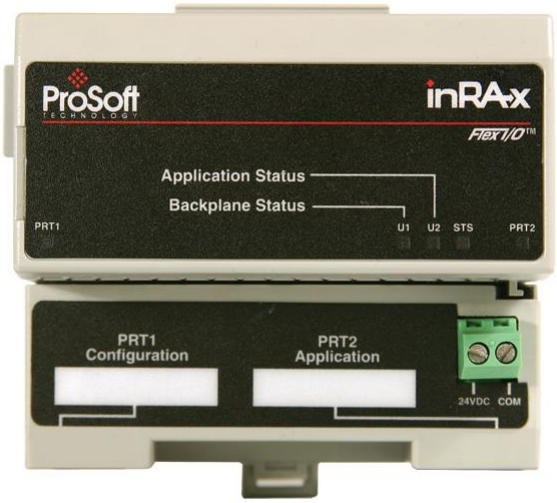
ProSoft TECHNOLOGY

Where Automation Connects.



MVI94-DNP

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

MVI94-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:

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

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.

Warnings

North America Warnings

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.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
C Suitable for use in Class I, division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

MVI94 Markings

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 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: 30g Operational; 50g non-operational; Vibration: 5 g from 10 Hz to 150 Hz
- Relative Humidity 5% to 95% without condensation)
- All phase conductor sizes must be at least 1.3 mm^2 and all earth ground conductors must be at least 4mm^2.

Contents

1 Quick Start

This section describes the procedure for installing and configuring the module for communication. These steps should be followed for successful implementation of a module in a user application.

1. Define the communication characteristics of the DNP master or slave port.
2. If a master device will be simulated, define the command lists to be used on the DNP master port.
3. Fill in the blank configuration form for application using the data sets defined in steps one and two.
4. Edit the configuration text file `Master.Cfg` or `Slave.Cfg` to reflect the desired data from the configuration form and save the file under a different name. These text files are provided as a starting point to ease application development.
5. Connect the MVI94-DNP module's Configuration/Debug Port to a computer containing HyperTerminal with a null-modem cable.
6. Connect the module to a 24 Vdc power source.
7. Select the directory containing the correct configuration file on the computer.
8. Start the terminal emulation program on the computer.
9. Press **[?]** to verify that the module is communicating with the computer and that the main menu mode is current.
10. Press **[>]** on the terminal emulator to begin receiving the configuration file. Press **[Y]** to confirm the request.
11. Press **[ALT-F3]** key (DOS Version) or open the Transfer menu and choose **SEND** (Windows Version) on the terminal emulator and enter the name of the configuration file to be loaded into the module. The configuration will be downloaded, and the module will restart using the new configuration.
12. Connect the module's DNP port to the DNP network. If everything is configured correctly and the cable connections are correct, communications should be present on the port.
13. Monitor the communication statistics for the port to verify that everything is working correctly.
14. View the database in the module using the terminal emulator.
15. Edit the ladder logic provided with the module for your system. The ladder logic is responsible for transferring the data between the module and processor.
16. Connect the module to the Flex I/O processor or ControlNet processor. If all is configured correctly, the data in the module should be visible in the processor.
17. Use the Configuration/Debug port to view the backplane transfer statistics.

2 Hardware Installation

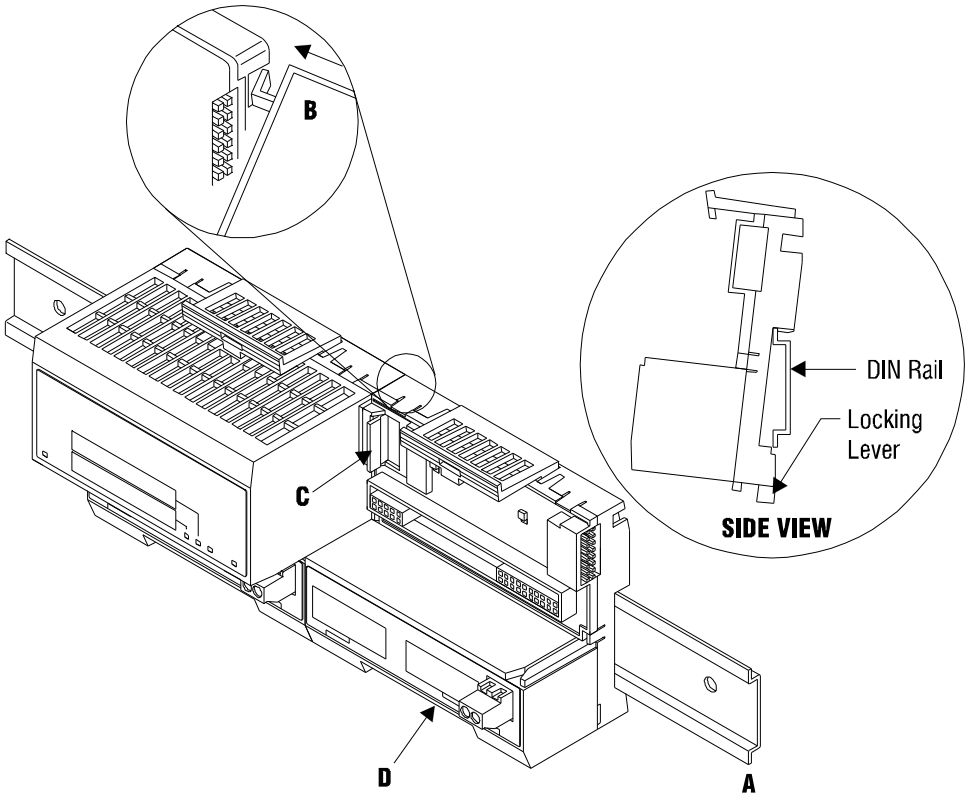
In This Chapter

2.1 Verify Package Contents

Make sure that you verify the contents of the product before you discard the packing material. The following components should be included with the product:

1 A MVI94 Flex I/O Base
2 A MVI94 Module with 3 jumpers installed
3 One Serial Adapter Cable

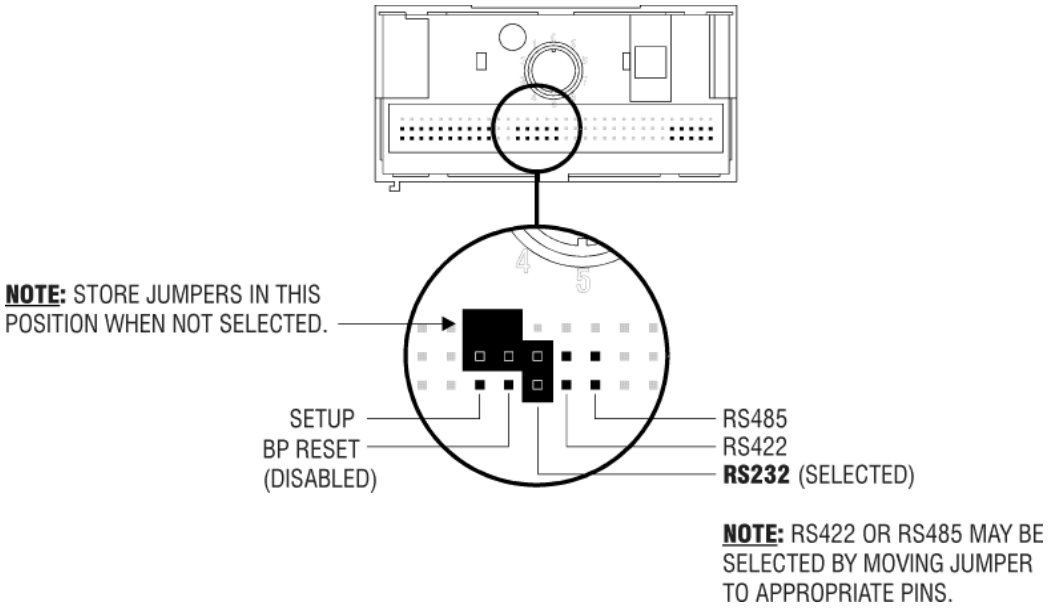
2.2 Mounting the MVI 94 Flex I/O Base



1. Remove the cover plug (if used) in the male connector of the unit to which you are connecting this Base.
2. Check to make sure that the 16 pins in the male connector on the adjacent device are straight and in line so that the mating female connector on this Base will mate correctly.
3. Make certain that the female flexbus connector **C** is **fully retracted** into the Base.
4. Position the Base on a 35 x 7.5mm DIN-rail **A** at a slight angle with the hook **B** on the left side of the Base hooked into the right side of the unit on the left.
5. Rotate the Base onto the DIN-rail with the top of the rail hooked under the lip on the rear of the Base. Use caution to make sure that the female flexbus connector does not strike any of the pins in the mating male connector.
6. Press the terminal base down onto the DIN-rail until flush. The locking tab **D** snaps into position and locks the terminal base to the DIN-rail.
7. If the Base does not lock in place, use a screwdriver or similar device to move the locking tab down, press the Base flush with the DIN-rail and release the locking tab to lock the base in place.
8. **Gently** push the female flexbus **C** connector into the adjacent base or adapter male connector to complete the flexbus connections.

2.3 Setting Jumpers

Before installing the MVI94 module onto its base, the module's configuration can be set using the jumpers on the bottom of the module as shown in this figure.



Port 2 RS-232/422/485: Select with jumper (shipped in 232).

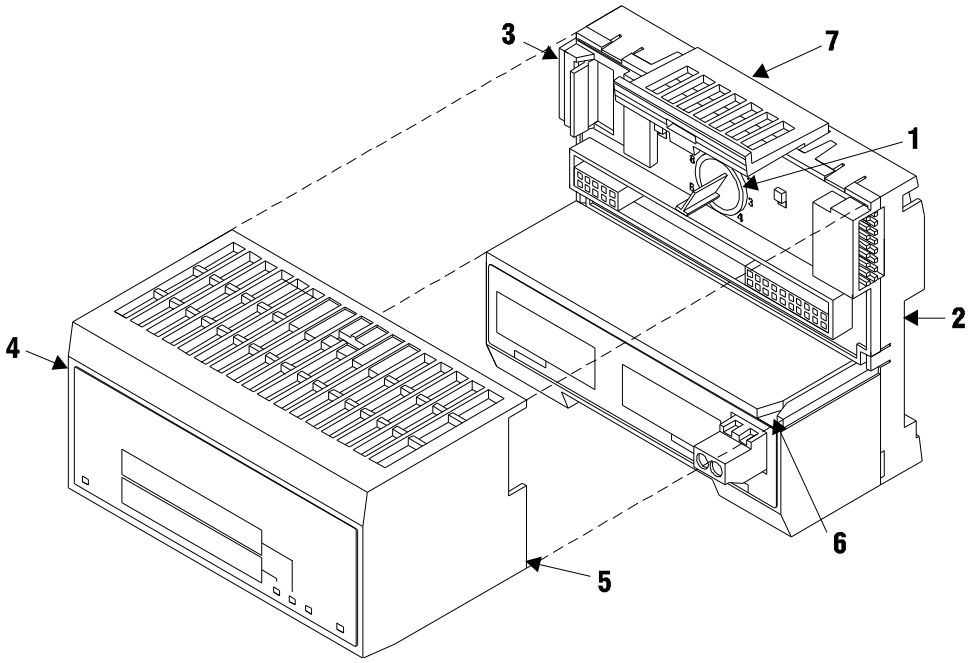
BP Reset: If the MVI94 module is to be reset when the Flex Bus is reset, install the BP RESET jumper in the Enabled position.

ATTENTION: Do not remove or replace a base unit when power is applied. Interruption of the flexbus can result in unintended operation or machine motion.

SETUP: To place the module in SETUP mode, install the jumper in the Selected position (DOS default). To prevent the module from being in Setup mode, leave the jumper in the disabled position.

2.4 Installing the Module onto the Base

1 Rotate the keyswitch **1** on the Base clockwise to position #1.
2 Make certain the flexbus connector **3** on the Base is pushed all the way to the left to connect with the neighboring base or adapter. **The Module cannot be installed unless the flexbus connector is fully extended.**
3 Make sure that the pins on the bottom of the Module are straight so they will align properly with the connector socket on the Base.
4 Position the Module with its alignment bar 5 aligned with the groove 6 on the Base.

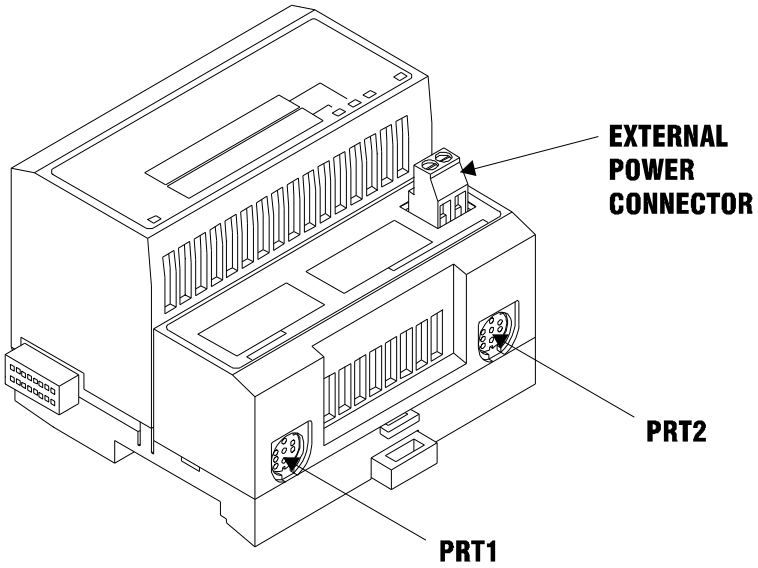


5 Press firmly and evenly to seat the Module in the Base. The Module is seated when the latch **7** on the Base is locked into the Module.

2.5 Installing the Serial Adapter Cables

Two identical serial adapter cables are supplied. Each cable has a locking-type 8 pin Mini-DIN plug on one end and a DB-9 male connector on the other end. The Mini-DIN connector on each cable is inserted into the Mini-DIN receptacles marked **PRT1** and **PRT2** on the Base.

To install the locking-type Mini-DIN connector, slide the spring-loaded sleeve back while inserting the plug into the receptacle on the Base, and then release the sleeve when fully seated. The locking mechanism prevents the cable from being removed during normal operation. To remove the cable, slide the sleeve back and remove the plug.



2.6 Wiring the Power Connections

External power is supplied to the Base on the 2 pin screw terminal block. The power supply can be either 24Vdc or 12Vdc, and should be located in close proximity of the base.

- Connect dc common to the **COM** terminal
- Connect +24V dc or +12V dc to the **24VDC** terminal

3 Configuration

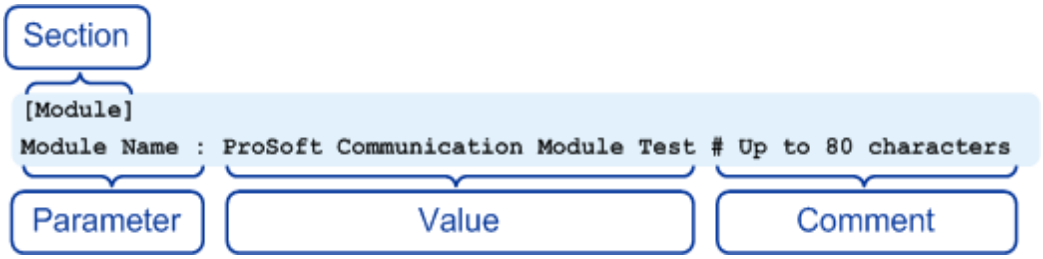
In This Chapter

3.1 Configuration File

The MVI94-DNP module stores its configuration in a text file called Master.Cfg or Slave.Cfg, located in the module's flash memory. When the module starts up, it reads the configuration file and uses the information to control how the Distributed Network Protocol protocol interacts with the module's application port(s).

The configuration file is arranged in *Sections,* with a heading in **[ ]** characters at the beginning of each section. Each *Section* contains a list of *Parameters* and *Values,* followed by an optional *Comment* that explains the parameter.

The following illustration shows an example of a *Section,* a *Parameter,* a *Value,* and a *Comment.*



The *Parameter* must be followed by a **[:]** (colon) character. The text following the **[:]** is a *Value.*

The module ignores *"comment"* text following the **[#]** character. Use comments to document your configuration settings.

You can get a sample configuration file for the module in the following places:

- Copy the Master.Cfg or Slave.Cfg from the module's flash memory to your PC
- Copy the Master.Cfg or Slave.Cfg from the ProSoft Solutions CD-ROM supplied with the module
- Download the Master.Cfg or Slave.Cfg from the ProSoft Technology web site at www.prosoft-technology.com

3.1.1 Editing the Configuration File

The DNPSNET_Q.CFG file consists of the following sections:

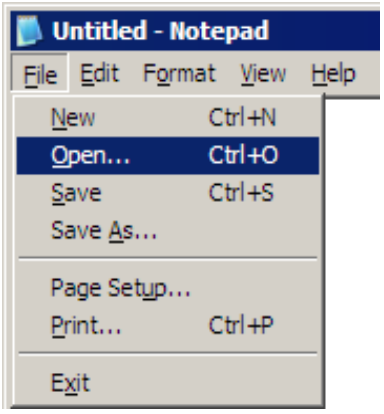
- [Module]
- [Backplane Configuration]
- [DNP ENET Slave]
- [DNP ENET IP Addresses]
- [DNP Slave Binary Inputs]
- [DNP Slave Analog Inputs]
- [DNP Slave Float Inputs]
- Important notes to consider when editing the sample configuration file:
- Comments within the file are preceded by the pound (#) sign. Any text on a line that occurs after the # character will be ignored.
- Do not use tabs or other non-printing characters instead of spaces to separate parameters (spacebar).
- Parameter names must begin in the first column of a line, and may not be preceded with a space (spacebar) or other non-printing character.

The Master.Cfg or Slave.Cfg file is a plain ASCII text file. Use a text editor such as Notepad.exe (included with Microsoft Windows) to open and edit the file.

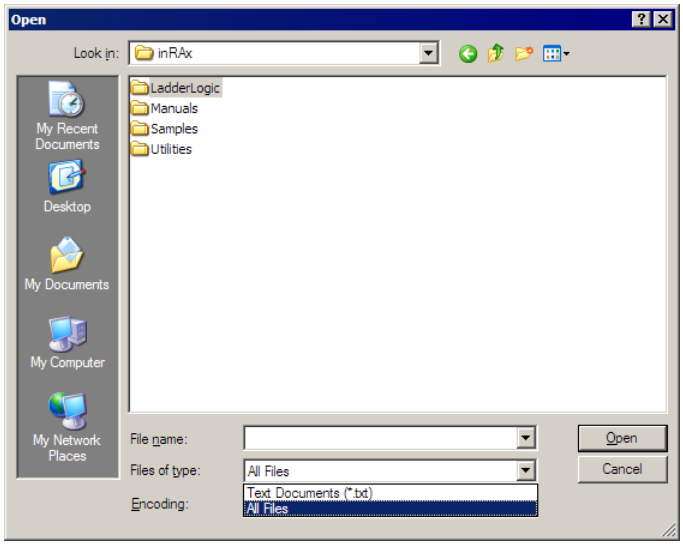
To open the configuration file in Notepad

1 Click the **START** button, and then choose **PROGRAMS**
2 Expand the *Programs* menu, and then choose **ACCESSORIES**.
3 On the *Accessories* menu, choose **NOTEPAD.**

4 In *Notepad*, open the **FILE** menu, and then choose **OPEN**



5 In the *Open* dialog box, select **ALL FILES** in the *Files of Type*: dropdown list.



Tip: Sample configuration files are stored under the LadderLogic folder on the ProSoft Solutions CD-ROM.

6 Navigate to the folder containing the configuration file, and then select the file to edit.
7 Click **OPEN** to open the file.
8 When you have finished editing, **SAVE** the file and close Notepad.

Important: Changes to the configuration file will not take effect until you download the file to the module, and then reboot the module.

3.2 [Module]

The [MODULE] section defines if the module will emulate a DNP master or slave device. The following table describes the parameters in this section.

Variable Name	Data Range	Description	IF Error
[Module]		This section header indicates the start of the module specific parameter set.	
Module Name:	80 Characters	This parameter sets the name or description of the module's application	"Not Configured"
Type:	M=Master or S=Slave	This parameter must be set to M for the DNP port to use as a master port. If the parameter is set to any other value, the port will be used as a slave port.	S

3.3 [DNP Slave]

The [DNP SLAVE] section is only required if the module is emulating a slave device. This section contains the information required to configure the communication port and set the protocol specific parameters. Refer to the configuration form (page 86) for a complete list of the parameters set in this section.

3.4 [DNP Slave Database]

The [DNP SLAVE DATABASE] section is only used if the module is simulating a slave device. This section contains the information to size the database to be used to interface with the remote master. The values entered in this section define the blocks to be transferred between the module and the processor over the backplane and the data to be interfaced with the remote DNP master device. The parameters defined in this section are shown in the configuration form (page 86).

3.4.1 Slave Class/Deadband Overrides

The [DNP SLAVE x INPUTS] sections are only used if the module is simulating a slave device. These sections contain the class and deadband information to override the default values stated under the [DNP Slave] section for each point in the input data sets. If no overrides are to be specified for a data type, its section can be omitted. The parameters defined in these sections are shown in the configuration form displayed in the Reference chapter.

3.5 [DNP Master]

The [DNP MASTER] section is only used if the module is simulating a DNP master device. This section contains information required to configure the communication port and set the protocol specific parameters. The parameters set in this section are shown in the configuration form (page 86).

3.6 [IED Database]

The [IED DATABASE] section is only used if the module is simulating a DNP master device. The size of each point type in the database is specified in the section. These data areas are transferred between the module and the processor using the module's I/O image. These data areas also interface to attached IED units connected to the port. Commands present in the master port command list use the database to control and monitor the attached devices. The parameters set in this section are described in the configuration form (page 86).

3.7 [DNP Master Slave List]

The [DNP MASTER SLAVE LIST] section is only used if the module is simulating a DNP master device. This list defines the protocol specific information for each IED unit to be interfaced with on the master port.

The slave list is formatted differently than the other sections of the configuration file. Each IED unit is present on a separate line in the section between the labels **START** and **END**. These labels inform the program where the lists starts and stops. The module's program will parse each IED unit's information from each line and place the data in the module and will continue reading records until the **END** label is reached or the maximum number of slave are processed.

The following table describes the format of each record.

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	

An example of the [DNP Master Slave List] section is as follows:

[DNP Master Slave List]		This section header defines the location of the DNP master slave definition list.

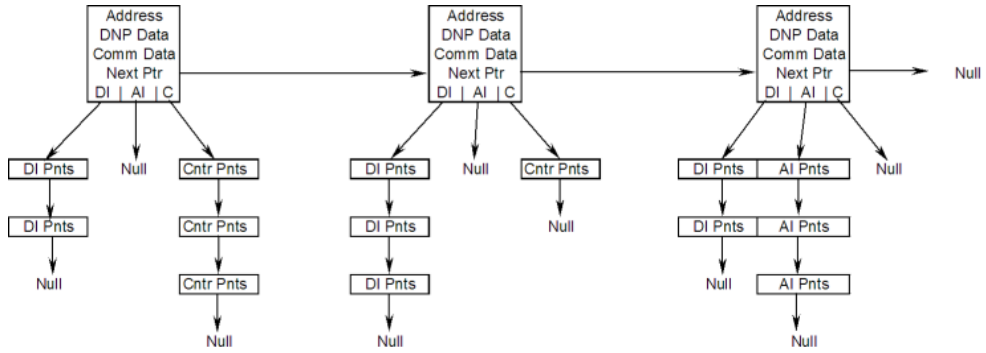
```
# DL Conf Mode ==> 0=Never, 1=Sometimes and 2=Always (select 0).
# Flag parameter is bit coded as follows:
# Bit 0 (decimal 1) ==> Enable the slave
# Bit 1 (decimal 2) ==> Use Unsolicited messaging with this slave
# Bit 2 (decimal 4) ==> Use delay measurement with this slave
# Bit 3 (decimal 8) ==> Auto time synchronization enabled
#
```

START		This string signals the start of the slave definition list.

```
# Node DL Conf Conf Conf App Rsp
# Address Mode Timeout Retry Timeout Flags
2 0 1000 0 5000 9
```

END		This string signals the end of the slave definition list.

A record is required for each IED unit on the master port. As the program reads in the information, it generates a list of slaves as shown in the following example:



When the commands are read in from the configuration, a link between the IED point and the module's database is generated and stored in this link list. This data is used by the module to handle event data transferred from the IED units to the module. When the master port receives an event from an IED unit, the data in the link list places the received data into the module's database.

3.7.1 Command List

The command list stores the command list used by the DNP master port. This list only must be defined 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

The definition of each parameter required for each command is provided in the following table.

Bits in the Port/Flags parameter are dependent on the data type. The following table defines the Port/Flags bits 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)	0 or 2
2	Enable/Disable Command (1=Enable, 0=Disable)	4
3	RBE Flag(0=Events from IED, 1=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)

The following table defines the Port/Flags bits 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 IED unit address on the DNP network to consider with the command. 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, 12, 20, 21, 30, 32, 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 to consider in the IED unit.

Point Count

This parameter defines the number of points in the IED unit. Refer to the discussion above for the interpretation of this parameter's values for the different object types.

DNP DB Address

This parameter defines the starting location 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 location 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 the field. A value of 0 for the parameter implies that the command should be executed every scan of the list.

3.8 [DNP Master Commands]

The [DNP MASTER COMMANDS] section of the configuration file sets the DNP master port command list. This list polls DNP slave devices attached to a simulated master port. The module supports numerous commands. This permits the module to interface with a wide variety of DNP protocol devices.

The command list is formatted differently than the other sections of the configuration file. Commands are present in a block between the labels **START** and **END**. These labels inform the program where the list resides. The module's program will parse all commands after the **START** label until it reaches the **END** label or maximum number of commands are processed.

The following table describes the structure of a single command.

[DNP Master Commands]		This section header defines the location of the DNP master command list.
START		This string signals the start of the command list.

```
# 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
6 2 1 0 1 0 -20 -1 0 0
```

END		This string signals the end of the command list.

Function codes used for each command are specific to the module and are defined to permit the module to interface with a wide variety of DNP communication devices.

3.9 Uploading and Downloading the Configuration File

ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you may transfer the file from the module to your PC or locate and load the file from the distribution CD-ROM supplied with the module. After editing, you must transfer the file back to the module for your changes to take effect.

This section describes these procedures.

Important: The illustrations of configuration/debug menus in this section are intended as a general guide and may not exactly match the configuration/debug menus in your own module. For specific information about the configuration/debug menus in your module, refer to The Configuration/Debug Menu (page 35).

3.9.1 Transferring the Configuration File to Your PC

1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.

```
******* 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 = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
```

2 Press **[>]** key (Send Module Configuration). The message "Press Y to confirm configuration send!" is displayed at the bottom of the screen.

```
******* 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 = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Receive Configuration File from Remote PC by pressing 'Y' key....
```

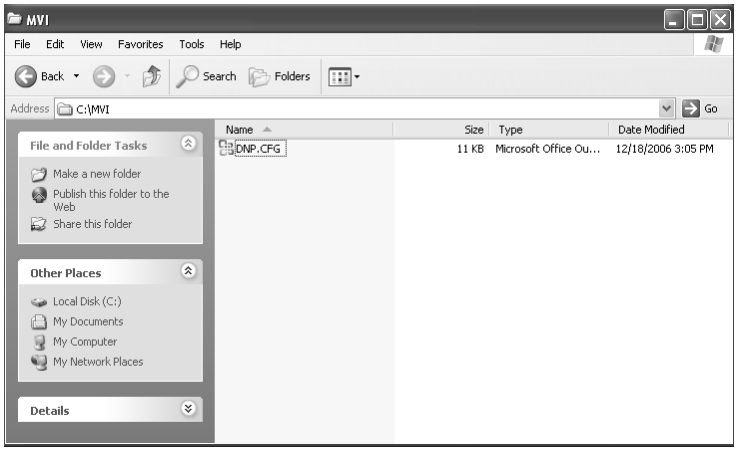
3 Press **[Y].** The module will automatically start a Zmodem file transfer. The configuration file will be stored in the default file transfer folder on your PC.

Note: ProSoft Technology suggests that you download the configuration file pre-loaded on your module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at www.prosoft-technology.com.

When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.

```
 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 = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Send Configuration File to Remote PC by pressing 'Y' key....

Sending configuration file:


TRANSFERRING CONFIGURATION FILES FROM MVI MODULE TO PC:


Select RECEIVE menu option and receive files *.*
ëOCONFIGURATION FILE TRANSFERRED TO PC.
```

The configuration file is now in a folder on your PC. To view the location of this folder, open the Transfer menu in HyperTerminal and choose Receive File.



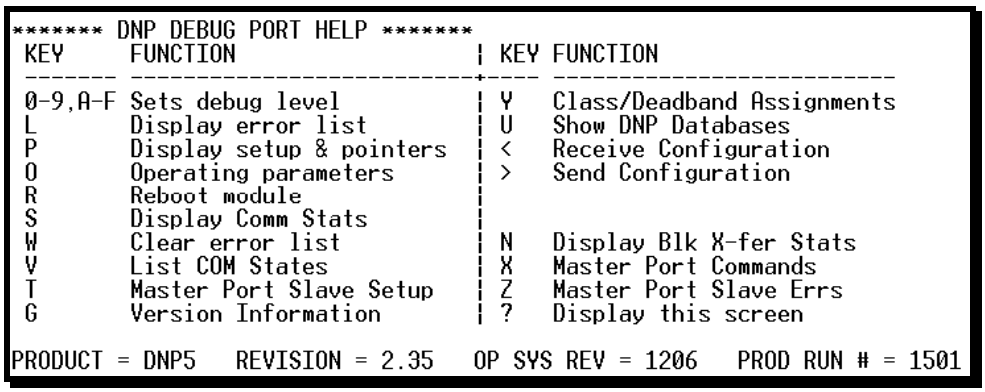
4 You can now open and edit the file in a text editor such as Notepad. When you have finished editing the file, save it and close Notepad.

Important: You must name this file DNP.CFG before you transfer it to the module. The module will not recognize configuration files with any other name or extension.

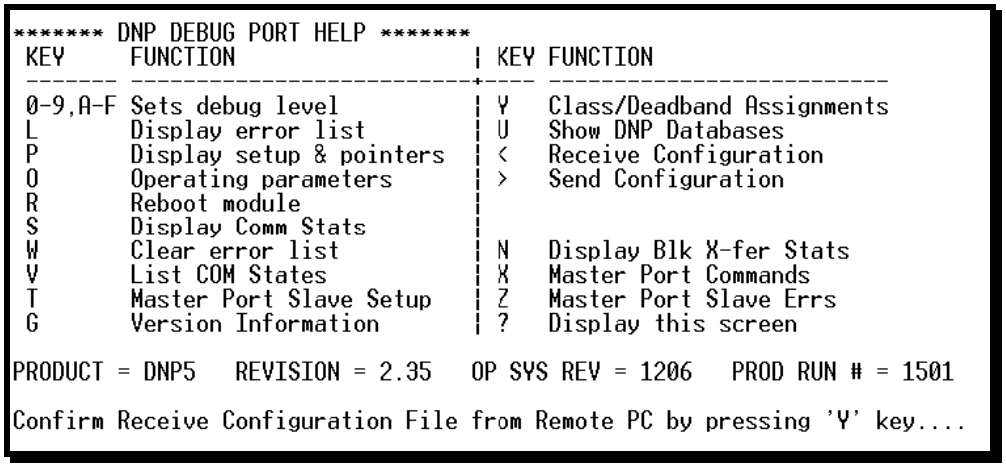
3.9.2 Transferring the Configuration File to the Module

Perform the following steps to transfer a configuration file from your PC to the module.

1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



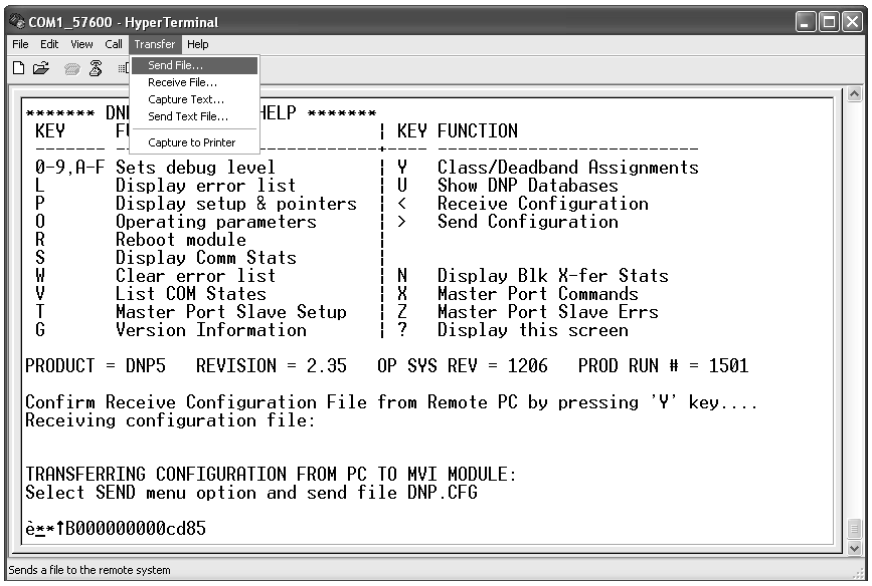
2 Press **[<]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



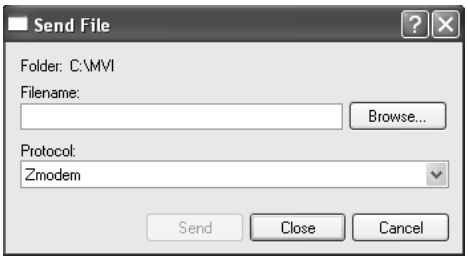
3 Press **[Y]**. The screen now indicates that the PC is ready to send.

```
******* 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 = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Receive Configuration File from Remote PC by pressing 'Y' key....
Receiving configuration file:


TRANSFERRING CONFIGURATION FROM PC TO MVI MODULE:
Select SEND menu option and send file DNP.CFG

è*↑B000000027fed4
```

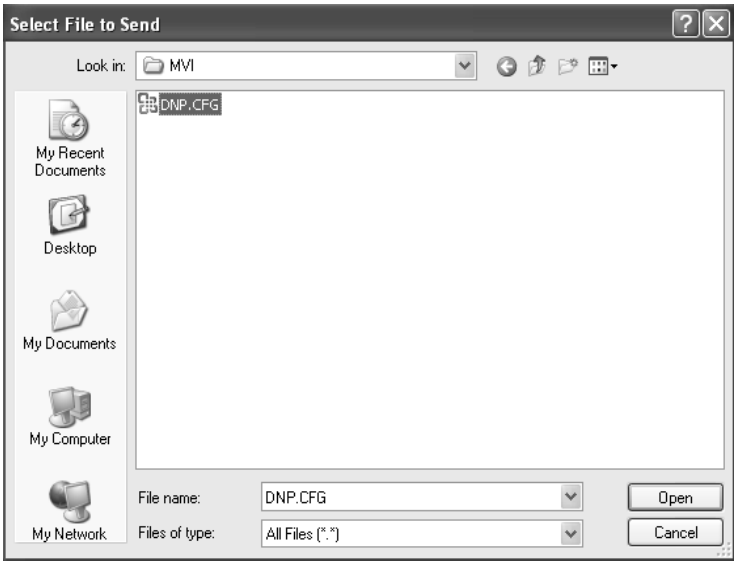
4 From the **Transfer** menu in HyperTerminal, select **Send File**.



The Send File dialog appears.

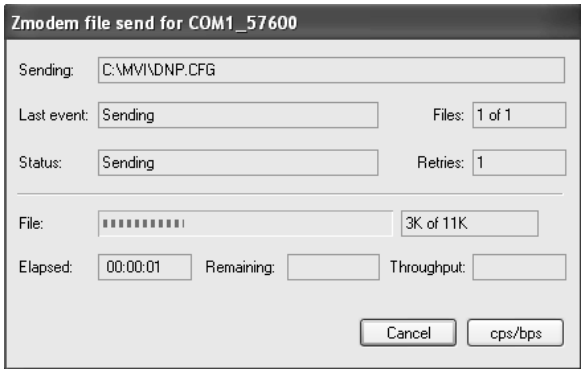


5 Use the Browse button to locate the configuration file your computer.

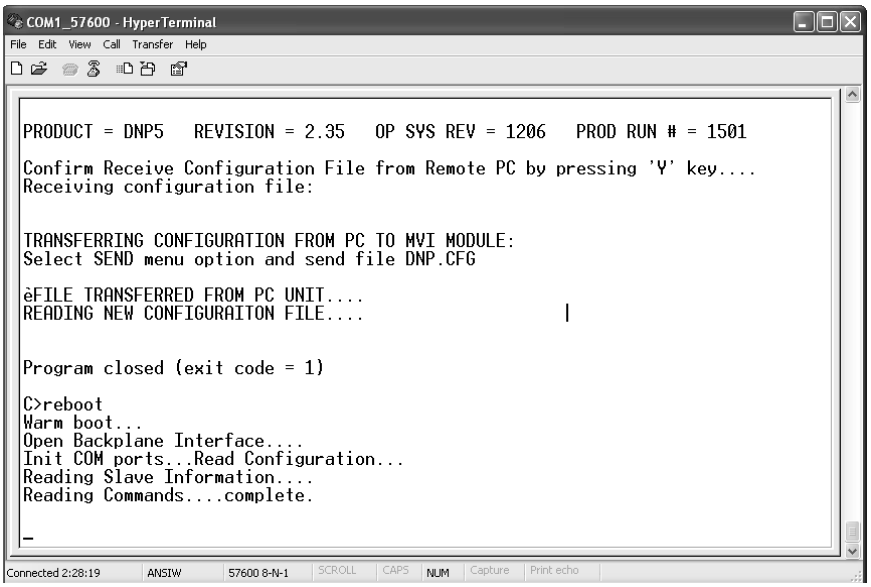


Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site.

6 Select Zmodem as the protocol.
7 Click the Send button. This action opens the Zmodem File Send dialog box.



When the upload is complete, the screen indicates that the module has reloaded program values and displays information about the module.



8 Your module now contains the new configuration. Press **[?]** to see the module's main menu.

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 *Microsoft Windows HyperTerminal*.
- 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

This section defines the indications provided on the MVI94-DNP module through LEDs.

4.1.1 PRT1

This LED indicates data transmit and receive activity on the configuration port. When the TXD or RXD pin is active on the port, the LED will be illuminated green. When the port is not active, the LED will be in the off state.

4.1.2 U1

This LED indicates backplane data transfer operation. When the module is successfully writing data to the FLEX I/O backplane, the LED will be in the off state. When the module is reading a new block of data from the FLEX I/O backplane, the LED will be in the on state (amber). During normal operation of the module, this LED should turn on and off at a very rapid rate. If the LED never turns on, check your ladder logic to verify that the data transfer is set up correctly.

4.1.3 U2

This LED indicates communication errors on the DNP application port. The LED is off when no error exists on the port. If a communication error is recognized on the port, the LED will be illuminated. If the LED is turned on, check for errors in the command list to determine the error condition recognized by the module.

4.1.4 STS

This LED indicates the "health" of the module. When power is applied to the module, the LED will be illuminated. If the LED is green, the program is working correctly and the user configuration is being used. If the LED is red, the program is halted. Try restarting the module by cycling power. This should cause the module to return to its normal state.The module should return to its normal state.

4.1.5 PRT2

This LED indicates data transmit and receive activity on the DNP master/slave port. When the TXD or RXD pin is active on the port, the LED will be illuminated green. When the port is not active, the LED will be in the off state.

4.2 The Configuration/Debug Menu

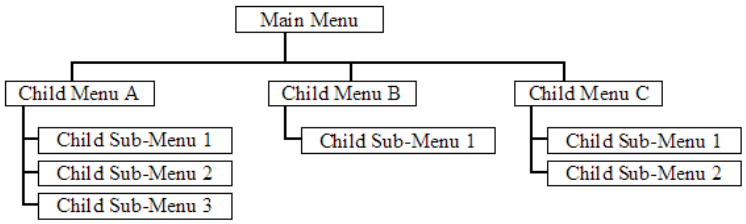
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in *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.

4.2.1 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.

4.2.2 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.

4.3 Required Software

In order to send and receive data over the serial port (COM port) on your computer to the module, you must use a communication program (terminal emulator).

A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have been tested by ProSoft Technology.

DOS	ProComm, as well as several other terminal emulation programs
Windows 3.1	Terminal
Windows 95/98	HyperTerminal
Windows NT/2000/XP	HyperTerminal

The module uses the Ymodem file transfer protocol to send and receive configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

4.4 Using the Configuration/Debug Port

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

1. Connect your computer to the module's port using a null modem cable.
2. Start the communication program on your computer and configure the communication parameters with the following settings:

Baud Rate	57,600
Parity	None
Data Bits	8
Stop Bits	1
Software Handshaking	None

3. Open the connection. When you are connected, press the **[?]** key on your keyboard. If the system is set up properly, you will see a menu with the module name followed by a list of letters and the commands associated with them.

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

1. 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.
2. Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC.
3. Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
4. 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, you can contact ProSoft Technology Technical Support for further assistance.

4.4.1 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
```

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 failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. 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 38) 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 all data associated with the DNP Server driver. For more information about the commands on this menu, refer to DNP Database View Menu (page 41).

Receiving the Configuration File

Press **[<] (SHIFT COMMA)** to download (receive) the current configuration file from the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File (page 27).

Sending the Configuration File

Press **[>] (SHIFT PERIOD)** to upload (send) an updated configuration file to the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File (page 27).

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
```

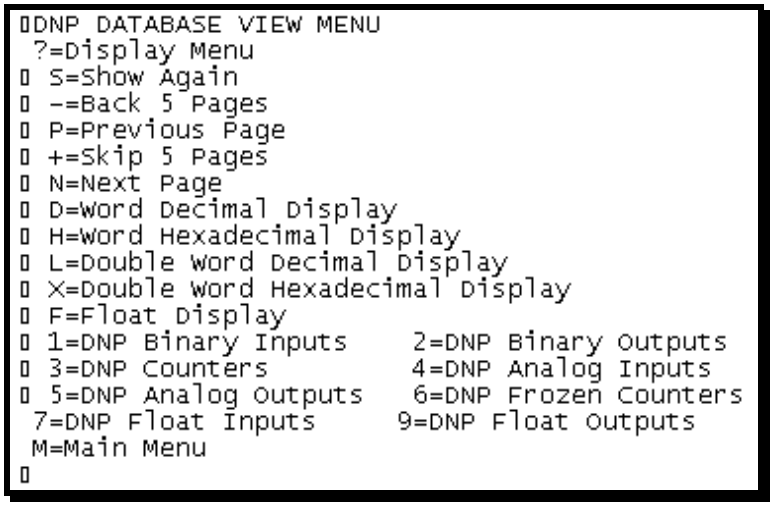
Opening the Command Error List Menu

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 _
```

4.4.2 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

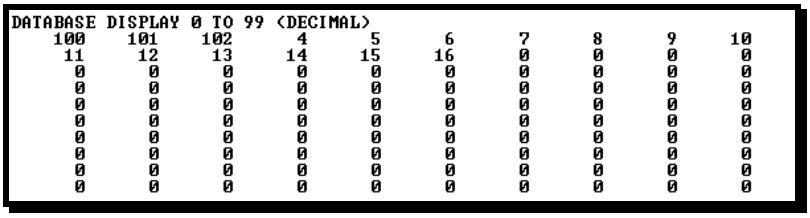
To view sets of register pages, use the keys described below:

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.



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 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.

4.4.3 The Class Assignment Menu

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

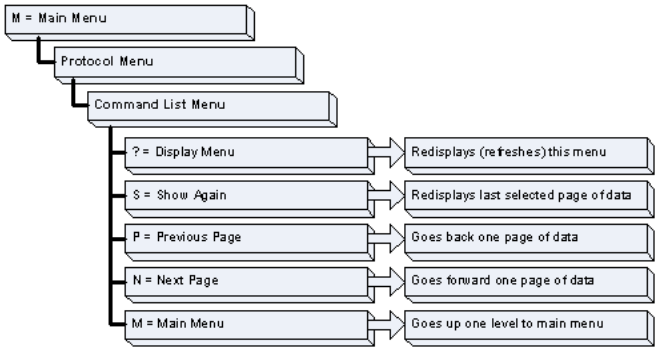
Press **[N]** to display the next page of data.

Returning to the Main Menu

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

4.4.4 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.



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

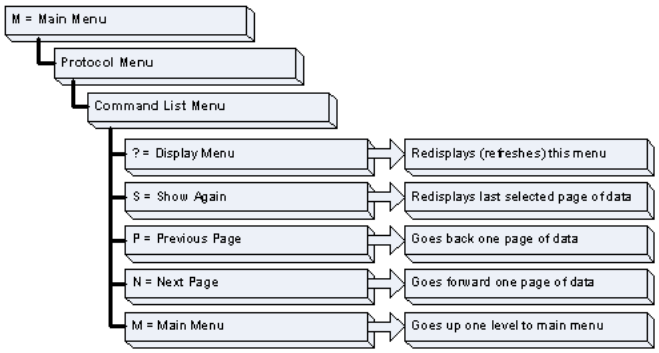
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.4.5 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.



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.

4.5 Reading Status Data from the Module

The module provides three sets of error/status data areas. The data sets are Error/Status Data, Slave Status Data, and Command Error List Data. This data is available for viewing through the Config/Debug port.

4.6 Error Status Data

The module error/status data areas are discussed in this section. Three sets of data are available with this type of data: Error/Status Data, Slave Status Data and Command Error List Data. Each of these datasets are discussed in the following topics:

4.6.1 Error Status Data

This data is provided to the processor using blocks 200 to 210 and is available for viewing using the configuration/debug port option. The module transfers this data to the processor at the end of each scan of normal data-block transfer. Each block contains 6-words of data. Ladder logic should be written to place the data received into a user data file. The following table describes the block numbers and data received in each block:

Word	Block Number	Block Offset	Variable Name	Description
0	200	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	200	3	DNP Slave Port last transmitted error code	This value represents the last error code transmitted to the master by this slave port.
2	200	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	200	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	200	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	200	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	201	2	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.

Word	Block Number	Block Offset	Variable Name	Description
7	201	3	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	201	4	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	201	5	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	201	6	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	201	7	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.
12	202	2	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
13	202	3	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	202	4	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
15	202	5	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	202	6	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	202	7	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	203	2	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	203	3	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	203	4	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.

Word	Block Number	Block Offset	Variable Name	Description
21	203	5	Total blocks transferred	Total BTR/BTW or side-connect interface transfers attempted by the module.
22	203	6	Successful blocks transferred	This value represents the total number of transfer operations between the PLC and module that are successful.
23	203	7	Total errors in block transfer	Total number of transfers that resulted in an error condition.
24	204	2	Total BTR or write errors	Total number of BTR or write transfers that resulted in an error.
25	204	3	Total BTW or read errors	Total number of BTW or read transfers that resulted in an error.
26	204	4	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
27	204	5	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	204	6	Block transfer error flag	This flag indicates that data is not being successfully transferred between the PLC and the module. This flag corresponds to the Device Trouble IIN bit.
29	204	7	Configuration Type	This is a coded field that defines the configuration of the module. The codes are as follows: 1=Slave Configuration, 2=Master Configuration
30 to 31	205	2 to 3	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
32 to 33	205	4 to 5	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
34 to 35	205	6 to 7	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
36 to 37	206	2 to 3	Production Run Number (ASCII)	These two words contain the production "batch" number for the particular chip in the module in ASCII format.
38	206	4	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	206	5	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.

Word	Block Number	Block Offset	Variable Name	Description
40	206	6	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	206	7	DNP Master Port Frame Block Count	This value represents the number of physical layer frame memory allocation blocks used by the program.
42	207	2	DNP Master Port Data Link Receive Block Count	This value represents the number of receive data link layer memory blocks allocated.
43	207	3	DNP Master Port Data Link Transmit Block Count	This value represents the number of transmit data link layer memory blocks allocated.
44	207	4	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer receive memory blocks allocated.
45	207	5	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer transmit memory blocks allocated.
46	207	6	DNP Master Port Device Memory Allocation Error Count	This value represents the number of memory allocation errors for device blocks.
47	207	7	DNP Master Port Physical Layer Memory Allocation Error Count	This value represents the number of memory allocation errors for physical layer frame blocks.
48	208	2	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	208	3	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.
50	208	4	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	208	5	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	208	6	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.

Word	Block Number	Block Offset	Variable Name	Description
53	208	7	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	209	2	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	209	3	Scan Counter LSB	Program scan counter
56	209	4	Scan Counter MSB	
57	209	5	Free Memory LSB	Free memory in module
58	209	6	Free Memory MSB	
59	209	7	Reserved	Future Use
60	210	2	DNP Slave Float Input Event count	This value contains the total number of floating-point input events which have occurred.
61	210	3	DNP Slave Double Float Input Event count	This value contains the total number of double floating-point input events which have occurred.
62	210	4	Reserved	Future Use
63	210	5	Reserved	Future Use
64	210	6	Reserved	Future Use
65	210	7	Reserved	Future Use

4.6.2 Command Error List

Each command in the command list has a reserved word value for a status/error code. This error data list can be read using the Configuration/Debug Port. Additionally, can be retrieved from the module by the processor using command block 9950.

The first word in the register location defined contains the status/error code for the first command in the port's command list. Each successive word in the command error list is associated with the next command in the list. Therefore, the size of the data area is dependent upon the number of commands defined.

Refer to the following Error Codes section to interpret the status/error codes present in the data area.

4.6.3 Slave Status Data

The slave status data contains the communication statistics for each IED unit interfaced with by the master port. This information is available for viewing using the configuration/debug port or by using command block 9949 requests from the processor.

4.7 Error Codes

The module error codes are listed in this section. Error codes are separated into module errors and command errors. The following two sections present the errors associated with each set of errors.

4.7.1 Module Error Codes

These error codes are generated by the module in response to communication problems on an emulated slave port or configuration errors. Review the error list to view the last set of 60 errors generated by the module. The error codes are listed in the following tables:

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.

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 input 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.

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.

4.7.2 Command Error Codes

Command error codes are generated by the module's program. These errors are generated when an error occurs when issuing a request or processing a response of a command list function. The following tables list the command error codes used in the module:

General Command Errors

Error Code	Name	Description
1	Device not defined	The IED slave address referenced in the command is not defined in the module. Check to make sure there is an entry in the slave table for each slave device referenced in the command list.
2	Invalid command	This command is not valid. Check to make sure the slave address parameter is greater than or equal to zero and that the point count is not set to zero.
3	Object not supported	The data object in the command is not supported by the module. Refer to the DNP subset for the Master Port.
4	Command function not supported	The function specified in the command is not supported for the object type selected. Refer to the DNP subset for the Master Port.
10	Invalid binary input poll command	This binary input object command is not valid.
11	Invalid binary input event poll command	This binary input event object poll command is not valid.
20	Invalid binary output command function	This binary output command function is not valid.
30	Invalid counter poll command function	The counter object poll command contains an invalid function code.
31	Invalid counter poll command	This counter object poll command is not valid.
40	Invalid frozen counter poll command	This frozen counter object poll command is not valid.
50	Invalid analog input poll command	This analog input poll command is not valid.
51	Invalid analog input event poll command	This analog input event poll command is not valid.
60	Invalid analog output poll command function	This analog output poll command contains an invalid function code.
61	Invalid analog output poll command	This analog output poll command is not valid.
70	Invalid time/date poll command	This time/date object poll command is not valid.
80	Invalid event poll command	This event poll command is not valid.

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.

Use the error codes returned for each command in the list to determine the success or failure of the command. If the command fails, use the error code to determine the cause of failure.

5 Reference

In This Chapter

5.1 Product Specifications

The MVI94 DNP 3.0 Master/Slave Communication Module is a single slot, backplane compatible DNP 3.0 interface solution for the FLEX 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 FLEX 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 FLEX platform. The module acts as an input/output module between the DNP network and the FLEX backplane. The data transfer from the FLEX 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

Some of the general specifications include:

- Operation via simple ladder logic
- Complete setup and monitoring of module through Debug port and user configuration file
- Flex backplane interface via I/O access

5.1.2 FLEX I/O Interfaces

Specification	Description
Form Factor	Single Slot 1794 Backplane compatible Locate in any slot of Backplane
Backplane current load	20 mA @ 5 V
External power supply	12 Vdc to 24 Vdc 340 mA to 170 mA
Operating temperature	0°C to 55°C (32°F to 131°F)
Storage temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational 5 g from 10150 Hz
Relative humidity	5% to 95% (without condensation)
LED indicators	Module status Backplane transfer status Application status Serial activity and error LED status
Configuration Serial port (PRT1)	Mini-DIN RS-232 Hardware handshaking
Application serial Port (PRT2)	Mini-DIN RS-232/422/485 jumper selectable 500V optical isolation from backplane
Dimensions (with Module installed in Base)	3.7H x 3.7W x 2.7D inches 94H x 94W x 69D mm

5.1.3 Functional Specifications

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

User defined internal register space is accessible to the protocol driver and to the FLEX processor memory.

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 databases. If a DNP Master port is also configured, a portion of these slave databases 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 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 (Date and Time stamping is not stored or used by module)
- Special command handling for Digital Output CROB under processor control for pulse output control

DNP 3.0 port (PRT2)

- User-definable module memory usage
- Full radio, modem and multi-drop support
- 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

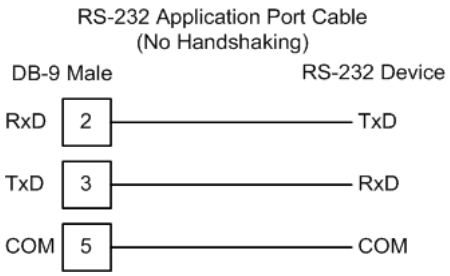
5.2 Cable Connections

The application ports on the MVI94-DNP module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

Note: When using RS-232 with radio modem applications, some radios or modems require hardware handshaking (control and monitoring of modem signal lines). Enable this in the configuration of the module by setting the UseCTS parameter to 1.

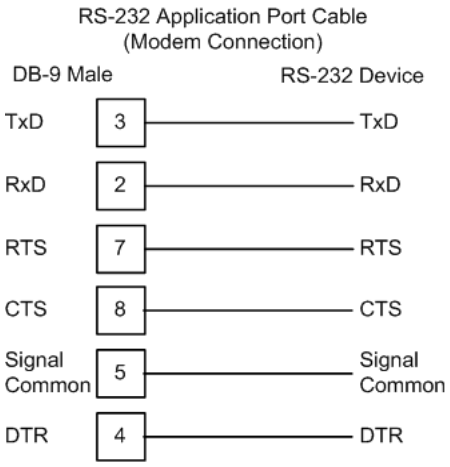
5.2.1 RS-232 Application Port(s)

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)

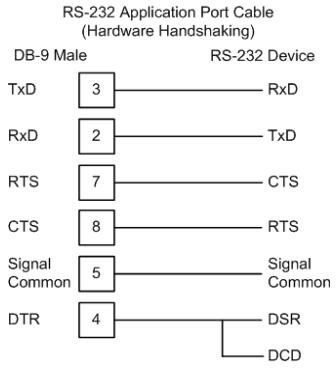
This type of connection is required between the module and a modem or other communication device.



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

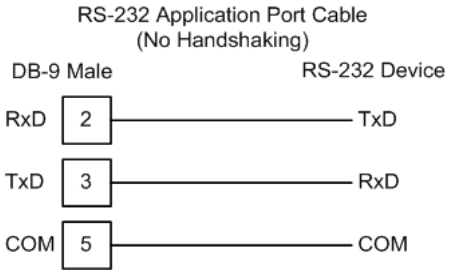
RS-232: Null Modem Connection (Hardware Handshaking)

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

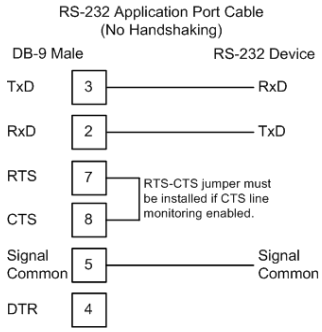


RS-232: Null Modem Connection (No Hardware Handshaking)

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

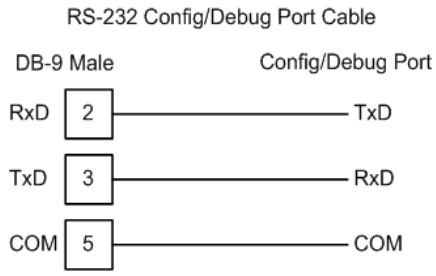


Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to **N** and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to **Y**, then a jumper is required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



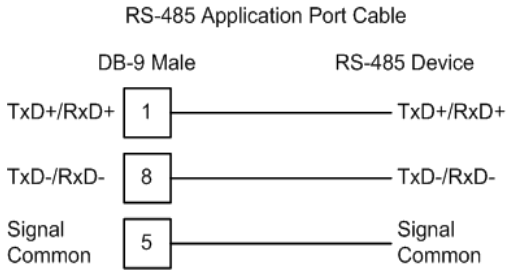
5.2.2 RS-232 Configuration/Debug Port

This port is physically an eight-pin, Mini-DIN8F connection. A Mini-DIN8M to DB9M 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. Here are the cable pinouts for RS-232 communication on this port.



5.2.3 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:

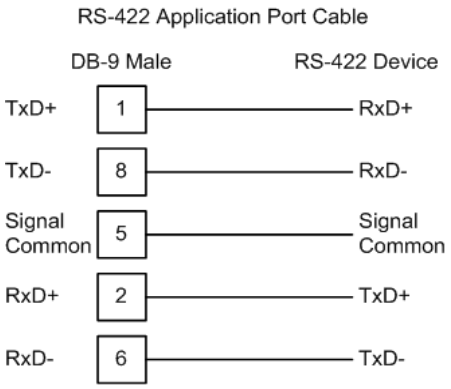


Note: This type of connection is commonly called a RS-485 half-duplex, 2-wire connection. If you have RS-485 4-wire, full-duplex devices, they can be connected to the gateway's serial ports by wiring together the TxD+ and RxD+ from the two pins of the full-duplex device to Pin 1 on the gateway and wiring together the TxD- and RxD- from the two pins of the full-duplex device to Pin 8 on the gateway. As an alternative, you could try setting the gateway to use the RS-422 interface and connect the full-duplex device according to the RS-422 wiring diagram. For additional assistance, please contact ProSoft Technical Support.

Note: Depending upon devices on the network, if there are problems in RS-485 communication that can be attributed to the signal echoes or reflections, then consider adding 120 OHM terminating resistors at both ends of the RS-485 line.

5.2.4 RS-422

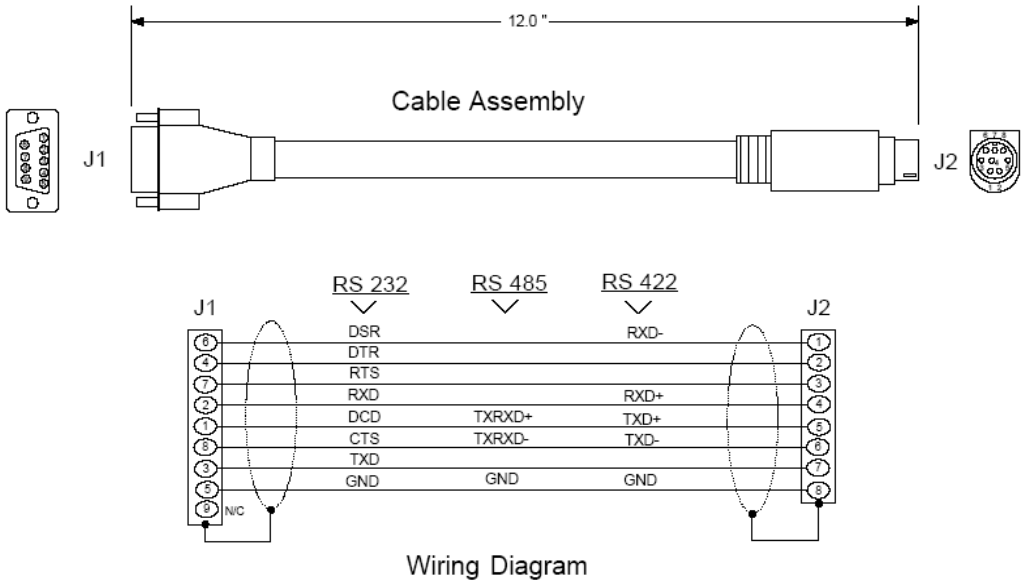
The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



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.

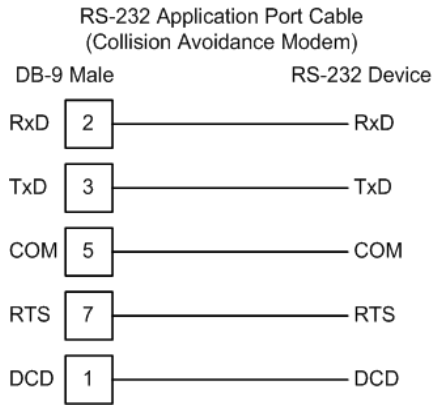
5.2.5 DB9 to Mini-DIN Adaptor (Cable 09)



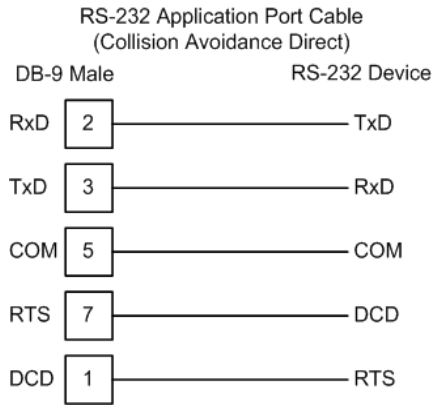
Wiring Diagram

5.3 Collision Avoidance

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.

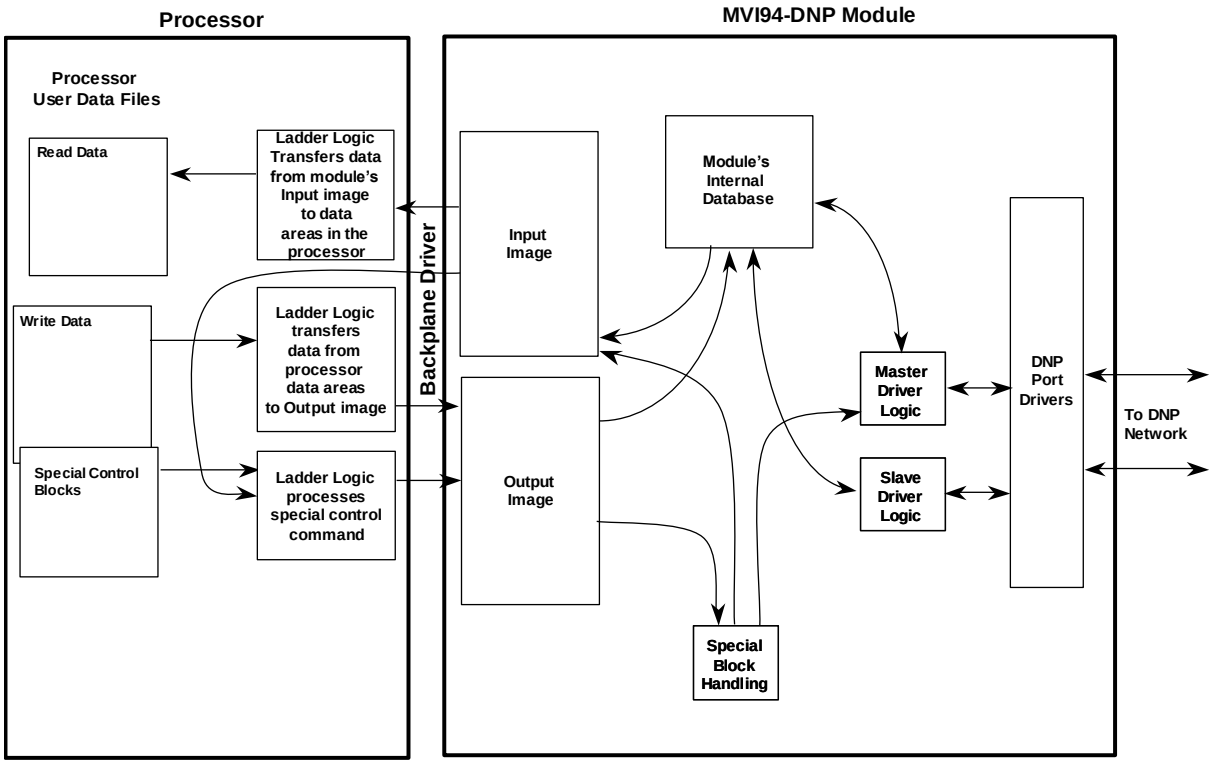


If collision avoidance is used in a point-to-point connection on the RS-232 interface, the following cable should be used.



5.4 Functional Overview

The MVI94-DNP communication module interfaces DNP slave or master devices with the Flex I/O system. The module contains a database defined by the user. This database is used for the request and command messages sent from the DNP master port to DNP slave devices. If the module is used as a virtual DNP slave device, the database is used to source read requests and is the destination of remote master write requests. The following illustration shows the data flow within the module and between the DNP network and backplane.



The database also interfaces with the Flex I/O system using the Flex I/O bus (backplane). Data is made available to the PLC or any processor on a ControlNet network using this backplane interface. Input and output image tables in the module present the data in the database to the backplane.

If the module is used as a DNP master device on the application port, it can continuously poll DNP slave devices. Up to 100 user-defined commands can be defined for the port. Data read from the slave devices are placed in the database. Any write requests for the slave devices are sourced with data from the database.

Commands can be activated in the module under processor control. When a command is activated, it is placed in the command queue for immediate execution. Normal command polling will begin after the command queue is completely processed.

If the module is used as a DNP slave device, it responds to requests from a remote DNP master device. All data in the module's database is available to the remote master device for read and write requests.

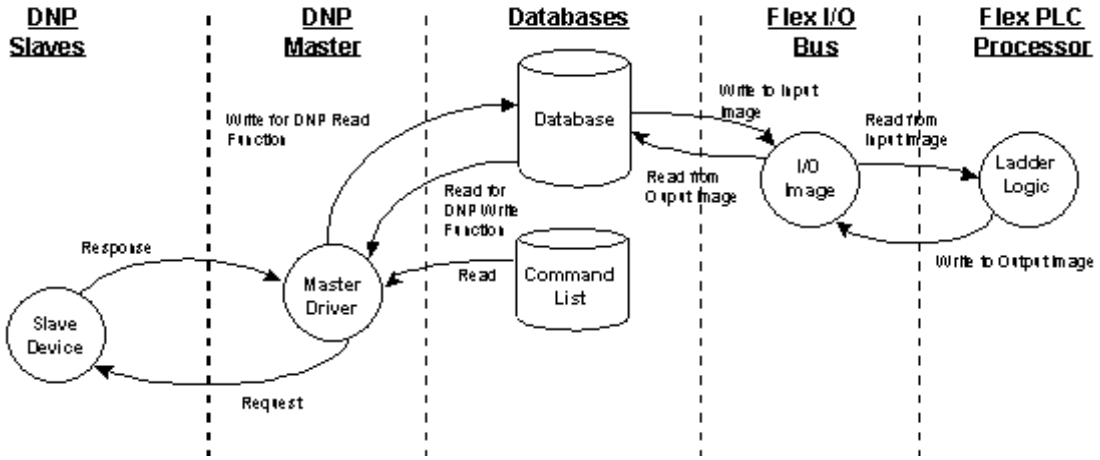
The module provides a Configuration/Debug port for use with an external computer executing a terminal emulation program. The terminal emulation program provided with the module permits uploading and downloading of the configuration information required by the module. Additionally, the Configuration/Debug port provides a view into the virtual database, communication statistics, and the configuration. Refer to Diagnostics and Troubleshooting for a full discussion of this port's functionality.

5.4.1 Database Information

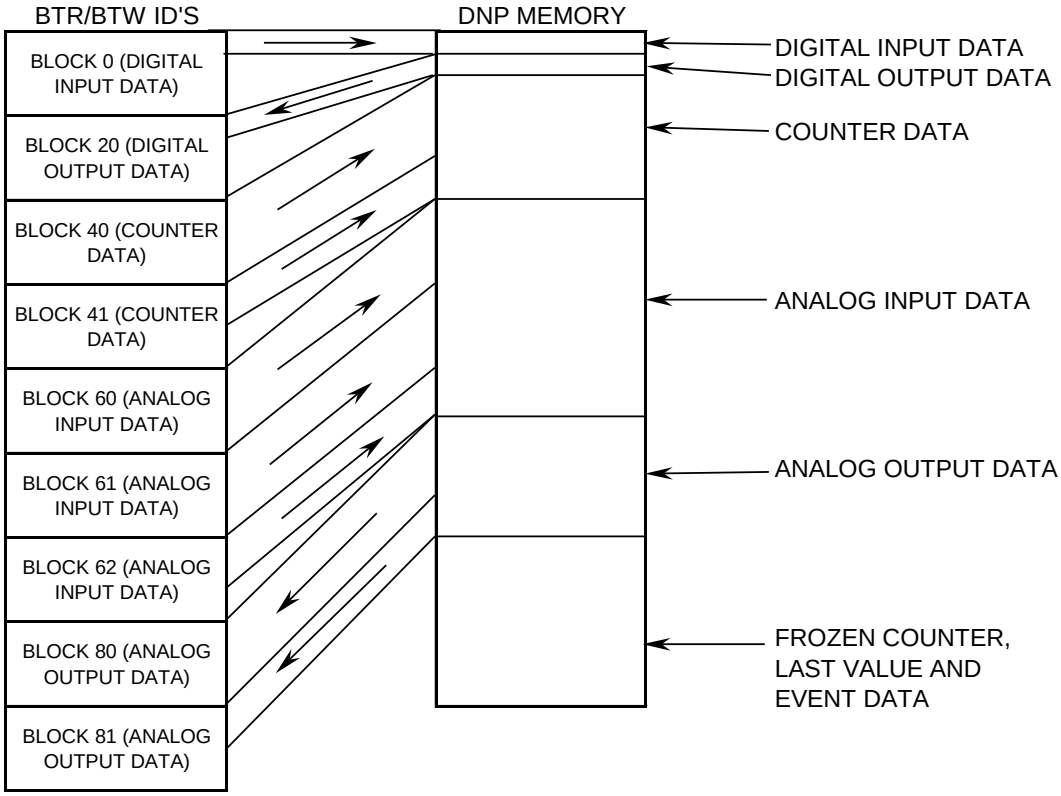
Central to the functionality of the module is the database. This database is used as the interface between remote DNP devices and the Flex I/O bus. The content and structure of the user data area of the database is completely user defined. The following illustration shows the general format of the module's database.

	DATA AREA		**BLOCKS**
DNP DATA	BINARY INPUTS	PLC DATA	0 to 9
	BINARY OUTPUTS	PLC DATA	20 to 29
	COUNTER DATA	PLC DATA	40 to 49
	ANALOG INPUTS	PLC DATA	60 to 79
	ANALOG OUTPUTS	PLC DATA	80 to 89
	FLOAT INPUTS	PLC DATA	300 to 319
	DOUBLE INPUTS	PLC DATA	320 to 349
	FLOAT OUTPUTS	PLC DATA	400 to 419
	DOUBLE OUTPUTS	PLC DATA	420 to 449
	FROZEN COUNTER DATA		
	BINARY INPUT EVENTS		
	ANALOG INPUT EVENTS		
LAST VALUE DATA	BINARY INPUTS		
	ANALOG INPUTS		
	DNP BINARY OUTPUTS		
	DNP ANALOG OUTPUTS		
	IED BINARY OUTPUTS		
	IED ANALOG OUTPUTS		
IED DATA	BINARY INPUTS		100 to 109
	BINARY OUTPUTS		120 to 129
	COUNTER DATA		140 to 149
	ANALOG INPUTS		160 to 179
	ANALOG OUTPUTS		180 to 189
RBE FLAGS	BINARY INPUT		
	ANALOG INPUT		

The Flex I/O bus reads data from and write data to the user data area using the backplane interface. The module interfaces data contained in remote DNP slave devices to the database using the DNP application port. User commands are issued out the master port from a command list. These commands gather or control data in the slave devices. The following illustration shows the relationships discussed above for a simulated master device:

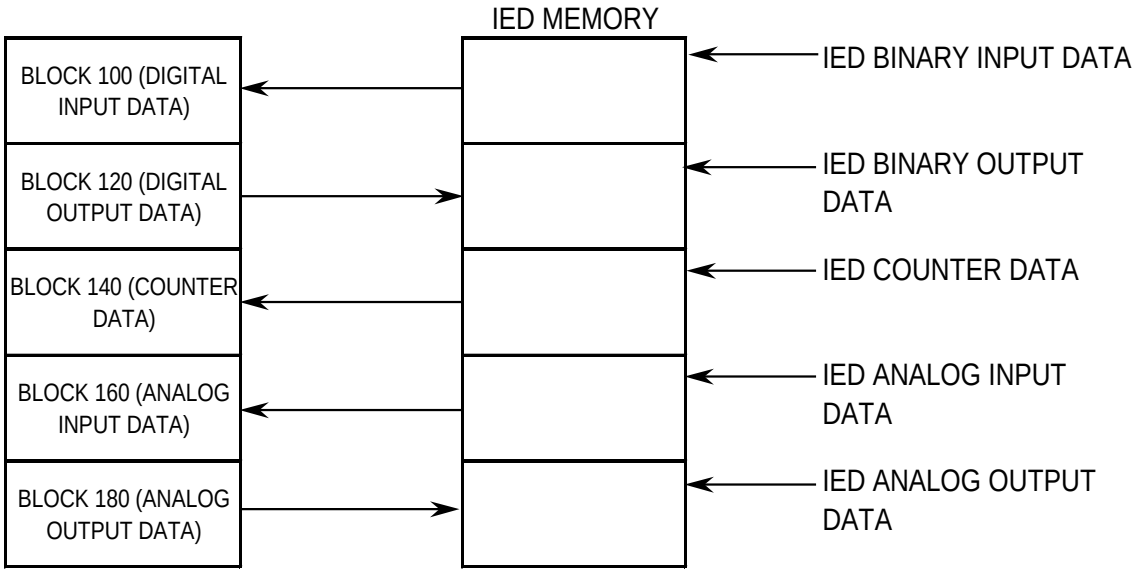


The database defined in the module depends on the DNP device type emulated. If the module emulates a DNP slave device, the database will be as shown in the following example:



Binary and analog input and counter data is transferred from the processor to the module. The remote DNP master device reads this data. Binary and analog output data is transferred from the remote DNP master device to the module. This data is then passed to the processor.

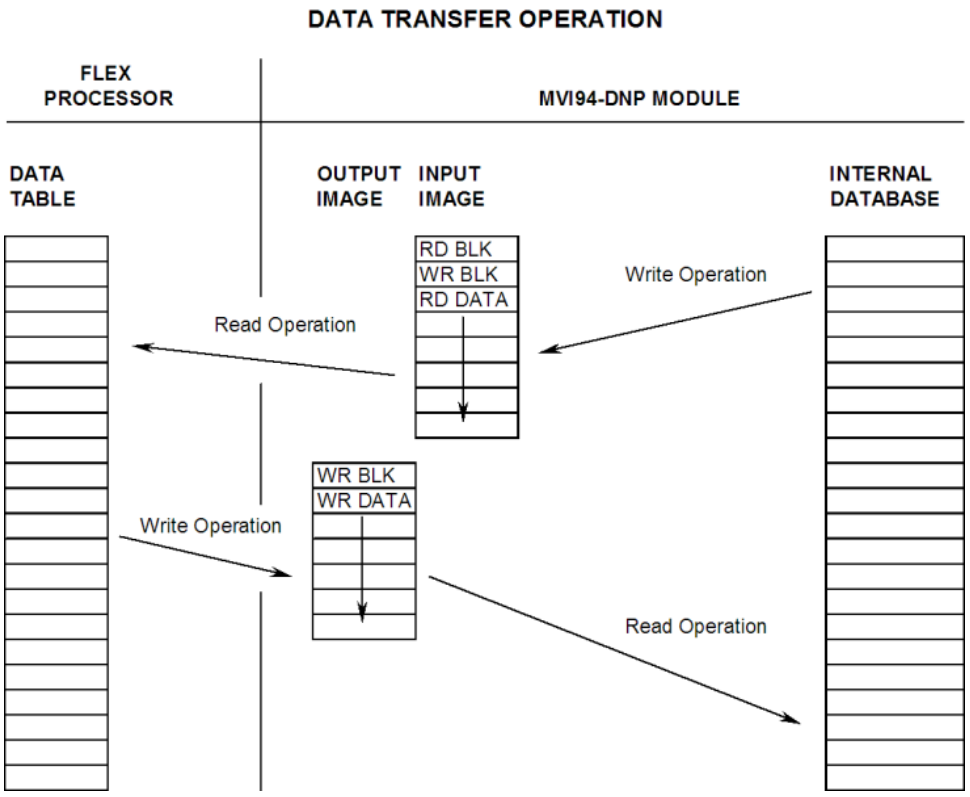
If the module emulates a DNP master device, the database will be as shown in the following example:



Binary and analog input and counter data is read by the DNP master port and placed in the internal database. This data is transferred to the processor. Binary and analog output data is received from the processor and placed in the database. This data is then used to control IED devices on the DNP network by the master command list.

5.4.2 Data Transfer

Data is transferred over the backplane using the module's input and output images. The module is configured with an eight-word input image and a seven-word output image. The module and the Flex processor use these images to page data and commands. The input image is set (written) by the module and is read by the Flex processor. The output image is set (written) by the Flex processor and read by the module. The following illustration shows this relationship:

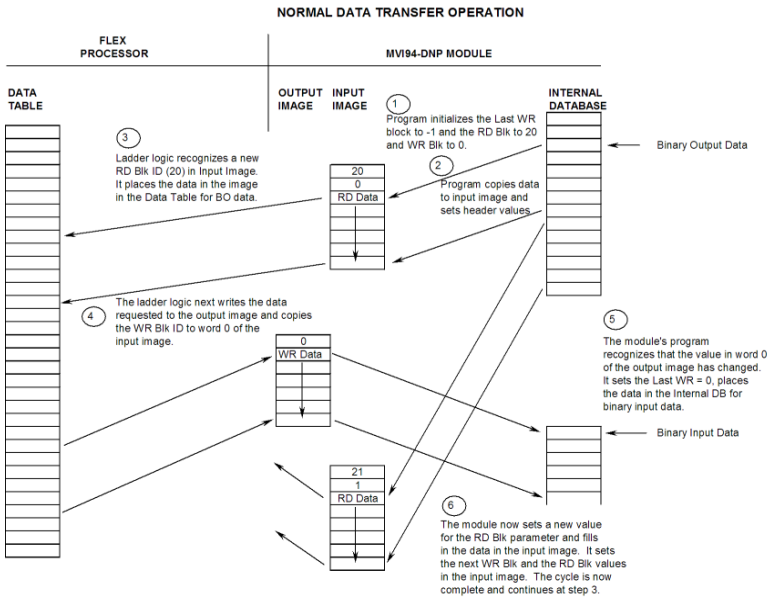


The module's program is responsible for setting the block identification code used to identify the data block written and the block identification code of the block it wants to read from the processor. User definition of the database to use with the module defines the blocks to be transferred between the module and the processor.

Each read and write operation transfers a six-word data area. The processor read operation contains a two-word header that defines the block identification code of the read data and the block identification code of the write block requested. These identification codes are in the range of 0 to 189, 300 to 349 and 400 to 449.

The module and the processor constantly monitor input and output images. How does either one know when a new block of data is available? Recognizing a change in the header information of the image (word 0) solves the problem. For example, when the module recognizes a different value in the first word of the output image, new processor write data is available. When the processor recognizes a new value in the first word of the input image, new processor read data is available. This technique requires the storage of the previously processed data block identification code.

The following illustration shows the normal sequence of events for data transfer:



1. The steps outlined in the diagram above are discussed below:
2. During program initialization, the write block identification code is set to 0 (binary input data) and read block identification codes are set to 20 (binary output data). The last block write variable is set to -1.
3. The program copies the first six-word block of the binary output database to the input image (words 2 to 7). It then sets the current write block code in word 1 of the input image. To "trigger" the write operation, the program places the current read block code into word 0 of the input image.
4. The Flex processor recognizes a new value in word 0 of the input image (based on the last_read_block_code not equal to read_block_code) in its ladder logic. The ladder logic computes the offset into the file based on the data block number (defines the data type and offset of block for data type).
5. The new data contained in the input image (words 2 to 7) is copied to the offset in the processor's user data file. The last_read_block_code storage register in the processor is updated with the new read_block_code.

Note: If the data area transferred from the module exceeds the size of a single user file in the Flex processor, logic will be required to handle multiple files.

6. Next the ladder logic examines the value of the write_block_code and determines the data type and offset into the data type data table where the data will be sourced.
7. The required six-word, write data is copied to the module's output image (words 1 to 6). To "trigger" the transfer operation, the ladder logic moves the write_block_code into word 0 of the output image.
8. The module's program recognizes the new write_block_code. It transfers the data to the correct offset in the database using the block number provided.
9. The module sets the last_write_block_code to the value of write_block_code.

10 The module now selects the next read and write blocks. The data for the processor read operation is placed in the input image and the write_block_code is set in word 1. The module "triggers" the transfer operation by setting the new read_block_code in word 0 of the input image.

11 The sequence continues at step 3.

The previous discussion is for normal data transfer operation. The following table lists the block identification codes used by the module for data transfer.

Data Type	Start Block #	Max Block #	Max # Of Points	Type
Digital Input	0	9	960	Slave
Digital Output	20	29	960	
Counters	40	49	30	
Analog Input	60	79	120	
Float Input	300	319	60	
Double Input	320	349	30	
Analog Output	80	89	60	
Float Output	400	419	60	
Double Output	420	449	30	
IED Digital Input	100	109	960	Master
IED Digital Output	120	129	960	
IED Counters	140	149	30	
IED Analog Input	160	179	120	
IED Analog Output	180	189	60	

Data is transferred between the processor and the module using the block identification codes of 0 to 189, 300 to 349 and 400 to 449. Other block codes control the module from the processor's ladder logic. They are implemented when the ladder logic needs to control the module. In order to use one of the blocks, the ladder logic inserts the data and code in the output image of the module. The data should be set before the code is placed in the block. This operation should be performed after the receipt of a new write block from the module.

The full list of block codes supported by the module are listed in the following table.

Block Number	Function/Description
-1 or -2	Dummy Blocks: Used by module when no data is to be transferred
0 to 89	DNP Data blocks
100 to 189	IED Data blocks
200 to 210	Error/Status List Blocks
300 to 349	DNP Data blocks for float and double input data
400 to 449	DNP Data blocks for float and double output data
9901	CROB Control Block for Digital Outputs
9902	Command Control Block (Add command to Command List Queue)
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
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)

Each of the special control blocks is discussed in the following topics.

CROB Control Block for Digital Outputs (Block 9901)

The CROB (Control Relay Output Block) controls a digital output on an IED unit connected to a master port under program control. This block provides more control than the binary output control offered in the command list. The selected point can use the pulse on/off feature of the DNP protocol using this block. Refer to the Basic 4 documentation for a full discussion of the CROB object and its functionality. The structure of the block sent from the processor to the module is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Slave Address	This is the IED node address for the slave to consider on the network.
2	Function	Function codes 3, 5 and 6 supported. Function code 4 is automatically sent after a successful function 3.
3	Address in Slave	Point in IED to consider with the CROB.
4	Control Code	This is a standard DNP protocol control code byte (see description below).
5	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.
6	Pulse On/Off Time	This parameter specifies the on-time and off-time intervals for pulse control. The same time value is used for both parameters in the constructed CROB message.

The Control Code field of the block defines the method of output control. This bit-mapped value has 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 and 4=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.

When the module receives this block, it will place the request in the command queue as long as there is room in the queue for the command. This will cause the command to be executed at a high priority. The module does not send a response message back to the processor for this block.

Command Control Block (Block 9902)

The Command Control Block places commands in the command list in the command queue. Commands in the command list need not have their enable bit set and can be executed using this block. The structure of the block built by the processor for the module is shown in the following table.

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 5.
2 to 6	Command Numbers to enable	These 5 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.

When the module receives the block, it places the commands represented in the block in the command queue until it is full. These commands will execute at a high priority. The module does not send a response block back to the processor for this command block.

Slave IED Unit Errors (Block 9949)

This command block is used by the processor to request slave status information for a specific IED unit. The format of the block constructed by the 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.
2	Slave Index	This parameter sets the index in the slave array to consider. 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 6	Spare	Not Used

When the module receives this block, it searches the slave list for the slave index contained in the block. If the index is valid for the module, it constructs a response block containing the status data for the indicated IED unit. The format of the response block 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 PLC.
1	Block ID	This is the next block requested by the module.
2	Bad CRC	This value represents the number of bad CRC values received from the slave device.
3	Buff Ovrflw	This value represents the number of buffer overflow messages received from the slave device.
4	Tran Seq#	This value represents the number of incorrect transport layer sequence number errors.
5	Conf Retry	This value represents the number of data link layer confirm request retries.
6	Conf Fail	This value represents the number of data link layer confirm request failures.
7	No App Rsp	This value represents the number of application layer no responses to requests.

Ladder logic must handle this response block and to place the data in a user data file.

Command List Error Data

This block is used by the processor to request the command list error data from the module. The format of the block constructed by the processor is shown in the following table.

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 4.
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 6	Spare	Not Used

When the module receives this block, it searches the command list to validate the request. If the request is validated, the module constructs a response block containing the last error code for each command requested. The format of the response block is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the PLC.
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 0 to 4.
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 7	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.

PLC Binary Input Event Data (Block 9958)

This block is used by the processor to send binary input event data from the processor to the module. This event is generated by the processor and not the module and only has validity when a slave device is emulated on the application port. When the processor recognizes a binary input event, it constructs a request block with the following format:

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	Sequence Counter	This field holds the sequence counter for each 9958 block transfer. This synchronizes and confirms receipt of the block by the module.
2	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
3	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.
4	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
5	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
6	Year	This is the four digit year for the event. Bits 12 and 13 can contain the class override values of 1 to 3.

When the processor receives the block, it places the information in the module's binary input event queue. This information is then made available to the remote DNP master unit on binary event object or class poll requests. The module informs the processor that the event has been processed by generating a response block with the following format:

Word Offset in Block	Data Field(s)	Description
0	Block ID	Identification code for block set to 9958.
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 7	Spare	Not used

The Sequence Counter and the Event Count fields in the response block can be utilized to insure that the module has processed the event. If the event buffer in the module is full, the event is not processed and the processor should resend the event to the module. If the module processes the event, the processor can remove the event from its buffer.

PLC Analog Input Event Data (Block 9959)

This block is used by the processor to send analog input event data from the processor to the module. This event is generated by the processor; not the module, and only has validity when a slave device is emulated on the application port. When the processor recognizes an analog input event, it constructs a request block with the following format:

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	DNP Analog Input Data point	This is the data point in the DNP analog input database represented by the event.
2	Analog Input Value	This is the new analog input value represented in the event.
3	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
4	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
5	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
6	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.

When the processor receives the block, it places the information in the module's analog input event queue. This information is then made available to the remote DNP master unit on analog event object or class poll requests. The module informs the processor that the event has been processed by generating a response block with the following format:

Word Offset in Block	Data Field(s)	Description
0	Block ID	Identification code for block set to 9959.
1	Block ID	Block identification code for request from PLC by the module.

Word Offset in Block	Data Field(s)	Description
2	Event Count	This field contains the number of events processed by the module.
3	DNP Analog Input Data point	This field contains the analog input point of the last successful block 9959 received.
4 to 7	Spare	Not used

The Sequence Counter and the Event Count fields in the response block can be utilized to insure that the module has processed the event. If the event buffer in the module is full, the event is not processed and the processor should resend the event to the module. If the module processes the event, the processor can remove the event from its buffer.

Set Processor Time Block (9970)

This block is constructed by the processor to request the module's time. The following table describes the format of this block.

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 6	Not Used	Not Used

When the module receives this block, it immediately constructs a response block containing the modules current time. The following table describes the format of this block.

Word Offset in Block	Data Field(s)	Description
0	Block Read ID	This field contains the block identification code of 9970 for the block.
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.

Set Module Time Block (9971)

This block is constructed by the processor to send its time to the module. The following table describes the format of this block.

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.

When the module receives the block, it immediately sets its time using the information contained in the block.

Warm Boot Block (9998)

This block does not contain any data. When the processor places a value of 9998 in word 0 of the output image, the module performs a warm-start. This involves clearing the configuration and all program status data. Finally, the program loads the configuration information from the flash disk and begin running. There is no positive response to this message other than the status data being set to zero and the block polling starting over. The format of this block is shown in the following table.

Word	Description	Length
0	9998	1
1 to 6	Not used	6

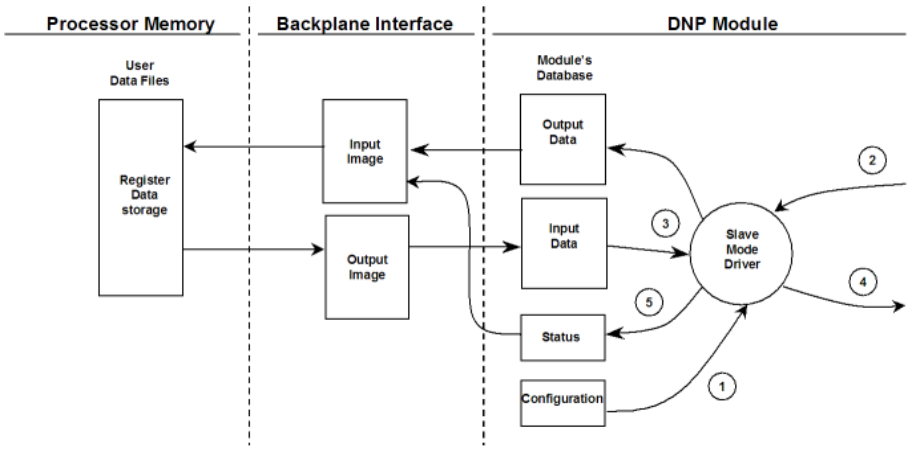
Cold Boot Block (9999)

This block does not contain any data. When the processor places a value of 9999 in word 0 of the output image, the module performs a hardware restart. This causes the module to reboot and reload the program. There is no positive response to this message other than the status data being set to zero and the block polling starting over. The format of this block is shown in the following table.

Word	Description	Length
0	9999	1
1 to 6	Not used	6

5.4.3 Slave Driver

The Slave Driver Mode allows the MVI94-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 Flash data area in the module. This information configures the serial port and define the slave node characteristics.
2	A Host device issues read or write commands 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 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.

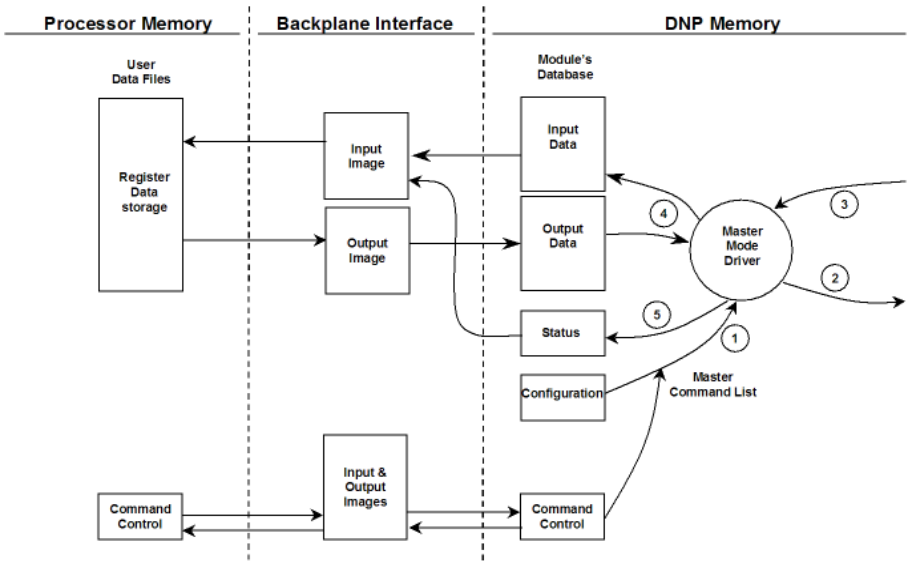
The slave driver supports object 110 (octet string data). Four points are pre-assigned values as defined in the following table.

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.

5.4.4 Master Driver

In the Master mode, the MVI94-DNP module is responsible for issuing read or write commands to slave devices on the DNP network. These commands are user-configured in the module via the Master Command List read from the module's Flash. Command status is returned to the processor for each individual command in the command list status block. The location of this status block in the module's internal database is user-defined. 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 Flash area of the module. The configuration data obtained includes the number of commands and the Master Command List. 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 the module's internal database 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 internal database, assuming a read command.
5	Status is returned to the processor for each command in the Master Command List.

Care must be taken in constructing each command in the list for predictable operation of the module. If two commands write to the same internal database address of the module, the results will not be as desired. All commands containing invalid data are ignored by the module.

5.4.5 IIN Bits

MVI94-DNP Module Internal Indication Bits (IIN Bits) for DNP Slave Port

The internal indication bits are stored in a word that follows the function code in all response messages. These bits report status and error information to the master DNP device. Below is a description of the 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.

5.5 MVI94-DNP Configuration Forms

This section contains configuration forms that will aid in the configuration of the module. If you design your system before trying to directly implement it, you will have a greater chance of success. Fill in the configuration forms for your application, and then edit the configuration text file. This section contains forms for both master and slave configuration:

5.5.1 Slave Configuration

Variable Name	Data Range	Description	IF Error	Config. Value
[Module]		This section header indicates the start of the module specific parameter set.		
Module Name:	80 Characters	This parameter sets the name or description of the module's application	"Not Configured "	
Type:	M=Master or S=Slave	This parameter must be set to M for the DNP port to be used as a master port. If the parameter is set to any other value, the port will be used as a slave port.	S	Slave

Variable Name	Data Range	Description	IF Error	Config. Value
[DNP Slave]		This section header defines the start of the DNP slave parameter set.		
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.	65534	
Baud Rate:	Baud Rate from Table	Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400), 576 (57600), 115 (115200)	9600	
RTS On:	0 to 65535	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.	NA	
RTS Off:	0 to 65535	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.	NA	
Min Response Delay:	0 to 65535	Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.	NA	
Modem:	Y or N	This parameter defines if a dial-up modem is used on the DNP slave port. If the value is set to N, no modem is used. If the parameter is set to Y, a modem is used.	N	
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.	NA	

Variable Name	Data Range	Description	IF Error	Config. Value
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.	NA	
Redial Delay Time:	0 to 65535	Defines the minimum number of milliseconds to wait before a redial attempt is made by the slave.	NA	
Redial Random Delay:	0 to 65535	Defines a random millisecond time range to be added to the redial delay time before the modem is accessed.	NA	
Idle Timeout:	0 to 65535	Defines the number of milliseconds the modem is inactive before it will disconnect.	NA	
Phone Number:	ASCII String Data	These fields 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.	NA	
Collision Avoidance:	Y or N	This parameter defines if the collision avoidance functionality is to be applied to the port. If the parameter is set to N, collision avoidance is not used. It will be used if set to Y. If collision avoidance is used, it requires a special cable.	N	
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.	32000	
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.	32000	
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.	NA	
BI Class:	0 to 3	This parameter sets the default class assignment for points in the binary input dataset. All points not specified in the override section of the configuration will be assigned to this class. For class 0, no events will be generated for the points.	0	
AI Class:	0 to 3	This parameter sets the default class assignment for points in the analog input dataset. All points not specified in the override section of the configuration will be assigned to this class. For class 0, no events will be generated for the points.	0	
Float Class:	0 to 3	This parameter sets the default class assignment for points in the float input dataset. All points not specified in the override section of the configuration will be assigned to this class. For class 0, no events will be generated for the points.	0	

Variable Name	Data Range	Description	IF Error	Config. Value
Double Class:	0 to 3	This parameter sets the default class assignment for points in the double input dataset. All points not specified in the override section of the configuration will be assigned to this class. For class 0, no events will be generated for the points.	0	
AI Deadband:	0 to 32767	This value sets the global deadband for all analog input points not specified in the override section. Events will be generated when the last saved value exceeds the limit of this parameter.	32767	
Float Deadband:	Any valid floating-point value	This value sets the global deadband for all float input points not specified in the override section. Events will be generated when the last saved value exceeds the limit of this parameter.	0	
Double Deadband:	Any valid double floating-point value	This value sets the global deadband for all double input points not specified in the override section. Events will be generated when the last saved value exceeds the limit of this parameter.	0	
Select/Operate Arm Time:	1 to 65535	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.	2000	
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.	1440	
Data Link Confirm Mode:	N, S or A	IED can request acknowledgement from master station when sending data. The codes are as follows: N=Never, S=Sometimes, A=Always	N	
Data Link Confirm Tout:	1 to 65535	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.	1	
Data Link Max Retry:	0 to 255	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.	255	
App Layer Confirm Tout:	1 to 65535	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.	2000	

Variable Name	Data Range	Description	IF Error	Config. Value
Unsolicited Response:	Y or N	Set if the slave unit will send unsolicited response messages. If set to N, the slave will not send unsolicited responses. If set to Y, the slave will send unsolicited responses. This feature requires collision avoidance on a multi-drop network.	N	
Class 1 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 1 required before an unsolicited response will be generated (not implemented).	10	
Class 2 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 2 required before an unsolicited response will be generated (binary input events).	25	
Class 3 Unsol Resp Min:	1 to 255 events	Minimum number of events in Class 3 required before an unsolicited response will be generated (analog input events).	25	
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.	NA	
Uresp Master Address:	0 to 65534	DNP destination address where unsolicited response messages are sent.	65534	
AI Events with time:	Y or N	This parameter determines if the analog input events are to be returned with a time value. If the parameter is set to N, no time will be returned with the event. If the parameter is set to Y, a time will be returned.	N	
Time Sync Before Events:	Y or N	This parameter defines if the remote DNP master must synchronize the time before events are generated by the slave. If the parameter is set to N, this synchronization is not required. If set to Y, the synchronization is required.	N	
Initialize DNP BO/AO:	Y or N	This parameter defines if the binary and analog outputs are to be read from the processor before the program starts its normal data transfer. If the parameter is set to N, this feature is not implemented. If the parameter is set to Y, the feature is used and special ladder logic must be implemented to transfer the data from the processor to the module.	N	

Variable Name	Data Range	Description	IF Error	Config. Value
[DNP Slave Database]		This section header defines the start of the DNP slave database definition parameter set.		
Binary Inputs:	0 TO 960	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.	0	

Variable Name	Data Range	Description	IF Error	Config. Value
Analog Inputs:	0 TO 120	Number of analog input points to configure in the DNP slave device. Each point will occupy a one word area in the module memory.	0	
Float Inputs:	0 TO 60	Number of float input points to configure in the DNP slave device. Each point will occupy a double-word area in the module memory.	0	
Double Inputs:	0 TO 30	Number of double input points to configure in the DNP slave device. Each point will occupy a four-word area in the module memory.	0	
Counters:	0 TO 30	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.	0	
Binary Outputs:	0 TO 960	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.	0	
Analog Outputs:	0 TO 60	Number of analog output points to configure in the DNP slave device. Each point will occupy a one word area in the module memory.	0	
Float Outputs:	0 TO 60	Number of float output points to configure in the DNP slave device. Each point will occupy a double-word area in the module memory.	0	
Double Outputs:	0 TO 30	Number of double output points to configure in the DNP slave device. Each point will occupy a four-word area in the module memory.	0	

Variable Name	Description
[DNP Slave Binary Inputs]	This section is used to override the default values of the class assignment for the binary input points.

oint# Class	
START	This string signals the start of the point definition list.
1	
2	
3	
END	This string signals the end of the point definition list.

Variable Name	Description

[DNP Slave Analog Inputs]	This section is used to override the default values of the class and deadband assignments for the analog input points.

Point# Class Deadband
START This string signals the start of the point definition list.
1 2000 #points 0-5=class 1, deadband = 1000
1 2000
2 1000
END This string signals the end of the point definition list.

Variable Name	Description
[DNP Slave Float Inputs]	This section is used to override the default values of the class and deadband assignments for the float input points.

Point# Class Deadband
START This string signals the start of the point definition list.
1 100.
2 12.34
3 1.23
END This string signals the end of the point definition list.

Variable Name	Description
[DNP Slave Double Inputs]	This section is used to override the default values of the class and deadband assignments for the double input points.

Point# Class Deadband
START This string signals the start of the point definition list.
1 100.
2 987.34
3 789.23
END This string signals the end of the point definition list.

5.5.2 Master Configuration

Variable Name	Data Range	Description	IF Error	Config. Value
[Module]		This section header indicates the start of the module specific parameter set.		
Module Name:	80 Characters	This parameter sets the name or description of the module's application	"Not Configured"	
Type:	M=Master or S=Slave	This parameter must be set to M for the DNP port to be used as a master port. If the parameter is set to any other value, the port will be used as a slave port.	S	Master

Variable Name	Data Range	Description	IF Error	Config. Value
[DNP Master]		This section header defines the start of the DNP master parameter set.		
Internal ID:	0 to 65534	This is the DNP master address for the unit represented on the port.	65534	
Baud Rate:	Baud rate value	Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400), 576 (57600), 115 (115200)	9600	
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.	NA	
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.	NA	
Min Response Delay:	0 to 65535 milliseconds	Minimum time between receiving a response and transmitting a request. Allows remote device time to disable transmitter on an RS-485 network.	NA	
Collision Avoidance:	Y or N	This parameter defines if the collision avoidance functionality is to be applied to the port. If the parameter is set to N, collision avoidance is not used. It will be used if set to Y. If collision avoidance is used, it requires a special cable.	N	
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.	32000	
CD Time Before Receive:	0 to 65535 milliseconds	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.	NA	

Variable Name	Data Range	Description	IF Error	Config. Value
Initialize IED Database:	Y or N	This parameter defines if the BI/AI/C data will be transferred from the processor to the module before normal data transfer occurs. Set the parameter to N to not use the feature. Set the parameter Y to use the feature. The use of this feature requires ladder logic to transfer the data.	N	

Variable Name	Data Range	Description	IF Error	Config. Value
[IED Database]		This section header defines the start of the DNP master parameter set.		
Binary Inputs:	0 to 960	Number of binary input points contained in the IED database to transfer to the PLC and obtained from the attached IED units..	0	
Binary Outputs:	0 to 960	Number of binary output points contained in the IED database which are transferred from the PLC and used by the attached IED units..	0	
Counters:	0 to 30	Number of counter points contained in the IED database to transfer to the PLC and obtained from the attached IED units..	0	
Analog Inputs:	0 to 120	Number of analog input points contained in the IED database to transfer to the PLC and obtained from the attached IED units..	0	
Analog Outputs:	0 to 60	Number of analog output points contained in the IED database which are transferred from the PLC and used by the attached IED units..	0	

Variable Name	Data Range	Description
[DNP Master Slave List]		This section header defines the location of the DNP master slave definition list.

```
)L Conf Mode ==> 0=Never, 1=Sometimes and 2=Always (select 0).
'lag parameter is bit coded as follows:
3it 0 (decimal 1) ==> Enable the slave
3it 1 (decimal 2) ==> Use Unsolicited messaging with this slave
3it 2 (decimal 4) ==> Use delay measurement with this slave
3it 3 (decimal 8) ==> Auto time synchronization enabled
```

START	This string signals the start of the slave definition list.
# Node DL Conf Conf Conf App Rsp	
# Address Mode Timeout Retry Timeout Flags	
2 0 1000 0 5000 9	
END	This string signals the end of the slave definition list.

Variable Name	Data Range	Description
[DNP Master Commands]		This section header defines the location of the DNP master command list.
START		This string signals the start of the command list.

```
. 2 3 4 5 6 7 8 9 10
.ags/ Node Data Data Cmd Device Point DNP DB IED DB Poll
ıable Address Object Variation Func Address Count Address Address
ıterval
2 1 0 1 0 -20 -1 0 0
```

END		This string signals the end of the command list.

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	

Column	Variable Name	Data Range	Description	IF Error	Config. Value
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	

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										
41										
42										
43										
44										
45										
46										
47										
48										
49										

5.6 DNP Master Subset Definition

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 does not use this object type and will not generate a message or process this type
	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

OBJECT			REQUEST		RESPONSE			
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**	**Data Size (bits)**	**NOTES**
	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
	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	

OBJECT			REQUEST		RESPONSE			
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**	**Data Size (bits)**	**NOTES**
	8	16-Bit Delta Counter Change Event With Time					72	
23	0	Frozen Counter Event - All Variations						
	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
	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
	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	
32	0	Analog Change Event - All Variations	1	06, 07, 08			24	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 Analog Change Event Without Time	1	06, 07, 08		17, 28	40	Master will generate and process this variation
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08		17, 28	24	Master will generate and process this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08		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 does not use this object type and will not generate a message or process this type
	1	32-Bit Analog Output Status					40	
	2	16-Bit Analog Output Status			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

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	Master will not process this variation
	2	Time Delay Fine			129	07, With Quant=1	16	Master will not process this variation
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	

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

5.7 DNP Slave Subset Definition

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

OBJECT			REQUEST	RESPONSE				
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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
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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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
	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 only be 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 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 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
	2	16-Bit Frozen Analog Event Without Time					24	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
	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 but data only 16-bit 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 but data only 16-bit
	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

OBJECT			REQUEST	RESPONSE				
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
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
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				Slave will respond to this variation (No class 1 data defined in application)
	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

OBJECT			REQUEST	RESPONSE				
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**	**Data Size (bits)**	**NOTES**
	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	
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

5.8 Device Profile

DNP V3.00 **DEVICE PROFILE DOCUMENT**	
Vendor Name: ProSoft Technology, Inc.	
Device Name: MVI94-DNP (VERSION 2.00)	
Highest DNP Level Supported : For Request: L2 For Responses: L2	Device Function: Slave or Master
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 and Configuration Error - User specified point or event count is too high for application (can only correct by changing configuration in module). Slave port on the module may be attached to a dial-up modem. 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. Slave module will not generate events until Restart IIN bit is cleared by remote DNP master.	
Maximum Data Link Frame Size (octets): Transmitted : 292 Received : 292	Maximum Application Fragment Size (octets): Transmitted : 2048 Received : 2048
Maximum Data Link Re-tries: Configurable from 0 - 255	Maximum Application Layer Re-tries: None
Requires Data Link Layer Confirmation: Configurable at module start-up (never, sometimes, & always)	
Requires Application Layer Confirmation: When reporting Event Data as a slave unit	

Time-outs while waiting for:

Data Link Confirm	: Configurable at module start-up (1 to 65535 mSec)
Complete Application Fragment	: Configurable at module start-up
Application Confirm	: Configurable at module start-up (1 to 65535 mSec)
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 65535)
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

6 Support, Service & Warranty

6.1 Contacting Technical Support

ProSoft Technology, Inc. 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 interfaced serial, Ethernet or Fieldbus devices

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

For additional ProSoft Technology contacts in your area, please visit:
www.prosoft-technology.com/About-Us/Contact-Us.

6.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS, please see the documents at:
www.prosoft-technology/legal