to be used in the block are contained in this object. Up to six of these objects can be contained in a single 9901 block command.

Name	Style	Description
Port_Flag	INT	This field is ignored in the current implementation
Slave_ID	INT	This is the DNP slave address to send the command to
Object	INT	This should always be 12
Variation	INT	This should always be 1
Function	INT	This should be 3, 5 or 6 depending on the write method
Address	INT	This is the binary output starting point to operate in the slave
Control_Code	INT	This determines the CROB operation
Pulse_Count	INT	This determines the number of pulses (0 to 255)
Pulse_On	INT	This determines the pulse on time
Pulse_Off	INT	This determines the pulse off time

3.3.3 DNPCROB_Data

The *DNPCROB_Data* object encapsulates the data for the *DNP CROB* (slave).

Name	Style	Description
Point_Number	INT	
Control_Code	SINT	
count	SINT	
On_Time	DINT	
Off_Time	DINT	

3.3.4 DNPEventMsg

The *DNPEventMsg* object stores event messages received on the DNP Master port to be passed to the processor.

Name	Style	Description
DevIndex	INT	Logical slave device index in module
IEDPoint	INT	Logical point address in IED database
DNPPoint	INT	Logical point address in DNP database
SlaveAddress	INT	Remote slave address that generated event
PointNum	INT	Point address in remote device
Object	INT	DNP object number for point
Variation	INT	DNP variation for event
LowTime	INT	Least-significant word of 48-bit DNP time
HighTime	DINT	Most-significant double-word of 48-bit DNP time
Value	DINT	Value for event

This information is passed to the processor from the module in a special read block with an identification code of 9903. Each block can send up to 20 event messages. Ladder logic must handle the receipt of this special data block and place the data received into controller tags.

3.3.5 DNPEvent_Analog

The *DNPEvent_Analog* object encapsulates all the data values used with Special Function Block 9959 to pass analog input event data from the processor to the module. Up to ten analog events can be reported in each block, so this UDT contains a ten-element array to preload the event data before sending the events from the DNP slave port to a remote DNP Master.

Name	Style	Description
Count	INT	Number of events (1 to 10)
SequenceCounter	INT	Sequence counter for each block transfer. It is used to synchronize and confirm receipt of the block by the module.
Data	DNPEvent_Analog_ Single[10]	Analog input event to be transferred to the module

3.3.6 DNP Event_Analog_DNPTime Block

rThe *DNPEvent_Analog_DNPTime* Block implements Special Function Block 9969, which transfers an analog event from the processor to the module

Name	Style	Description
EventCount	INT	Number of events in the block (1 to 10)
SequenceCounter	INT	Used to synchronize and confirm receipt of the block by the module.
Event	DNPEvent_Analog_DNPTime_Element[10]	10-element array of UDT DNPEvent_Analog_DNPTime_Element UDT

3.3.7 DNP Event_Analog_DNPTime_Element

The *DNP Event_Analog_DNPTime_Element* object is used to build the 10-element Event array used in the *DNPEvent_Analog_DNPTime_Block* object. This object encapsulates all the data associated with one analog input event.

Name	Style	Description
DNP_Analog_Input	INT	Data Point in the DNP analog input database represented by the event
Value	INT	Event input value
DNP_Time	INT[3]	DNP Time as number of miliseconds since Jan 1970
Reserved	INT	Bits 0 and 1 are used for class override values of 1 to 3

3.3.8 DNPEvent_Analog_Single

The *DNPEvent_Analog_Single* object stores the information for a single analog input event to be sent from the processor to the module in a special function block 9959. The structure shown contains all the parameters required for an analog input event.

Name	Data Type	Description
Point	INT	Data point in the DNP analog input database represented by event
Value	INT	Point value
ClassOverride	INT	Class Override Value (1 through 3)
Year	INT	Year
Month	INT	Month
Day	INT	Day
Hour	INT	Hour
Minute	INT	Minute
Seconds	INT	Seconds
Milliseconds	INT	Milliseconds

3.3.9 DNP Event_Binary

The *DNPEvent_Binary* object encapsulates all the data values used with Special Function Block 9958 to pass binary input event data from the processor to the module. Up to twelve binary events can be reported in each block, so this UDT contains a twelve-element array to preload the event data before sending the events from the DNP slave port to a remote DNP Master.

Name	Style	Description
Count	INT	Event Count (1 to 12)
SequenceCounter	INT	Used to synchronize and confirm receipt of the block by the module
Data	DNPEvent_Binary_Single[12]	Binary event data

3.3.10 DNPEvent_Binary_DNPTime_Block

The *DNPEvent_Binary_DNPTime_Block* object implements the Special Function Block 9968 for binary input time data.

Name	Style	Description
EventCount	INT	Number of events in the block (1 to 10)
SequenceCounter	INT	Used to synchronize and confirm receipt of the block by the module.
Event	DNPEvent_Binary_DNPTime_Element[10]	10-element array of UDT DNPEvent_Binary_DNPTime_Element UDT

3.3.11 DNP Event_Analog_DNPTime_Element

The *DNPEvent_Binary_DNPTime_Element* object is used to build the 10-element *Event* array used in the *DNPEvent_Binary_DNPTime_Block* object. This object encapsulates all the data associated with one binary input event.

Name	Style	Description
DNP_Binary_Input	INT	Data Point in the DNP binary input database represented by the event
Value	INT	Event input value
DNP_Time	INT[3]	DNP Time as number of miliseconds since Jan 1970
Reserved	INT	Bits 0 and 1 are used for class override values of 1 to 3

3.3.12 DNPEvent_Binary_Single

The *DNPEvent_Binary_Single* object stores the information for a single binary input event to be sent from the processor to the module in a special function block 9958. The structure shown contains all the parameters required for a binary input event.

Name	Data Type	Description
Point	INT	Data point in the DNP binary database represented by event
Value	BOOL	Point value
ClassOverride	INT	Class Override Value (1 through 3)
Year	INT	Year
Month	INT	Month
Day	INT	Day
Hour	INT	Hour
Minute	INT	Minute
Seconds	INT	Seconds
Milliseconds	INT	Milliseconds

3.3.13 DNPSlave_Err

The *DNPSlave_Err* object stores the slave status information returned from the module after a 9949 block request from the ladder logic. An array of this object should be defined to hold the status data for each slave used by the module.

Name	Style	Description
Device_Index	INT	Index in the slave array for the Master port
Slave_ID	INT	Slave address for device
Err_CRC	INT	Number of CRC errors
Err_Overflow	INT	Number of overflow errors
Err_Seq	INT	Number of sequence errors
Err_DLConf	INT	Number of data-link confirm retry errors
Err_DLCFail	INT	Number of data-link confirm failures
Err_AppResp	INT	Number of application response errors

3.3.14 DNP_SOE_BIEvntBlk

The *DNP_SOE_BIEvntBlk* object encapsulates all the data values used with Special Function Block 9961 to pass binary input event data created by a Sequence of Events (SOE) module in the processor chassis to the module. Up to 20 binary events can be reported in each block, so this UDT contains a 20-element array to preload the event data before sending the events from the DNP slave port to a remote DNP Master.

Name	Style	Description
EventCount	INT	Event Count
SeqCounter	INT	Sequence Counter
Data	DNP_SOE_BIEvntData[20]	Event array

3.3.15 DNP_SOE_BIEvntBlk

The *DNP_SOE_BIEvntData* object holds all data associated with one SOE-generated binary event and is used in the *DNP_SOE_BIEvntBlk* object to create a 20-element array used to concentrate SOE-generated event data into an easily managed data object.

Name	Style	Description
DataPoint	INT	DNP Binary Input Data Point
Time_64Bit	DINT(2)	This is the 64-Bit time
Value	INT	Value for the Event Data (0 - 1)

4 Diagnostics and Troubleshooting

In This Chapter

The module provides information on diagnostics and troubleshooting in the following forms:

- LED status indicators on the front of the module provide general information on the module's status.
- Status data contained in the module can be viewed through the Configuration/Debug port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- Status data values can be transferred from the module to processor memory and can be monitored there manually or by customer-created logic.

4.1 LED Status Indicators

The LEDs indicate the module's operating status as follows:

LED	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the DNP network on its DNP Primary slave port.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the DNP network on its Port 3. This may either be configured as a DNP Master or backup slave port.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI56-DNP is working normally.
		On	The MVI56-DNP module program has recognized a communication error on one of its DNP ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

4.1.1 Clearing a Fault Condition

Typically, if the OK LED on the front of the module turns RED for more than ten seconds, a hardware problem has been detected in the module or the program has exited.

To clear the condition, follow these steps:

1. Turn off power to the rack.
2. Remove the card from the rack.
3. Verify that all jumpers are set correctly.
4. If the module requires a Compact Flash card, verify that the card is installed correctly.
5. Re-insert the card in the rack and turn the power back on.
6. Verify correct configuration data is being transferred to the module from the ControlLogix controller.

If the module's OK LED does not turn GREEN, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Technical Support.

4.1.2 Troubleshooting

Use the following troubleshooting steps if you encounter problems when the module is powered up. If these steps do not resolve your problem, please contact ProSoft Technology Technical Support.

Processor Errors

Problem description	Steps to take
Processor fault	Verify that the module is plugged into the slot that has been configured for the module in the I/O Configuration of RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56-DNP. Verify that all modules in the rack are correctly configured in the ladder logic.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56E modules with scrolling LED display: <*Backplane Status*> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert it, and then restore power to the rack.

4.2 Reading Status Data from the Module

The MVI56-DNP module returns a Status Data block that can be used to determine the module's operating status. This data is located in the module's database in status and error controller tags. This data is transferred using backplane transfer Read Blocks with an identification code of 100.

4.2.1 Using ProSoft Configuration Builder (PCB) for Diagnostics

The *Configuration and Debug* menu for this module is arranged as a tree structure, with the *Main* menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

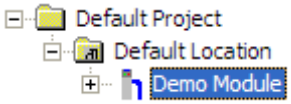
Because this is a text-based menu system, you enter commands by typing the [command letter] from your computer keyboard in the *Diagnostic* window in *ProSoft Configuration Builder (PCB)*. The module does not respond to mouse movements or clicks. The command executes as soon as you press the **[COMMAND LETTER]** – you do not need to press **[ENTER].** When you type a **[COMMAND LETTER]**, a new screen will be displayed in your terminal application.

Using the Diagnostic Window in ProSoft Configuration Builder

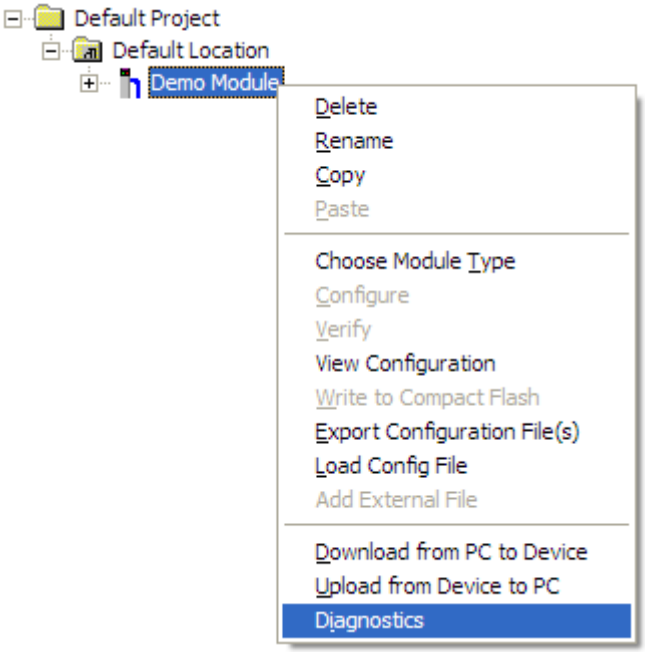
Tip: You can have a ProSoft Configuration Builder Diagnostics window open for more than one module at a time.

To connect to the module's Configuration/Debug serial port

1 Start *PCB*, and then select the module to test. Click the right mouse button to open a shortcut menu.

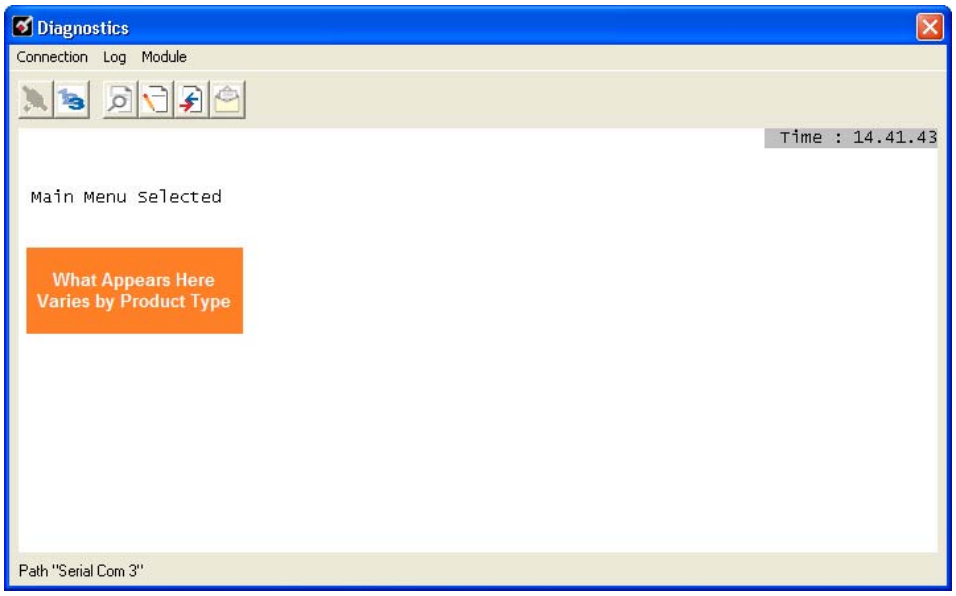


2 On the shortcut menu, choose **DIAGNOSTICS**.



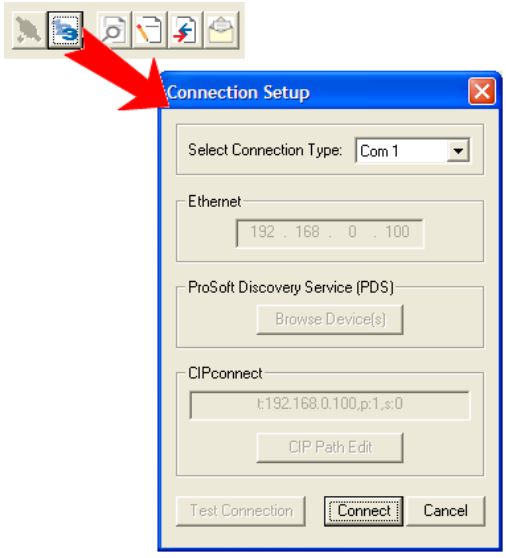
This action opens the *Diagnostics* dialog box.

3 Press **[?]** to open the *Main* menu.



If there is no response from the module, follow these steps:

1 Click to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.



2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

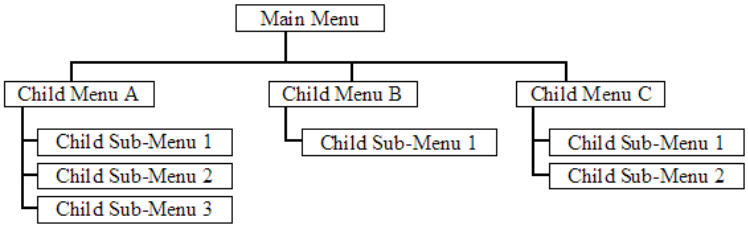
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

Navigation

All of the submenus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

The menus use a few special characters (**?**, **-**, **+**, @) that must be entered exactly as shown. Some of these characters will require you to use the **SHIFT**, **CTRL**, or **ALT** keys to enter them correctly. For example, on US English keyboards, enter the **?** command as **SHIFT** and */*.

Also, take care to distinguish the different uses for uppercase letter "eye" (**I**), lowercase letter "el" (**L**), and the number one (**1**). Likewise, uppercase letter "oh" (**O**) and the number zero (**0**) are not interchangeable. Although these characters look alike on the screen, they perform different actions on the module and may not be used interchangeably.

4.2.2 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank. To activate the main menu, press the **[?]** key on your computer's keyboard. If the module is connected properly, the following menu will appear.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P      □Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP                     OP SYS REV = 1206   PROD RUN # = 1501
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other communication failures. Use these commands only if you fully understand their potential effects, or if you are specifically directed to do so by ProSoft Technology Technical Support Engineers.
There may be some special command keys that are not listed on the menu but that may activate additional diagnostic or debugging features. If you need these functions, you will be advised how to use them by Technical Support. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Setting the Debug Level

You can increase or decrease the level of debug messages sent from the module to the Debug Menu. The following table shows the type of debugging information for each key **[0]** to **[9], [A]** to **[F]**

Key	None	DNP Statistics	Data Link Layer Messages	DPA Level Messages
0	X			
1		X		
2			X	
3		X	X	
4				
5		X		
6			X	
7		X	X	
8				X
9		X		X
A			X	X
B		X	X	X
C				X
D		X		X
E			X	X
F		X	X	X

Viewing the Error List

Press **[L]** to display the last 60 errors for the DNP slave port. Refer to the error list section of the user manual to interpret each error recorded by the module.

If there are no errors present for the module, the message "NO ERRORS FOR SYSTEM!" is displayed.

Viewing DNP Set Up & Pointers

Press **[P]** to display the memory allocation and the database setup parameters.

Viewing Operating Parameters

Press **[O]** to view the DNP Protocol setup information (Operating Parameters) for the module.

Warm Booting the Module

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other communication failures. Use these commands only if you fully understand their potential effects, or if you are specifically directed to do so by ProSoft Technology Technical Support Engineers.

There may be some special command keys that are not listed on the menu but that may activate additional diagnostic or debugging features. If you need these functions, you will be advised how to use them by Technical Support. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[R]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must cause the module to re-boot.

Viewing Comm Status

Press **[S]** to view the communication status for the DNP port.

Clearing the Error List

Press **[W]** to clear the error list. Use this command after viewing the error list (page 78) to delete the current list of errors and start a new list.

Viewing COM States

Press **[V]** to view the current state of the DNP application port and the port configuration information.

Viewing Master Port Slave Setup

Press **[T]** to view configuration information for the Master Port Slave.

Viewing Version Information

Press **[G]** to view Version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The Program Scan Counter value is incremented each time a module's program cycle is complete.

Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Opening the Class/Deadband Assignment Menu

Press **[Y]** to view the class and deadband override values for the binary, analog, float and double input DNP database.

Opening the DNP Database View Menu

Press **[U]** to open the DNP Database View Menu. This menu allows you to view the data and quality flags (if enabled) associated with the DNP Server driver. For more information about the commands on this menu, refer to DNP Database View Menu (page 83).

Receiving the Configuration File

Press **[<] (SHIFT COMMA)** to upload (receive) the current configuration file from a personal computer (PC) to the module.

Sending the Configuration File

Press **[>] (SHIFT PERIOD)** to download (send) an updated configuration file to the module from a PC.

Viewing Block Transfer Statistics

Press **[N]** from the Main Menu to view the Block Transfer Statistics screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

Tip: Repeat this command at one-second intervals to determine the number of blocks transferred each second.

Opening the Command List Menu

Press **[X]** to open the Command List menu. Use this command to view the configured command list for the module.

```
***** DNP MASTER PORT COMMAND SELECTION MENU *****
PRODUCT = DNP    REVISION = 1.00   OP SYS REV = 0900   PROD RUN # = 1501
SELECT RANGE OF COMMANDS TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS
   0 -   0-19     1 -  20-39     2 -  40-59     3 -  60-79     4 -  80-99
   5 - 100-119    6 - 120-139    7 - 140-159    8 - 160-179    9 - 180-199
   A - 200-219    B - 220-239    C - 240-259    D - 260-279    E - 280-299
```

Opening the Command Error List Menu

Press **[Z]** to open the Command Error List. This list consists of multiple pages of command list error/status data. Press **[?]** to view a list of commands available on this menu.

```
***** DNP MASTER PORT SLAVE SELECTION MENU *****
PRODUCT = DNP    REVISION = 1.00   OP SYS REV = 0900   PROD RUN # = 1501
SELECT RANGE OF SLAVES TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-SLAVES   KEY-SLAVES   KEY-SLAVES   KEY-SLAVES
   0 -  0-9     1 - 10-19    2 - 20-29    3 - 30-39 _
```

4.2.3 The Class Assignment Menu [Y]

This menu allows you to view the class and deadband override values for the binary, analog, float and double input DNP database. Press **[?]** to display the commands available on this menu.

```
CLASS ASSIGNMENT MENU
 ?=Display Menu
 0=Binary Inputs
 1=Analog Inputs
 2=Float Inputs
 3=Double Inputs
 S=Show Again
 P=Previous Page
 N=Next Page
 M=Main Menu
```

The following illustration shows the output for the Analog data set (menu key **[1]**)

```
CLASS ASSIGNMENT DISPLAY 0 TO 19 <ANALOG INPUTS>
 POINT#  CLASS  DEADBAND
      0      1      1000
      1      1      1000
      2      1      1000
      3      1      1000
      4      1      1000
      5      1      1000
      6      1      2000
      7      1      2000
      8      2      1000
      9      3      2000
     10      2      1000
     11      2      1000
     12      1      1000
     13      1      1000
     14      1      1000
     15      1      1000
     16      1      1000
     17      1      1000
     18      1      1000
     19      1      1000
```

Viewing Binary Inputs

Press **[0]** to view the override values for Binary Input Data.

Viewing Analog Inputs

Press **[1]** to view the override values for Analog Input Data.

Viewing Float Inputs

Press **[2]** to view the override values for Float Input Data.

Viewing Double Inputs

Press **[3]** to view the override values for Double Input Data.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing the Previous Page of Data

Press **[P]** to display the previous page of data.

Viewing the Next Page of Data

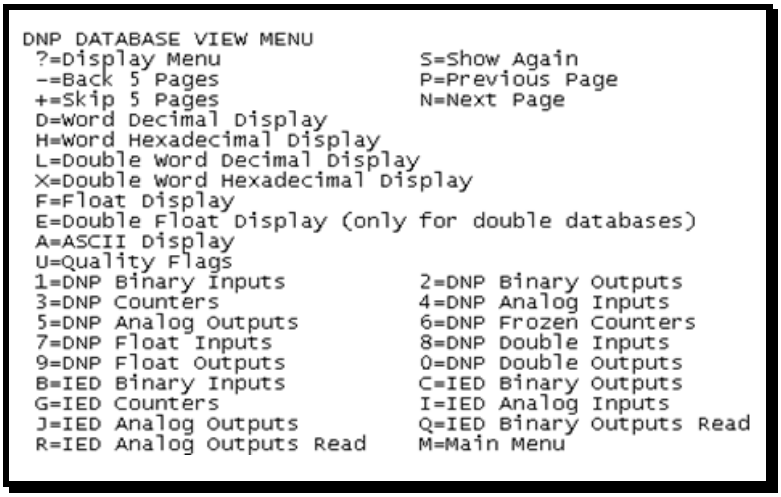
Press **[N]** to display the next page of data.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.2.4 DNP Database View Menu

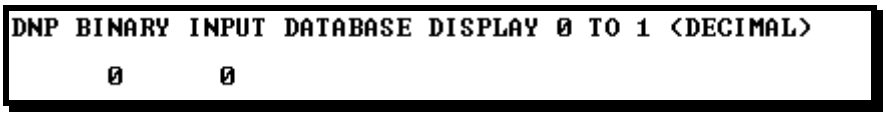
Use this menu command to view the current contents of the selected database. Press **[D]** to view a list of commands available on this menu.



Viewing Data Type Databases

Press **[D]** from the DNP menu, then hold down the **[SHIFT]** key and press the **[/]** key.

Use the number keys 1 to 6 to select the display of the data type you wish to view. For example, if the **[1]** key is pressed, the following is displayed:



Viewing Register Pages

To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again

Press **[S]** from the *Database View* menu to show the current page of registers again.

```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
   100   101   102     4     5     6     7     8     9    10
    11    12    13    14    15    16     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the *Database View* menu to skip five pages back in the database to see the 100 registers of data starting 500 registers before the currently displayed page.

Viewing the Previous Page of Registers

Press **[P]** from the *Database View* menu to display the previous page of data.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the *Database View* menu to skip five pages ahead in the database to see 100 registers of data 500 registers ahead of the currently displayed page.

Viewing the Next Page of Registers

Press **[N]** from the *Database View* menu to display the next page of data.

Viewing Data in Decimal Format

Press **[D]** from the *Database View* menu to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** from the *Database View* menu to display the data on the current page in hexadecimal format.

Viewing Data in Floating-Point Format

Press **[F]** from the *Database View* menu to display the data on the current page in floating-point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

Press **[A]** from the *Database View* menu to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Viewing Data in Double Word Decimal Format

Press **[L]** to display the data on the current page in Double Word Decimal format. This is useful for regions of the database that contain Double Word Decimal data.

Viewing Data in Double Word Decimal Format

Press **[X]** to display the data on the current page in Double Word Hexadecimal format. This is useful for regions of the database that contain Double Word Hexadecimal data.

Viewing Quality Flags

Press **[U]** to view the DNP point Quality Flags.

```
Quality Flags Menu Selected

QUALITY FLAGS MENU
 ?=Display Menu              S=Show Again
 -=Back 5 Pages              P=Previous Page
 +=Skip 5 Pages              N=Next Page
 1=DNP Binary Inputs         2=DNP Binary Outputs
 3=DNP Analog Inputs         4=DNP Analog Outputs
 5=DNP Float Inputs          6=DNP Double Inputs
 7=DNP Float Outputs         8=DNP Double Outputs
 M=Main Menu
```

Viewing DNP Binary Inputs

Press **[1]** to view a list of DNP Binary Inputs.

Viewing DNP Binary Outputs

Press **[2]** to view a list of DNP Binary Outputs.

Viewing DNP Counters

Press **[3]** to view a list of DNP Counters.

Viewing DNP Analog Inputs

Press **[4]** to view a list of DNP Analog Inputs.

Viewing DNP Analog Outputs

Press **[5]** to view a list of DNP Analog Outputs.

Viewing DNP Frozen Counters

Press **[6]** to view a list of DNP Frozen Counters.

Viewing DNP Float Inputs

Press **[7]** to view a list of DNP Float Inputs.

Viewing DNP Float Outputs

Press **[9]** to view a list of DNP Float Outputs.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.2.5 DNP Master Command List Menu

Use this menu to view the command list for the module. Press **[?]** to view a list of commands available on this menu.

```
***** DNP MASTER PORT COMMAND SELECTION MENU *****
PRODUCT = DNP    REVISION = 1.00   OP SYS REV = 0900   PROD RUN # = 1501
SELECT RANGE OF COMMANDS TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS   KEY-COMMANDS
   0 -   0-19     1 -  20-39     2 -  40-59     3 -  60-79     4 -  80-99
   5 - 100-119    6 - 120-139    7 - 140-159    8 - 160-179    9 - 180-199
   A - 200-219    B - 220-239    C - 240-259    D - 260-279    E - 280-299
```

Use keys **[0]** through **[E]** to view each range of commands. The following illustration shows the status of Command 0.

```
***** DNP MASTER PORT COMMANDS *****
PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
COMMAND INFORMATION (FIRST COMMAND INDEX =0)
  CMD PORT   DEV                  PNT POINT   DNP   IED  POLL  LAST  LAST
 NUMB /FLG  ADDR OBJ# VAR FUNC  ADDR COUNT  ADDR  ADDR  INTV  POLL ERROR
    0   31     2   12   1    6     0    10    -1     0     0     0 65535
```

Refer to Command List (page 45) for a description of the fields on this list.

- The Last Poll field is the count timer compared to the user configured poll interval. When the Last Poll value is >= to the poll interval, the command is ready to execute.
- The Last Error field contains the value 65535 when the next command is being executed. Refer to Command Error Codes for an explanation of other values that may appear in this field.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing the Previous 50 Commands

Press **[-]** to view the previous 50 commands.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next 50 Commands

Press **[+]** to view the next 50 commands from the Master command list.

Viewing the Next Page of Commands

Press **[N]** to display the next page of commands.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.2.6 DNP Master Command Error List Menu

Use this menu to view the command error list for the module. Press **[?]** to view a list of commands available on this menu.

```
***** DNP MASTER PORT SLAVE SELECTION MENU *****
PRODUCT = DNP    REVISION = 1.00   OP SYS REV = 0900   PROD RUN # = 1501
SELECT RANGE OF SLAVES TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-SLAVES   KEY-SLAVES   KEY-SLAVES   KEY-SLAVES
   0 -  0-9     1 - 10-19    2 - 20-29    3 - 30-39 _
```

Use keys **[0]** through **[3]** to view the command list for each group of slaves. The following illustration shows the status of slaves 0 and 1.

```
***** DNP MASTER PORT SLAVE STATISTICS *****
PRODUCT = DNP    REVISION = 1.00   OP SYS REV = 0900   PROD RUN # = 1501
ERROR COUNTERS FOR EACH DEVICE (FIRST DEVICE =0)
  DEVICE  DEVICE     BAD  BUFFER    TRANSPORT  CONFIRM  CONFIRM   NO APPL.
  INDEX   ADDRESS    CRC  OVERFLOW  SEQUENCE#  RETRIES  FAILURES  RESPONSE
      0     65535      0         0          0        0         0         0
      1         2      0         0          0        0         0         0
```

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Moving Back Through 5 Pages of Commands

Press **[-]** to display data for last 5 page commands.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Moving Forward (Skipping) Through 5 Pages of Commands

Press **[+]** to display data for the next page of commands.

Viewing the Next Page of Commands

Press **[N]** to display the next page of commands.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.3 Error Status Table

The program maintains an error/status table. This table of data is transferred to the ControlLogix processor automatically through Read Block 100. Ladder logic should be programmed to accept this block of data and place it in the module's controller tag. You can use the error/status data to determine the "health" of the module.

Word	Block Offset	Variable Name	Description
0	2	Current DNP Slave Port status	This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a slave. The possible values are described in the application documentation.
1	3	DNP Slave Port last transmitted error code	This value represents the last error code transmitted to the Master by this slave port.
2	4	DNP Slave Port total number of message frames received by slave	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.
3	5	DNP Slave Port total number of response message frames sent from slave	This value represents the number of good (non-error) responses that the slave has sent to the Master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
4	6	DNP Slave Port total number of message frames seen by slave	This value represents the total number of message frames received by the slave, regardless of the slave address.
5	7	DNP Slave synchronization error count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
6	8	DNP Slave overrun error count (Physical Layer Error)	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.
7	9	DNP Slave length error count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
8	10	DNP Slave bad CRC error (Data Link Layer Error)	This value counts the number of times a bad CRC value is received in a message.
9	11	DNP Slave user data overflow error (Transport Layer Error)	This value counts the number of times the application layer receives a message fragment buffer which is too small.
10	12	DNP Slave sequence error (Transport Layer Error)	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
11	13	DNP Slave address error (Transport Layer Error)	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.

Word	Block Offset	Variable Name	Description
12	14	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
13	15	DNP Slave Binary Input Event count in buffer	This value represents the number of binary input events which are waiting to be sent to the Master.
14	16	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
15	17	DNP Slave Analog Input Event count in buffer	This value represents the number of analog input events which are waiting to be sent to the Master.
16	18	DNP Slave bad function code error (Application Layer Error)	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
17	19	DNP Slave object unknown error (Application Layer Error)	This value counts the number of times a request for an unsupported object is received by the slave device.
18	20	DNP Slave out of range error (Application Layer Error)	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
19	21	DNP Slave message overflow error (Application Layer Error)	This value counts the number of times an application response message from the slave is too long to transmit.
20	22	DNP Slave multi-frame message from DNP Master error (Application Layer Error)	This value counts the number of times the slave receives a multi-frame message from the Master. The application does not support multi-frame Master messages.
21	23	Total blocks transferred	Total BTR/BTW or side-connect interface transfers attempted by the module.
22	24	Successful blocks transferred	This value represents the total number of transfer operations between the ControlLogix processor and module that are successful.
23	25	Total errors in block transfer	Total number of transfers that resulted in an error condition.
24	26	Total BTR or write errors	Total number of BTR or write transfers that resulted in an error.
25	27	Total BTW or read errors	Total number of BTW or read transfers that resulted in an error.
26	28	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
27	29	Continuous block error counter	Count of sequential data transfer errors. When this value exceeds that specified for the data transfer operation, the error flag below will be set.
28	30	Reserved	Not used
29	31	Configuration Type	This is a coded field that defines the configuration of the module. The codes are as follows: 0=Single Slave Configuration, 1=Dual Slave Configuration, 2=Slave/Master Configuration

Word	Block Offset	Variable Name	Description
30 to 31	32 to 33	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
32 to 33	34 to 35	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
34 to 35	36 to 37	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
36 to 37	38 to 39	Production Run Number (ASCII)	These two words contain the production "batch" number for the particular chip in the module in ASCII format.
38	40	DNP Master Port Slave Count	This is the total number of slaves configured for the DNP Master port. This may not represent the number of active slaves as it includes slaves that are not enabled.
39	41	DNP Master Port Command Count	This is the total number of commands configured for the DNP Master port. This may not represent the number of active commands as it includes commands that are disabled.
40	42	DNP Master Port Device Memory Block Count	This value represents the number of memory allocation blocks for slave devices. This number should be one greater than the number of slave devices. The extra device is held for the broadcast device.
41	43	DNP Master Port Frame Block Count	This value represents the number of physical layer frame memory allocation blocks used by the program.
42	44	DNP Master Port Data Link Receive Block Count	This value represents the number of receive data link layer memory blocks allocated.
43	45	DNP Master Port Data Link Transmit Block Count	This value represents the number of transmit data link layer memory blocks allocated.
44	46	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer receive memory blocks allocated.
45	47	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer transmit memory blocks allocated.
46	48	DNP Master Port Device Memory Allocation Error Count	This value represents the number of memory allocation errors for device blocks.
47	49	DNP Master Port Physical Layer Memory Allocation Error Count	This value represents the number of memory allocation errors for physical layer frame blocks.
48	50	DNP Master Port Data Link Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer receive blocks.
49	51	DNP Master Port Data Link Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer transmit blocks.

Word	Block Offset	Variable Name	Description
50	52	DNP Master Port Application Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer receive blocks.
51	53	DNP Master Port Application Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer transmit blocks.
52	54	DNP Master Synchronization Error Count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
53	55	DNP Master Length Error Count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
54	56	DNP Master Bad CRC Error Count (Physical Layer Error)	This value counts the number of times a bad CRC value is received in a message.
55	57	Scan Counter LSB	Program scan counter
56	58	Scan Counter MSB	
57	59	Free Memory LSB	Free memory in module
58	60	Free Memory MSB	
59	61	DNP Slave Port Transmit State	Value of the DNP Slave state machine for transmit.
60	62	DNP Float Event Count	Total number of events generated for analog floating-point input data points.
61	63	DNP Double Event Count	Total number of events generated for analog double, floating-point input data points.
62	64	Event Message Queue Count	Number of event messages waiting to send to processor.
63	65	Event Message Queue Overflow	Flag to indicate if the event message queue has overflowed. If more than 200 event messages are received on the Master port and they are not sent to the processor, this flag will be set (1). The flag will clear after the messages are sent to the processor.
64	66	Slave IIN Bits	Local slave IIN bits being reported to controlling station
65 to 77	67 to 79	Reserved	Future Use
78	80	Error_List[0]	First value in error list
79	81	Error_List[1]	Second value in error list
-	-	-	-
137	139	Error_List[59]	Last value in error list
138	140	Slave[0]	IIN Bits received from slave
139	141	Slave[1]	IIN Bits received from slave
-	-	-	-
177	179	Slave[39]	IIN Bits received from slave

4.4 Internal Indication Word

First Byte

Bit	Description
0	All stations message received. Set when a request is received with the destination address set to 0xffff. Cleared after next response. Used to let Master station know broadcast received.
1	Class 1 data available. Set when class 1 data is ready to be sent from the slave to the Master. Master should request class 1 data when this bit is set.
2	Class 2 data available. Set when class 2 data is ready to be sent from the slave to the Master. Master should request class 2 data when this bit is set.
3	Class 3 data available. Set when class 3 data is ready to be sent from the slave to the Master. Master should request class 3 data when this bit is set.
4	Time synchronization required from Master. The Master should write the date and time when this bit is set. After receiving the write command the bit will be cleared.
5	Slave digital outputs are in local control. This bit is not used in this application.
6	Not Used
7	Device restart. This bit is set when the slave either warm or cold boots. It is cleared after a Master writes a 0 to the bit.

Second Byte

Bit	Description
0	Bad function code. The function code contained in the Master request is not supported for the specified object/variation.
1	Requested object(s) unknown. Object requested by Master is not supported by the application.
2	Parameters in the qualifier, range or data fields are not valid or out of range for the slave.
3	Event buffer(s) or other application buffers have overflowed. This bit is also set if the slave receives a multi-frame message from the Master.
4	Request understood but requested operation is already executing. The slave will never set this bit.
5	Bad configuration. The slave configuration is invalid and should be re-configured. If the configuration is invalid, the slave will set the invalid parameters to default values and continue to run. Check error log using debug port.
6	Reserved, always 0.
7	Reserved, always 0.

4.5 Module Error Codes

If the module's program encounters a DNP protocol-related error during execution, it will log the error to the error list. This list is transferred to the ControlLogix processor using block identification code 100 (see section above) in at offsets 62 to 119. This data is also available for viewing on the debug monitor port. The following tables list the error codes generated by the program with their associated description. Use the errors to help define where problems exist in the system.

4.5.1 Slave Port Communication Errors

Error Code	Name	Description
0	OK	The module is operating correctly and there are no errors.
10	DNP synchronization error (Physical Layer Error)	Extra bytes are received before the start bytes (0x05 and 0x64).
11	DNP overrun error (Physical Layer Error)	Mainline Data Link Layer routine could not read data received on DNP port before it was overwritten.
12	DNP length error (Physical Layer Error)	Length of message does not match length value in message.
13	DNP bad CRC error (Data Link Layer Error)	Computed CRC value for message does not match that received in message.
14	DNP user data overflow error (Transport Layer Error)	Application layer received a message fragment buffer which is too small.
15	DNP sequence error (Transport Layer Error)	Sequence numbers of multi-frame request fragments do not increment correctly.
16	DNP address error (Transport Layer Error)	Source addresses contained in multi- frame request fragments do not match.
17	DNP bad function code error (Application Layer Error)	Function code received from DNP Master is not supported for selected object/variation.
18	DNP object unknown error (Application Layer Error)	Slave does not have the specified objects or there are no objects assigned to the requested class.
19	DNP out of range error (Application Layer Error)	Qualifier, range or data fields are not valid or out of range for the selected object/variation.
20	DNP message overflow error (Application Layer Error)	Application response buffer overflow condition. The response message from the slave is too long to transmit.
21	DNP Master multi-frame message error (Application Layer Error)	Received a multi-frame message from the DNP Master. This application does not support multi-frame messages from the Master.

4.5.2 System Configuration Errors

Error Code	Name	Description
100	Too many binary input points	Too many binary input points are configured for the module. Maximum value is 15360.
101	Too many binary output points	Too many binary output points are configured for the module. Maximum value is 15360.
102	Too many counter points	Too many counter points are configured for the module. Maximum value is 480.
103	Too many analog input points	Too many analog input points are configured for the module. Maximum value is 960.
104	Too many analog output points	Too many analog output points are configured for the module. Maximum value is 960.
105	Too many binary input events	Too many binary input events are configured for the module. Maximum value is 400.
106	Too many analog input events	Too many analog input events are configured for the module. Maximum value is 400.
107	Invalid analog input deadband	Deadband value for analog input events is out of range. Value must be in the range of 0 to 32767.
108	Not enough memory	There is not enough memory in the module to configure the module as specified.
109	Invalid block transfer delay for blocks 251 and 252 (error/status blocks)	Block transfer delay value specified is too low.
110	File count invalid	The file count must be in the range of 0 to 6.
111	Invalid file record size	The file record size must be in the range of 1 to 120.
112	Invalid block identification code for file	The file block transfer code must be in the range of 100 to 120.

4.5.3 DNP Port Configuration Errors

Error Code	Name	Description
212	Invalid DNP address	The DNP address specified in the configuration is not valid (0 to 65534).
213	Invalid DNP port baud rate	The baud rate code specified in the configuration is not valid.
219	Invalid DNP data link layer confirm mode	The data link confirmation mode code is not valid in the configuration.
220	Invalid DNP data link confirm time-out	The data link time-out period specified in the configuration is 0. It must be an integer in the range of 1 to 65535.
222	Invalid DNP select/operate arm time duration	The select/operate arm timer is set to 0. It must be an integer in the range of 1 to 65535.
223	Invalid DNP application layer confirm time- out	The application layer confirm time-out value is set to 0. It must be an integer in the range of 1 to 65535.
224	Invalid DNP write time interval	The write time interval is not in the data range in the configuration. The value must be in the range of 0 to 1440.
225	Invalid DNP unsolicited response mode	The unsolicited response mode code is not valid in the configuration.
226	Invalid DNP unsolicited response minimum quantity for Class 1	The unsolicited response minimum quantity for Class 1 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
227	Invalid DNP unsolicited response minimum quantity for Class 2	The unsolicited response minimum quantity for Class 2 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
228	Invalid DNP unsolicited response minimum quantity for Class 3	The unsolicited response minimum quantity for Class 3 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
230	Invalid DNP unsolicited response destination address	The unsolicited response destination address is not valid in the configuration. Value must be in the range of 1 to 65534.

4.6 Command Error Codes

4.6.1 General Command Errors

Error Code	Name	Description
1000	Device index invalid	The device index in the request or response message is not found in the slave list.
1001	Duplicate request in application layer queue	The newly submitted message to the application layer already exists in the queue. The message is ignored.
1002	COM port device removed from system	The communication port for the message has been uninstalled on the system. This error should never occur as the communication ports are only uninstalled when the module's program is terminated.
1003	Sequence number error	The application sequence number in the response message does not match that based on the last request message. This indicates application layer messages are received out of order.
1004	Response to select before operate does not match	The select response message received from the slave module is not that expected from the last select request. This indicates a synchronization problem between the Master and slave devices.
1005	Response does not contain date/time object	The response message from the slave device does not contain a date/time object. The Master expects this object for the response message.
1006	Time-out condition on response	The slave device did not respond to the last request message from the Master within the time-out set for the IED device. The application layer time-out value is specified for each IED unit in the slave configuration table in the module. This table is established each time the module performs the restart operation.
1007	Function code in application layer message not supported	The function code returned in the response message is not valid for the application layer or not supported by the module.
1008	Read operation not supported for object/variation	The application layer response message contains an object that does not support the read function.
1009	Operate function not supported for the object/variation	The application layer response message contains an object that does not support the operate function.
1010	Write operation not supported for the object/variation	The application layer response message contains an object that does not support the write function.

4.6.2 Application Layer Errors

Error Code	Name	Description
1000	Device index invalid	The device index in the request or response message is not found in the slave list.
1001	Duplicate request in application layer queue	The newly submitted message to the application layer already exists in the queue. The message is ignored.
1002	COM port device removed from system	The communication port for the message has been uninstalled on the system. This error should never occur as the communication ports are only uninstalled when the module's program is terminated.
1003	Sequence number error	The application sequence number in the response message does not match that based on the last request message. This indicates application layer messages are received out of order.
1004	Response to select before operate does not match	The select response message received from the slave module is not that expected from the last select request. This indicates a synchronization problem between the Master and slave devices.
1005	Response does not contain date/time object	The response message from the slave device does not contain a date/time object. The Master expects this object for the response message.
1006	Time-out condition on response	The slave device did not respond to the last request message from the Master within the time-out set for the IED device. The application layer time-out value is specified for each IED unit in the slave configuration table in the module. This table is established each time the module performs the restart operation.
1007	Function code in application layer message not supported	The function code returned in the response message is not valid for the application layer or not supported by the module.
1008	Read operation not supported for object/variation	The application layer response message contains an object that does not support the read function.
1009	Operate function not supported for the object/variation	The application layer response message contains an object that does not support the operate function.
1010	Write operation not supported for the object/variation	The application layer response message contains an object that does not support the write function.

5 Reference

In This Chapter

5.1 Product Specifications

The MVI56-DNP allows Rockwell Automation ControlLogix I/O compatible processors to interface easily with other DNP protocol compatible devices. The module supports DNP Subset Level 2 features and some of the Level 3 features.

The MVI56 DNP 3.0 Master/Slave Communication Module is a single-slot, backplane-compatible DNP 3.0 interface solution for the ControlLogix platform. This module provides highly configurable support of both DNP 3.0 Master and slave implementations (level 2 minimum), allowing the many SCADA and field devices supporting the DNP protocol to be integrated into the powerful ControlLogix platform.

The module supports DNP Subset Level 2 features and some of the Level 3 features, allowing the many SCADA and field devices supporting the DNP protocol to be integrated into the ControlLogix platform. The module acts as an input/output module between the DNP network and the ControlLogix backplane. The data transfer from the ControlLogix processor is asynchronous from the actions on the DNP network. Databases are user-defined and stored in the module to hold the data required by the protocol.

5.1.1 General Specifications

- Single Slot - 1756 backplane compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module.
- Ladder Logic is used for data transfer between module and processor. Sample ladder file included.
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included
- Local or remote rack

5.1.2 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 Vdc 3 mA @ 24 Vdc
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational Vibration: 5 g from 10 Hz to 150 Hz
Relative Humidity	5% to 95% (without condensation)
LED Indicators	Module Status Backplane Transfer Status Application Status Serial Activity
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Application ports (PRT1 & PRT2) Full hardware handshaking control, providing radio, modem and multi-drop support	
Software configurable communication parameters	Baud rate: 110 to 115,200 baud, depending on protocol RS-232, 485 and 422 Parity: none, odd or even Data bits: 5, 6, 7, or 8 Stop bits: 1 or 2 RTS on/off delay: 0 to 65535 milliseconds
App Ports (P1, P2) (Serial modules)	RJ45 (DB-9M with supplied cable) RS-232 handshaking configurable 500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port 6-foot RS-232 configuration cable

5.1.3 Functional Specifications

The module has two DNP protocol ports that can be user-configured to operate either in a Master/Slave or in a Slave/Redundant Slave configuration.

User-defined internal register space is accessible to the protocol driver and to the ControlLogix processor.

DNP 3.0 Slave Protocol Specifications

The DNP Slave port(s) accepts DNP commands to control and monitor data stored in the module's DNP Slave database. If a DNP Master port is also configured, a portion of these slave database can be derived from or can control IED devices connected to the DNP Master port.

- Report-by-Exception data is logged to the module's database
- Supports unsolicited messaging
- Each DNP point type is user-configurable by point
- Class assignments are completely user-definable on a Type and point basis (BI, AI, FI, DI point types)
- Supports clock synchronization from a remote Master or from the processor
- Up to 400 events are stored for Floats, Binary In, Analog In and Double Inputs
- Collision avoidance algorithm per DNP organization for redundant port switching (redundant slave mode)
- Special modem AT command string and timing support for dialing out on redundant port (redundant slave mode)

DNP 3.0 Master Protocol Specifications

The DNP 3.0 Master port can be configured as a virtual DNP Master device that actively issues user-defined DNP commands to nodes on the network.

- The Master port supports 300 user defined commands, each one containing its own set of data link and application layer characteristics
- Master port logically supports up to 40 slave devices
- Individual command configuration includes conditional or continuous polling and Poll Delay Time
- Slave status and Command status available for transfer to the processor
- Event data received from the slave devices updates the module database with the latest data values. Optionally date and time stamped data can be passed to the processor through a special block 9903 that is enabled/disabled through the parameter **Event Messages to PLC** within the configuration of the module. When this option is used, events from the attached slaves are passed to an array in the ladder logic containing the event data (slave device, point index, point value) as well as the time stamp of the event from the attached slave device (value is presented as the 48 bit DNP time).
- Special command handling for Digital Output CROB under processor control for pulse output control

DNP 3.0 ports (PRT1 & PRT2)

- User-definable module memory usage
- Support for the storage and transfer of all DNP data types across the backplane
- Communication parameters
  - Address: 0 to 65534 (slave mode)
  - Baud rate: 110 to 115K
  - Parity: none, data bits: 8, Stop bit: 1
  - RTS on delay: 0 to 65535 milliseconds
  - RTS off delay: 0 to 65535 milliseconds

5.2 Functional Overview

5.2.1 General Concepts

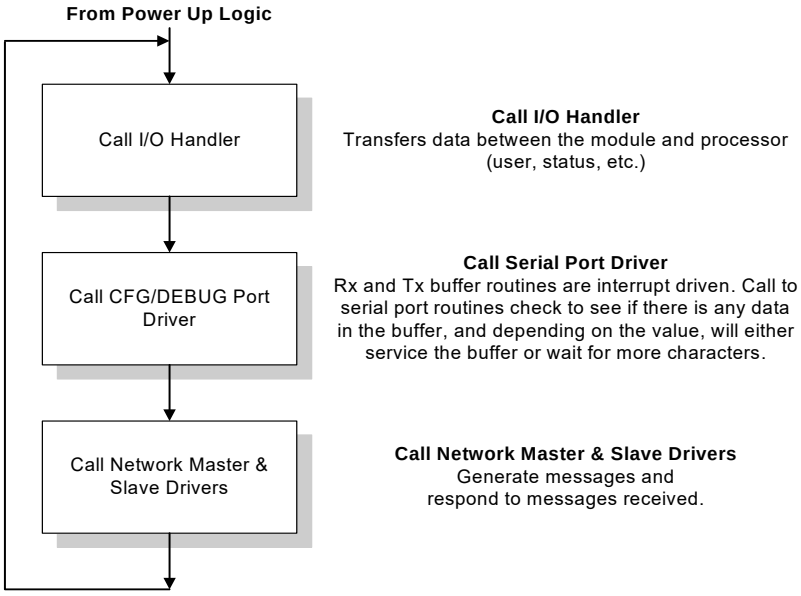
The following topics describe several concepts that are important for understanding the operation of the MVI56-DNP module.

1. On power up the module begins performing the following logical functions:
2. Initialize hardware components
   - Initialize ControlLogix backplane driver
   - Test and Clear all RAM
   - Initialize the serial communication ports
3. Reads configuration from the DNP.CFG file
4. Allocate and initialize Module Register space
5. Enable Slave Driver on selected ports
6. Enable Master Driver on selected port if configured

After the module has received the Module Configuration, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the following functions:

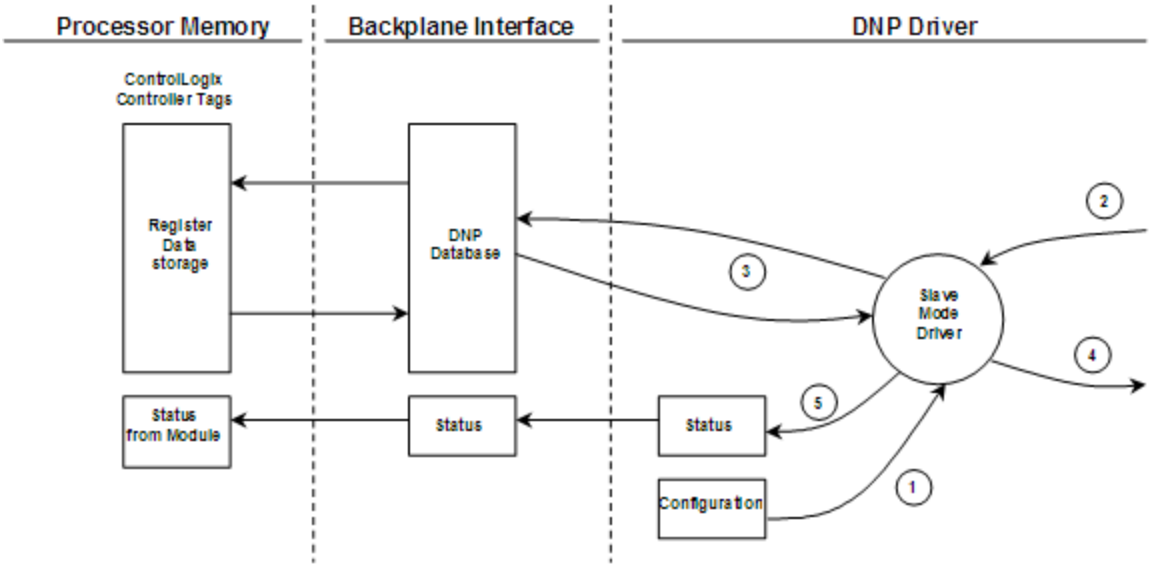


Data Flow Between MVI56-DNP Module and ControlLogix Processor

The following topics describe the flow of data between the two pieces of hardware (ControlLogix processor and MVI56-DNP module) and other nodes on the DNP network under the module's different operating modes. Each port on the module is configured to emulate a DNP Master device or a DNP slave device. The operation of each port depends on this configuration. The following topics discuss the operation of each mode.

Slave Driver

The Slave Driver Mode allows the MVI56-DNP module to respond to data read and write commands issued by a Master on the DNP network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The DNP slave port driver receives the configuration information from the DNP.CFG file. This information configures the serial port and define the slave node characteristics. Additionally, the configuration information contains data that can be used to offset data in the database to addresses requested in messages received from Master units.
2	A Host device (DNP Master unit) issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the command, the data is immediately transferred to or from the appropriate internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating Master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

Review the Installing and Configuring the Module section for a complete list of the parameters that must be defined for a slave port. The response messages from the slave driver include an IIN (internal indication word) defined in the Reference chapter Internal Indication Word (page 92).

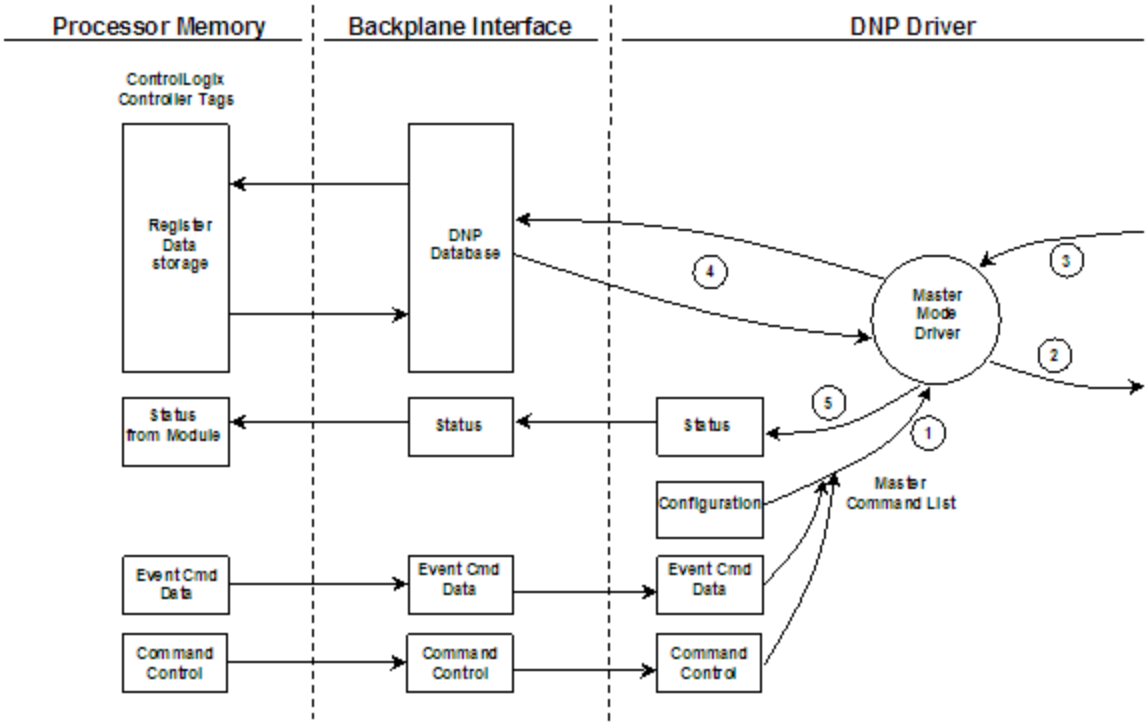
The slave driver supports object 110 (octet string data). Four points are pre-assigned values as defined in the following table.

Point #	Description
0	Module Name as assigned in configuration file.
1	Product Name
2	Version Information in format: wwww xxxx yyyy zzzz Where wwww is product code, xxxx is the revision, yyyy is the operating system number, and zzzz is the run number.
3	Manufacturer name for module.

The variation used in the request message determines the length of the string returned for each point. The maximum string length used by the module is 100.

Master Driver Mode

In the Master mode, the MVI56-DNP module issues read or write commands to slave devices on the DNP network. These commands are user configured in the module via the Master Command List received from the ControlLogix processor or issued directly from the ControlLogix processor (Special Function). Command status is returned to the processor for each individual command in the command list status block. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Master driver obtains configuration data from the DNP.CFG file. The configuration data obtained includes the Master Slave and Command Lists. These values are used by the Master driver to determine the type of commands to be issued to the other nodes on the DNP network.
2	After configuration, the Master driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from one of the module's internal databases to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the Master driver for processing.
4	Data received from the node on the network is passed into the module's appropriate internal database, assuming a read command.
5	Status is returned to the ControlLogix processor for each command in the Master Command List.

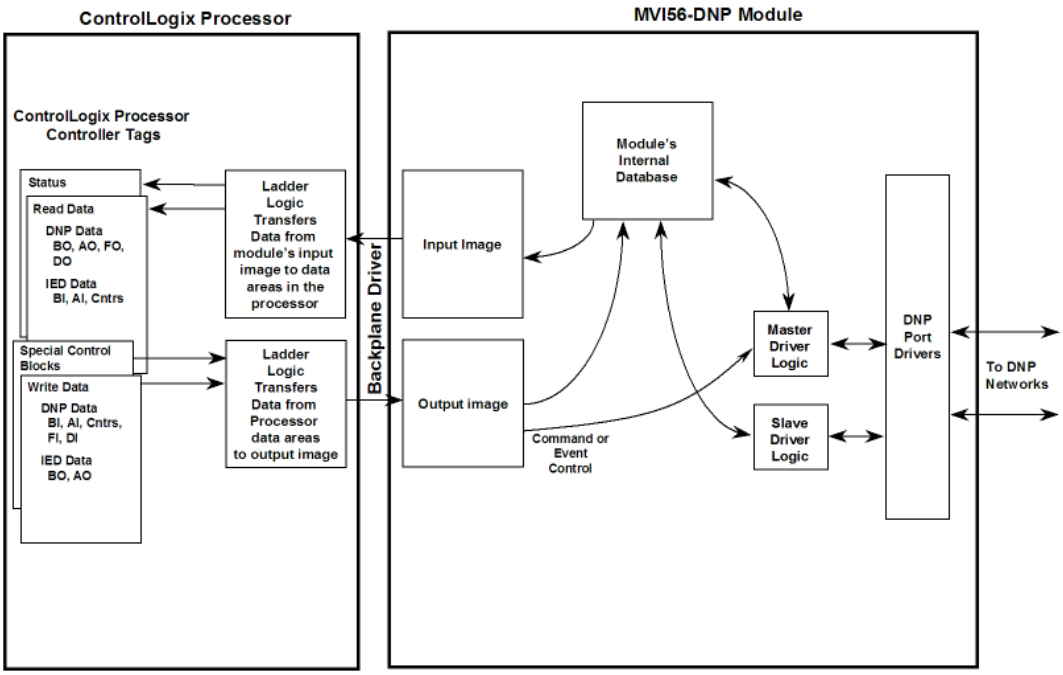
Backplane Data Transfer

The MVI56-DNP module communicates directly over the ControlLogix backplane. Data is paged between the module and the ControlLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 1 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is set to 250 words. This large data area permits fast throughput of data between the module and the processor.

The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module is set to 248 words. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-DNP module and the DNP Network.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the ControlLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal databases. These databases are defined as a virtual DNP data tables with addresses from 0 to the maximum number of points for each data type.

Data Area			Blocks
DNP Data	Binary Inputs	PLC Data	0 to 1
		IED Data	
	Binary Outputs	PLC Data	4 to 6
		IED Data	
	Counter Data	PLC Data	8 to 10
		IED Data	
	Analog Inputs	PLC Data	12 to 14
		IED Data	
	Analog Outputs	PLC Data	16 to 18
		IED Data	
	Float Inputs	PLC Data	40 to 42
	Double Inputs	PLC Data	44 to 46
	Float Outputs	PLC Data	48 to 50
	Double Outputs	PLC Data	52 to 54
	Frozen Counter Data		
	Binary Input Events		
	Analog Input Events		
	Float Input Events		
	Double Input Events		
Last Value Data	Binary Inputs		
	Analog Inputs		
	Float Inputs		
	Double Inputs		
	DNP Binary Outputs		
	DNP Analog Outputs		
	IED Binary Outputs		
	IED Analog Outputs		
IED Data	Binary Inputs		20 to 22
	Binary Outputs		24 to 26
	Counter Data		28 to 30
	Analog Inputs		32 to 34
	Analog Outputs		36 to 38
RBE Flags	Binary Input		
	Analog Input		

Data contained in this database is paged through the input and output images by coordination of the ControlLogix ladder logic and the MVI56-DNP module's program. Up to 248 words of data can be transferred from the module to the processor at a time. Up to 247 words of data can be transferred from the processor to the module.

Each block transferred from the module to the processor or from the processor to the module contains a block identification code that describes the content of the block.

Block Number	Function/Description
-1 or -2	Dummy Blocks: Used by module when no data is to be transferred
-4 to -6, -16 to -18, -48 to -50 and -52 to -54	DNP Initialize Output Blocks
0 to 18, 40 to 54	DNP Data blocks
20 to 38	IED Data blocks
100	Error/Status and Error List Block
9901	CROB Special Function Block for Digital Outputs
9902	Command Control Block (Add command to Command List Queue)
9903	Event Messages from Master port
9910	CROB Data received on DNP Port
9949	Slave IED unit errors on Master port
9950	Command List Error data
9958	PLC Binary Input Event data
9959	PLC Analog Input Event data
9961	SOE Binary Input Event with 64-bit time (1972 year base)
9962	SOE Binary Input Event with 64-bit time (1970 year base)
9968	Binary Input Event Data with DNP Time
9969	Analog Input Event Data with DNP Time
9970	Set PLC time using module's DNP time
9971	Set module's time using PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Blocks -1 and -2 are empty blocks used during module startup, when there is no data to transfer. Negative numbered blocks -4 to -54 are used to initialize the DNP slave database to specific values during startup, before the port begins to respond to requests from a remote Master. These negative numbered blocks are used only if the configuration parameter, *Initialize DNP Database,* is set to **YES** in the configuration file. Otherwise, the DNP database will be initialized with all values equal to zero.

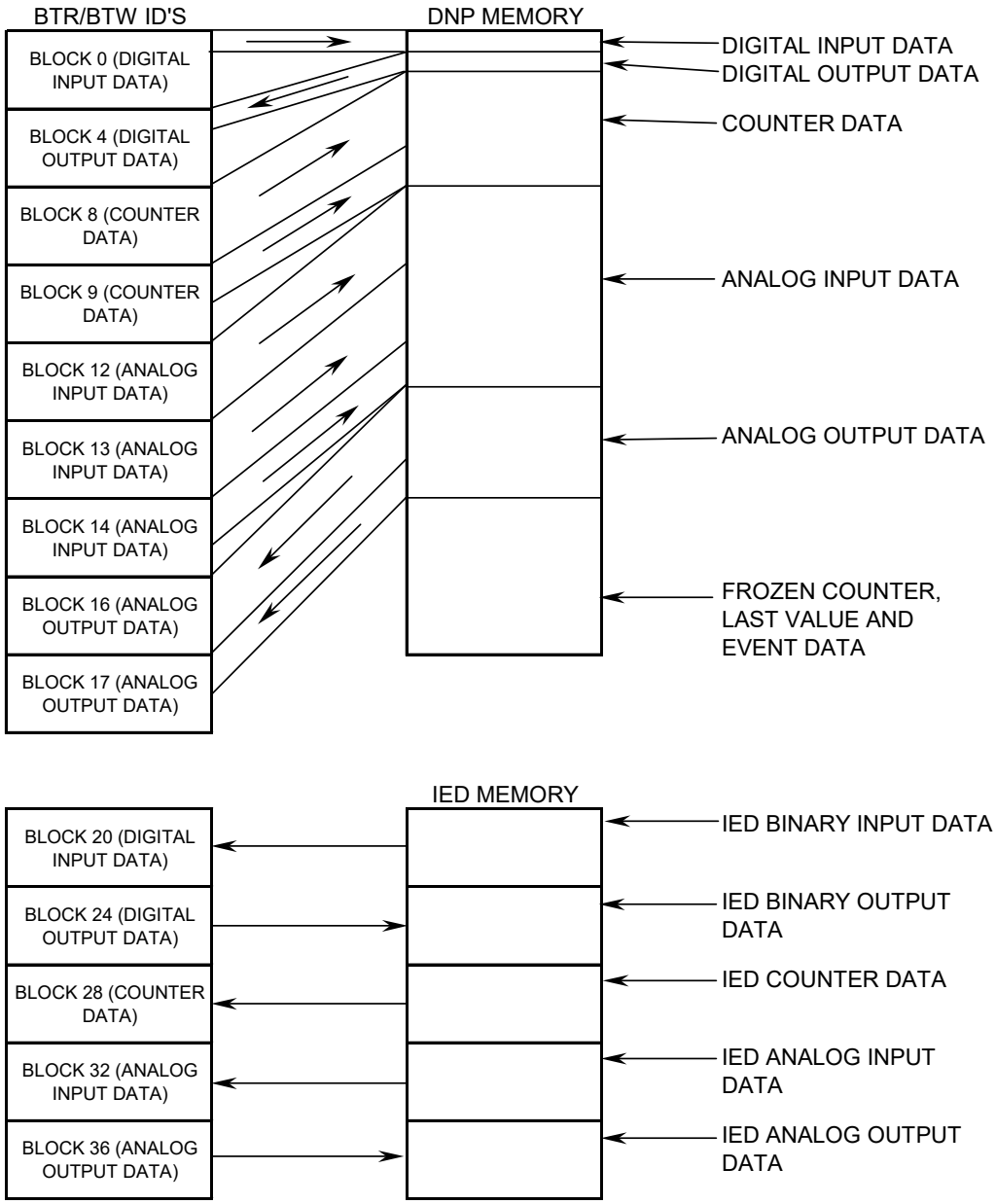
Blocks 0 to 54 are used to transfer the various kinds of DNP and IED process data. Block 100 transfers error and status data. Blocks 9901 to 9999 are used for Special Functions.

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal databases between the module and the controller. These data are transferred through read (input image) and write (output image) blocks. Refer to the Installing and Configuring the Module section for a description of the data objects used with the blocks and the ladder logic required. Each data block transferred between the module and the processor has a specific block identification code that defines the data type contained in the block. The following table lists the block identification codes used for data transfer by the module.

Data Type	Start Block #	Max Block #	Max # Of Points
Digital Input	0	2	8000
Digital Output	4	6	8000
Counters	8	10	250
Analog Input	12	14	500
Analog Output	16	18	500
IED Digital Input	20	22	8000
IED Digital Output	24	26	8000
IED Counters	28	30	250
IED Analog Input	32	34	500
IED Analog Output	36	38	500
DNP Float Input	40	42	250
DNP Double Input	44	46	125
DNP Float Output	48	50	250
DNP Double Output	52	54	125

The following illustration shows the direction of movement of these data types between the module and the processor.



Read Block

READ Blocks transfer information from the module to the ControlLogix processor. The following table describes the basic block structure of an input image.

Block Offset	Content
0	Reserved
1	Write block ID
2 to 241	Read data
242 to 248	Spare (Not used)
249	Read block ID

The *Read Block ID* is an index value used to determine the location of where the data will be placed in the ControlLogix processor read data controller tag array. Each transfer can move up to 240 words (block offsets 2 to 241) of data. The value of the Read Block identification code indentifies the type of data contained in the block, so the sample ladder logic can move it to the correct controller tag array.

The *Write Block ID* contained in the Read Block tells the ladder logic which block of data the module is expecting to receive from the ControlLogix processor during the next backplane transfer. Under normal program operation, the module sequentially sends read blocks and requests write blocks. For example, if one block each of binary and analog output data and one block of binary input data, two blocks of counter data and two blocks of analog input data are used with the application, the backplane transfer sequence block numbers will be:

R4W0⟶R16W8⟶R4W9⟶R16W12⟶R4W13⟶R16W0⟶R4W8⟶

This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the DNP network or operator control through the module's Configuration/Debug port. This sequence is occasionally interrupted by the read block identification code 100. This block passes the error/status and error list information from the module to the processor. Refer to the Error/Status section of this document for the structure and data contained in a Status Read block.

Write Block

WRITE blocks transfer information from the ControlLogix processor to the module. The following table describes the structure of a typical output image Write Block.

Block Offset	Content
0	Write block ID
1 to 240	Write data
241 to 247	Spare (Not used)

The *Write Block ID* is an index value used to determine the location in the module's database where the data will be placed as defined in the table presented in the previous section. Each transfer can move up to 240 words (block offsets 1 to 240) of data.

In cases where the ladder logic uses Special Function Blocks, the normal *Read Block IDs* and *Write Block IDs* will be replaced with a *Special Function Block ID*. Once the Special Function has been processed, the module will resume the normal data *Read Block ID* and *Write Block ID* sequence, starting from where the sequence was interrupted.

Trip/Close

The MVI56-DNP module supports Trip/Close functionality for Binary Output points.

This allows Trip/Close commands to be sent to the MVI56-DNP module, for dual point control. Each DNP Trip/Close command will occupy 2 bits within the module memory.

This does overlap the regular pulse on/off and latch on/off Binary Output database, therefore special consideration must be used to make sure that points are not used twice.

The following table describes the address mapping for the module using Latch and Pulse commands, and Trip/Close functionality.

DNP BO Database Point	BO Latch/Pulse Point	BO Trip/Close Point
0	BO 0	Close BO 0
1	BO 1	Trip BO 0
2	BO 2	Close BO 1
3	BO 3	Trip BO 1
4	BO 4	Close BO 2
5	BO 5	Trip BO 2
100	BO 100	Close BO 50
101	BO 101	Trip BO 50
1000	BO 1000	Close BO 500
1001	BO 1001	Trip BO 500
2000	BO 2000	Close BO 1000
2001	BO 2001	Trip BO 1001
3000	BO 3000	Close BO 1500
3001	BO 3001	Trip BO 1501
...and so on...		
7998	BO 7998	Close BO 3998
7999	BO 7999	Trip BO 3999

As you can see from the above chart, trip/close requires 2 points within the module's DNP database. A Trip is represented by the binary value of '10' for those 2 points, and a Close is represented by the binary value of '01' for those same 2 points.

The module can process only 4000 trip/close dual points, as the database for the DNP BO is limited to 8000 total bits.

5.2.3 Special Function Blocks

Special Function blocks are special blocks used to control the module or request special data from the module. The current version of the software supports several Special Function blocks, each of which is discussed in the following topics.

CROB Control Block for Digital Output (9901)

If the ControlLogix processor sends a block 9901, the module places the digital output control commands into the command queue. Commands placed in the queue with this method are not contained in the normal command list. Data contained in the block completely defines the command to the system.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Command Count	This field defines the number of CROB blocks to generate. The valid range for the field is 1 to 6.
2 to 11	Command #1	Data for the command relay block (CROB) to be generated.
12 to 21	Command #2	Data for the command relay block (CROB) to be generated.
22 to 31	Command #3	Data for the command relay block (CROB) to be generated.
32 to 41	Command #4	Data for the command relay block (CROB) to be generated.
42 to 51	Command #5	Data for the command relay block (CROB) to be generated.
52 to 61	Command #6	Data for the command relay block (CROB) to be generated.
62 to 247	Spare	Not Used

Up to six commands can be contained in a single block. The 10-word data area for each command is defined in the following table.

Word Offset	Definitions	Description
0	Port/Flags	This field is currently ignored as all 9901 blocks are sent immediately out the Master port.
1	Slave Address	This is the IED node address for the slave to consider on the network.
2	Object	Object type always 12
3	Variation	Variation always 1
4	Function	Function codes 3 (select/operate), 5 (direct operate with ACK), and 6 (direct operate no ACK) supported. Function code 4 is automatically sent after a successful function 3.
5	Address in Slave	Point in IED to consider with the CROB.
6	Control Code	This is a standard DNP protocol control code byte (see description below).
7	Pulse Count	This parameter specifies the number of pulses to generate for pulse output control. This parameter has a range of 0 to 255 as the value is a byte parameter in the CROB. If a value of zero is entered, the operation will not execute.
8	Pulse On Time	This parameter specifies the on-time interval for pulse control.
9	Pulse Off Time	This parameter specifies the off-time interval for pulse control.

The control code in the command is a bit coded byte value with the following definition:

Bits	Definitions	Description
0 to 3	Code	These bits determine the control operation to be performed by the command: 0=No operation, 1=Pulse on, 2=Pulse off, 3=Latch on and4=Latch off. All other values are undefined in the DNP protocol.
4	Queue	0=Normal (execute once), 1=Requeue (place at end of queue after operation).
5	Clear	This parameter clears the queue. If the value is set to zero, the queue is not affected. If the value is set to 1, the queue will be cleared.
6 to 7	Trip/Close	These two bits select the trip or close relay. For close relay control, set the bits to 01. For trip relay control, set the bits to 10. A value of 00 for the bits is used for single point control of normal digital output points.

The command list for the Master port does not provide the means for all the possible CROB operations. It only supports the latch on and off operations. With the use of this block, outputs can be pulsed on or off for user specified count and time intervals. Additionally, this command provides support for trip/close relay control.

Command Control Block (9902)

If the ControlLogix processor sends a block 9902, the module will place the commands referenced in the block in the command queue. Commands placed in the queue with this method need not have their enable bit set. Only valid commands will be placed in the queue.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9902 identifying the enable command to the module.
1	Command count	This field contains the number of commands to enable in the command list. Valid values for this field are 1 to 60.
2 to 61	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to enable. The commands in the list will be placed in the command queue for immediate processing by the module. The first command in the list has an index of 0.
62 to 247	Spare	Not Used

Up to 60 commands can be enabled and placed in the command queue with one write request from the ControlLogix processor.

Event Message Block (9903)

If the module contains a DNP Master port and is configured to pass event messages from the port to the processor, block identification 9903 will be utilized. When the Master port receives an event message, it will place the data in the message into the event message queue. This queue has room for up to 200 messages. When the backplane task in the modules recognizes data in this queue, it will form 9903 blocks to transfer the data to the processor. Ladder logic extracts the event data from the 9903 block and place it in controller tags.

Word Offset in Block	Data Field(s)	Description
0	Reserved	
1	Block ID	This is the next block requested by the module.
2	Event Count	This field contains the number of events present in the block. Values of 1 to 20 are valid.
3 to 14	Event 1	Event message
15 to 26	Event 2	Event message
27 to 38	Event 3	Event message
39 to 50	Event 4	Event message
51 to 62	Event 5	Event message
63 to 74	Event 6	Event message
75 to 86	Event 7	Event message
87 to 98	Event 8	Event message
99 to 110	Event 9	Event message
111 to 122	Event 10	Event message
123 to 134	Event 11	Event message
135 to 146	Event 12	Event message
147 to 158	Event 13	Event message
159 to 170	Event 14	Event message
171 to 182	Event 15	Event message
183 to 194	Event 16	Event message
195 to 206	Event 17	Event message
207 to 218	Event 18	Event message
219 to 230	Event 19	Event message
231 to 242	Event 20	Event message
243 to 248	Spare	Not Used
249	Block ID	This field contains the block identification code of 9903 for the block.

Up to 20 events are passed to the processor in each block. The format of each event message is the block is as shown in the following table.

Word Offset	Definitions	Description
0	Device Index	This field contains the module's device index for the IED the message was received from (0 to 39).
1	IED Address	This field contains the IED database index for the point. If set to -1, then not in database.
2	DNP Address	This field contains the DNP database index for the point. If set to -1, then not in database.
3	Slave Address	This field contains the remote slave address for the IED unit from which the message was received.
4	Point Number	This field contains the point number in the remote IED unit for the event message.
5	Object	This field contains the object code for the point and event.
6	Variation	This field contains the variation code for the point and event.
7	Low Time	This field contains the least-significant word of the 48-bit DNP time for the event.
8 to 9	High Time	This field contains the most- significant double word of the 48-bit time for the event.
10 to 11	Value	This field contains the a double word value for the point associated with the event message.

CROB Data received on DNP Port Block (9910)

Block identification code 9910 is used to send CROB messages received on the DNP slave port to the processor.

Block Format for Read

Word Offset in Block		Data Field(s)	Description
Start	Stop		
0	0	Reserved	
1	1	Block ID	This is the next block requested by the module.
2	2	CROB count	This field contains the number of CROB records that are contained in this block. The range is between 1 and 40.
3	8	CROB 1	CROB block data as defined below
9	14	CROB 2	CROB block data as defined below
15	20	CROB 3	CROB block data as defined below
21	26	CROB 4	CROB block data as defined below
27	32	CROB 5	CROB block data as defined below
33	38	CROB 6	CROB block data as defined below
39	44	CROB 7	CROB block data as defined below
45	50	CROB 8	CROB block data as defined below
51	56	CROB 9	CROB block data as defined below
57	62	CROB 10	CROB block data as defined below
63	68	CROB 11	CROB block data as defined below
69	74	CROB 12	CROB block data as defined below
75	80	CROB 13	CROB block data as defined below
81	86	CROB 14	CROB block data as defined below
87	92	CROB 15	CROB block data as defined below
93	98	CROB 16	CROB block data as defined below
99	104	CROB 17	CROB block data as defined below
105	110	CROB 18	CROB block data as defined below
111	116	CROB 19	CROB block data as defined below
117	122	CROB 20	CROB block data as defined below
123	128	CROB 21	CROB block data as defined below
129	134	CROB 22	CROB block data as defined below
135	140	CROB 23	CROB block data as defined below
141	146	CROB 24	CROB block data as defined below
147	152	CROB 25	CROB block data as defined below
153	158	CROB 26	CROB block data as defined below
159	164	CROB 27	CROB block data as defined below
165	170	CROB 28	CROB block data as defined below

Word Offset in Block Start	Stop	Data Field(s)	Description
171	176	CROB 29	CROB block data as defined below
177	182	CROB 30	CROB block data as defined below
183	188	CROB 31	CROB block data as defined below
189	194	CROB 32	CROB block data as defined below
195	200	CROB 33	CROB block data as defined below
201	206	CROB 34	CROB block data as defined below
207	212	CROB 35	CROB block data as defined below
213	218	CROB 36	CROB block data as defined below
219	224	CROB 37	CROB block data as defined below
225	230	CROB 38	CROB block data as defined below
231	236	CROB 39	CROB block data as defined below
237	242	CROB 40	CROB block data as defined below
243	248	Spare	Not Used
249	249	Block ID	This field contains the block identification code of 9910 for the block.

The format of each 6 word data region in the block is as follows:

Word Offset	Definitions	Description
0	Point Number	This field contains the BO point number for the following CROB command.
1	Control Code/Count	This field contains the two bytes of the CROB message receved that contain the control code and count. Control code 81 = trip, 41 = close, and 1 = pulse.
2 to 3	On Time	This double-word contains the on time received for the CROB block.
4 to 5	Off Time	This double-word contains the off time received for the CROB block.

Read Slave Communication Error Table Block (9949)

If the ControlLogix processor sends a block 9949, the module will respond with a slave communication error listing. The module can be set up for 40 slaves on its DNP Master port. A 10-word data area is established in the module for each slave. This requires 400 words of data storage.

Value	Description
Index	This value corresponds to the index in the device array for the slave.
Slave Addr	This value corresponds to the DNP slave address for the device.
Bad CRC	This value represents the number of bad CRC values received from the slave device.
Buff Ovrflw	This value represents the number of buffer overflow messages received from the slave device.
Tran Seq#	This value represents the number of incorrect transport layer sequence number errors.
Conf Retry	This value represents the number of data link layer confirm request retries.
Conf Fail	This value represents the number of data link layer confirm request failures.
No App Rsp	This value represents the number of application layer no responses to requests.
Spare	Reserved for future use.
Spare	Reserved for future use.

These data values are updated after each command processed by the module. The block 9949 request is structured to retrieve data for up to 30 slave units each call. The format of the block sent from the ControlLogix processor is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9949 identifying the block type to the module.
1	Number of slaves	This field contains the number of slaves to report in the response message. The value has a range of 1 to 30.
2	Start Slave Index	This parameter sets the index in the slave array where to start. The first slave in the array has a value of 0. The last index in the array has a value of MaxSlaves -1.
3 to 247	Spare	Not Used

Using the data in this block, the module responds with a read block 9949 with the following format:

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	This is the next block requested by the module.
2	Slave Count	This field contains the number of slave records contained in the block that must be processed by the PLC. This field will have a value of 1 to 30.
3	Slave Start Index	This field contains the index in the slave array for the first record in the file. This field will have a value of 0 to MaxSlaves-1.
4 to 11	Slave Data #1	This is the slave data for the first slave in the block. The slave index for the data is the Slave Start Index given in word 3.
12 to 19	Slave Data #2	This is the slave data for the second slave in the block.
20 to 27	Slave Data #3	This is the slave data for the third slave in the block.
236 to 243	Slave Data #30	Last slave requested.
244 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9949 identifying the block type to the PLC.

The 8-word data area for each slave is that shown in the table above. You can sequentially read through the list of all IED units up to 30 at a time to retrieve all the error information. This data can be transferred to the module's controller tag in the processors ladder logic.

Read Command Error List Block (9950)

If the ControlLogix processor sends a block number of 9950 to the module, the application will respond with a command error list. Each command in the system has a data word set aside for its last error code. This value is set by the DNP Master port command list task and the values correspond to the errors listed in the error section of this documentation. This 300-word data area can be accessed 200 commands at a time.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the module.
1	Number of Commands to report	This field contains the number of commands to report in the response message. The value has a range of 1 to 200.
2	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands -1.
3 to 247	Spare	Not Used

The module responds to the request with a read block 9950 with the following format:

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved(0)
1	Block ID	This is the next block requested by the module.
2	Number of Commands reported	This field contains the number of commands contained in the block that must be processed by the PLC. This field will have a value of 1 to 200.
3	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 0 to MaxCommands-1.
4 to 203	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the block. The number of valid command errors in the block is set in word 2 of the block. Refer to the command error list to interpret the error codes reported.
204 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9950 identifying the block type to the PLC.

The ControlLogix program can sequentially step through the list of commands to retrieve the value of the last error. To retrieve all 300 commands, this requires two transfers.

PLC Binary Input Event Block (9958)

If the ControlLogix processor sends a block 9958, the module will place the binary input event data from the block into the event buffer and alter the data values for the points in the DNP binary input database.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9958 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 12.
2	Sequence Counter	This field is used to hold the sequence counter for each 9958 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4	Month/Day/State	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month, bit 15 = digital state for point. All other bits are ignored.
5	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
6	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
7	Year	This is the four digit year for the event (0-4095). Bits 12 and 13 can contain the class override values of 1 to 3.
8 to 12		Five words of data for Event #2.
13 to 17		Five words of data for Event #3.
18 to 22		Five words of data for Event #4.
23 to 27		Five words of data for Event #5.
28 to 32		Five words of data for Event #6.
33 to 37		Five words of data for Event #7.
38 to 42		Five words of data for Event #8.
43 to 47		Five words of data for Event #9.
48 to 52		Five words of data for Event #10.
53 to 57		Five words of data for Event #11.
58 to 62		Five words of data for Event #12.
63 to 247	Spare	Not Used

Up to 12 events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9958 to the ControlLogix processor. The following table describes the format of the block.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9958 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9958.

The sequence counter field in the returned block is set to the last successfully processed block 9958 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor. If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9958 block, the ControlLogix processor must re-send the block.

Analog Input Event Block (9959)

If the ControlLogix processor sends a block 9959, the module will place the analog input event data in the block into the event buffer and alter the data values for the points in the DNP analog input database.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9959 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
2	Sequence Counter	This field is used to hold the sequence counter for each 9959 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Analog Input Data point	This is the data point in the DNP analog input database represented by the event.
4	Analog Input Value	This is the new analog input value represented in the event.
5	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
6	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
7	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
8	Year	This is the four digit year for the event (0-4095). Bits 12 and 13 can contain the class override values of 1 to 3.
9 to 14		Six words of data for Event #2.
15 to 20		Six words of data for Event #3.
21 to 26		Six words of data for Event #4.
27 to 32		Six words of data for Event #5.
33 to 38		Six words of data for Event #6.
39 to 44		Six words of data for Event #7.
45 to 50		Six words of data for Event #8.
51 to 56		Six words of data for Event #9.
57 to 62		Six words of data for Event #10.
63 to 247	Spare	Not Used

Up to 10 events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9959 to the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved(0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9959 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9959.

The sequence counter field in the returned block is set to the last successfully processed block 9959 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor. If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9959 block, the ControlLogix processor must re-send the block.

SOE Binary Input Events Block (9961) with 1972 Time Base (GMT Time)

Block Identification Code 9961 is used by the PLC to send a set of SOE binary input events to the module using 64-bit time code based at a starting point of January 1st, 1972. This block is for RS Logix versions prior to version 16.

Block Request from Processor to Module (9961)

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9961 identifying the SOE event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 20.
2	Sequence Counter	This field is used to hold the sequence counter for each 9961 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4 to 7	64-bit time	This is the 64-bit time value generated by the SOE module.
8	Value	This is the value for the event data. It is either a 0 or 1.
9 to 14		Six words of data for Event #2.
15 to 20		Six words of data for Event #3.
21 to 26		Six words of data for Event #4.
27 to 32		Six words of data for Event #5.
33 to 38		Six words of data for Event #6.
39 to 44		Six words of data for Event #7.
45 to 50		Six words of data for Event #8.
51 to 56		Six words of data for Event #9.
57 to 62		Six words of data for Event #10.
63 to 68		Six words of data for Event #11.
69 to 74		Six words of data for Event #12.
75 to 80		Six words of data for Event #13.
81 to 86		Six words of data for Event #14.
87 to 92		Six words of data for Event #15.
93 to 98		Six words of data for Event #16.
99 to 104		Six words of data for Event #17.
105 to 110		Six words of data for Event #18.
111 to 116		Six words of data for Event #19.
117 to 122		Six words of data for Event #20.
123 to 247	Spare	Not Used

To insure the receipt of this block of information, the module returns a BTR block 9961 with the sequence counter set to the value of the last successful block 9961 received.

Block Response from Module to Processor (9961)

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9961 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9961.

SOE Binary Input Events Block (9962) with 1970 Time Base (UTC Time)

Block Identification Code 9962 is used by the PLC to send a set of SOE binary input events to the module using 64-bit time based at a starting point of January 1st, 1970. This block is for RS Logix versions 16 and later.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9962 identifying the SOE event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 20.
2	Sequence Counter	This field is used to hold the sequence counter for each 9962 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4 to 7	64-bit time	This is the 64-bit time value generated by the SOE module.
8	Value	This is the value for the event data. It is either a 0 or 1.
9 to 14		Six words of data for Event #2.
15 to 20		Six words of data for Event #3.
21 to 26		Six words of data for Event #4.
27 to 32		Six words of data for Event #5.
33 to 38		Six words of data for Event #6.
39 to 44		Six words of data for Event #7.
45 to 50		Six words of data for Event #8.
51 to 56		Six words of data for Event #9.
57 to 62		Six words of data for Event #10.
63 to 68		Six words of data for Event #11.
69 to 74		Six words of data for Event #12.
75 to 80		Six words of data for Event #13.
81 to 86		Six words of data for Event #14.
87 to 92		Six words of data for Event #15.
93 to 98		Six words of data for Event #16.
99 to 104		Six words of data for Event #17.
105 to 110		Six words of data for Event #18.
111 to 116		Six words of data for Event #19.
117 to 122		Six words of data for Event #20.
123 to 247	Spare	Not Used

To insure the receipt of this block of information, the module returns a BTR block 9962 with the sequence counter set to the value of the last successful block 9962 received.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9962 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9962.

Binary Input Event with DNP Time (9968)

The Block 9968 identification code is used by the PLC to send a set of binary input events to the module with DNP network time.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9968 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
2	Sequence Counter	This field is used to hold the sequence counter for each 9968 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4	State	Formatted: bit 15 = digital state for point. All other bits are ignored.
5 to 7	DNP Time	DNP time as received from Block ID 9903
8	Reserved	Bits 0 and 1 are used for class override values of 1 to 3.
9 to 14		Five words of data for Event #2.
15 to 20		Five words of data for Event #3.
21 to 26		Five words of data for Event #4.
27 to 32		Five words of data for Event #5.
33 to 38		Five words of data for Event #6.
39 to 44		Five words of data for Event #7.
45 to 50		Five words of data for Event #8.
51 to 56		Five words of data for Event #9.
57 to 62		Five words of data for Event #10.
63 to 247	Spare	Not Used

Up to ten (10) events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9968 to the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9968 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9968.

The sequence counter field in the returned block is set to the last successfully processed block 9968 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor. If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9968 block, the ControlLogix processor must re-send the block.

Analog Input Event Data with DNP Time (9969)

The Block 9969 identification code is used by the PLC to send a set of analog input events to the module with DNP network time.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9969 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
2	Sequence Counter	This field is used to hold the sequence counter for each 9969 block transfer. This is used to synchronize and confirm receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4	State	Formatted: bit 15 = digital state for point. All other bits are ignored.
5 to 7	DNP Time	DNP time as received from Block ID 9903
8	Reserved	Bits 0 and 1 are used for class override values of 1 to 3.
9 to 14		Six words of data for Event #2.
15 to 20		Six words of data for Event #3.
21 to 26		Six words of data for Event #4.
27 to 32		Six words of data for Event #5.
33 to 38		Six words of data for Event #6.
39 to 44		Six words of data for Event #7.
45 to 50		Six words of data for Event #8.
51 to 56		Six words of data for Event #9.
57 to 62		Six words of data for Event #10.
63 to 247	Spare	Not Used

Up to ten (10) events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9969 to the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9969 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9969.

The sequence counter field in the returned block is set to the last successfully processed block 9969 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor. If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9969 block, the ControlLogix processor must re-send the block.

Set Processor Time Block (9970)

This block transfers the module's time to the ControlLogix processor. Ladder logic must be used to set the processor's clock using the data received.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to 247	Not Used	Not Used

The module responds to the request with a read block 9970 with the following format.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year for the new time value.
3	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
9	Remote Time Synchronization	This field informs the PLC if the date and time passed has been synchronized with a remote DNP Master device on the module's slave port. 1 = time has been set on the DNP network. 0 = waiting for time sync from Master.
10 to 248	Not Used	Not Used
249	Block Read ID	This field contains the block identification code of 9970 for the block.

Set Module Time Block (9971) MVI56-DNP

Block identification code 9971 passes the clock time in the PLC to the module. The date and time provided will be used to set the module's DNP clock.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9971 for the block.
1	Year	This field contains the four-digit year for the new time value.
2	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
4	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
5	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
8 to 247	Not Used	Not Used

Warm Boot Block (9998)

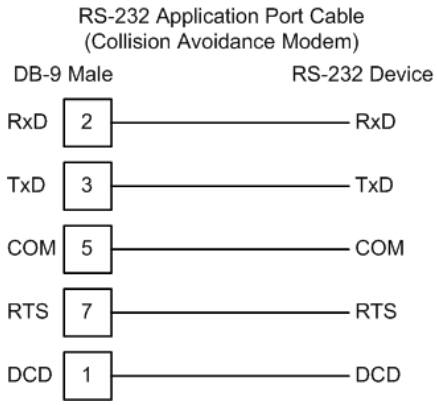
If the ControlLogix sends a block number 9998, the module will perform a warm-boot operation. The module will reconfigure the communication ports and reset the error and status counters.

Cold Boot Block (9999)

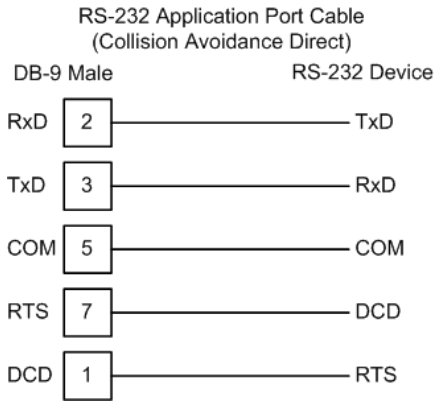
If the ControlLogix processor sends a block number 9999, the firmware will perform a cold-boot operation. The firmware will reload the configuration file and reset all DNP memory, error and status data.

5.3 Collision Avoidance (DNP modules only)

The RTS line is controlled by the RTS on and off parameters set for the port. If the CTS line is used (usually only required for half-duplex modems and not defined for use in the DNPS specification), the RTS and CTS lines must either be connected together or connected to the modem. The following illustration shows the cable required when connecting the port to a modem.



If collision avoidance is used in a point-to-point connection on the RS-232 interface, the following cable should be used.



5.4 DNP 3.0 Device Profile Document

<table>
<tr><td colspan="2">DNP V3.00
DEVICE PROFILE DOCUMENT</td></tr>
<tr><td colspan="2">Vendor Name: ProSoft Technology, Inc.</td></tr>
<tr><td colspan="2">Device Name: MVI56-DNP (VERSION 2.00)</td></tr>
<tr><td>Highest DNP Level Supported:
For Request: L2
For Responses: L2</td><td>Device Function:
Slave & Master</td></tr>
<tr><td colspan="2">Notable objects, functions, and/or qualifiers supported in addition to the highest DNP level stated above (see attached table for complete list).

Definition of selected IIN bits:
Device Trouble - PLC data transfer operation is not taking place
Configuration Error - User specified point or event count is too high for application (can correct only by changing configuration in PLC).

Support for a redundant slave port on the module which may be attached to a dial-up modem is provided. Auto switching is provided by the module to switch between the primary and secondary ports.

The following features are configurable on the module:
Collision avoidance, time sync before events are generated and default analog input events, Obj32V4 or O32V2, select option. Floating-point variations are supported for analog input and output objects (both single and double floating-point types). Support for Obj110 (octet string) available only using read function.

Events generated by IED units attached to a Master port may pass their events directly to the slave port. These events may not occur in the correct time sequence. They are placed in the event buffer as the module receives them. This provides the greatest time resolution for remote events.

Counter Freeze with reset will not zero values in the processor. Therefore, this function should only be used on the Master port.

Module will not generate events until Restart IIN bit is cleared by DNP Master except for events passed through module from attached IED units.</td></tr>
<tr><td>Maximum Data Link Frame Size (octets):
Transmitted: 292
Received: 292</td><td>Maximum Application Fragment Size (octets):
Transmitted: 2048
Received: 2048</td></tr>
<tr><td>Maximum Data Link Re-tries:
Configurable from 0 - 255</td><td>Maximum Application Layer Re-tries:
None</td></tr>
<tr><td colspan="2">Requires Data Link Layer Confirmation:
Configurable at module start-up (never, sometimes, & always)</td></tr>
<tr><td colspan="2">Requires Application Layer Confirmation:
When reporting Event Data as a slave unit</td></tr>
<tr><td colspan="2">Time-outs while waiting for:
Data Link Confirm : Configurable at module start-up (1 to 65535 milliseconds)
Complete Application Fragment : Configurable at module start-up</td></tr>
</table>

DNP V3.00 DEVICE PROFILE DOCUMENT Application Confirm : Configurable at module start-up (1 to 65535 milliseconds) Complete Application Response : None	
Sends/Executes Control Operations: WRITE Binary Outputs : Never SELECT/OPERATE : Always DIRECT OPERATE : Always DIRECT OPERATE-NO ACK : Always Count > 1 : Always (1 to 255) Pulse On : Always Pulse Off : Always Latch On : Always Latch Off : Always Queue : Never Clear Queue : Never	
Reports Binary Input Change Events when no specific variation requested: Only time-tagged	Reports time-tagged Binary Input Change Events when no specific variation requested: Binary Input Change with Time
Sends Unsolicited Responses: This is configurable at module start-up. If the number of events for the Binary or Analog Input Events is greater than 0, unsolicited responses are supported. Use the Enable/Disable Unsolicited function code from the DNP Master for control.	Sends Static Data in Unsolicited Responses: Never
Default Counter Object/Variation: Object : 20 Variation : 5	Counters Roll Over at: 32 Bits
Sends Multi-Fragment Responses: Yes	

5.5 DNP Subset Definition - Slave

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
1	0	Binary Input - All Variations	1	06			1	Slave will return variation 1 data
	1	Binary Input	1	06	129, 130	00, 01	1	Slave will return this variation
	2	Binary Input with Status			129, 130	00, 01	8	Slave will return Unknown Object to this request
2	0	Binary Input Change - All Variations	1	06, 07, 08			56	Slave will return variation 2 data
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Slave will return this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Slave will return this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Slave will parse this message and return no data
10	0	Binary Output - All Variations	1	06			8	Slave will return variation 2 data
	1	Binary Output					1	Slave will return Unknown Object to this request
	2	Binary Output Status	1	06	129, 130	00, 01	8	Slave will return this variation
12	0	Control Block - All Variations					88	Slave will use variation 1 control
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Slave will respond correctly to this variation
	2	Pattern Control Block					88	Slave will return Unknown Object to this request
	3	Pattern Mask					16	Slave will return Unknown Object to this request
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10	06			32	Slave will return variation 5 data
	1	32-Bit Binary Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	2	16-Bit Binary Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	3	32-Bit Delta Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	4	16-Bit Delta Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	32	Slave will return this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
21	0	Frozen Counter - All Variations	1	06			32	Slave will return variation 9 data
	1	32-Bit Frozen Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	9	32-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation
	10	16-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	11	32-Bit Frozen Delta Counter Without Flag					32	Slave will return Unknown Object to this request
	12	16-Bit Frozen Delta Counter Without Flag					16	Slave will return Unknown Object to this request
22	0	Counter Change Event - All Variations	1	06, 07, 08				Slave will parse this request and return no data
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Slave will return Unknown Object to this request
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Slave will return Unknown Object to this request
	3	32-Bit Delta Counter Change Event Without Time					40	Slave will return Unknown Object to this request
	4	16-Bit Delta Counter Change Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Counter Change Event With Time					88	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	6	16-Bit Counter Change Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Delta Counter Change Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Change Event With Time					72	Slave will return Unknown Object to this request
23	0	Frozen Counter Event - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Counter Event Without Time					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Counter Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter Event Without Time					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter Event With Time					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter Event With Time					72	Slave will return Unknown Object to this request
30	0	Analog Input - All Variations	1	06			16	Slave will respond with variation 4 data
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation (Note: Data will be only 16-bit)
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Slave will return this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation (Note: Data will only be 16-bit)
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation
	5	Short Floating Point Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation
	6	Long Floating Point Analog Input	1	06	129, 130	00, 01	72	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
31	0	Frozen Analog Input - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Input					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Analog Input					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Input With Time To Freeze					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Input With Time To Freeze					72	Slave will return Unknown Object to this request
	5	32-Bit Frozen Analog Input Without Flag					32	Slave will return Unknown Object to this request
	6	16-Bit Frozen Analog Input Without Flag					16	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Input					40	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Input					72	Slave will return Unknown Object to this request
32	0	Analog Change Event - All Variations	1	06, 07, 08			24	Slave will return variation 2 data
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation (Note: Data will be only 16-bit)
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Slave will return this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation (Note: Data will be only 16-bit)
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	5	Short Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation
	6	Long Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	7	Short Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation
	8	Long Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	120	Slave will return this variation
33	0	Frozen Analog Event - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Event Without Time					40	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	16-Bit Frozen Analog Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Event With Time					72	Slave will return Unknown Object to this request
	5	Short Floating Point Frozen Analog Event					40	Slave will return Unknown Object to this request
	6	Long Floating Point Frozen Analog Event					72	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Event With Time					120	Slave will return Unknown Object to this request
40	0	Analog Output Status - All Variations	1	06			24	Slave will return variation 2 data
	1	32-Bit Analog Output Status	1	06	129,130	00,01	40	Slave will return this variation with only 16-bit data accuracy
	2	16-Bit Analog Output Status	1	06	129, 130	00, 01	24	Slave will return this variation
	3	Short Floating Point Analog Output Status	1	06	129, 130	00, 01	40	Slave will return this variation
	4	Long Floating Point Analog Output Status	1	06	129, 130	00, 01	72	Slave will return this variation
41	0	Analog Output Block - All Variations					24	Slave will respond to this request using variation 2 data
	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129,130	00,01	40	Slave will respond to this request with only 16-bit data
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Slave will respond to this request
	3	Short Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	40	Slave will respond to this request
	4	Long Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	72	Slave will respond to this request
50	0	Time and Date - All Variations	2	07, With Quant=1			48	Slave will use variation 1
	1	Time and Date	2	07, With Quant=1			48	Slave will respond to this variation
	2	Time and Date With Interval					80	Slave will return Unknown Object to this request
51	0	Time and Date CTO - All Variations						Slave will return Unknown Object to this request
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
52	0	Time Delay - All Variations						
	1	Time Delay Coarse			129	07, With Quant=1	16	Slave will never return this variation
	2	Time Delay Fine			129	07, With Quant=1	16	Slave will return this variation to functions 0D, 0E, and 17
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Slave will respond to this variation with all static data
	2	Class 1 Data	1	06, 07, 08				
	3	Class 2 Data	1	06, 07, 08				Slave will respond to this variation with all class 2 data (binary input events)
	4	Class 3 Data	1	06, 07, 08				Slave will respond to this variation with all class 3 data (analog input events)
70	0	Not Defined						Not Defined in DNP
	1	File Identifier						Slave will return Unknown Object to this request
80	0	Not Defined						Not Defined in DNP
	1	Internal Indications	2	00, Index=7			24	Slave will respond to this variation
81	0	Not Defined						Not Defined in DNP
	1	Storage Object						
82	0	Not Defined						Not Defined in DNP
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
110	0	Not Defined						Not Defined as the variation determines the string length
	1 to 100	Octet String	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 07, 08, 17, 28	8 * Var #	The module will return this variation for the points defined in the module. The variation determines the returned string length.
No Object			13					Slave supports the Cold Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			14					Slave supports the Warm Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			20					Slave supports the Enable Unsolicited Function
			21					Slave supports the Disable Unsolicited Function
			23					Slave supports the Delay Measurement & Time Synchronization Function and will return Obj 52, Var 2, Qual 7, Cnt 1

5.6 DNP Subset Definition - Master

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
1	0	Binary Input - All Variations	1	06			1	Master will generate this variation
	1	Binary Input	1	06	129, 130	00, 01	1	Master will generate and process this variation
	2	Binary Input with Status	1	06	129, 130	00, 01	8	Master will generate and process this variation
2	0	Binary Input Change - All Variations	1	06, 07, 08			56	Master will generate this variation
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Master will generate and process this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Master will generate and process this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Master will generate and process this variation
10	0	Binary Output - All Variations	1	06			8	Master will generate and process these variations
	1	Binary Output					1	
	2	Binary Output Status			129, 130	00, 01	8	
12	0	Control Block - All Variations					88	
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Master will generate this variation and parse the response
	2	Pattern Control Block					88	
	3	Pattern Mask					16	
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10	06			32	Master will generate this variation
	1	32-Bit Binary Counter			129, 130	00, 01	40	Master will process this variation
	2	16-Bit Binary Counter			129, 130	00, 01	24	Master will process this variation
	3	32-Bit Delta Counter			129, 130	00, 01	40	Master will process this variation
	4	16-Bit Delta Counter			129, 130	00, 01	24	Master will process this variation
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	32	Master will generate and process this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	16	Master will generate and process this variation
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Master will process this variation
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Master will process this variation
21	0	Frozen Counter - All Variations	1	06			32	Master will generate this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	32-Bit Frozen Counter			129, 130	00, 01	40	Master will process this variation
	2	16-Bit Frozen Counter			129, 130	00, 01	24	Master will process this variation
	3	32-Bit Frozen Delta Counter					40	
	4	16-Bit Frozen Delta Counter					24	
	5	32-Bit Frozen Counter With Time Of Freeze					88	
	6	16-Bit Frozen Counter With Time Of Freeze					72	
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	
	9	32-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	32	Master will generate and process this variation
	10	16-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	16	Master will generate and process this variation
	11	32-Bit Frozen Delta Counter Without Flag					32	
	12	16-Bit Frozen Delta Counter Without Flag					16	
22	0	Counter Change Event - All Variations	1	06, 07, 08				Master will not generate a request for this variation
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Master will process this variation
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Master will process this variation
	3	32-Bit Delta Counter Change Event Without Time					40	
	4	16-Bit Delta Counter Change Event Without Time					24	
	5	32-Bit Counter Change Event With Time					88	
	6	16-Bit Counter Change Event With Time					72	
	7	32-Bit Delta Counter Change Event With Time					88	
	8	16-Bit Delta Counter Change Event With Time					72	
23	0	Frozen Counter Event - All Variations						

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	32-Bit Frozen Counter Event Without Time					40	
	2	16-Bit Frozen Counter Event Without Time					24	
	3	32-Bit Frozen Delta Counter Event Without Time					40	
	4	16-Bit Frozen Delta Counter Event Without Time					24	
	5	32-Bit Frozen Counter Event With Time					88	
	6	16-Bit Frozen Counter Event With Time					72	
	7	32-Bit Frozen Delta Counter Event With Time					88	
	8	16-Bit Frozen Delta Counter Event With Time					72	
30	0	Analog Input - All Variations	1	06			16	Master will generate this variation
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Master will generate and process this variation. Data returned to ladder logic will be Least Significant 16 bits only.
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Master will generate and process this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Master will generate and process this variation. Data returned to ladder logic will be Least Significant 16 bits only.
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Master will generate and process this variation
31	0	Frozen Analog Input - All Variations						
	1	32-Bit Frozen Analog Input					40	
	2	16-Bit Frozen Analog Input					24	
	3	32-Bit Frozen Analog Input With Time To Freeze					88	
	4	16-Bit Frozen Analog Input With Time To Freeze					72	
	5	32-Bit Frozen Analog Input Without Flag					32	
	6	16-Bit Frozen Analog Input Without Flag					16	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
32	0	Analog Change Event - All Variations	1	06, 07, 08			24	Master will generate this variation
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Master will generate and process this variation.
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Master will generate and process this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Master will generate and process this variation.
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Master will generate and process this variation
33	0	Frozen Analog Event - All Variations						
	1	32-Bit Frozen Analog Event Without Time					40	
	2	16-Bit Frozen Analog Event Without Time					24	
	3	32-Bit Frozen Analog Event With Time					88	
	4	16-Bit Frozen Analog Event With Time					72	
40	0	Analog Output Status - All Variations	1	06			24	Master will generate these variations and parse the responses. For variation 1, only 16-bit value is stored.
	1	32-Bit Analog Output Status	1	06	129, 130	00,01	40	
	2	16-Bit Analog Output Status	1	06	129, 130	00, 01	24	
41	0	Analog Output Block - All Variations					24	
	1	32-Bit Analog Output Block					40	
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Master will generate this variation and parse the response
50	0	Time and Date - All Variations					48	
	1	Time and Date	2	07, With Quant=1			48	Master will generate this variation
	2	Time and Date With Interval					80	
51	0	Time and Date CTO - All Variations						
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Master will process this variation
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Master will process this variation
52	0	Time Delay - All Variations						
	1	Time Delay Coarse			129	07, With Quant=1	16	Master will not process this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	Time Delay Fine			129	07, With Quant=1	16	Master will process this variation for propagation delay calculation
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Master will generate this variation
	2	Class 1 Data	1	06, 07, 08				Master will generate this variation
	3	Class 2 Data	1	06, 07, 08				Master will generate this variation
	4	Class 3 Data	1	06, 07, 08				Master will generate this variation
70	0	Not Defined						
	1	File Identifier						
80	0	Not Defined						
	1	Internal Indications	2	00, Index=7			24	The Master will generate this variation
81	0	Not Defined						
	1	Storage Object						
82	0	Not Defined						
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
No Object			13					Master supports the Cold Restart Function
			14					Master supports the Warm Restart Function
			20					Master supports the Enable Unsolicited Function

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
			21					Master supports the Disable Unsolicited Function

5.7 Master Port DNP Slave Configuration Values (DNP Master Slave List)

Column	Variable Name	Data Range	Description	IF Error	Config. Value
1	DNP Slave Address	0 to 65534	This is the slave address for the unit to override the default values.	Ignore	
2	Data Link Confirm Mode	Coded Value (0=Never, 1=Sometimes, 2=Always)	This value specifies if data link frames sent to the remote device require a data link confirm. This value should always be set to zero for almost all applications.	0	
3	Data Link Confirm Time-out	1 to 65535 milliseconds	This parameter specifies the time to wait for a data link confirm from the remote device before a retry is attempted.	300	
4	Maximum Retries for Data Link Confirm	0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.	3	
5	Application Layer Response Time-out	1 to 65535 milliseconds	Time-out period the Master will wait for each response message fragment. If data link confirms are enabled, make sure the time-out period is set long enough to permit all data confirm retries.	5000	
6	Slave Mode	Coded Value (Bit 0=Enable, Bit 1=Unsol Msg, Bit 2=Use DM, Bit 3=Auto Time Sync)	This word contains bits that define the slave mode. The slave mode defines the functionality of the slave device and can be combined in any combination. The fields have the following definition: Enable-- determines if this slave will be used. Unsol Msg-- causes an enabled unsolicited response message to be sent to the slave when its RESTART IIN bit is set. This parameter is also required for unsolicited message reporting by the IED unit.Use DM--uses delay measurement. Auto Time Sync--time synchronization used when NEED TIME IIN bit set.	5	

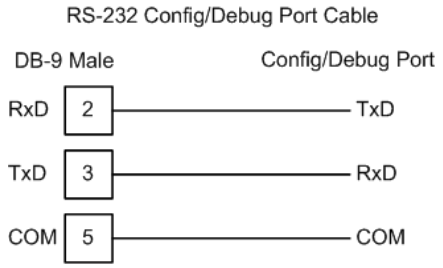
5.8 Cable Connections

The application ports on the MVI56-DNP module support RS-232, RS-422, and RS-485 interfaces. Please ensure that the jumpers are set correctly for the type of interface you are using.

Note for modules with RS-232 connection to a radio or modem: Some radios or modems require hardware handshaking (control and monitoring of modem signal lines) on the RTS and CTS lines of an RS-232 connection. Enable this by setting the *UseCTS* parameter in the module configuration to 1.

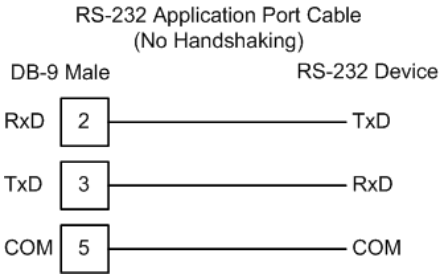
5.8.1 RS-232 Configuration/Debug Port

This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC-based terminal emulation program to view configuration and status data in the module and to control the module. The cable pinout for communications on this port is shown in the following diagram.



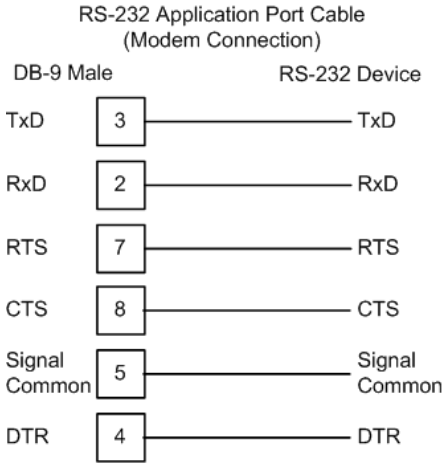
5.8.2 RS-232 Application Port(s)

When the RS-232 interface is selected, the use of hardware handshaking (control and monitoring of modem signal lines) is user definable. If no hardware handshaking will be used, here are the cable pinouts to connect to the port.



RS-232: Modem Connection (Hardware Handshaking Required)

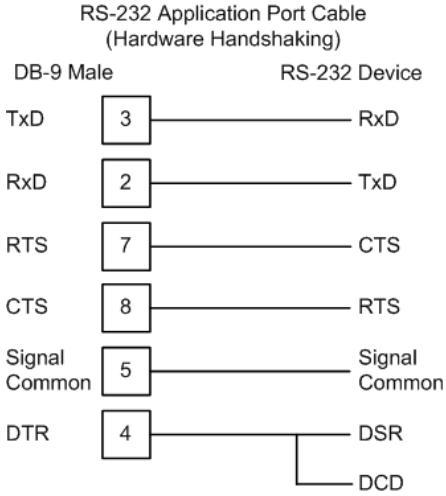
This type of connection is required between the module and a modem or other communication device.



The "Use CTS Line" parameter for the port configuration should be set to 'Y' for most modem applications.

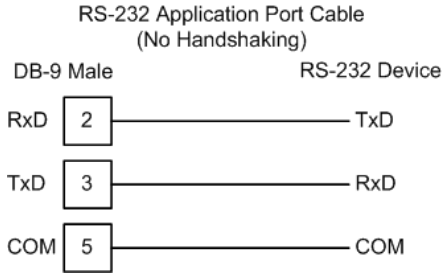
RS-232: Null Modem Connection (Hardware Handshaking)

This type of connection is used when the device connected to the module requires hardware handshaking (control and monitoring of modem signal lines).

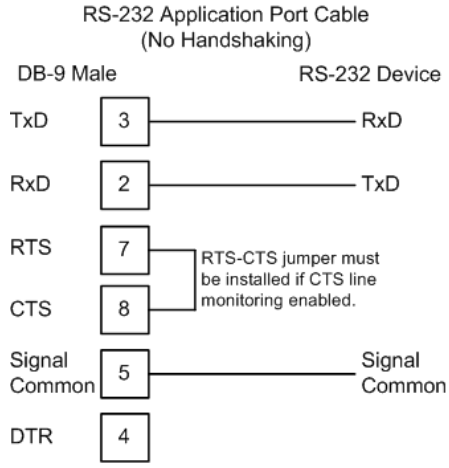


RS-232: Null Modem Connection (No Hardware Handshaking)

This type of connection can be used to connect the module to a computer or field device communication port.

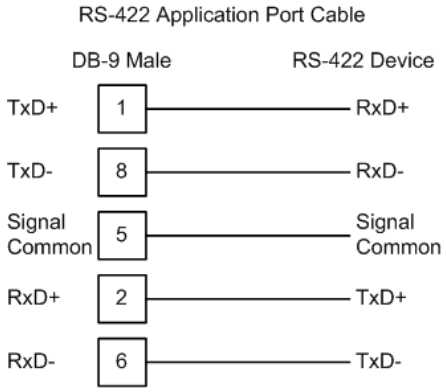


Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to N and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to Y, then a jumper is required between the RTS and the CTS lines on the port connection.



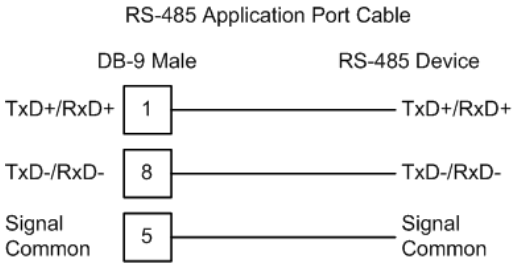
5.8.3 RS-422

The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



5.8.4 RS-485 Application Port(s)

The RS-485 interface requires a single two or three wire cable. The Common connection is optional, depending on the RS-485 network devices used. The cable required for this interface is shown below:

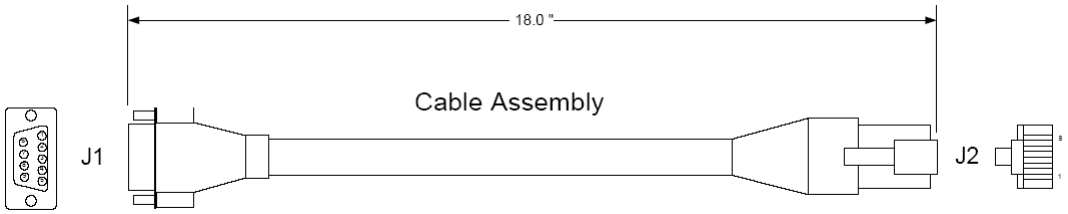


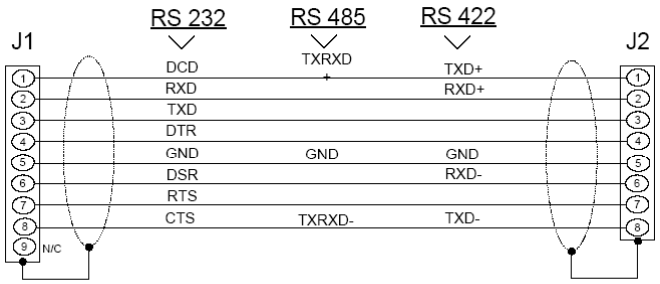
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In these cases, installing a 120-ohm terminating resistor between pins 1 and 8 on the module connector end of the RS-485 line may improve communication quality.

RS-485 and RS-422 Tip

If communication in the RS-422 or RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret + and -, or A and B, polarities differently.

5.8.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.9 Command List Entry Form

	0	1	2	3	4	5	6	7	8	9
#	Port/ Flags	Slave Add.	Object	Variation	Function	Address	Pnt Count	DNP DB	IED DB	Poll Interval
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										

6 Support, Service & Warranty

Contacting Technical Support

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1. Product Version Number
2. System architecture
3. Network details

If the issue is hardware related, we will also need information regarding:

1. Module configuration and associated ladder files, if any
2. Module operation and any unusual behavior
3. Configuration/Debug status information
4. LED patterns
5. Details about the serial, Ethernet or fieldbus devices interfaced to the module, if any.

Note: For technical support calls within the United States, ProSoft Technology's 24/7 after-hours phone support is available for urgent plant-down issues.

North America (Corporate Location)	**Europe / Middle East / Africa Regional Office**
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com