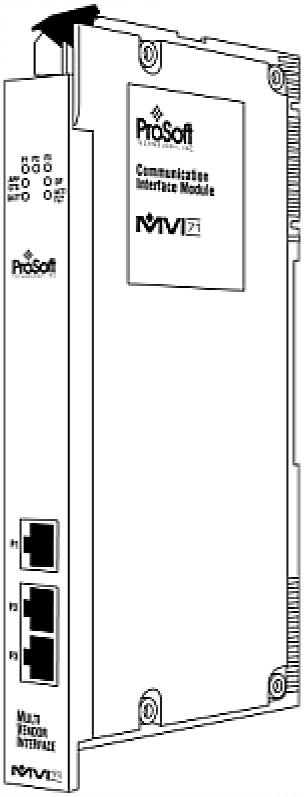
inRAx



MVI71-101S

PLC Platform

IEC 60870-5-101 Slave Communication Module

User Manual

February 05, 2008

ProSoft TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation PLC hardware, the MVI71-101S Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

This manual is provided to assist the user. Every attempt has been made to ensure that the information provided is accurate and a true reflection of the product's installation requirements. In order to ensure a complete understanding of the operation of the product, the user should read all applicable Rockwell Automation documentation on the operation of the Rockwell Automation hardware.

Under no conditions will ProSoft Technology be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

Reproduction of the contents of this manual, in whole or in part, without written permission from ProSoft Technology is prohibited.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology Improvements and/or changes in this manual or the product may be made at any time. These changes will be made periodically to correct technical inaccuracies or typographical errors.

Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

Power, Input, and Output (I/O) wiring must be in accordance with Class 1, 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.

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class 1, Division 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.

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 the product, documentation or support, please write or call us.

ProSoft Technology
1675 Chester Avenue, Fourth Floor
Bakersfield, CA 93301
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
http://www.prosoft-technology.com

MVI71-101S User Manual
February 05, 2008
PSFT.101S.MVI71.UM.08.02.05

Contents

Guide to the MVI71-101S User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 19, page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 55) Diagnostics and Troubleshooting (page 55)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 71) Functional Overview (page 73) Product Specifications	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 113)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI71-101S module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI71-101S Module and the application in which they will be used.

Caution: It is important that those responsible for implementation can complete the application without exposing personnel, or equipment, to unsafe or inappropriate working conditions. Safety, quality and experience are key factors in a successful installation.

1.1 System Requirements

The MVI71-101S module requires the following minimum hardware and software components:

- Rockwell Automation PLC processor, with compatible power supply and one free slot in the rack, for the MVI71-101S module. The module requires 800mA of available power.
- The PLC Processor must provide for at least 64 words of BTR/BTW area, otherwise the module may not function correctly.
- Rockwell Automation RSLogix 5 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® 100 MHz minimum. Pentium III 700 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP
  - Microsoft Windows 2000
  - Microsoft Windows NT v4.0 with Service Pack 3 or greater
  - Microsoft Windows ME
  - Microsoft Windows 98
- 64 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)

- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- CD-ROM drive
- 3.5 inch floppy disk drive
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

1.2 Package Contents

The following components are included with your MVI71-101S module, and are all required for installation and configuration.

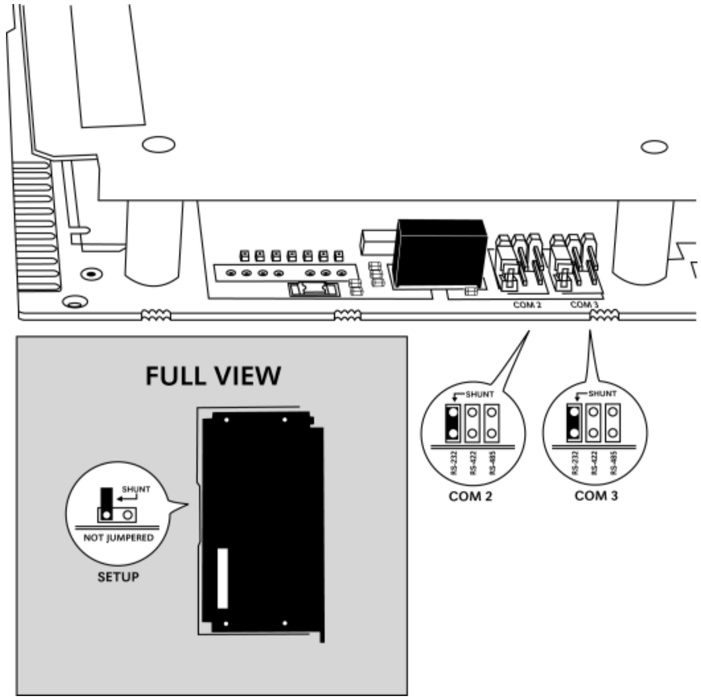
> Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI71-101S Module	MVI71-101S	IEC 60870-5-101 Slave Communication Module
1	Cable	RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI71-101S module.

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

1.3 Setting Jumpers

The following illustration shows the jumper configurations for the various RS interfaces. If you are using an interface other than RS-232 (default), you must change the jumpers as shown:



The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

1.4 Install the Module in the Rack

If you have not already installed and configured your PLC processor and power supply, please do so before installing the MVI71-101S module. Refer to your Rockwell Automation product documentation for installation instructions.

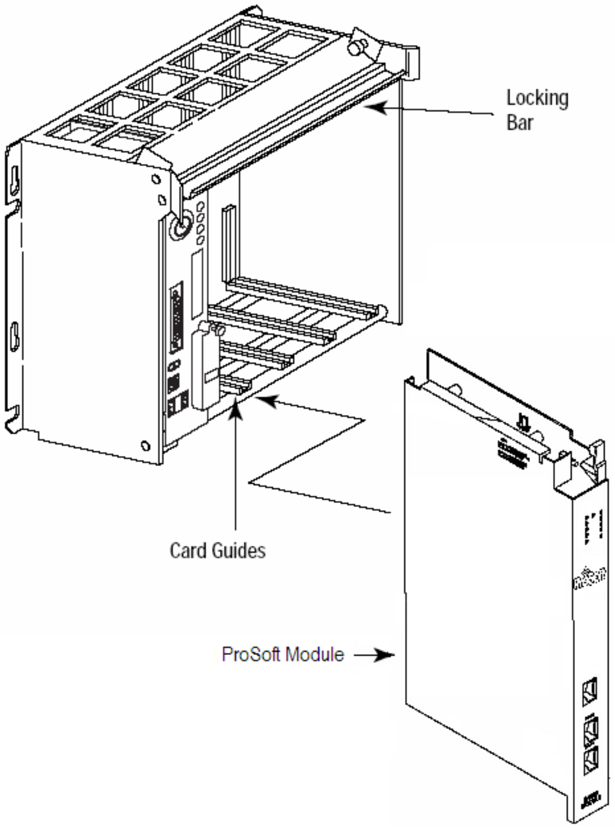
Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify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

After you have checked the placement of the jumpers, insert MVI71-101S into the PLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install PLC modules.

Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1 Turn power OFF.

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



3 With a firm but steady push, snap the module into place.

4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.

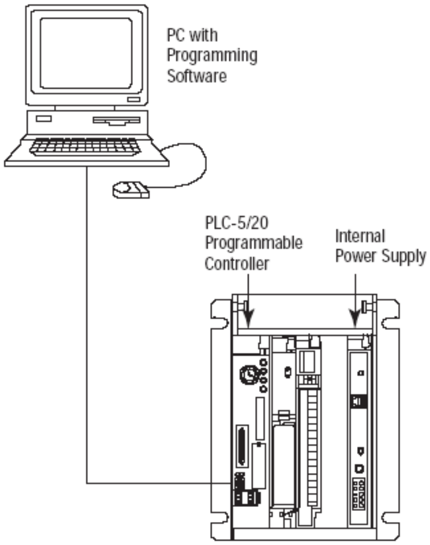
5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample program to work correctly. Slot numbers are identified on the green circuit board (backplane) of the PLC rack.

6 Turn power ON.

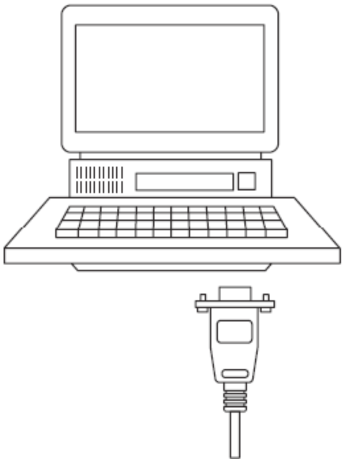
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.5 Connect your PC to the Processor

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



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

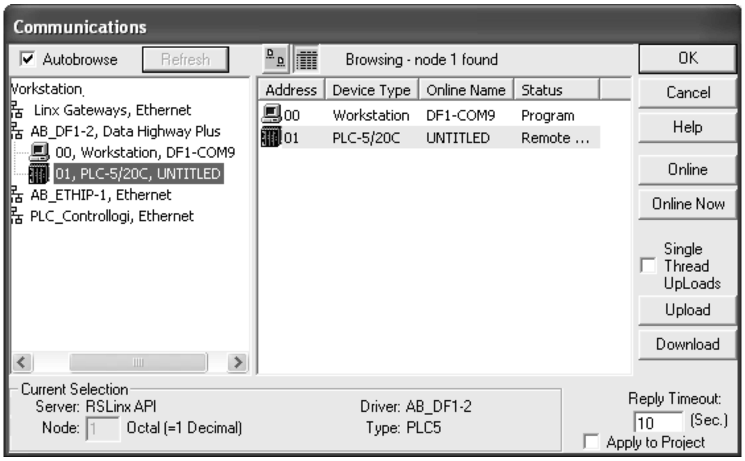


1.6 Download the Sample Program to the Processor

To download the sample program from RSLogix 5 to the PLC processor:

> Note: The key switch on the front of the PLC processor must be in the REM position.

1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



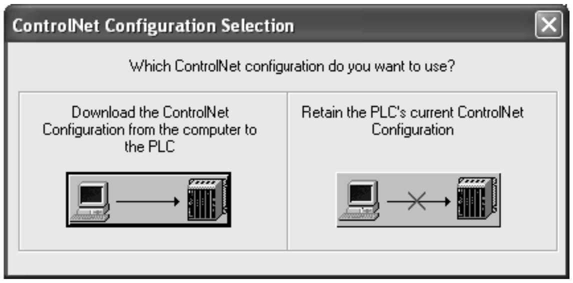
Vorkstation
- Linx Gateways, Ethernet
- AB_DF1-2, Data Highway Plus
  - 00, Workstation, DF1-COM9
  - 01, PLC-5/20C, UNTITLED
- AB_ETHIP-1, Ethernet
- PLC_Controllogi, Ethernet

Address	Device Type	Online Name	Status
00	Workstation	DF1-COM9	Program
01	PLC-5/20C	UNTITLED	Remote ...

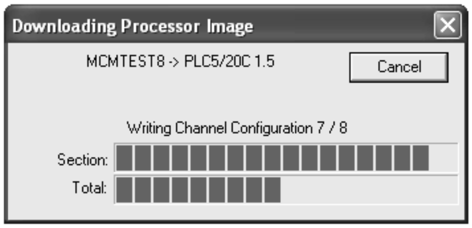
Current Selection
Server: RSLinx API
Node: 1 Octal (=1 Decimal)
Driver: AB_DF1-2
Type: PLC5
Reply Timeout: 10 (Sec.)

2 Click the Download button to transfer the sample program to the processor.

3 When prompted, choose Computer to PLC


ControlNet Configuration Selection
Which ControlNet configuration do you want to use?
Download the ControlNet Configuration from the computer to the PLC
Retain the PLC's current ControlNet Configuration

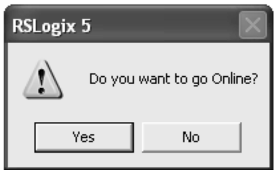
4 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



MCMTEST8 -> PLC5/20C 1.5

Writing Channel Configuration 7 / 8

5 When the download is complete, RSLogix will open another confirmation dialog box. Click OK to switch the processor from Program mode to Run mode.



RSLogix 5

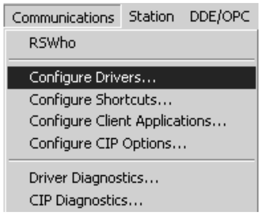
Do you want to go Online?

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

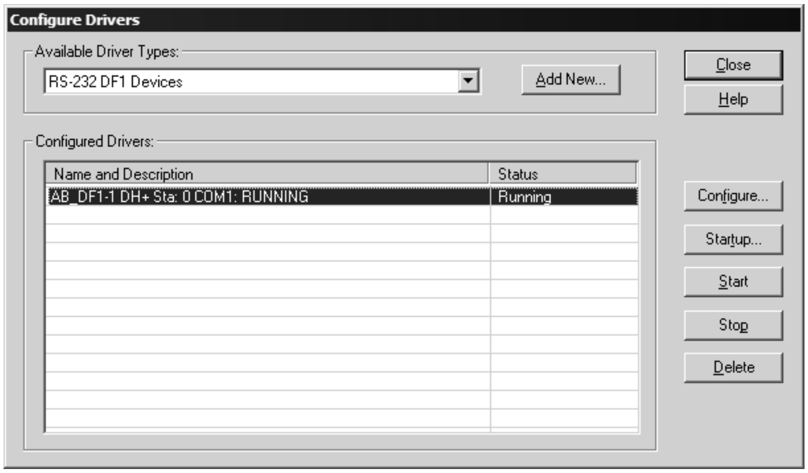
1.6.1 Configuring RSLinx

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

1 Open RSLinx.
2 Open the Communications menu, and choose Configure Drivers.



This action opens the Configure Drivers dialog box.



Configure Drivers

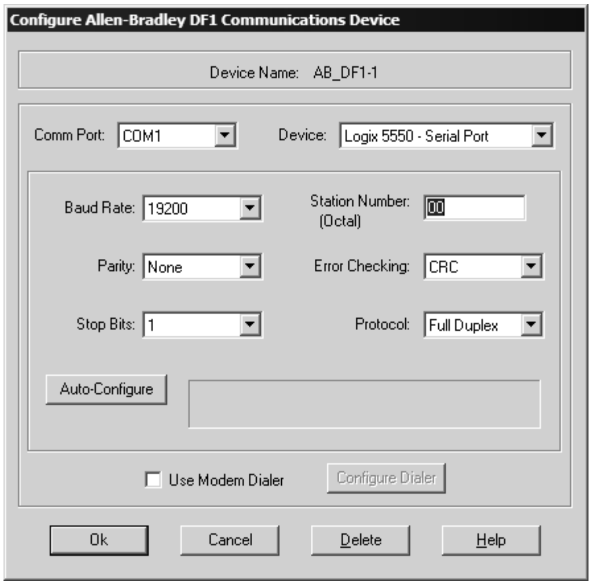
Available Driver Types: RS-232 DF1 Devices

Configured Drivers:

Name and Description	Status
AB_DF1-1 DH+ Sta: 0 COM1: RUNNING	Running

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

3 Click to select the driver, and then click Configure. This action opens the Configure Allen-Bradley DF1 Communications Device dialog box.



Configure Allen-Bradley DF1 Communications Device

Device Name: AB_DF1-1

Comm Port: COM1

Device: Logix 5550 - Serial Port

Baud Rate: 19200

Station Number: (Octal) 00

Parity: None

Error Checking: CRC

Stop Bits: 1

Protocol: Full Duplex

□ Use Modem Dialer

4 Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.

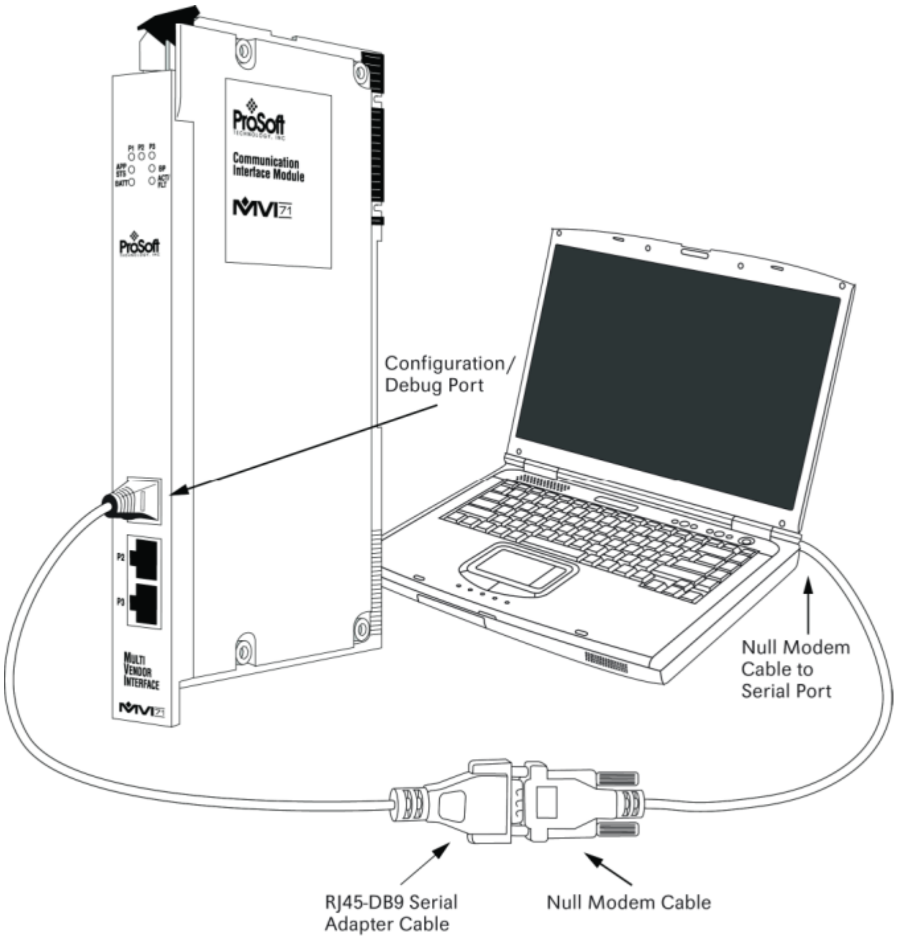
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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

1.7 Connect your PC to the Module

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

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



2 Module Configuration

In This Chapter

This section contains the setup procedure, data and ladder logic requirements for successful application of the MVI71-101S module. Each step in the setup procedure is defined in order to simplify the use of the module. Additionally, this document contains a discussion on configuring the module using the IEC8701S.CFG file. All configuration information used by the module is stored in this file. Go to the **Configuration File** section to begin setting up this file.

The document also contains a discussion of the data areas defined for the module. These areas contain the read and write data from the module and status related to the module. It is important to understand each element of the data areas for proper application of the module.

The ladder logic requirement section defines the minimum ladder logic required to apply the module in a user application. The logic is simple to understand and implement. Go to the **Ladder Logic** section for a complete discussion of the ladder logic requirements.

2.1 Installing and Configuring the Module

This chapter describes how to install and configure the module to work with your application. The configuration process consists of the following steps.

1 Modify the module's configuration files to meet the needs of your application, and copy the updated configuration to the module. Example configuration files are provided on the CD-ROM. Refer to the Modifying the Example Configuration File section, later in this chapter, for more information on the configuration files.

2 Modify the example ladder logic to meet the needs of your application, and copy the ladder logic to the processor. Example ladder logic files are provided on the CD-ROM.

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

The rest of this chapter describes these steps in more detail.

Before installing and configuring the module, design the application. Determine the number points for each data type and complete the forms found in the Reference chapter. Make certain that the point index assigned to each point is unique. This is a requirement of the IEC-60870-5-101 specification.

It is now time to edit the IEC8701S.CFG file to set up the module for the specific application. Refer to the **Configuration File** section of this document. Download this configuration to the module along with the associated ladder logic.

The next step in installing and configuring the module is to define whether the block-transfer or side-connect interface will be utilized. If the block transfer interface is to be used you should be ready to connect the module to the IEC-60870-5-101 network if the ladder logic is defined correctly.

If the side-connect interface is utilized, make sure the file SC_DATA.TXT on the Compact Flash Disk contains the correct first file number. You can run the setdnpsc.exe program to set the file number to be used with your application. Install the module in the rack and turn on the power. Connect the terminal server to the module's debug/configuration port and exit the program by pressing the Esc key followed by the 'X' key. This will cause the program to exit and remain at the operating system prompt. Run the setdnpsc.exe program with a command line argument of the file number to use for the first file. For example, to select N10: as the first file, enter the following:

```
SETDNPSC 10
```

The program will build the SC_DATA.TXT on the Compact Flash Disk (C: drive in the root directory).

The next step in module setup is to define the data files to be used with the application. If the block transfer interface is used, define the data files to hold the user data (read and write data). Enter the ladder logic to handle the blocks transferred between the module and the PLC. Download the program to the PLC and test the program with the module.

If the side-connect interface is used, no ladder logic is required for data transfer. The user data files to interface with the module must reside in contiguous order in the processor. The first file to be used by the interface is the status/control file. This is file number set in the SC_DATA.TXT file using the SETDNPSC.EXE program. The following table lists the files used by the side-connect interface:

File Number	Example	Size	Description
Cfg File	N10	100	Control/Status File
Cfg File+1	N11	to 1000	Data transferred from the module to the processor
			Other files for read data
Cfg File+1+n	N12	to 1000	Data transferred from the processor to the module
Cfg File+1+n+m			Other files for write data

n is the number of read data files minus one. Each file contains up to 1000 words.

m is the number of write data files minus one. Each file contains up to 1000 words.

More than one read and/or write file may exist in an application. This is required when more than 1000 words of data are required. The following two examples are provided for the files used with different data set sizes.

2.1.1 Example of 240 words of read and write data (cfg file=10)

Data Files	Description
N11:0 to 239	Read data
N12:0 to 239	Write data

Example of 2300 read and 3500 write data registers (cfg file=10)

Data Files	Description
N11:0 to 999	Read data words 0 to 999
N12:0 to 999	Read data words 1000 to 1999
N13:0 to 299	Read data words 2000 to 2299
N14:0 to 999	Write data words 0 to 999
N15:0 to 999	Write data words 1000 to 1999
N16:0 to 999	Write data words 2000 to 2999
N17:0 to 499	Write data words 3000 to 3499

Even if the files are not required for an application, they still are reserved and should only be used for that purpose. The read and write data contained in the last set of files possess the data transferred between the module and the processor. The read data file (Cfg File + 1) will contain data transferred from the module to the processor and should be associated with control data types. The write data file (Cfg File + 1 + n) will contain data passed to the module from the processor and should be associated with monitor data types.

Special care must be taken when defining the files for the side-connect interface. Because the module directly interacts with the PLC processor and its memory, any errors in the configuration may cause the processor to fault and it may even lose its configuration and program. After defining the files and populating them with the correct data, download the program to the processor, and place the processor in run mode. If everything is configured correctly, the module should start its normal operation.

The module is now and ready to use with your application. Insert the module in the rack (with the power turned off) and attach the serial communication cables. Download the new application to the controller and place the processor in run mode. Download the new IEC8701S.CFG file to the module using a terminal emulation program. If all the configuration parameters are set correctly and the module is attached to a network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink very rapidly. Refer to the **Diagnostics and Troubleshooting** section if you encounter errors. Attach a computer or terminal to Port 0 on the module and look at the status of the module using the Configuration/Debug Menu in the module. Refer to the **Diagnostics and Troubleshooting** section for a complete discussion of the use of this feature.

2.2 Module Data

All data related to the MVI71-101S module is stored in a user defined data file. It is the responsibility of the ladder logic program to construct all the data files required by the program and to write the ladder logic required to interface to these files.

2.2.1 Status Data

When the side-connect interface is employed in the application, the status data is automatically transferred from the module to the first file used by the interface. The data is placed at an offset of 0 in the file and has the following format.

Offset	Parameter	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
14	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
15	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
16	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
17	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
18	RX Frames	This word contains the total number of frames received by the module
19	TX Frames	This word contains the total number of frames transmitted by the module.

When the block transfer interface is used, the status data is placed in the module's internal database at the location specified by the Error Offset parameter in the configuration file. If this data area is transferred to the processor in the read data area, it will be passed from the module to the processor in a normal BTR block. This will be placed in the normal read data area. The format of the data is exactly the same as shown above, but the user determines its position. Refer to the Reference chapter for a complete listing of the data stored in this object.

2.2.2 User Data

When the side-connect interface is utilized, the read and write data is moved between the module and the processor without any ladder logic. The size of the data area and position of the data areas in the module's database is determined by the parameters set in the configuration file.

When the block transfer interface is used, ladder logic is required to page the data between the module and the processor. The size of the data area and position of the data areas in the module's database is determined by the parameters set in the configuration file.

The read data area should be set to match the value entered in the **Read Register Count** parameter of the IEC8701S.CFG file. For ease of use, this array should be dimensioned as an even increment of 60 words. This data is paged up to 60 words at a time from the module to the processor. The ReadData task places the data received into the proper position in the read data array. Use this data for status and control in the ladder logic of the processor.

The write data area should be set to match the value entered in the **Write Register Count** parameter of the IEC8701S.CFG file. For ease of use, this array should be dimensioned as even increments of 60 words. This data is paged up to 60 words at a time from the processor to the module. The WriteData task places the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.

2.3 Modifying the Configuration File

2.3.1 Obtain the Sample Configuration Files

The ProSoft Solutions CD is organized in folders by module name. In the folder for the module you are using, you will find sample configuration files and other information.

1. Use Windows Explorer to locate the sample configuration files for your MVI71 module on the MVI71 CD.
2. When you have located the correct configuration files, use the Copy and Paste commands to move the files to a location on your PC's hard drive. We recommend C:\temp.

3 Files copied from a CD-ROM are read-only. The next step is to make the files writable. Navigate to the directory where you copied the files, then select the files and click the right mouse button to open a shortcut menu. On the shortcut menu, select Properties, and clear (uncheck) the Read Only check box.

4 The next step is to open the configuration files in a text editor such as Notepad, which comes with Windows. To start Notepad, click the Start button, and then choose **Programs / Accessories / Notepad**.

5 When Notepad starts, open the File menu, and then choose **Open**. Navigate to the folder where you copied the configuration file on your PC and select the file. Click **Open**. The configuration file will open in Notepad, ready for editing.

Note: We do not recommend opening the configuration file in a word processor such as Microsoft Word, because the file may be saved in a format that cannot be read by the module.

2.3.2 Modifying the Configuration File

In order for the module to operate, a configuration file (IEC8701S.CFG) is required. This configuration file contains information to set the data transfer characteristics between the module and the processor, to configure the communication ports, to establish the IEC 60870-5-101 protocol parameters and to define the databases required to hold the protocol data sets. Each parameter in the file must be set carefully in order for the application to be implemented successfully.

The configuration file is separated into sections with topic header names enclosed in the **[ ]** characters. The configuration file consists of the following topics:

[Section]	**Description**
[Backplane Configuration]	Backplane transfer parameter section
[IEC-870-5-101 Port 0]	Slave port communication and protocol parameters
[IEC-870-5-101 Database]	Size definitions for each point database
[M_SP_NA_1]	Definition of monitored single-point database
[M_DP_NA_1]	Definition of monitored dual-point database
[M_ST_NA_1]	Definition of monitored step-point database
[M_ME_NA_1]	Definition of monitored normalized-point database
[M_ME_NB_1]	Definition of monitored scaled-point database
[M_IT_NA_1]	Definition of monitored integrated total database
[C_SC_NA_1]	Definition of command single-point database
[C_DC_NA_1]	Definition of command dual-point database
[C_RC_NA_1]	Definition of command step-point database
[C_SE_NA_1]	Definition of command normalized-point database
[C_SE_NB_1]	Definition of command scaled-point database
[C_SE_NC_1]	Definition of command short-float setpoint database

After each section header, the file contains a set of parameters. Unique labels are used under each section to specify a parameter. Each label in the file must be entered exactly as shown in the file for the parameter to be identified by the program. If the module is not considering a parameter, look at the label for the data item. Each parameter's value is separated from the label with the '**:**' character. This character is used by the program to delimit the position in the data record where to start reading data. All data for a parameter must be placed after the '**:**' character. For numeric parameter values any text located after the value will not be used. There must be at least one space character between the end of the parameter value and the following text. The following is an example of a parameter entry:

```
Baud Rate  : 38400  #Baud rate for port 300 to 38400
```

The parameter label is "Baud Rate" and the parameter value is 38400. The characters after the parameter value are ignored and are used for internal documentation of the configuration file.

Any record that begins with the '#' character is considered to be a comment record. These records can be placed anywhere in the file as long as the '#' character is found in the first column of the line. These lines are ignored in the file and can be used to provide documentation within the configuration file. Liberal use of comments within the file can ease the use and interpretation of the data in the file.

The database definition sections in the database are formatted differently than the other sections. These sections contain lists of parameters to be used with each point in the database. Each list of points in begins with the label **START** and ends when the number of points specified for the data type is reached or when the **END** label is reached. When entering the point list data, make certain that the first character in each line is left blank. The point number should never be present as the first character in the line it should be a space or tab character.

One aspect of the point configuration database that leads to confusion is the group definition field. This assignment for each point assigns a point to one or more interrogation groups. Use of interrogation groups permits the controlling unit to interface with a specific set of data. Refer to the IEC 60870-5-101 standard for a full discussion of interrogation groups. A specific group, Periodic data group, reports data points on a set frequency. The frequency is set in the **Cyclic Data Transmission** parameter in the configuration file. Remember that a point can be assigned to more than one group:

Group Code	Description
0x00000001	Interrogated by general interrogation (station or global)
0x00000002	Interrogated by group 1 interrogation
0x00000004	Interrogated by group 2 interrogation
0x00000008	Interrogated by group 3 interrogation
0x00000010	Interrogated by group 4 interrogation
0x00000020	Interrogated by group 5 interrogation
0x00000040	Interrogated by group 6 interrogation
0x00000080	Interrogated by group 7 interrogation
0x00000100	Interrogated by group 8 interrogation

Group Code	Description
0x00000200	Interrogated by group 9 interrogation
0x00000400	Interrogated by group 10 interrogation
0x00000800	Interrogated by group 11 interrogation
0x00001000	Interrogated by group 12 interrogation
0x00002000	Interrogated by group 13 interrogation
0x00004000	Interrogated by group 14 interrogation
0x00008000	Interrogated by group 15 interrogation
0x00010000	Interrogated by group 16 interrogation
0x00020000	Interrogated by general counter request
0x00040000	Interrogated by group 1 counter request
0x00080000	Interrogated by group 2 counter request
0x00100000	Interrogated by group 3 counter request
0x00200000	Interrogated by group 4 counter request
0x40000000	Disable event scanning of this point
0x80000000	Periodic/cyclic data returned from unit

If the highest bit (bit 31) is set, data will be produced by the driver for the specified point at the rate set for periodic data generation. Bit 30 (0x40000000) enables scanning of this point for event generation. If the bit is clear and the data type is set for scanning, events will be generated for the point. If the bit is set, events will not be generated for the point. This feature can be used to select which points will generate events for the controlling station and can get rid of event data that is not important to the application.

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.

2.3.3 [Backplane Configuration]

This section provides the module with:

- a unique name,
- designates database addresses for input and output on the module and on the processor,
- identifies the method of failure for the communications for the module if the PLC is not in run mode
- describes how to initialize the module upon startup.

The following example shows a sample [Backplane Configuration] section:

```
[Backplane Configuration]
Module Name: MVI71-101S Demo Unit
Read Register Start :  600 #Starting DB address where read data stored
Read Register Count :  600 #Number of regs to read from processor
Write Register Start:    0 #Starting DB address where write data accessed
Write Register Count:  600 #Number of regs to write to processor
Failure Flag Count  :    0 #Determines if BP failure will cause protocol to be
                            #disabled (0=Ignore, >0 = failure count to disable)
Error Offset        : 1000 #Location of where to write status data (-1=disable)
```

Modify each of the parameters according to the needs of your application.

Module Name

0 to 80 characters

This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.

Read Register Start

```
Read Register Start :  600 #Starting DB address where read data stored
```

The Read Register Start parameter assigns the starting address for data to send to the processor.

Read Register Count

```
Read Register Count :  600 #Number of regs to read from processor
```

The Read Register Count parameter defines the size of the module's input database, up to a maximum value of 4000 words.

Write Register Start

```
Write Register Start:    0 #Starting DB address where write data accessed
```

The Write Register Start parameter assigns the starting address for data to retrieve from the processor.

Write Register Count

```
Write Register Count:  600 #Number of regs to write to processor
```

The Write Register Count parameter defines the size of the module's output database, up to a maximum value of 4000 words.

Failure Flag Count

```
Failure Flag Count  :    0 #Determines if BP failure will cause protocol to
                            #be disabled (0=Ignore, >0 = failure count to
                            #disable)
```

This parameter determines if backplane failure will cause protocol to be disabled. Valid values are 0 = Ignore, >0 = failure count before disable.

Error Offset

0 to 8899

This parameter specifies the register location in the module's database where module status data will be stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 8966, the data will be placed in the modules database.

2.3.4 [IEC-870-5-101 Port 0]

This section provides information required to configure a slave application with the module. Most entries contained within this section are self-explanatory. Modify each parameter according to the needs of your application:

Enabled

```
Enabled                  :     1 #0=Disable port, 1=Enable port for
                               #protocol
```

This parameter enables or disables the protocol port. 0 = Disable port, 1 = Enable the port.

Time DB Offset

```
Time DB Offset           :  2000 #DB location of IEC time
```

This parameter defines the location in the database where the time maintained for the IEC protocol is copied. This time is updated when ever a time synchronization command is received from the host and continually as the program runs.

Data Link Address

```
Data link address        :     1 #Range is 0 to 65535
```

This parameter defines the data link address for the device emulated on the module. This address identifies the module on the network along with the common address of ASDU. Valid values are 0 to 65535.

Data Link Address Length

0, 1 or 2

This parameter sets the number of octets used to define the data link address for the session. A value of 0 is only permitted when balanced mode is used.

Common Address of ASDU

```
Common Address of ASDU   :     1 #Range 0 to 65535
```

This parameter specifies the common address of the ASDU (section address) for access to data in the module. There is only one value entered for access to all data in the module.

Common Address of ASDU Len

1 or 2

This parameter specifies the number of octets used for the common address of ASDU. This parameter must be set the same for all devices on the network.

Inform. Object Address Len

1, 2 or 3

This parameter sets the number of octets used to specify the address for an information object in each sector for this session.

Select/Operate Timeout

```
Select/Operate Timeout    :  2000 #Milliseconds before select timeout
```

This parameter sets the number of milliseconds after a select command is received in which to wait for a valid execute command. If the parameter is set to 0, the feature will be disabled.

Cyclic Data Transmission

```
Cyclic data transmission  : 60000 #Numb of milliseconds between cyclic
                                  #updates
```

This parameter defines the number of milliseconds between cyclic updates. If the parameter is set to 0, cyclic data reporting will be disabled.

Use ACTTERM with Setpoint

```
Use ACTTERM with setpoint :     1 #1=Yes, 0=No ACTCON is last response
                                  #to cmd
```

This parameter determines if an ACTTERM will be sent. If the parameter is set to Y, then setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to N, ACTCON is the last response to a setpoint command.

Use ACTTERM with Step

```
Use ACTTERM with step     :     1 #1=Yes, 0=No ACTCON is last response
                                  #to cmd
```

This parameter determines if an ACTTERM will be sent. If the parameter is set to Y, then step commands will issue an ACTTERM when the command is complete. If the parameter is set to N, ACTCON is the last response to a step command.

Single Char ACK F0, 1 or 3

Yes or No

If set to Y, a single character ACK (0xE5) will be sent instead of a fixed length ACK (secondary function code 0) in response to a primary link function code 0, 1 or 3 if there is no access demand for class 1 data (ACD=1). If set to N, the fixed length ACK will be sent.

Single Char ACK C1 or C2

```
Single char ACK C1 or C2  :     1 #1=Single E5 resp when no C1 or C2 data
```

If set to Y, a single character ACK (0xE5) will be sent instead of a fixed length NACK (secondary function code 9) when no response user data is available. If set to N, the fixed length NACK will be sent.

Maximum ASDU Resp Len

```
Maximum ASDU Resp Len     :   249 #Max ASDU response message length may
                                   #have value from 25 to 252 (usually 252).
```

This parameter limits the maximum size of the ASDU portion of a response message. Most applications will use a value of 252.

Cause of Trans Octets

```
Cause of Trans Octets     :     2 #Number of COT octets (1 or 2)
```

This parameter sets the COT length to 1 or 2. The second octet stores the originator address passed in the register. Spontaneous and cyclic data will always respond with the originator address set to 0.

Event Scan Delay

```
Event Scan delay    :      1 #MSec between event scanning (0-65535)
                           #0=Disable
```

If set to 0, the feature will be disabled and the module will not generate any events. If set from 1 to 65535, the parameter represents the number of milliseconds between event scanning. This parameter defines how often the program will scan for new events in the databases.

Scan Events

```
M_SP_NA Scan Events :      1 #0=No scanning, 1=scan for events
M_DP_NA Scan Events :      1 #0=No scanning, 1=scan for events
M_ST_NA Scan Events :      1 #0=No scanning, 1=scan for events
M_ME_NA Scan Events :      1 #0=No scanning, 1=scan for events
M_ME_NB Scan Events :      1 #0=No scanning, 1=scan for events
M_ME_NC Scan Events :      1 #0=No scanning, 1=scan for events
```

Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.

Time Type

```
M_SP_NA Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_DP_NA Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_ST_NA Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_ME_NA Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_ME_NB Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_ME_NC Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
M_IT_NA Time Type   :      2 #0=None, 1=CP24, 2=CP56 time
```

This parameter defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.

Use Balanced Mode

Yes or No

This parameter specifies if the port will use balanced mode. If balanced mode is used, only one controlled station will be permitted on the port. If unbalanced mode is used, multiple controlled stations can be used on a port. Select Yes to use balanced mode and No to use unbalanced mode.

Retry Count

```
Retry Count         :      2 #0 to 255 retries for balanced mode if no
                              #response
```

In balanced mode, this parameter specifies the number of retries (0 to 255) if a response is not received. In unbalanced mode, this parameter is ignored.

Response Timeout

```
Response Timeout    :   1000 #0 to 65535 milliseconds to wait for response
```

This parameter specifies the minimum number of milliseconds to wait for a response to a primary message. Do not set this parameter too small or timeout conditions may prevent successful data transmission. If the timeout is recognized, the message will be retransmitted up to the number of times specified in the Retry Count parameter. This parameter is only used in balance mode.

Baud Rate

This parameter specifies the baud rate to be used on the communication channel (port). Values from 110 to 38.4K are permitted.

Parity

None, Odd, Even

This parameter specifies the parity for this port using the following code definitions: N=none, O=odd, E=even.

RTS On

0 to 65535

The parameter sets the RTS pre-send delay. The value entered represents the number of milliseconds the module will wait after setting the RTS modem line before sending the data.

RTS Off

1 to 65535

This parameter sets the RTS off delay. The value entered represents the number of milliseconds the module will wait after the data packet is sent before dropping the RTS modem line.

Minimum Delay

1 to 65535

This parameter specifies the minimum number of milliseconds to delay before sending the message (setting RTS high). This can be used when the serial network requires time for units to turn off their transmitters.

Receive Timeout

1 to 65535

This value represents the number of milliseconds to wait on a port from the time the first character is received until the last character in the longest message received on the port. This parameter should be set dependent on the baud rate. A value of 2000 should work with most applications.

Backup Port Enabled

```
Backup Port Enabled  :      1 #0=Disable port, 1=Enable port for
protocol
```

This parameter enables or disables the backup port (Port 2).

Backup Port Baud Rate

This parameter specifies the baud rate to be used on the backup communication channel (port). Values from 110 to 38.4K are permitted.

Backup Port Parity

```
Backup Port Parity   :      2 #0=None, 1=Odd, 2=Even, 3=Mark, 4=Space
```

This parameter sets parity for the backup port.

Backup Port RTS On

```
Backup Port RTS On   :      0 #0-65536 mSec before message
```

This parameter specifies the number of milliseconds to delay after asserting the RTS line before data will be sent from the primary port.

Backup Port RTS Off

```
Backup Port RTS Off  :      0 #0-65536 mSec after message
```

This parameter specifies the number of milliseconds to delay after sending the data frame before the RTS line is dropped.

Backup Port Min Delay

```
Backup Port Min Delay:     20 #Minimum # of mSec before response sent
```

This parameter defines the minimum number of milliseconds to wait before a response is sent from the unit.

2.3.5 [IEC-870-5-101 Database]

This section describes the [IEC-870-5-101 Database] Section. An example is shown:

```
[IEC-870-5-101 Database]
Short Pulse Time    :   2000 #MSec for short pulse command
Long Pulse Time     :  10000 #MSec for long pulse command

M_SP_NA point count :     10 #Number of monitored single-points
M_DP_NA point count :     10 #Number of monitored dual-points
M_ST_NA point count :     10 #Number of monitored step-points
M_ME_NA point count :     10 #Number of monitored normalized-points
M_ME_NB point count :     10 #Number of monitored scaled-points
M_ME_NC point count :      5 #Number of monitored short-float points
M_IT_NA point count :      3 #Number of monitored counter-points
C_SC_NA point count :     10 #Number of command single-points
C_DC_NA point count :     10 #Number of command dual-points
C_RC_NA point count :     10 #Number of command step-points
C_SE_NA point count :     10 #Number of command normalized-points
C_SE_NB point count :     10 #Number of command scaled-points
C_SE_NC point count :      5 #Number of command short-float points

M_SP_NA Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_DP_NA Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_ME_NA Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_ME_NB Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_ME_NC Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_IT_NA Sequence    : Yes  #Y=ASDU in sequence with SQ=1, N=report separate
                            #(SQ=0)
M_ME_NA Parameter Offset  :  2000 #M_ME_NA IOA offset for parameter data
M_ME_NB Parameter Offset  :  2000 #M_ME_NB IOA offset for parameter data
M_ME_NC Parameter Offset  :  2000 #M_ME_NC IOA offset for parameter data
```

Each parameter is described below. Edit the configuration file according to the needs of your application.

Short Pulse Time

```
Short Pulse Time    :   2000 #MSec for short pulse command
```

This parameter defines the number of milliseconds to be associated with a short pulse command.

Long Pulse Time

```
Long Pulse Time     :  10000 #MSec for long pulse command
```

This parameter defines the number of milliseconds to be associated with a long pulse command.

Point Count

Point Count configuration ranges in the following configuration items are based on the assumption that you will be using only one of the available data types for your application. The number of point counts you configure will have an effect on module performance, in particular the accuracy of the module's internal clock.

```
M_SP_NA point count :     10 #Number of monitored single-points
```

This parameter specifies the number of point values assigned in monitored single-point database. Range is 0 to 1000.

```
M_DP_NA point count :     10 #Number of monitored dual-points
```

This parameter specifies the number of point values assigned in monitored dual-point database. Range is 0 to 1000.

```
M_ST_NA point count :     10 #Number of monitored step-points
```

This parameter specifies the number of point values assigned in monitored step-point database. Range is 0 to 1000.

```
M_ME_NA point count :     10 #Number of monitored normalized-points
```

This parameter specifies the number of point values assigned in monitored normalized-point database. Range is 0 to 1000.

```
M_ME_NB point count :     10 #Number of monitored scaled-points
```

This parameter specifies the number of point values assigned in monitored scaled-point database. Range is 0 to 1000.

```
M_ME_NC point count :     10 #Number of monitored scaled-points
```

This parameter specifies the number of point values assigned in monitored scaled short-float point database. Range is 0 to 50.

```
M_IT_NA point count :     10 #Number of monitored counter-points
```

This parameter specifies the number of point values assigned in monitored counter-point database. Range is 0 to 99.

```
C_SC_NA point count :     10 #Number of command single-points
```

This parameter specifies the number of point values assigned in command single-point database. Range is 0 to 1000.

```
C_DC_NA point count :     10 #Number of command dual-points
```

This parameter specifies the number of point values assigned in command dual-point database. Range is 0 to 1000.

```
C_RC_NA point count :     10 #Number of command step-points
```

This parameter specifies the number of point values assigned in command step-point database. Range is 0 to 1000.

```
C_SE_NA point count :     10 #Number of command normalized-points
```

This parameter specifies the number of point values assigned in command normalized-point database. Range is 0 to 1000.

```
C_SE_NB point count :     10 #Number of command scaled-points
```

This parameter specifies the number of point values assigned in command scaled-point database. Range is 0 to 1000.

```
C_SE_NC point count :      5 #Number of command short-float points
```

This parameter specifies the number of point values assigned in command short-float point database. Range is 0 to 50.

Parameter Offset

```
M_ME_NA Parameter Offset  :  2000 #M_ME_NA IOA offset for parameter data
M_ME_NB Parameter Offset  :  2000 #M_ME_NB IOA offset for parameter data
M_ME_NC Parameter Offset  :  2000 #M_ME_NC IOA offset for parameter data
```

This parameter specifies the IOA offset to the parameter data for the normalized parameter data. The value entered is added to the Information Object Address for the associated point to compute the parameter IOA address.

The Master may send a "Parameter of Measured Normalized" or "Parameter of Measured Scaled" command using the parameter IOA in order to change the deadband values for specific points.

Note: The Low Limit and High Limit values are always calculated based on the deadband value as described in the following table:

Point	Value
Threshold	Determined by the deadband set in the configuration file or altered by the write command.
Low	Last reported event value - threshold.
High	Last reported event value + threshold.

For example, if the MVI71-101S configuration sets two M_ME_NA points with IOA (Point #) of 600 and 601 and a M_ME_NA Parameter Offset value of 3000, it would result in the parameter points shown in the table.

2.3.6 [M_SP_NA_1]

This section defines the monitored single-point database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies 1 bit. (1 = On, 0 = Off state).

The following is an example of the [M_SP_NA_1] section:

```
[M_SP_NA_1]
#
# Each point is one bit and the DB Address value corresponds to the bit
# offset into the database.
#
# Point #   DB Address   Group(s)  IV DB Bit
# -------   ----------   --------  ---------
START
      100            0   0fffffff          0 # P1-PSHH -- Discharge pressure
                                             # SD
      101            1   00000001          0 # P1-PSH  -- High discharge
                                             # pressure
      102            2   00000003          0 # P1-PSL  -- Low suction
                                             # pressure
      103            3   00000001          0 # P1-FSL  -- Low flow
      104            4   000000f1          0 # P2-PSHH -- Discharge pressure
                                             # SD
      105            5   00000001          0 # P2-PSH  -- High discharge
                                             # pressure
      106            6   00000001          0 # P2-PSL  -- Low suction
                                             # pressure
      107            7   00000003          0 # P2-FSL  -- Low flow
      108            8   00000000          0 #
      109            9   00000000          0 #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- IV DB Bit

Each point is one bit and the DB address value corresponds to the bit offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.7 [M_DP_NA_1]

This section defines the monitored dual-point database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies two bits. (00 = intermediate, 01 = off, 10 = on and 11 = intermediate).

The following is an example of a [M_DP_NA_1] section:

```
[M_DP_NA_1]
#
# Each point is TWO bits and the DB Address value corresponds to the bit
# offset into the database.
#
# Point #   DB Address   Group(s)  IV DB Bit
# -------   ----------   --------  ---------
START
      200           16   0fffffff           0 # Pump 1 Status
      201           18   00000001           0 # MOV101 -- position switch
      202           20   00000001           0 # MOV102 -- position switch
      203           22   00000001           0 # MOV103 -- position switch
      204           24          1           0 # Pump 2 Status
      205           26          1           0 # MOV201 -- position switch
      206           28          1           0 # MOV202 -- position switch
      207           30          1           0 # MOV203 -- position switch
      208           32          1           0 # MOV401 -- position switch
      209           34          1           0 # MOV402 -- position switch
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- IV DB Bit

Each point is two bits and the DB address value corresponds to the bit offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.8 [M_ST_NA_1]

This section defines the monitored step database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies one byte.

The following shows an example of the [M_ST_NA_1] section:

```
[M_ST_NA_1]
#
# Each point is one byte and the DB Address value corresponds to the byte
# offset into the database.
#
# Point #   DB Address   Group(s)  IV DB Bit
# -------   ----------   --------  ---------
START
      300           10   0fffffff          0 #
      301           11   00000001          0 #
      302           12          1          0 #
      303           13          1          0 #
      304           14          1          0 #
      305           15          1          0 #
      306           16          1          0 #
      307           17          1          0 #
      308           18          1          0 #
      309           19          1          0 #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- IV DB Bit

Each point is one byte and the DB Address value corresponds to the byte offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.9 [M_ME_NA_1]

This section defines the monitored measured value, normalized database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies a word position in the database. The IOA for the parameters are for each object and are determined by adding the Point # below to the value of the M_ME_NA parameter offset parameter set in the previous section.

The following shows an example of the [M_ME_NA_1] section:

```
[M_ME_NA_1]
#
# Each point is one word and the DB Address value corresponds to the word
# offset into the database.
#
#                                  Default
# Point #   DB Address   Group(s)  Deadband  IV DB Bit
# -------   ----------   --------  --------  ---------
START
      400           10   0fffffff      2000          0 # P1 suction pressure
      401           11   00000001      1000          0 # P1 discharge
                                                       # pressure
      402           12   00000001      5000          0 # P2 suction pressure
      403           13          1       100          0 # P2 discharge
                                                       # pressure
      404           14          1       100          0 # Station discharge
                                                       # pressure
      405           15          1       100          0 # VSD speed
      406           16          1       100          0 #
      407           17          1       100          0 #
      408           18          1       100          0 #
      409           19          1       100          0 #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB Address value corresponds to the word offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.10 [M_ME_NB_1]

This section defines the monitored measured value, scaled database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies a word position in the database. The IOA for the parameters for each object are determined by adding the Point # below to the value of the M_ME_NB parameter offset parameter set in the previous section.

The following shows an example [M_ME_NB_1] section:

```
[M_ME_NB_1]
#
# Each point is one word and the DB Address value corresponds to the word
# offset into the database.
#
# Point #   DB Address   Group(s)  Deadband  IV DB Bit
# -------   ----------   --------  --------  ---------
START
      500           20   0fffffff       100          0 # P1 inboard bearing
                                                       # temp
      501           21   00000001       100          0 # P1 outboard bearing
                                                       # temp
      502           22   00000001       100          0 # P1 winding Temp
      503           23   00000001       100          0 # P1 current
      504           24          2       100          0 # P2 inboard bearing
                                                       # temp
      505           25          2       100          0 # P2 outboard bearing
                                                       # temp
      506           26          2       100          0 # P2 winding Temp
      507           27          2       100          0 # P2 current
      508           28          0       100          0 #
      509           29          0       100          0 #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB Address value corresponds to the word offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.11 [M_IT_NA_1]

This section defines the monitored integrated totals (counter) database for the slave emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies two words in the database (4 bytes).

The following shows an example [M_IT_NA_1] section:

```
[M_IT_NA_1]
#
# Each point is two words and the DB Address value corresponds to the
# double-word offset into the database.
#
# Point #   DB Address   Group(s)  IV DB Bit
# -------   ----------   --------  ---------
      600           20   0fffffff           # Meter 1 totalizer
      601           21   0fff0000           # Meter 2 totalizer
      602           22   0fff0000           # Meter 3 totalizer
      603           23   0fff0000           # Meter 4 totalizer
END
```

This section takes the following parameters:

- Point #
- DB Address
- Groups
- IV DB Bit

Each point is two words and the DB Address value corresponds to the double-word offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the PLC processor to the MVI71-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.3.12 [C_SC_NA_1]

This section defines the single point command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a single bit position in the database. You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Address values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

The following shows an example of a [C_SC_NA_1] section:

```
[C_SC_NA_1]
#
# Each point is one bit and the DB Address value corresponds to the bit
# offset into the database.
#
```

```
#                        Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START
      700         1600       110     1600        1   #Alarm
      701         1601         0        0        0   #Blower #1
      702         1602         0        0        0   #Blower #2
      703         1603         0        0        0   #Panel Light
      704         1604         0        0        0   #
      705         1605         0        0        0   #
      706         1606         0        0        0   #
      707         1607         0        0        0   #
      708         1608         0        0        0   #
      709         1609         0        0        0   #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is one bit and the DB Address value corresponds to the bit offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.3.13 [C_DC_NA_1]

This section defines the double point command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies two bits in the database. You can associate a command with a monitored double point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Addr values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

The following shows an example of a [C_DC_NA_1] section:

```
[C_DC_NA_1]
#
# Each point is TWO bits and the DB Address value corresponds to the bit
# offset into the database.
#
#                        Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START
      800         1616         0        0        1   # P1 control
      801         1618         0        0        1   # MOV101 control
      802         1620         0        0        0   # MOV102 control
      803         1622         0        0        0   # MOV103 control
      804         1624         0        0        0   # P2 control
      805         1626         0        0        0   # MOV201 control
      806         1628         0        0        0   # MOV202 control
      807         1630         0        0        0   # MOV203 control
      808         1632         0        0        0   # MOV401 control
      809         1634         0        0        0   # MOV402 control
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is two bits and the DB Address value corresponds to the bit offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.3.14 [C_RC_NA_1]

This section defines the step command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a byte in the database. The control value can be associated with a monitored point as described in the previous example.

The following shows an example of a [C_RC_NA_1] section:

```
[C_RC_NA_1]
#
# Each point is one byte and the DB Address value corresponds to the byte
# offset into the database.
#
#                        Monitor  Monitor
# Point #   DB Address   Monitor  DB Addr
# -------   ----------   -------  -------
START
      900          220         0        0     #
      901          221         0        0     #
      902          222         0        0     #
      903          223         0        0     #
      904          224         0        0     #
      905          225         0        0     #
      906          226         0        0     #
      907          227         0        0     #
      908          228         0        0     #
      909          229         0        0     #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr

Each point is one byte and the DB Address value corresponds to the byte offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.3.15 [C_SE_NA_1]

This section defines the normalized setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a word position in the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

The following shows an example of a [C_SE_NA_1] section:

```
[C_SE_NA_1]
#
# Each point is one word and the DB Address value corresponds to the word
# offset into the database.
#

#                                         Require
# Point #   DB Address   Reserved Reserved Select
# -------   ----------   -------  -------  -------
START
     1000          120         0        0        0    # VSD Speed
     1001          121         0        0        0    #
     1002          122         0        0        0    #
     1003          123         0        0        0    #
     1004          124         0        0        0    #
     1005          125         0        0        0    #
     1006          126         0        0        0    #
     1007          127         0        0        0    #
     1008          128         0        0        0    #
     1009          129         0        0        0    #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is one word and the DB Address value corresponds to the word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.3.16 [C_SE_NB_1]

This section defines the scaled setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

The following shows an example of a [C_SE_NB_1] section:

```
[C_SE_NB_1]
#
# Each point is one word and the DB Address value corresponds to the word
# offset into the database.
#
#                                           Require
# Point #   DB Address   Reserved Reserved Select
# -------   ----------   -------  -------  -------
START
     1100          130        0        0        0    #
     1101          131        0        0        0    #
     1102          132        0        0        0    #
     1103          133        0        0        0    #
     1104          134        0        0        0    #
     1105          135        0        0        0    #
     1106          136        0        0        0    #
     1107          137        0        0        0    #
     1108          138        0        0        0    #
     1109          139        0        0        0    #
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is one word and the DB Address value corresponds to the word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.3.17 [C_SE_NC_1]

This section defines the short-float setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a double-word position in the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

The following shows an example of a [C_SE_NC_1] section:

```
[C_SE_NC_1]
#
# Each point is two words and the DB Address value corresponds to the
# double-word offset into the database.
#
#           DB Address                      Require
# Point #   (word*2)     Reserved Reserved  Select
# -------   ----------   -------  -------  -------
START
     1200           70         0        0        0    #db word offset 140
     1201           71         0        0        0    #db word offset 142
     1202           72         0        0        0    #db word offset 144
     1203           73         0        0        0    #db word offset 146
     1204           74         0        0        0    #db word offset 148
END
```

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is two words and the DB Address value corresponds to the double-word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI71-101S to the PLC processor. You should configure the DB Address parameter in an area that is on a Backplane Command Function 2.

2.4 Uploading and Downloading the Configuration File

ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you must transfer the file from the module to your PC. After editing, you must transfer the file back to the module.

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 55).

2.4.1 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

2.4.2 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).

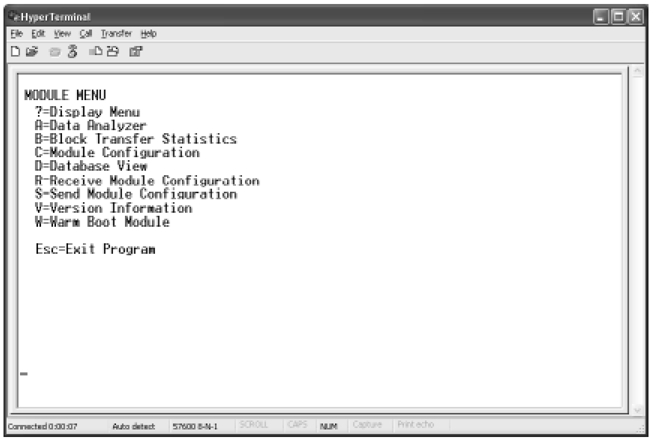
A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have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

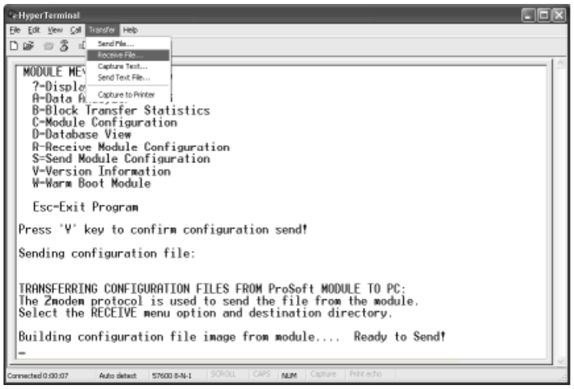
2.4.3 Transferring the Configuration File to Your PC

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.



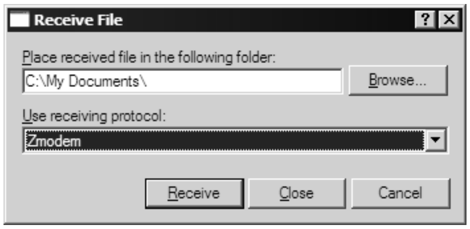
```
MODULE MENU
 ?=Display Menu
 A=Data Analyzer
 B=Block Transfer Statistics
 C=Module Configuration
 D=Database View
 R=Receive Module Configuration
 S=Send Module Configuration
 V=Version Information
 W=Warm Boot Module

 Esc=Exit Program
```

2 From the **Transfer** menu in HyperTerminal, select **Receive File**.



```
MODULE ME
 ?-Displa
 A-Data A
 B-Block Transfer Statistics
 C-Module Configuration
 D-Database View
 R-Receive Module Configuration
 S=Send Module Configuration
 V-Version Information
 W-Warm Boot Module

 Esc-Exit Program

Press 'Y' key to confirm configuration send!

Sending configuration file:


TRANSFERRING CONFIGURATION FILES FROM ProSoft MODULE TO PC:
The Zmodem protocol is used to send the file from the module.
Select the RECEIVE menu option and destination directory.

Building configuration file image from module....  Ready to Send!
```

3 In the Receive File dialog box, browse to the location on your PC where the configuration file should be stored, and select Zmodem (or Zmodem with Crash Recovery) as the receiving protocol.

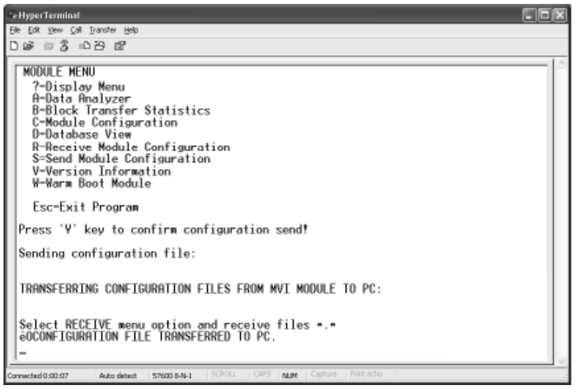


Place received file in the following folder: C:\My Documents\

Use receiving protocol: Zmodem

When you have completed your selections, click Close.

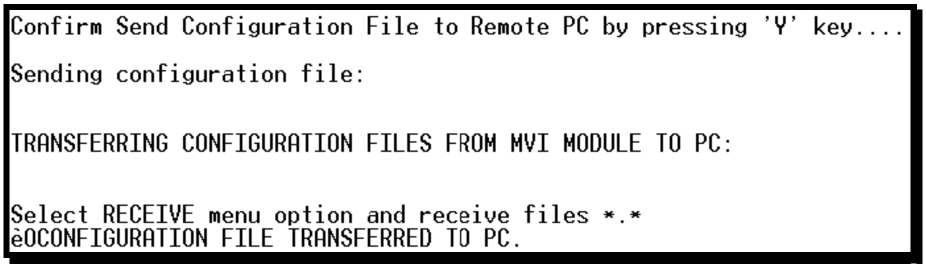
4 Press **[S]** (Send Module Configuration), and then press **[Y]** to confirm the transfer.



```
MODULE MENU
  ?-Display Menu
  A-Data Analyzer
  B-Block Transfer Statistics
  C-Module Configuration
  D-Database View
  R-Receive Module Configuration
  S=Send Module Configuration
  V-Version Information
  W-Warm Boot Module

  Esc-Exit Program
Press 'Y' key to confirm configuration send!

Sending configuration file:


TRANSFERRING CONFIGURATION FILES FROM MVI MODULE TO PC:


Select RECEIVE menu option and receive files *.*
è0CONFIGURATION FILE TRANSFERRED TO PC.
_
```

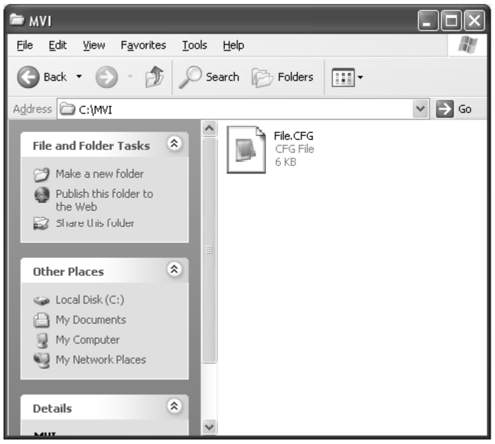
The file transfer will then begin automatically, using the protocol and location you specified in Step 3.

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



```
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 *.*
è0CONFIGURATION FILE TRANSFERRED TO PC.
```

The configuration file is now on your PC at the location you specified.

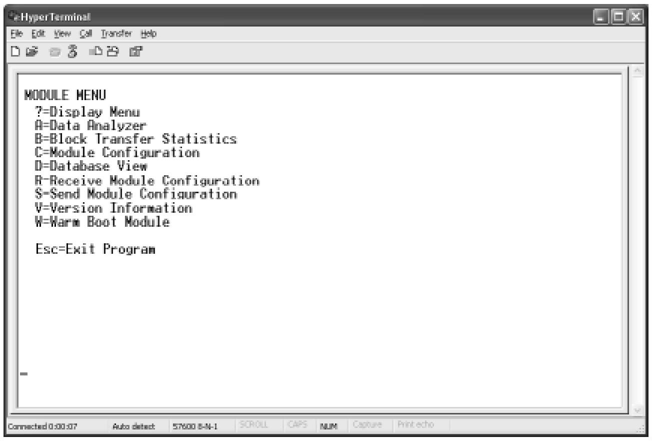


5 You can now open and edit the file in a text editor such as Notepad. When you have finished editing the file, save it and close Notepad.

2.4.4 Transferring the Configuration File to the Module

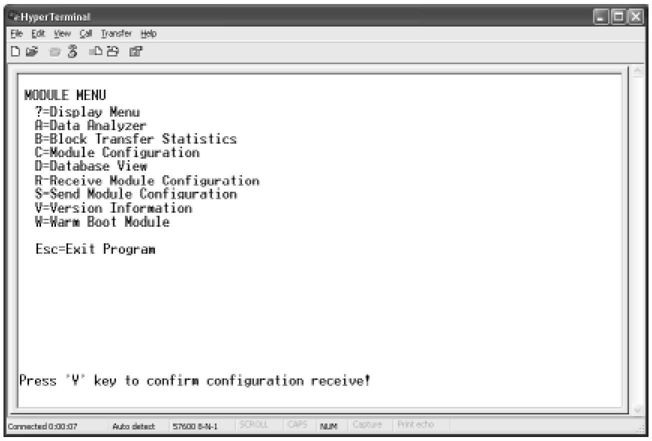
Perform the following steps to transfer a configuration file from your PC to the module.

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.



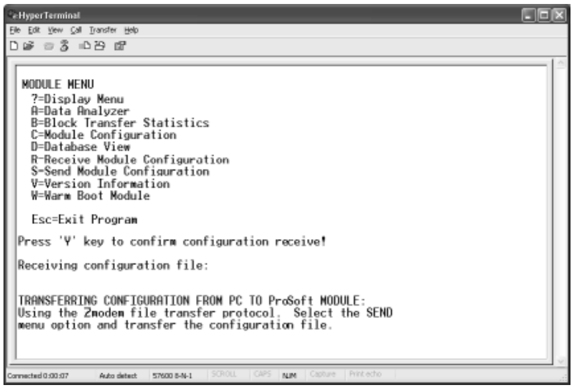
```
MODULE MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Receive Module Configuration
  S=Send Module Configuration
  V=Version Information
  W=Warm Boot Module

  Esc=Exit Program
```

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



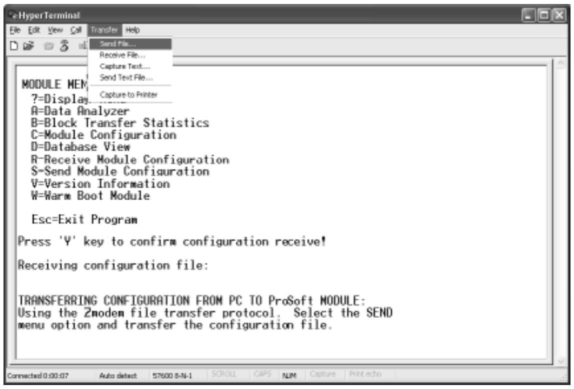
```
MODULE MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Receive Module Configuration
  S=Send Module Configuration
  V=Version Information
  W=Warm Boot Module

  Esc=Exit Program






Press 'Y' key to confirm configuration receive!
```

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


HyperTerminal
File Edit View Call Transfer Help
Connected 0:00:07 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

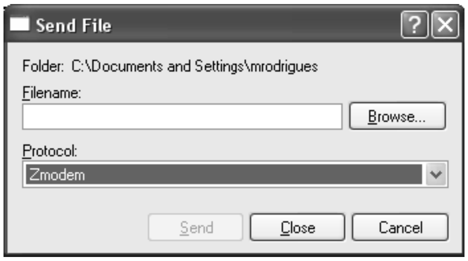
```
MODULE MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R-Receive Module Configuration
  S-Send Module Configuration
  V=Version Information
  W=Warm Boot Module

  Esc=Exit Program
Press 'Y' key to confirm configuration receive!

Receiving configuration file:


TRANSFERRING CONFIGURATION FROM PC TO ProSoft MODULE:
Using the Zmodem file transfer protocol.  Select the SEND
menu option and transfer the configuration file.
```

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


HyperTerminal
File Edit View Call Transfer Help
Send File...
Receive File...
Capture Text...
Send Text File...
Capture to Printer
Connected 0:00:07 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

```
MODULE MEN
  ?=Displa
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R-Receive Module Configuration
  S-Send Module Configuration
  V=Version Information
  W=Warm Boot Module

  Esc=Exit Program
Press 'Y' key to confirm configuration receive!

Receiving configuration file:


TRANSFERRING CONFIGURATION FROM PC TO ProSoft MODULE:
Using the Zmodem file transfer protocol.  Select the SEND
menu option and transfer the configuration file.
```

The Send File dialog appears.

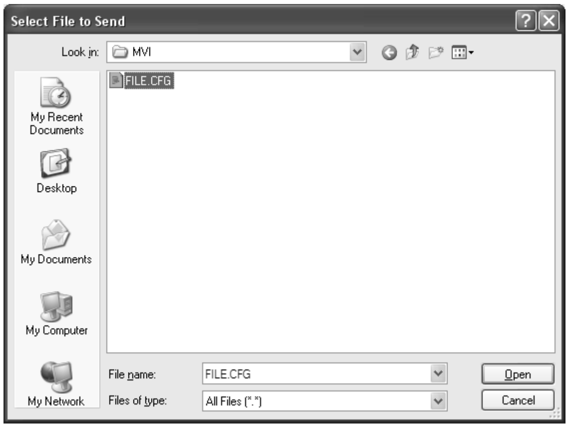


Folder: C:\Documents and Settings\mrodrigues

Filename:

Protocol: Zmodem

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



Look in: MVI

FILE.CFG

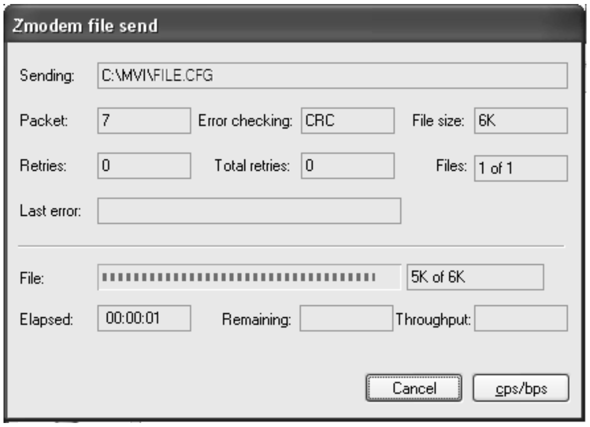
File name: FILE.CFG

Files of type: All Files (*.*)

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.



Sending: C:\MVI\FILE.CFG

Packet: 7 Error checking: CRC File size: 6K

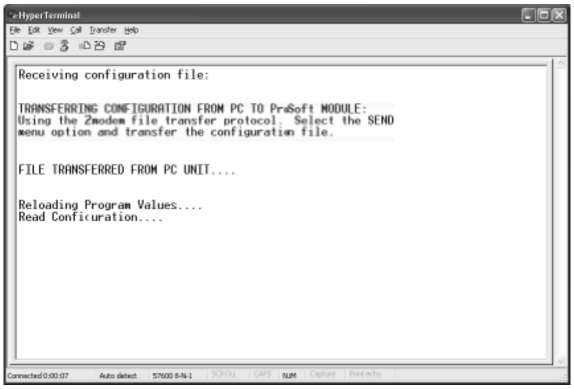
Retries: 0 Total retries: 0 Files: 1 of 1

Last error:

File: 5K of 6K

Elapsed: 00:00:01 Remaining: Throughput:

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



```
Receiving configuration file:


TRANSFERRING CONFIGURATION FROM PC TO ProSoft MODULE:
Using the Zmodem file transfer protocol.  Select the SEND
menu option and transfer the configuration file.


FILE TRANSFERRED FROM PC UNIT....


Reloading Program Values....
Read Conficuration....
```

8 Your module now contains the new configuration.

3 Ladder Logic

Ladder logic is required for application of the MVI71-101S module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

4 Diagnostics and Troubleshooting

In This Chapter

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

- Status data values are transferred from the module to the processor.
- Data contained in the module can be viewed through the Configuration/Debug port attached to a terminal emulator.
- LED status indicators on the front of the module provide information on the module's status.

4.1 Reading Status Data From the Module

The MVI71-101S module returns a 20-word Status Data block that can be used to determine the module's operating status. This data can be located in the module's database at registers at the location specified in the configuration.

4.1.1 The Configuration/Debug Menu

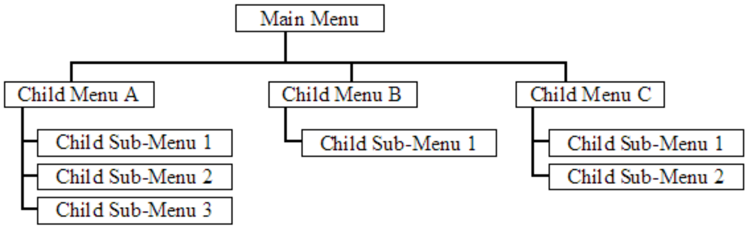
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the terminal application (for example, HyperTerminal). 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.

Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu 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 you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital 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][/]**.

Also, take care to distinguish capital letter **[I]** from lower case letter **[l]** (L) and number **[1]**; likewise for capital letter **[O]** and number **[0]**. Although these characters look nearly the same on the screen, they perform different actions on the module.

4.1.2 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

4.1.3 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 (upload) and receive (download) 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.

4.1.4 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.

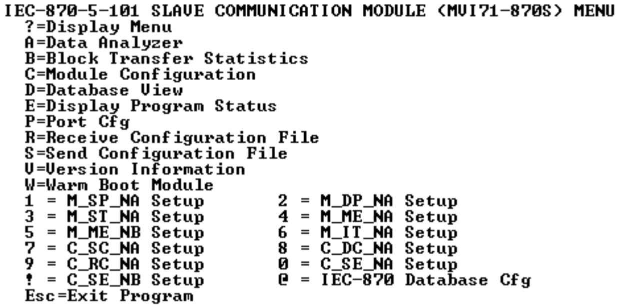
If there is no response from the module, follow these steps:

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 (page 87).
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.

4.1.5 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 on your terminal screen:



```
IEC-870-5-101 SLAVE COMMUNICATION MODULE (MVI71-870S) MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  E=Display Program Status
  P=Port Cfg
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  W=Warm Boot Module
  1 = M_SP_NA Setup       2 = M_DP_NA Setup
  3 = M_ST_NA Setup       4 = M_ME_NA Setup
  5 = M_ME_NB Setup       6 = M_IT_NA Setup
  7 = C_SC_NA Setup       8 = C_DC_NA Setup
  9 = C_RC_NA Setup       0 = C_SE_NA Setup
  ! = C_SE_NB Setup       @ = IEC-870 Database Cfg
  Esc=Exit Program
```

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.

Viewing Block Transfer Statistics

Press **[B]** 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: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

Press **[C]** to view the Module Configuration screen.

Use this command to display the current configuration and statistics for the module.

Viewing Program Status

Press **[E]** to view the error/status data for the module.

Viewing Port Configuration

Press **[6]** or **[7]** from the Main Menu to view configuration information for ports 1 and 2.

Use this command to display detailed configuration information for the selected port.

Receiving the Configuration File

Press **[R]** 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 45).

Sending the Configuration File

Press **[S]** 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 45).

Viewing Version Information

Press **[V]** 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.

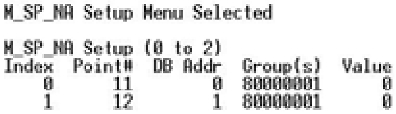
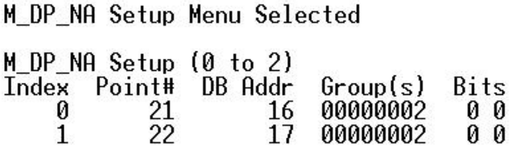
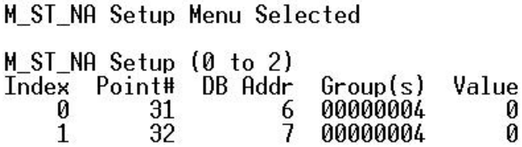
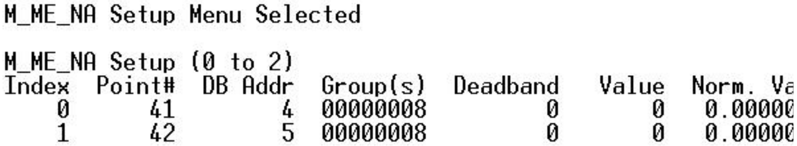
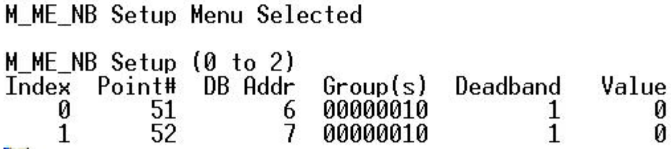
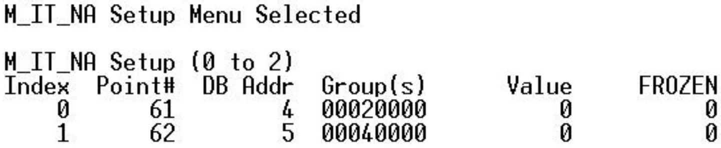
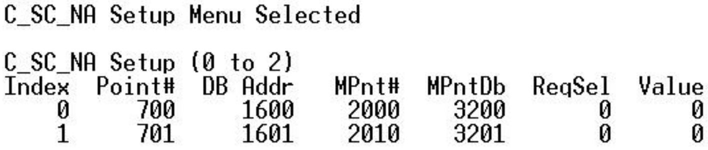
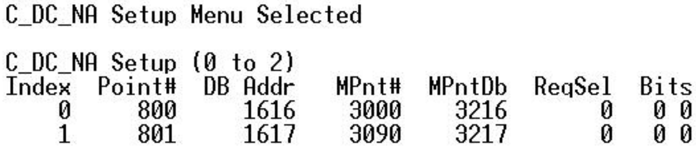
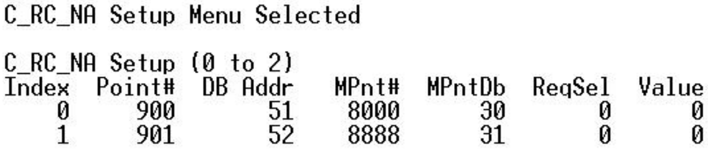
Warm Booting the Module

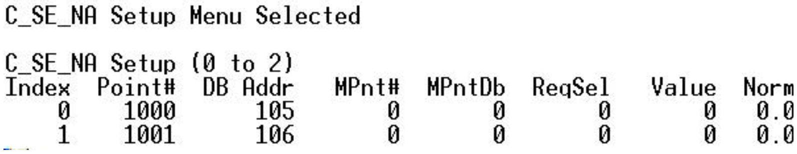
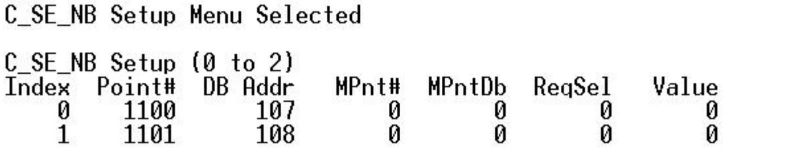
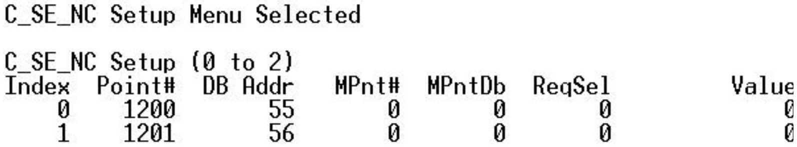
Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically 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 **[W]** 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 force the module to re-boot.

Viewing Data Type Setup

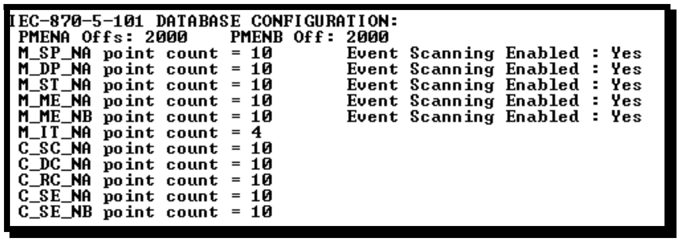
To view setup information for each data type, press the matching key from the Menu.

Key	Data Type	Screen Example
1	M_SP_NA	 M_SP_NA Setup Menu Selected M_SP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value 0 11 0 80000001 0 1 12 1 80000001 0
2	M_DP_NA	 M_DP_NA Setup Menu Selected M_DP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Bits 0 21 16 00000002 0 0 1 22 17 00000002 0 0
3	M_ST_NA	 M_ST_NA Setup Menu Selected M_ST_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value 0 31 6 00000004 0 1 32 7 00000004 0
4	M_ME_NA	 M_ME_NA Setup Menu Selected M_ME_NA Setup (0 to 2) Index Point# DB Addr Group(s) Deadband Value Norm. Va 0 41 4 00000008 0 0 0.00000 1 42 5 00000008 0 0 0.00000
5	M_ME_NB	 M_ME_NB Setup Menu Selected M_ME_NB Setup (0 to 2) Index Point# DB Addr Group(s) Deadband Value 0 51 6 00000010 1 0 1 52 7 00000010 1 0
6	M_IT_NA	 M_IT_NA Setup Menu Selected M_IT_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value FROZEN 0 61 4 00020000 0 0 1 62 5 00040000 0 0
7	C_SC_NA	 C_SC_NA Setup Menu Selected C_SC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 700 1600 2000 3200 0 0 1 701 1601 2010 3201 0 0
8	C_DC_NA	 C_DC_NA Setup Menu Selected C_DC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Bits 0 800 1616 3000 3216 0 0 0 1 801 1617 3090 3217 0 0 0
9	C_RC_NA	 C_RC_NA Setup Menu Selected C_RC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 900 51 8000 30 0 0 1 901 52 8888 31 0 0

Key	Data Type	Screen Example
0	C_SE_NA	 C_SE_NA Setup Menu Selected C_SE_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value Norm 0 1000 105 0 0 0 0 0.0 1 1001 106 0 0 0 0 0.0
Shift 1	C_SE_NB	 C_SE_NB Setup Menu Selected C_SE_NB Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 1100 107 0 0 0 0 1 1101 108 0 0 0 0
Shift 4	C_SE_NC	 C_SE_NC Setup Menu Selected C_SE_NC Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 1200 55 0 0 0 0 1 1201 56 0 0 0 0

Viewing IEC 60870 Database Configuration

Press **[@]** to view the database size configuration information.



```
IEC-870-5-101 DATABASE CONFIGURATION:
 PMENA Offs: 2000    PMENB Off: 2000
 M_SP_NA point count = 10       Event Scanning Enabled : Yes
 M_DP_NA point count = 10       Event Scanning Enabled : Yes
 M_ST_NA point count = 10       Event Scanning Enabled : Yes
 M_ME_NA point count = 10       Event Scanning Enabled : Yes
 M_ME_NB point count = 10       Event Scanning Enabled : Yes
 M_IT_NA point count = 4
 C_SC_NA point count = 10
 C_DC_NA point count = 10
 C_RC_NA point count = 10
 C_SE_NA point count = 10
 C_SE_NB point count = 10
```

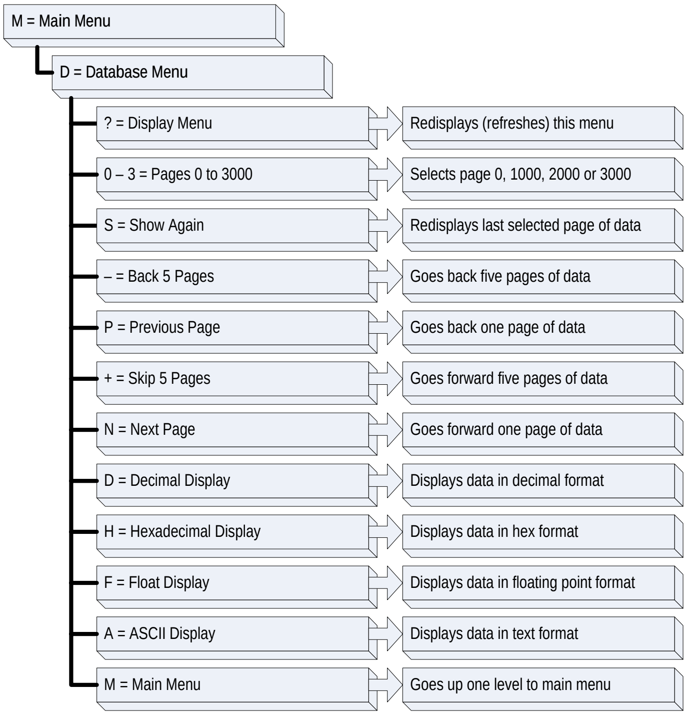
Exiting the Program

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 **[Esc]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash ROM to configure the module.

4.1.6 Database View Menu

Press **[D]** from the Main Menu to open the Database View menu. Use this menu command to view the current contents of the module's database. Press **[?]** to view a list of commands available on this menu.


M = Main Menu
D = Database Menu
? = Display Menu
Redisplays (refreshes) this menu
0 – 3 = Pages 0 to 3000
Selects page 0, 1000, 2000 or 3000
S = Show Again
Redisplays last selected page of data
– = Back 5 Pages
Goes back five pages of data
P = Previous Page
Goes back one page of data
+ = Skip 5 Pages
Goes forward five pages of data
N = Next Page
Goes forward one page of data
D = Decimal Display
Displays data in decimal format
H = Hexadecimal Display
Displays data in hex format
F = Float Display
Displays data in floating point format
A = ASCII Display
Displays data in text format
M = Main Menu
Goes up one level to main menu

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.

Moving Back Through 5 Pages of Registers

Press **[-]** from the Database View menu to skip back to the previous 500 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the Database View menu to display the previous 100 registers of data.

Skipping 500 Registers of Data

Hold down **[Shift]** and press **[=]** to skip forward to the next 500 registers of data.

Viewing the Next 100 Registers of Data

Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Returning to the Main Menu

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

4.1.7 Data Analyzer

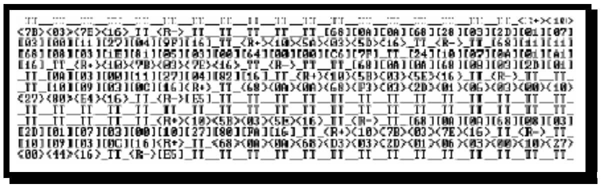
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

Press **[1]** to display I/O data for the first application port in the Data Analyzer. The following illustration shows an example of the Data Analyzer output.



```
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_<R+><10>
<7B><03><7E><16>_TT_<R->_TT__TT__TT__TT__TT_[68][0A][0A][68][28][03][2D][01][07]
[03][00][11][27][04][9F][16]_TT_<R+><10><5A><03><5D><16>_TT_<R->_TT_[68][11][11]
[68][08][03][1E][81][05][03][00][64][00][00][C6][7F]_TT_[24][10][07][0A][01][A1]
[16]_TT_<R+><10><7B><03><7E><16>_TT_<R->_TT__TT_[68][0A][0A][68][08][03][2D][01]
_TT_[0A][03][00][11][27][04][82][16]_TT_<R+><10><5B><03><5E><16>_TT_<R->_TT__TT_
_TT_[10][09][03][0C][16]<R+>_TT_<68><0A><0A><68><F3><03><2D><01><06><03><00><10>
<27><00><E4><16>_TT_<R->[E5]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT_<R+><10><5B><03><5E><16>_TT_<R->_TT_[68][0A][0A][68][08][03]
[2D][01][07][03][00][10][27][80][FA][16]_TT_<R+><10><7B><03><7E><16>_TT_<R->_TT_
[10][09][03][0C][16]<R+>_TT_<68><0A><0A><68><D3><03><2D><01><06><03><00><10><27>
<00><44><16>_TT_<R->[E5]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
```

Analyzing Data for the second application port

Press **[2]** to display I/O data for the second application port in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

You can display timing marks for a variety of intervals in the data analyzer screen. These timing marks can help you determine communication-timing characteristics.

Key	Interval
[5]	1 milliseconds ticks
[6]	5 milliseconds ticks
[7]	10 milliseconds ticks
[8]	50 milliseconds ticks
[9]	100 milliseconds ticks
[0]	Turn off timing marks

Removing Timing Marks in the Data Analyzer

Press **[0]** to turn off timing marks in the Data Analyzer screen.

Viewing Data in Hexadecimal Format

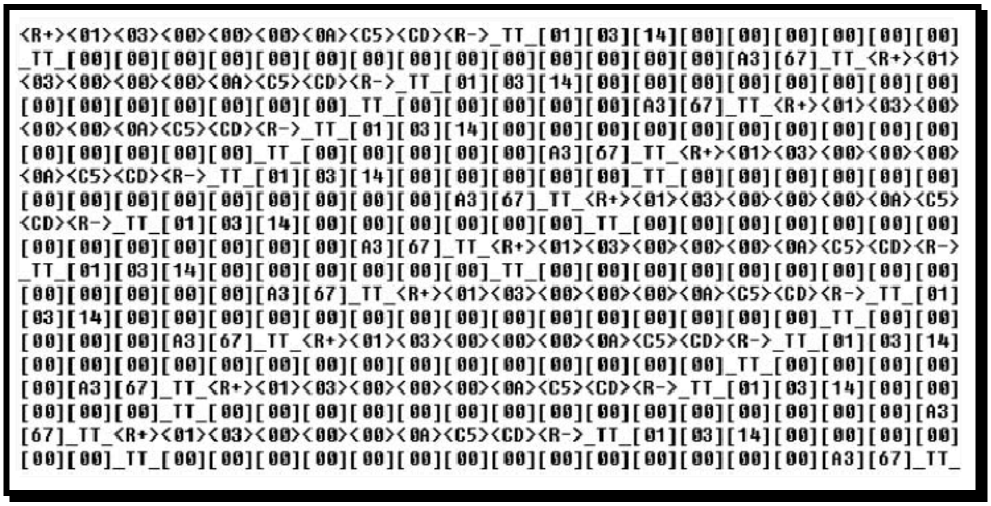
Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

Press **[B]** to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port will be displayed. An example display is shown below:



```
<R+><01><03><00><00><00><0A><C5><CD><R->_TT_[01][03][14][00][00][00][00][00][00]
_TT_[00][00][00][00][00][00][00][00][00][00][00][00][00][00][A3][67]_TT_<R+><01>
<03><00><00><00><0A><C5><CD><R->_TT_[01][03][14][00][00][00][00][00][00][00][00]
[00][00][00][00][00][00][00]_TT_[00][00][00][00][00][A3][67]_TT_<R+><01><03><00>
<00><00><0A><C5><CD><R->_TT_[01][03][14][00][00][00][00][00][00][00][00][00][00]
[00][00][00][00][00]_TT_[00][00][00][00][00][A3][67]_TT_<R+><01><03><00><00><00>
<0A><C5><CD><R->_TT_[01][03][14][00][00][00][00][00]_TT_[00][00][00][00][00][00]
[00][00][00][00][00][00][00][00][00][A3][67]_TT_<R+><01><03><00><00><00><0A><C5>
<CD><R->_TT_[01][03][14][00][00][00][00][00][00]_TT_[00][00][00][00][00][00][00]
[00][00][00][00][00][00][00][A3][67]_TT_<R+><01><03><00><00><00><0A><C5><CD><R->
_TT_[01][03][14][00][00][00][00][00][00]_TT_[00][00][00][00][00][00][00][00][00]
[00][00][00][00][00][A3][67]_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT_[01]
[03][14][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00]_TT_[00][00]
[00][00][00][A3][67]_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT_[01][03][14]
[00][00][00][00][00][00][00][00][00][00][00][00][00][00][00]_TT_[00][00][00][00]
[00][A3][67]_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT_[01][03][14][00][00]
[00][00][00]_TT_[00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][A3]
[67]_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT_[01][03][14][00][00][00][00]
[00][00]_TT_[00][00][00][00][00][00][00][00][00][00][00][00][00][00][A3][67]_TT_
```

The Data Analyzer displays the following special characters:

Character	Definition
[ ]	Data enclosed in these characters represent data received on the port.
< >	Data enclosed in these characters represent data transmitted on the port.
<R+>	These characters are inserted when the RTS line is driven high on the port.
<R->	These characters are inserted when the RTS line is dropped low on the port.
<CS>	These characters are displayed when the CTS line is recognized high.
_TT_	These characters are displayed when the timing mark interval has been reached. This parameter is user defined.

Stopping the Data Analyzer

Press **[S]** to stop the data analyzer. Use this option to freeze the display so the data can be analyzed. To restart the analyzer, press **[B]**.

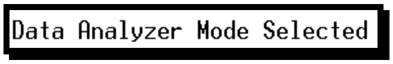
Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Returning to the Main Menu

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

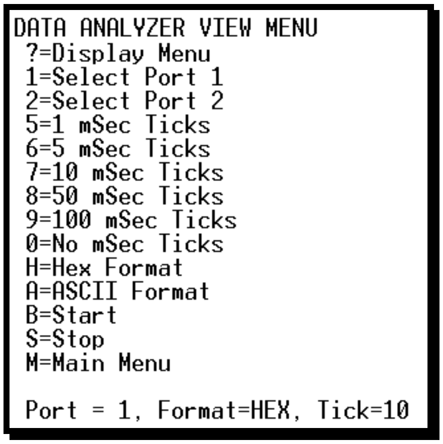
4.1.8 Data Analyzer Tips

From the main menu, press **[A]** for the "Data Analyzer". You should see the following text appear on the screen:



```
Data Analyzer Mode Selected
```

After the "Data Analyzer" mode has been selected, press **[?]** to view the Data Analyzer menu. You will see the following menu:



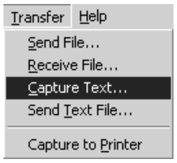
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

 Port = 1, Format=HEX, Tick=10
```

From this menu, you can select the "Port", the "format", and the "ticks" that you can display the data in.

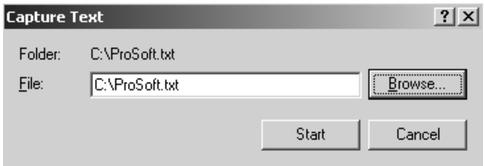
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

The Tick value is a timing mark. The module will print a _TT for every xx milliseconds of no data on the line. Usually 10milliseconds is the best value to start with.

After you have selected the Port, Format, and Tick, we are now ready to start a capture of this data. The easiest way to do so is to go up to the top of you HyperTerminal window, and do a **Transfer / Capture Text** as shown below:



After selecting the above option, the following window will appear:

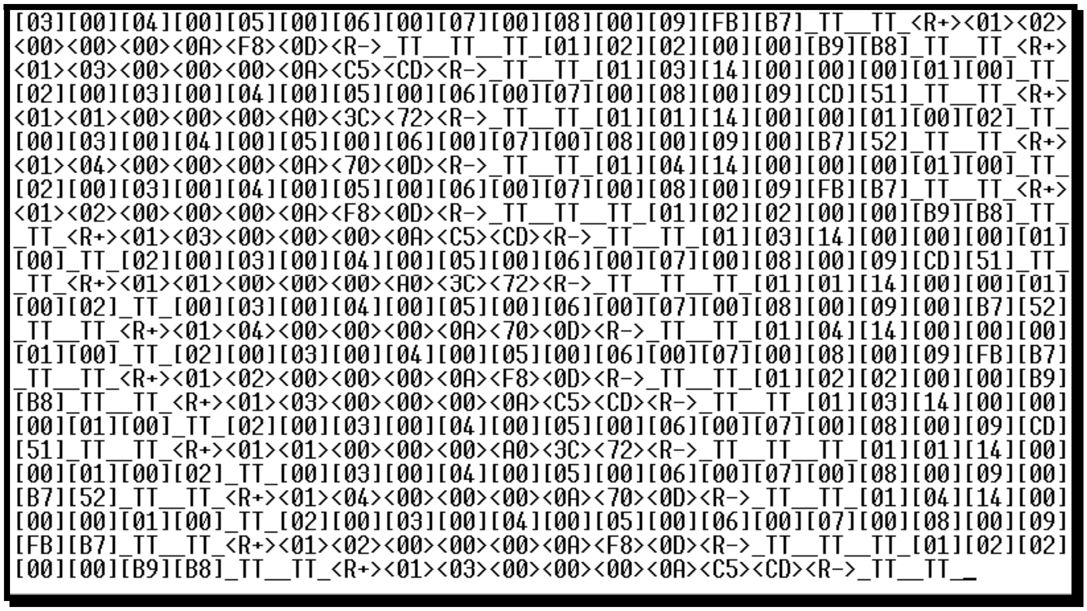


Folder: C:\ProSoft.txt
File: C:\ProSoft.txt

Next name the file, and select a directory to store the file in. In this example, we are creating a file ProSoft.txt and storing this file on our root C: drive. After you have done this, press the Start button.

Now you have everything that shows up on the HyperTerminal screen being logged to a file called ProSoft.txt. This is the file that you will then be able to email to ProSoft Technical Support to assist with issues on the communications network.

To begin the display of the communications data, you will then want to press 'B' to tell the module to start printing the communications traffic out on the debug port of the module. After you have pressed 'B', you should see something like the following:



```
[03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+><01><02>
<00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT__TT_<R+>
<01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT__TT_<R+>
<01><01><00><00><00><A0><3C><72><R->_TT__TT_[01][01][14][00][00][01][00][02]_TT_
[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]_TT__TT_<R+>
<01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+>
<01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT_
_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01]
[00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT_
_TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00][00][01]
[00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]
_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00]
[01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]
_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT_[01][02][02][00][00][B9]
[B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00]
[00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD]
[51]_TT__TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00]
[00][01][00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00]
[B7][52]_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00]
[00][00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09]
[FB][B7]_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02]
[00][00][B9][B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT__
```

The <R+> means that the module is transitioning the communications line to a transmit state.

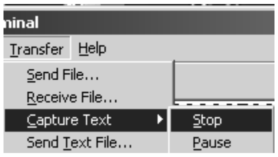
All characters shown in <> brackets are characters being sent out by the module.

The <R-> shows when the module is done transmitting data, and is now ready to receive information back.

And finally, all characters shown in the [ ] brackets is information being received from another device by the module.

After taking a minute or two of traffic capture, you will now want to stop the "Data Analyzer". To do so, press the 'S' key, and you will then see the scrolling of the data stop.

When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

4.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the IEC 60870-5-101 network on Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the IEC 60870-5-101 network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI71-101S is working normally.
		On	The MVI71-101S module program has recognized a communication error on one of its ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

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

4.2.1 Clearing a Fault Condition

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

To clear the condition, follow these steps:

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

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

4.2.2 Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic.

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

5 Reference

In This Chapter

5.1 Standards

The standards used in developing the product are listed in the following table:

PUBLICATION	TITLE
IEC 60870-5-101	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-101 Amendment 1	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

Refer to these standards for any questions on the protocol supported.

The MVI71 IEC 60870-5-101 Slave Communication Module allows Rockwell Automation PLC I/O compatible processors to interface easily with IEC 60870-5-101 protocol compatible hosts. The module's two powerful and highly configurable ports allow the many SCADA host systems supporting the IEC protocol to be integrated into the PLC platform.

5.1.1 Features and Benefits

The MVI71-101S module is the fastest and easiest way to add IEC 60870-5-101 protocol interface support to the PLC platform. It is a single slot, backplane compatible solution for the PLC platform. This module has two powerful and highly configurable IEC 60870-5-101 Slave ports, allowing the many SCADA and field devices supporting the IEC protocol to be integrated into the PLC platform.

The MVI71-101S module acts as an input/output module between the IEC 60870-5-101 telecontrol network and the PLC backplane. The data transfer from the PLC processor is asynchronous from the actions on the network. A 5000-word register space in the module exchanges data between the processor and the telecontrol network.

5.1.2 General Specifications

- Single Slot - 1771 backplane compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module
- Ladder Logic is used for data transfer between module and processor. Sample ladder file included.
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included.

5.1.3 Hardware Specifications

Specification	Description
Form Factor	Single Slot 1771 chassis compatible BTR/BTW data transfer Local or remote rack
Backplane current load	800 mA @ 5 V
Operating temperature	0 to 60°C (32 to 140°F)
Storage temperature	-40 to 85°C (-40 to 185°F)
Shock	30g operational 50g non-operational
Vibration	5 g from 10150 Hz
Relative humidity	5% to 95% (non-condensing)
LED Indicators	Module status Backplane transfer status Application status Serial activity and error LED status
Debug/Configuration port (CFG)	
CFG Port (P1)	RJ45 (DB-9M with supplied cable) RS-232 only
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Application Serial port (P2, P3) (Serial Modules)	Two RJ45 RS-232/422/485 Application ports

5.1.4 Functional Specifications

The MVI71-101S module accepts commands from an attached master unit. A port configured as a virtual slave permits a remote master to interact with all data contained in the module. This data can be derived from the PLC processor. The remote master device uses the fully-configured databases in the module to control outputs and monitor inputs. The module can operate in balanced or unbalanced mode.

- Supports time stamp events
- Supports time and data synchronization from a master or the processor
- Supports monitored data
- Event queue supports 99 points for each data type
- Reports events by configurable priority order
- Order monitored points by interrogation groups
- Configurable deadband for monitored measured points
- Supports Master Class 1 and Class 2 polls with configurable parameters
- Acknowledgement transmission is handled internally by the module
- Configurable data link address, Common ASDU address and Information Object Address
- Configurable pulse duration

5.2 Functional Overview

This section provides an overview of how the MVI71-101S module transfers data using the 101S protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

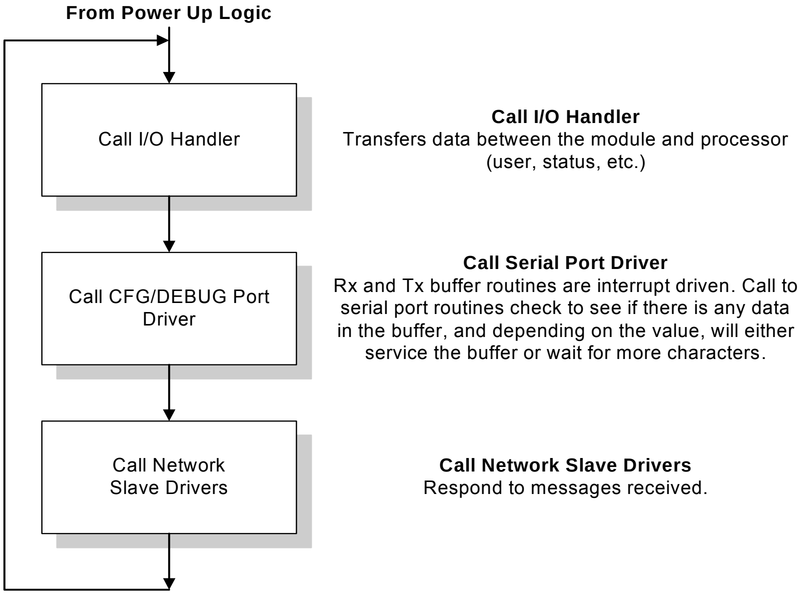
The following topics describe several concepts that are important for understanding the operation of the MVI71-101S module.

1. On power up the module begins performing the following logical functions:
   - **a** Initialize hardware components
   - **b** Initialize PLC backplane driver
   - **c** Test and Clear all RAM
   - **d** Initialize the serial communication ports
2. Reads configuration from Compact Flash Disk
3. Initialize Module Register space
4. Enable Slave Driver on selected ports

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

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.


From Power Up Logic
Call I/O Handler
Call CFG/DEBUG Port Driver
Call Network Slave Drivers
Call I/O Handler
Transfers data between the module and processor (user, status, etc.)
Call Serial Port Driver
Rx and Tx buffer routines are interrupt driven. Call to serial port routines check to see if there is any data in the buffer, and depending on the value, will either service the buffer or wait for more characters.
Call Network Slave Drivers
Respond to messages received.

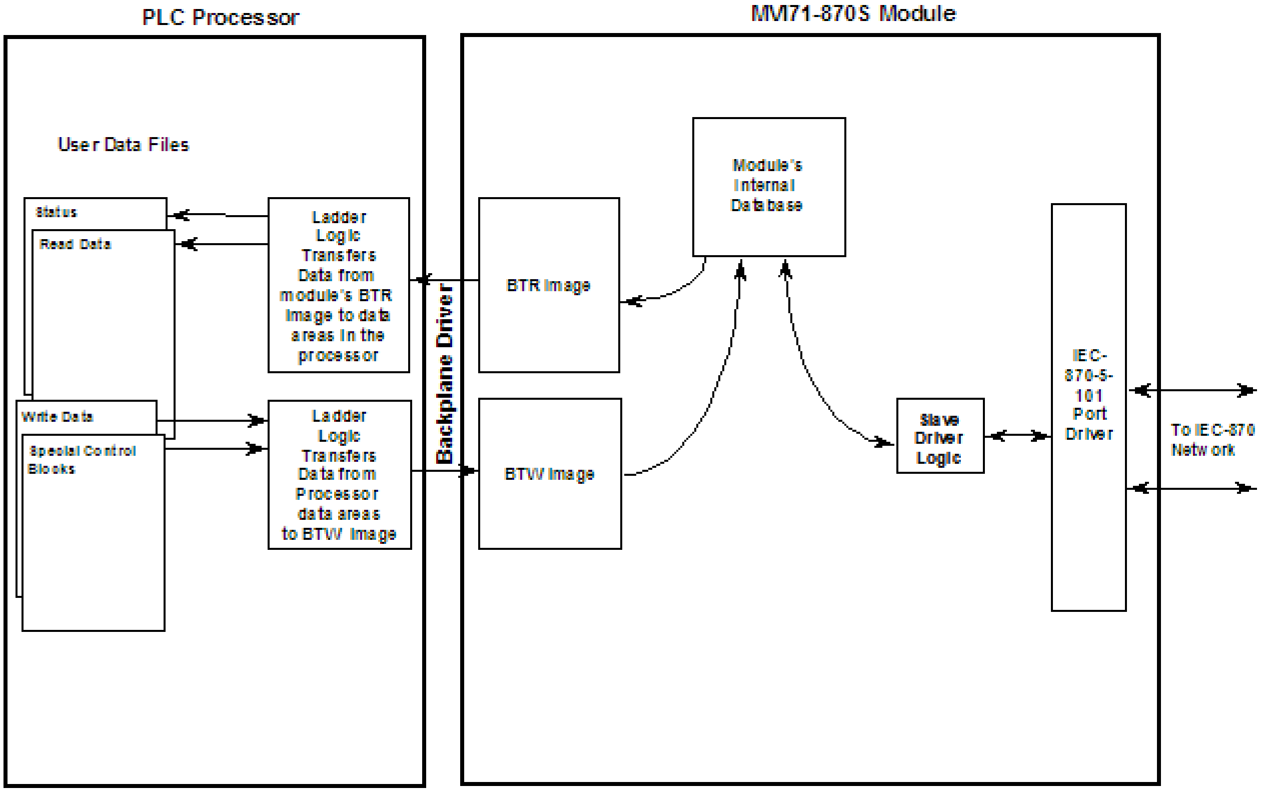
PLC Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (that is, Fault or PGM), the protocol ports can be shut down as prescribed in the user configuration. When the processor is returned to a running state, the module will resume communications on the network.

Block Transfer Backplane Data Transfer

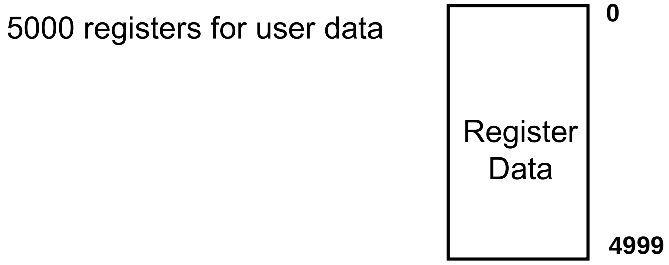
The MVI71-101S module communicates directly over the PLC backplane for the block transfer interface. Data is paged between the module and the PLC processor across the backplane using BTR and BTW operations. Data is transferred from the module to the processor using the BTR blocks, and data is transferred from the processor to the module using the BTW blocks.

The following diagram shows the data transfer method used to move data between the PLC processor, the MVI71-101S module, and the IEC 60870-5-101 network.


PLC Processor
MVI71-870S Module
User Data Files
Status
Read Data
Write Data
Special Control Blocks
Ladder Logic Transfers Data from module's BTR image to data areas in the processor
Ladder Logic Transfers Data from Processor data areas to BTW image
Backplane Driver
BTR Image
BTW Image
Module's Internal Database
Slave Driver Logic
IEC-870-5-101 Port Driver
To IEC-870 Network

As shown in the previous diagram, all data transferred between the module and the processor over the backplane is through the BTR and BTW blocks. Ladder logic must be written in the PLC processor to interface the block data with user data files. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Module's Internal Database Structure



Data contained in this database is paged through the block transfer interface by coordination of the PLC ladder logic and the MVI71-101S module's program. Up to 64 words of data can be transferred from the module to the processor at a time. Up to 64 words of data can be transferred from the processor to the module. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The block identification codes used by the module are listed in the following table:

Block Range	Descriptions
-1	Null Block
0	Null Block
1 to 84	Read or Write Data
1000 to 1083	Request Output Data from Processor
9958	Event Messages from Processor
9970	Set PLC Time Using Module's Time
9971	Set Module's Time Using PLC's Time
9998	Warm Boot Control Block
9999	Cold Boot Control Block

Each image has a defined structure depending on the data content and the function of the data transfer as defined in the following topics:

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999 and the status data. These data are transferred through read (input image) and write (output image) blocks. Refer to the **Module Configuration** and **Ladder Logic** sections for a description of the data objects used with the blocks and the ladder logic required. The structure and function of each block is discussed in the following topics.

Status Block

Status block information informs the processor of the activity and "health" of the module. When the side-connect interface is used, this data set is automatically placed in the application file at offset 200. The format of the block is shown below:

Offset	Description	Length
0	Read Block ID	1
1	Write Block ID	1
2	Program Scan Counter	1
3 to 4	Product Name	2
5 to 6	Product Version	2
7 to 8	Operating System	2
9 to 10	Run Number	2
11 to 17	Port 1 Error Status	7
18 to 24	Port 2 Error Status	7
25 to 30	Data Transfer Status	6

Offset	Description	Length
31	Port 1 Current Error/Index	1
32	Port 1 Last Error/Index	1
33	Port 2 Current Error/Index	1
34	Port 2 Last Error/Index	1
35 to 63	Not Used	20

When the BTR/BTW interface is utilized, the status data will be transferred to the module in the normal read data blocks. The data will reside in the block in the normal receive data area of the block. This data is placed in blocks that have a receive data length parameter of zero for a port.

Note: A block may contain status data for one port and receive data for another port.

Write Block

These blocks of data transfer information from the PLC processor to the module and source the input (monitored) data to be used by the remote master. The structure of the image used to transfer this data is shown in the following table:

Offset	Description	Length
0	Write Block ID	1
1 to 60	Write Data	60
61 to 63	Spare	3

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed. Each transfer can move up to 60 words (block offsets 1 to 60) of data.

5.2.3 Output Data Initialization

The module can be configured to have the output data in the module set to that stored in the PLC processor. This feature requires ladder logic support. When the module performs a restart operation, it will send a set of one or more special block requests to the processor. These blocks must be handled by the ladder logic and return the requested output data. The format of the blocks used by the module to request this data has the following format.

Offset	Description	Length
0	1000 to 1083	1
1	1000 to 1083	1
2 to 62	Spare	62

The value in the block at offset 0 represents the output data set required. The module will only request the data set in the user configuration. Each block requested must receive a response block from the processor. The format of the response block is as follows:

Offset	Description	Length
0	1000 to 1083	1
1 to 60	Output Data	60
61 to 63	Spare	3

The module places the received output data in the block into the module's database.

5.2.4 Sending Events

The module operates by sending data when the master sends Class 2 polls. When the slave has an event, it sends the event to the module queue. If the communication mode is unbalanced, the slave will notify the master that it has an event to be sent and the master's next poll should be a Class 1 poll. The slave then replies to this poll by sending the first event to the queue.

There are two ways of sending timestamp events to the module queue. The first is when a configured point changes its value in the module's database. This is actually the way the module would normally send events. That is, every time a configured point changes its value, the module sends this event to the queue.

The second method is using block 9958 (refer to the next section), in which case ladder logic should be used to send events in this manner. This method is limited by the number of events that can be sent to the master. An important concern about events is that the slave should have its clock synchronized with the master. The master should send a Sync. Command to the MVI module in order to synchronize both clocks. You can also synchronize the module and the processor clocks using blocks 9970 and 9971 (Refer to the next section).

5.2.5 Command Control Blocks

Command control blocks are special blocks used to control the module. The current version of the software supports five command control blocks: event messages, get module time, set module time, warm boot and cold boot.

Event Messages (Block 9958)

The Block 9958 identification code sends event messages from the processor to the module.

Block Format for Write with Block Transfer Interface

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9958 identifying the block type to the module.
1	Event Count	Number of events present in the block. This field can have a value from 1 to 10.
2 to 16	Event #1	Event data to add to event message queue.
17 to 31	Event #2	Event data to add to event message queue.
32 to 46	Event #3	Event data to add to event message queue.
47 to 61	Event #4	Event data to add to event message queue.
62 to 63	Spare	Not Used

Block Format for Write with Side-Connect Interface

Word Offset in Block	Data Field(s)	Description
50	Block ID	This field contains the value of 9958 identifying the block type to the module.
51	Event Count	Number of events present in the block. This field can have a value from 1 to 10.

Word Offset in Block	Data Field(s)	Description
52 to 66	Event #1	Event data to add to event message queue.
67 to 81	Event #2	Event data to add to event message queue.
82 to 96	Event #3	Event data to add to event message queue.
97 to 111	Event #4	Event data to add to event message queue.
112 to 126	Event #5	Event data to add to event message queue.
127 to 141	Event #6	Event data to add to event message queue.
142 to 156	Event #7	Event data to add to event message queue.
157 to 171	Event #8	Event data to add to event message queue.
172 to 186	Event #9	Event data to add to event message queue.
187 to 201	Event #10	Event data to add to event message queue.

The structure of each event record in the block is shown in the following table:

Word Offset in Event Record	Data Field(s)	Description
0	DB Index	This is the index for the point in the module's database. This corresponds to the order of point definition for the module data types. This is not the point address for the event.
1	ASDU	This is the ASDU data type for the event message. Valid entries are as follows: 1=single-point 3=double-point 5=step 9=normalized 11=scaled 15=integrated total
2	Qualifier	This is the qualifier code to be used with the event message. Refer to the IEC Protocol Specification for a full listing of valid qualifier codes for each ASDU type.
3	Year	This field contains the four-digit year to be used with the event.
4	Month	This field contains the month value for the event. Valid Values: 1 to 12
5	Day	This field contains the day value for the event. Valid Values: 1 to 31
6	Hour	This field contains the hour value for the event. Valid Values: 0 to 23
7	Minute	This field contains the minute value for the event. Valid Values: 0 to 59
8	Seconds and Milliseconds	This field contains the seconds and milliseconds value for the event. Valid Values: 0 to 59,999
9 to 14	Data	These words contain the data to be used with the event. For single- and double-point, step and measured value events, the first word is used. For integrated total events, the first two words are used.

Get Module Time (Block 9970)

The Block 9970 identification code requests the module's date and time. Use this data to set the PLC clock.

Block Format for Write

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 63	Not Used	Not Used

The module responds to a valid block 9970 request with a block containing the requested date and time. The following table shows the format for this block:

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block Read ID	This field contains the block identification code of 9970 to identify the block.
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year to be used with the new time value.
3	Month	This field contains the month value for the new time. Valid Values: 1 to 12
4	Day	This field contains the day value for the new time. Valid Values: 1 to 31
5	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
6	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59
7	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59
8	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999
9 to 63	Not Used	Not Used

Side Connect Block Format for Read

Word Offset in Block	Data Field(s)	Description
50	Done Flag	This word contains a value of 0 to indicate that the operation is complete.
51	Block Read ID	This field contains the block identification code of 9970 for the block`
52	Year	This field contains the four-digit year to be used with the new time value.
53	Month	This field contains the month value for the new time. Valid Values: 1 to 12
54	Day	This field contains the day value for the new time. Valid Values: 1 to 31
55	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
56	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59
57	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59
58	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999

Set Module Time (Block 9971)

The Block identification code of 9971 passes the clock time in the PLC to the module. The date and time provided sets the module's clock.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9971 to identify the block.
1	Year	This field contains the four-digit year to be used with the new time value.
2	Month	This field contains the month value for the new time. Valid Values: 1 to 12
3	Day	This field contains the day value for the new time. Valid Values: 1 to 31
4	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
5	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59
6	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59
7	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999
8 to 63	Not Used	Not Used

Block Format for Write with Side-Connect Interface

Word Offset in Block	Data Field(s)	Description
50	Block ID	This field contains the block identification code of 9971 for the block.
51	Year	This field contains the four-digit year to be used with the new time value.
52	Month	This field contains the month value for the new time. Valid Values: 1 to 12
53	Day	This field contains the day value for the new time. Valid Values: 1 to 31
54	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
55	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59
56	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59
57	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999

Warm Boot (Block 9998)

This block is sent from the PLC processor to the module (BTW block) when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made. This will force the module to read the new configuration information and to restart. The structure of the control block is shown in the following table:

Offset	Description	Length
0	9998	1
1 to 63	Spare	63

Because many of the parameters set in the configuration file require the program to restart, this function works the same as the cold-boot function. It will cause the module to exit and restart the program.

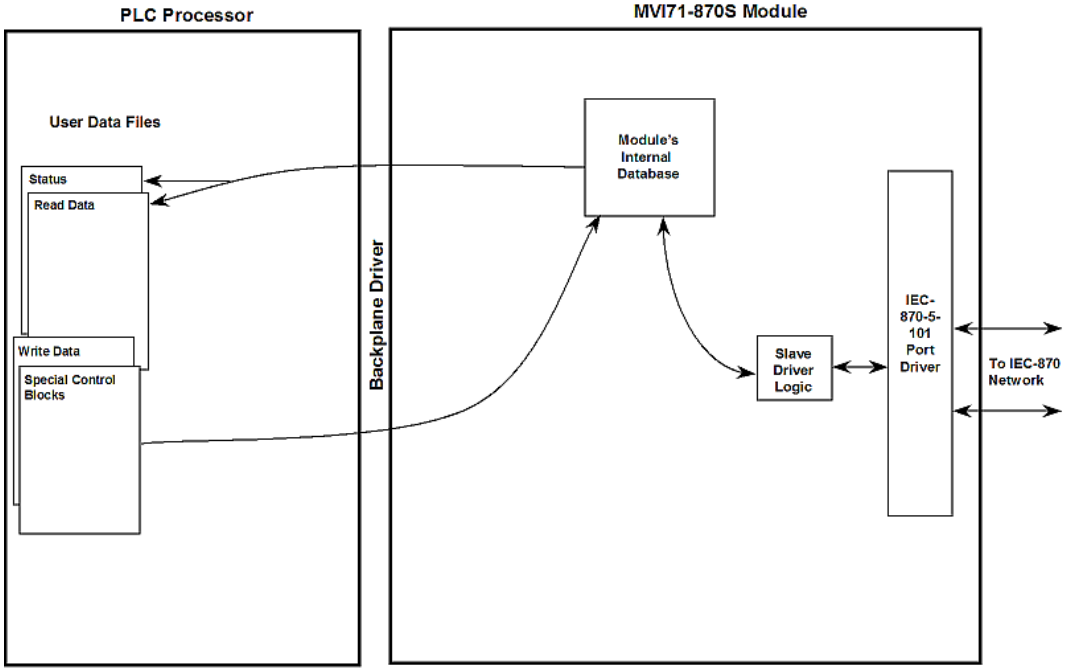
Cold Boot (Block 9999)

This block is sent from the PLC processor to the module (BTW block) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The structure of the control block is shown in the following table:

Offset	Description	Length
0	9999	1
1 to 63	Spare	63

Side-Connect Backplane Data Transfer

The side-connect interface is the simplest method to implement the module. No ladder logic is required for the interface because the driver handles data movement between the module and the processor automatically. The following illustration shows the data flow associated with this interface.



The configuration information for the module determines the size of the read and write data areas and the locations of these data sets in the module's internal database. Therefore, to use this interface, just set up the files required by the module. The following table lists the files required for the side-connect interface.

File Number	Example	Size	Description
Cfg File	N10	100	Control/Status File
Cfg File+1	N11	to 1000	Data transferred from the module to the processor
			Other files for read data
Cfg File+1+n	N12	to 1000	Data transferred from the processor to the module
Cfg File+1+n+m			Other files for write data

n is the number of read data files minus one. Each file contains up to 1000 words.

m is the number of write data files minus one. Each file contains up to 1000 words.

The number of read and write files are dependent on the modules configuration. The following provides two examples:

Example of 240 words of read and write data (cfg file=10)

Data Files	Description
N11:0 to 239	Read data
N12:0 to 239	Write data

Example of 2300 read and 3500 write data registers (cfg file=10)

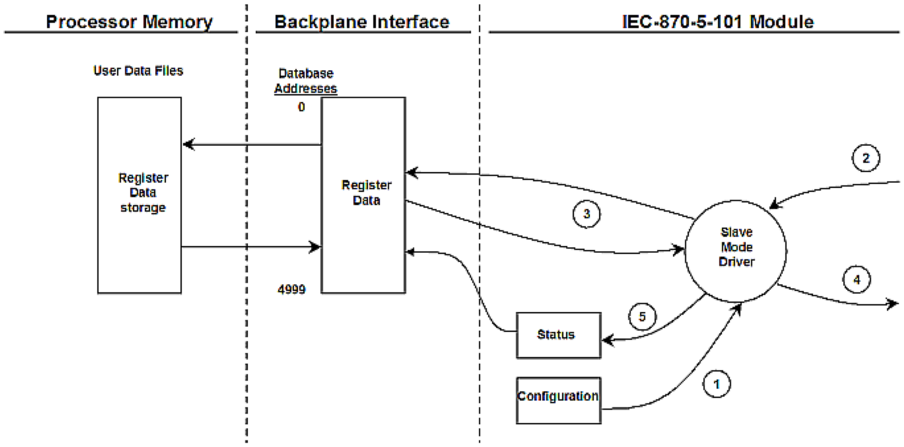
Data Files	Description
N11:0 to 999	Read data words 0 to 999
N12:0 to 999	Read data words 1000 to 1999
N13:0 to 299	Read data words 2000 to 2299
N14:0 to 999	Write data words 0 to 999
N15:0 to 999	Write data words 1000 to 1999
N16:0 to 999	Write data words 2000 to 2999
N17:0 to 499	Write data words 3000 to 3499

5.2.6 Data Flow Between MVI71-101S Module and PLC Processor

The following topics describe the flow of data between the two pieces of hardware (PLC processor and MVI71-101S module) and the IEC 60870-5-101 master unit. Each port on the module is configured to emulate a common slave device. The database used in the module is used for both ports and only one port can be utilized at one time.

Slave Driver

The Slave Driver allows the MVI71-101S module to respond to data read and write commands issued by a master unit on the telecontrol network. The following flow chart and the associated table, shows the flow of data into and out of the module.



Step	Description
1	The slave port driver receives the configuration information from the Compact Flash Disk in the module. This information configures the serial port and define the slave node characteristics.
2	A Host device issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the message, 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.

Review the **Module Set Up** section for a complete list of the parameters that must be defined for a slave port. The IEC 60870-5-101 Interoperability Document for the MVI71-101S Slave Module contains a listing of the protocol support supplied in the module.

5.2.7 Databases

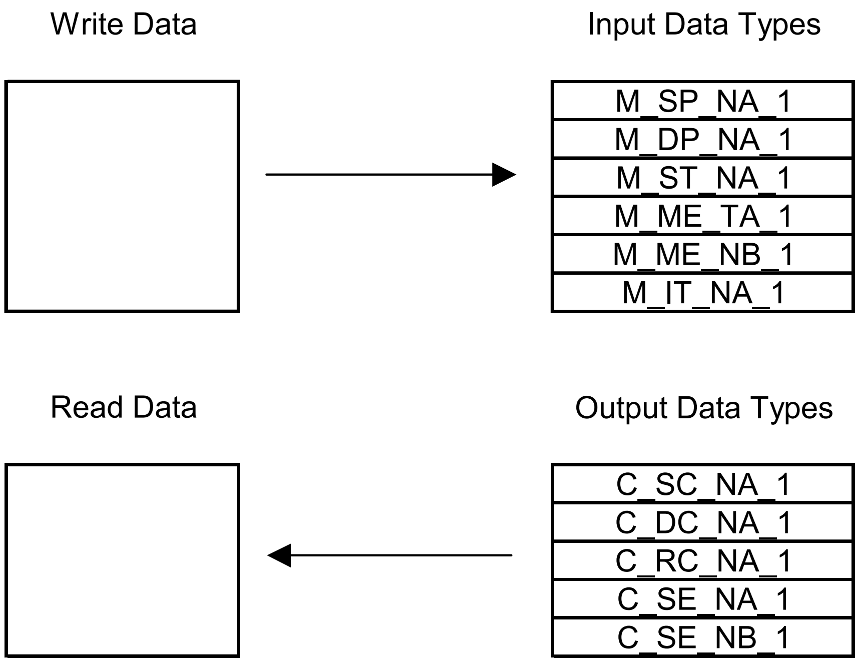
This section discusses the databases utilized by the module to support the IEC 60870-5-101 protocol. The user is responsible for defining the databases for their specific application. In the module's configuration file, the size of each database and definition of each point is established. Refer to the **Module Configuration** and **Configuration File** sections for a complete discussion of the configuration file. The following table summarizes the data types used in each of the 11 individual databases.

Type ID	Type	Description	Data Representation
1	M_SP_NA_1 (7.3.1.1)	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value with (7.2.6.1)0=Off and 1=On.
3	M_DP_NA_1 (7.3.1.3)	Monitored Dual-point Information: This data type stores a dual-point binary input value (that is, valve status). Associated time-tagged event information for this type are M_DP_TA_1 (4) and M_DP_TB_1 (31).	Dual-bit status (7.2.6.2) with 00b (0 decimal) = indeterminate or intermediate, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = indeterminate.
5	M_ST_NA_1 (7.3.1.5)	Monitored Step-point Information: This data type is used for step position of transformers or other step position information. The value for the position ranges from -64 to 63. Associated time-tagged event information for this type are M_ST_TA_1 (6) and M_ST_TB_1 (32).	Step data (7.2.6.5) is stored in a single character value with bits 0 to 6 (-64 to +63) representing the step position and bit 7 representing the following states: 0 = Equipment is not in transient state 1 = Equipment in transient state
9	M_ME_TA_1 (7.3.1.9)	Monitored Normalized Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TA_1 (10) and M_ME_TD_1 (34).	Normalized values (7.2.6.6) are stored in a (16-bit) word data area with a range of -1..+1-2^{-15}
11	M_ME_NB_1 (7.3.1.11)	Monitored Scaled Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TB_1 (12) and M_ME_TE_1 (35).	Scaled values (7.2.6.7) are stored in a (16-bit) word data area with a range of -215.. +215-1
15	M_IT_NA_1 (7.3.1.15)	Monitored Integrated Total-point Information: This data type stores meter or other count data. Associated time-tagged event information for this type are M_IT_TA_1 (15) and M_IT_TB_1 (37).	Binary counter data (7.2.6.9) is stored in a double-word (32-bit) value with a range of -2^{31}..+2^{31}-1.
45	C_SC_NA_1 (7.3.2.1)	Single-point Command: This command controls a single binary point such as a relay.	Single bit value (7.2.6.15) with 0 = Off and 1 = On
46	C_DC_NA_1 (7.3.2.2)	Double-point Command: This command controls a dual-point binary control device such as a trip/close relay.	Double (7.2.6.16)Command with 0 = Not permitted 1 = Off 2 = On 3 = Not permitted
47	C_RC_NA_1 (7.3.2.3)	Regulating Step Command: This command controls a stepping device such as a transformer.	Regulating Step Command with (7.2.6.17)0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted

Type ID	Type	Description	Data Representation
48	C_SE_NA_ 1 (7.3.2.4)	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values (7.2.6.6) are stored in a (16-bit) word data area with a range of -1..+1-2-^{15}
49	C_SE_NB_ 1 (7.3.2.5)	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values (7.2.6.7) are stored in a (16-bit) word data area with a range of -2^{15} .. +2^{15}-1

A key concept in interfacing the protocol with the PLC processor is the relationship between the databases and the data transfer operation between the module and the processor. The module transfers data to the processor in read blocks. These blocks should contain the information received from the controlling unit (output data) and includes the following data types: C_SC_NA_1, C_DC_NA_1, C_RC_NA_1, C_SE_NA_1 and C_SE_NB_1. This data is all sourced from the master unit and passed to the processor for control. Databases associated with these data types should place the points in the read data area of the module's database. The Read Register Start and Read Register Count parameters in the configuration file establish the portion of the database to transfer to the processor. Ladder logic extracts the data from the read data area and places it in the proper location for use by the processor when the block transfer interface is used.

Similarly, data to be monitored (input data) by the master unit (all databases associated with the "M_" data types) must all be placed in the write data area of the module. The Write Register Start and Write Register Count parameters establish the portion of the database to receive data from the processor. This data is sourced from the processor and passed through the module to the remote controlling unit. Ladder logic is required to place the data in the correct position in the write data area when the block transfer interface is utilized. The relationship between the data types and the read and write data areas is displayed in the following diagram:



The read and write areas can be placed anywhere in the module's 5000-word database area. Because each point is defined individually to the module, the data for a specific type need not be contiguous in the module's database. This means that the module error/status data area can be passed to the controlling station using the M_ME_NB_1 database. In the database definition for the type, establish a point for each status value to be monitored by the controlling station and set the module's database address for the point in the definition.

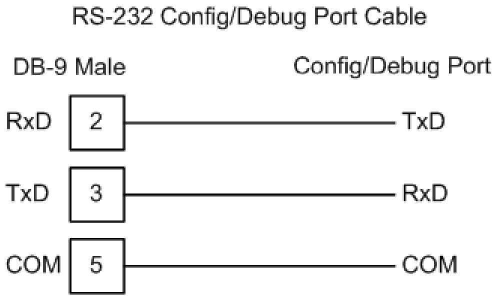
5.3 Cable Connections

The application ports on the MVI71-101S 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.

5.3.1 RS-232 Configuration/Debug Port

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

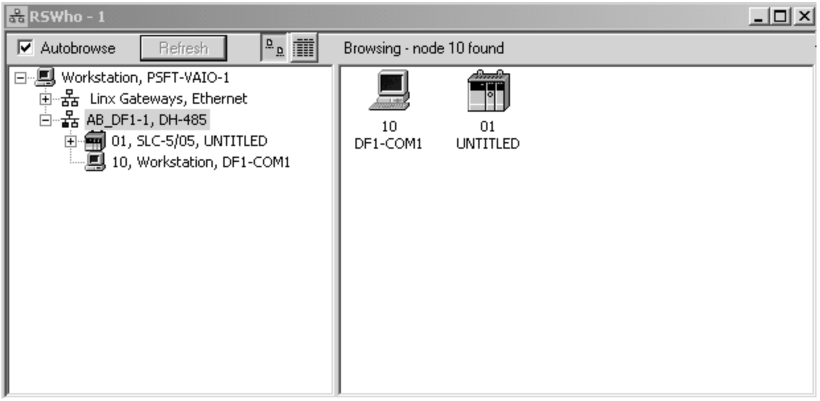


Disabling the RSLinx Driver for the Com Port on the PC

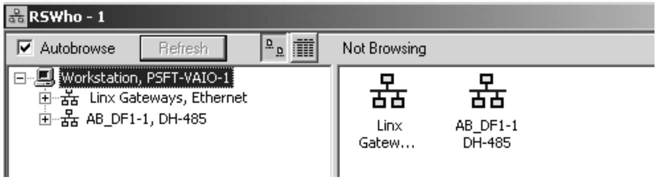
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable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

1 Open RSLinx and go to Communications>RSWho

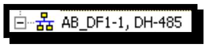
2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

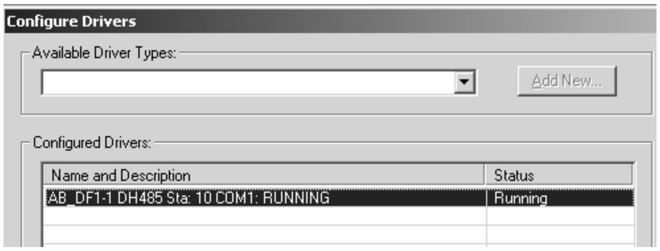


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



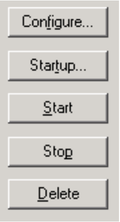
4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:

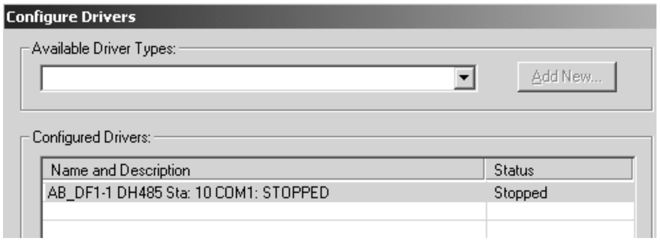


Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: RUNNING	Running

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



5 After you have stopped the driver you will see the following:



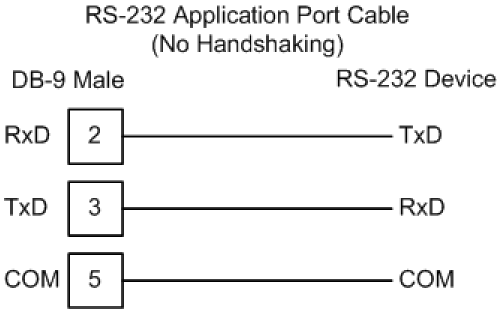
Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: STOPPED	Stopped

6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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

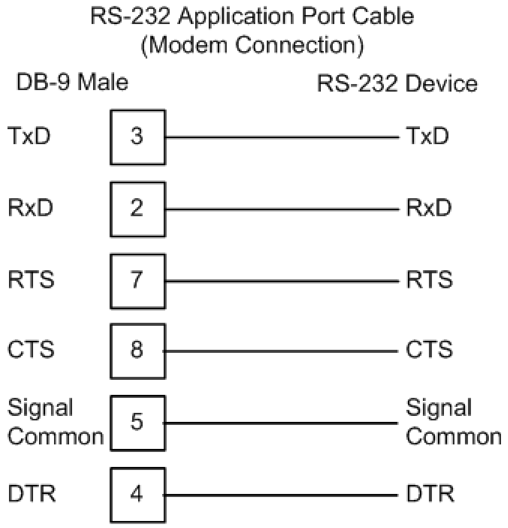
5.3.2 RS-232

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, the cable to connect to the port is as shown below:



RS-232: Modem Connection

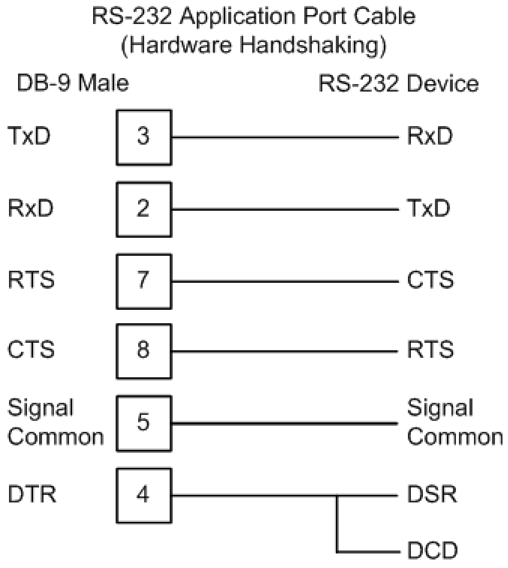
This type of connection is required between the module and a modem or other communication device.



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

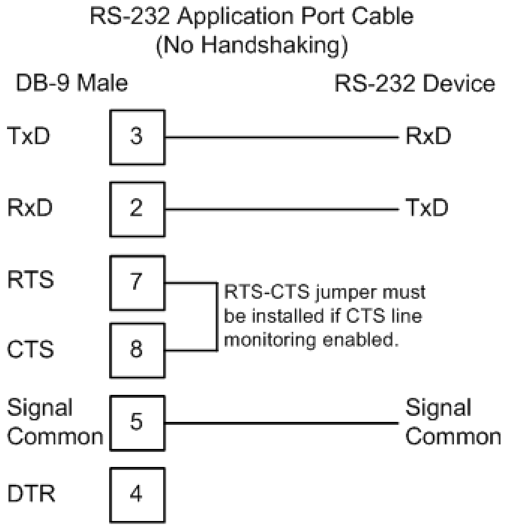
RS-232: Null Modem Connection (Hardware Handshaking)

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



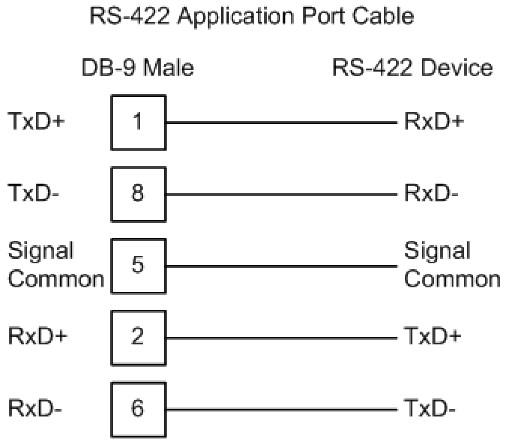
RS-232: Null Modem Connection (No Hardware Handshaking)

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


RS-232 Application Port Cable
(No Handshaking)
DB-9 Male
RS-232 Device
TxD
3
RxD
RxD
2
TxD
RTS
7
CTS
8
RTS-CTS jumper must be installed if CTS line monitoring enabled.
Signal Common
5
Signal Common
DTR
4

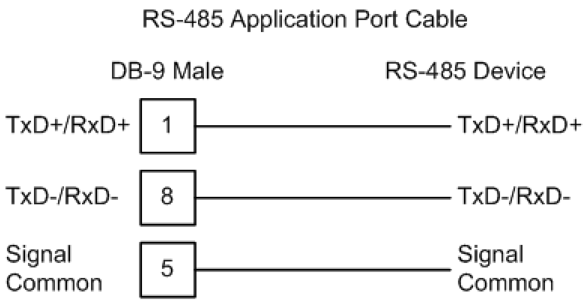
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.3 RS-422



5.3.4 RS-485

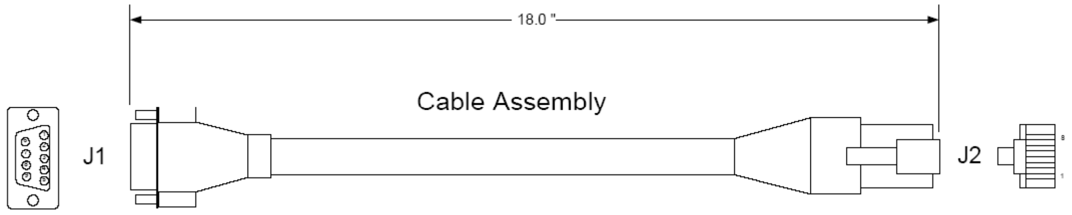
The RS-485 interface requires a single two or three wire cable. The Common connection is optional and dependent on the RS-485 network. The cable required for this interface is shown below:

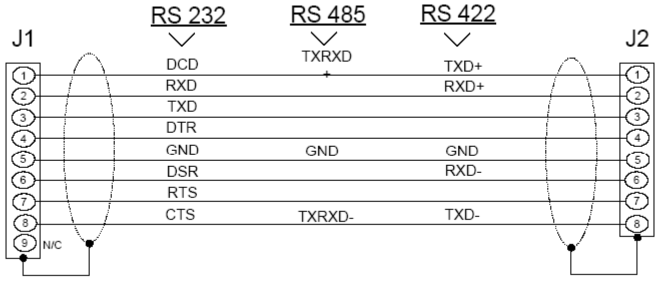


RS-485 and RS-422 Tip

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

5.3.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 MVI71-101S Configuration Form

This section contains a configuration form to be used when designing an application. Use of this form will aid in defining a successful application.

[Section]/Item	Value	Range	Description
[Backplane Configuration]			Backplane transfer parameters
Module Name:		0 to 80 characters	This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.
Read Register Start:		0 to 3999	This parameter specifies the starting register in the module where data will be transferred from the module to the processor. Valid range for this parameter is 0 to 3999.
Read Register Count:		0 to 4000	This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 4000.
Write Register Start:		0 to 3999	This parameter specifies the starting register in the module where the data transferred from the processor will be placed. Valid range for this parameter is 0 to 3999.
Write Register Count:		0 to 4000	This parameter specifies the number of registers to transfer from the processor to the module. Valid entry for this parameter is 0 to 4000.
Failure Flag Count:		0 to 65535	This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to 0, the communication ports will continue to operate under all conditions. If the value is set larger than 0 (1 to 65535), communications will cease if the specified number of failures occur.
Error Offset:		0 to 8899	This parameter specifies the register location in the module's database where module status data will be stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 8966, the data will be placed in the modules database.

[Section]/Item	Value	Range	Description
[IEC-870-5-101 Port 0]			Slave port communication and protocol parameters
Enabled:		Y or N	This parameter determines if the primary port will be utilized. If the port is not enabled (N), then the module will not use the port. If the port is enabled (Y), the module will emulate a IEC-870-101 slave device on the port.
Time DB Offset:		-1 or 0 to 3994	This parameter defines the location in the database where the time maintained for the IEC protocol is copied. This time is updated when ever a time synchronization command is received from the host and continually as the program runs.
Data link address:		0 to 65535	This parameter defines the data link address for the device emulated on the module. This address identifies the module on the network along with the common address of ASDU.
Data link address length:		0, 1 or 2	This parameter specifies the number of octets used for the data link address. This parameter must be set the same for all devices on the network. A value of 0 is only valid when the balanced mode is used. If unbalanced mode is used, a value of 1 or 2 must be used.
Common Address of ASDU:		0 to 65535	This parameter specifies the common address of the ASDU (section address) for access to data in the module. There is only one value entered for access to all data in the module.

[Section]/Item	Value	Range	Description
Common Address of ASDU Len:		1 or 2	This parameter specifies the number of octets used for the common address of ASDU. This parameter must be set the same for all devices on the network.
Inform. Object Address Len:		1, 2 or 3	This parameter specifies the number of octets used to define the address of an information object (point address).
Cyclic data transmission:		0 to 2^32	This parameter defines the number of milliseconds between cyclic updates. The range of values for this parameter permit update times of 1 millisecond to 49.7 days. If the parameter is set to 0, cyclic data reporting will be disabled.
Select/Operate Timeout:		0 to 2^32	This parameter sets the number of milliseconds after a select command is received in which to wait for a valid execute command. The range of values for this parameter permit times of 1 millisecond to 49.7 days. If the parameter is set to 0, the feature will be disabled.
Use ACTTERM with setpoint:		Y or N	This parameter determines if an ACTTERM will be sent. If the parameter is set to Y, then setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to N, ACTCON is the last response to a setpoint command.
Use ACTTERM with step:		Y or N	This parameter determines if an ACTTERM will be sent. If the parameter is set to Y, then step commands will issue an ACTTERM when the command is complete. If the parameter is set to N, ACTCON is the last response to a step command.
Single char ACT F0,1 or 3:		Y or N	If set to Y, a single character ACK (0xE5) will be sent instead of a fixed length ACK (secondary function code 0) in response to a primary link function code 0, 1 or 3 if there is no access demand for class 1 data (ACD=1). If set to N, the fixed length ACK will be sent.
Single char ACK C1 or C2		Y or N	If set to Y, a single character ACK (0xE5) will be sent instead of a fixed length NACK (secondary function code 9) when no response user data is available. If set to N, the fixed length NACK will be sent.
Maximum ASDU Resp Len		25 to 252	This parameter limits the maximum size of the ASDU portion of a response message. Most applications will use a value of 252.
Cause of Trans Octets		1 or 2	This parameter sets the COT length to 1 or 2. The second octet stores the originator address passed in the register. Spontaneous and cyclic data will always respond with the originator address set to 0.
Event Scan Delay:		0 to 65535	If set to 0, the feature will be disabled. If set from 1 to 65535, the parameter represents the number of milliseconds between event scanning. This parameter defines how often the program will scan for new events in the databases.
M_SP_NA Scan Events:		0 or 1	Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.
M_SP_NA Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.
M_DP_NA Scan Events:		0 or 1	Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.
M_DP_NA Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.
M_ST_NA Scan Events:		0 or 1	Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.
M_ST_NA Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.

[Section]/Item	Value	Range	Description
M_ME_NA Scan Events:		0 or 1	Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.
M_ME_NA Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.
M_ME_NB Scan Events:		0 or 1	Determines if events of this point type will be generated by the module. If 0, then events will not be generated. If 1, events will be scanned and generated on change.
M_ME_NB Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.
M_IT_NA Time Type:		0, 1 or 2	This parameters defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.
Use Balanced Mode:		Y or N	This parameter specifies if the port will support balanced mode (requires point-to-point connection). If set to N, the module will only function in unbalanced mode. If set to Y, the module will function in balanced mode.
Retry Count:		0 to 255	In balanced mode, this parameter specifies the number of retries (0 to 255) if a response is not received. In unbalanced mode, this parameter is ignored.
Response Timeout:		0 to 65535	This parameter specifies the minimum number of milliseconds to wait for a response to a primary message. Do not set this parameter too small or timeout conditions may prevent successful data transmission. If the timeout is recognized, the message will be retransmitted up to the number of times specified in the Retry Count parameter. This parameter is only used in balance mode.
Baud Rate:		300 to 38400	This parameter specifies the baud rate for the primary port on the module. Baud rates from 300 to 38400 are supported on the module.
Parity:		N, O, E, M or S	This parameter specifies the parity for this port using the following code definitions: N=none, O=odd, E=even, M=mark and S=space.
RTS On:		0 to 65535	This parameter specifies the number of milliseconds to delay after asserting the RTS line before data will be sent from the primary port.
RTS Off:		0 to 65535	This parameter specifies the number of milliseconds to delay after sending the data frame before the RTS line is dropped.
Minimum Delay:		0 to 65535	This parameter defines the minimum number of milliseconds to wait before a response is sent from the unit.
Receive Timeout:		0 to 65535	This parameter specifies the minimum number of milliseconds to wait after the first byte of a frame is received before a timeout condition is set. Be careful not to set this parameter too small. If the timeout condition is set, all bytes in the frame received will be discarded.
Backup Port Enabled		0 or 1	0 = Disable port, 1 = Enable port for protocol
Backup Port Baud Rate:		300 to 38400	This parameter specifies the baud rate for the primary port on the module. Baud rates from 300 to 38400 are supported on the module.
Backup Port Parity:		N, O, E, M or S	This parameter specifies the parity for this port using the following code definitions: N=none, O=odd, E=even, M=mark and S=space.
Backup Port RTS On:		0 to 65535	This parameter specifies the number of milliseconds to delay after asserting the RTS line before data will be sent from the primary port.
Backup Port RTS Off:		0 to 65535	This parameter specifies the number of milliseconds to delay after sending the data frame before the RTS line is dropped.
Backup Port Min Delay		0 to 65535	This parameter defines the minimum number of milliseconds to wait before a response is sent from the unit.

[Section]/Item	Value	Range	Description
[IEC-870-5-101 Database]			Size definitions for each point database
Short Pulse Time:		0 to 2^31-1	This parameter defines the number of milliseconds to be associated with a short pulse command. The valid range of numbers for this parameter are 0 to 2,147,483,647.
Long Pulse Time:		0 to 2^31-1	This parameter defines the number of milliseconds to be associated with a long pulse command. The valid range of numbers for this parameter are 0 to 2,147,483,647.
M_SP_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored single-point database.
M_DP_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored dual-point database.
M_ST_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored step-point database.
M_ME_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored normalized-point database.
M_ME_NB point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored scaled-point database.
M_IT_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in monitored counter-point database.
C_SC_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in command single-point database.
C_DC_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in command dual-point database.
C_RC_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in command step-point database.
C_SE_NA point count:		0 to 65535	This parameter specifies the number of point values assigned in command normalized-point database.
C_SE_NB point count:		0 to 65535	This parameter specifies the number of point values assigned in command scaled-point database.
C_SE_NC point count		0 to 65535	This parameter specifies the number of point values assigned in the command short float database
M_ME_NA Parameter Offset:		Application dependent	This parameter specifies the IOA offset to the parameter data for the normalized parameter data. The value entered is added to the Information Object Address for the associated point to compute the parameter IOA address.
M_ME_NB Parameter Offset:		Application dependent	This parameter specifies the IOA offset to the parameter data for the scaled parameter data. The value entered is added to the Information Object Address for the associated point to compute the parameter IOA address.

[Section]/Item		**Description**
[M_SP_NA_1]		Definition of monitored single-point database
# Point # DB Address Group(s)		
# ------- ---------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[M_DP_NA_1]	[redacted]	Definition of monitored dual-point database
# Point # DB Address Group(s)		
# ------- ---------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[M_ST_NA_1]	[redacted]	Definition of monitored step-point database
# Point # DB Address Group(s)		
# ------- ---------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[M_ME_NA_1]	[redacted]	Definition of monitored normalized-point database
# Default		
# Point # DB Address Group(s) Deadband		
# ------- ---------- -------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[M_ME_NB_1]	[redacted]	Definition of monitored scaled-point database
# Default		
# Point # DB Address Group(s) Deadband		
# ------- ---------- -------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[M_IT_NA_1]	[redacted]	Definition of monitored integrated total database
# Point # DB Address Group(s)		
# ------- ---------- --------		
START		
END		

[Section]/Item	[redacted]	**Description**
[C_SC_NA_1]	[redacted]	Definition of command single-point database
# Monitor Monitor Require		
# Point # DB Address Point # DB Addr Select		
# ------- ---------- ------- ------- -------		
START		
END		

[Section]/Item	[redacted]	**Description**
[C_DC_NA_1]	[redacted]	Definition of command dual-point database

```
#                  Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START

END
```

[Section]/Item	[redacted]	**Description**
[C_RC_NA_1]	[redacted]	Definition of command step-point database

```
#                  Monitor  Monitor
# Point #   DB Address   Point #  DB Addr
# -------   ----------   -------  -------
START

END
```

[Section]/Item	[redacted]	**Description**
[C_SE_NA_1]	[redacted]	Definition of command normalized-point database

```
#                  Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START

END
```

[Section]/Item	[redacted]	**Description**
[C_SE_NB_1]	[redacted]	Definition of command scaled-point database

```
#                  Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START

END
```

[Section]/Item	[redacted]	**Description**
[C_SE_NC_1]	[redacted]	Definition of command short float database

```
#                  Monitor  Monitor  Require
# Point #   DB Address   Point #  DB Addr   Select
# -------   ----------   -------  -------  -------
START

END
```

5.5 MVI71-101S Status Data Definition

This section contains a description of the members present in the status data area. This data is automatically transferred to the PLC when the side-connect interface is used. When the block transfer interface is employed, this data can be mapped into the read data area transferred to the processor. The format of the status data block is shown in the following table:

Offset	Parameter	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
14	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
15	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
16	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
17	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
18	RX Frames	This word contains the total number of frames received by the module.
19	TX Frames	This word contains the total number of frames transmitted by the module.

5.6 MVI71-101S Database Design Forms

This section contains a set of forms that can be used to design the databases required by the module.

5.6.1 Form for use with the data types M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 and M_IT_NA_1

Point Number	Database Address	Group Assignment

5.6.2 Form to be used with data types M_ME_NA_1 and M_ME_NB_1

Point Number	Database Address	Group Assignment	Default Deadband

5.6.3 Forms to be used with all command data types except C_RC_NA_1

Point Number	Database Address	Monitor Point #	Monitor Database Address	Require Select

5.6.4 Form to be used with C_RC_NA_1 data type

Point Number	Database Address	Monitor Point #	Monitor Database Address

5.7 MVI71-101S Error Status Table

This section contains a listing of the MVI71-101S module's status data area. This file is located at the MVI71-101S database starting at address 4000. You may also configure an additional area using the "Error Offset" parameter.

Offset	Parameter	Description
4000	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
4001 to 4002	Product Name	This two-word data area contains the text values representing the product name.
4003 to 4004	Revision	This two-word data area contains the text values for the revision number.
4005 to 4006	Op Sys #	This two-word data area contains the text values for the operating system number.
4007 to 4008	Run Number	This two-word data area contains the text values for the run number.
4009	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
4010	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
4011	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
4012	Error Blk Cnt	This word contains the total number of block transfer errors.
4013	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
4014	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
4015	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
4016	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
4017	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
4018	RX Frames	This word contains the total number of frames received by the module.
4019	TX Frames	This word contains the total number of frames transmitted by the module.
4020	MSP Event Buffer	This parameter shows the number of events available in the event buffer for M_SP_NA points.
4021	MDP Event Buffer	This parameter shows the number of events available in the event buffer for M_DP_NA points
4022	MST Event Buffer	This parameter shows the number of events available in the event buffer for M_ST_NA points
4023	MMENA Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NA points

Offset	Parameter	Description
4024	MMENB Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NB points
4025	MMENC Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NC points.
4026	MIT Event Buffer	This parameter shows the number of events available in the event buffer for M_IT_NA points

5.8 IEC 60870-5-101 Slave Interoperability Document

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the COMMON ADDRESS of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment originating from different manufacturers it is necessary that all partners agree on the selected parameters.

Note: In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

□ Function or ASDU is not used

☒ Function or ASDU is used as standardized (default)

A black check box indicates that the option cannot be selected in this companion standard.

Network Configuration

(Network-specific parameter)

☒ Point-to-point ☒ Multipoint-party line

☒ Multiple point-to-point ☒ Multipoint-star

Physical Layer

(Network-specific parameter)

Transmission speed (control direction)

Unbalanced interchange Unbalanced interchange Balanced interchange

circuit V.24/V.28 circuit V.24/V.28 circuit X.24/X.27

Standard Recommended if >1 200 bit/s

☐ 100 bit/s ☒ 2400 bit/s ☒ 2400 bit/s ☐
56000 bit/s

☐ 200 bit/s ☒ 4800 bit/s ☒ 4800 bit/s ☐
64000 bit/s

☒ 300 bit/s ☒ 9600 bit/s ☒ 9600 bit/s

☒ 600 bit/s ☒ 19200 bit/s

☒ 1200 bit/s ☒ 38400 bit/s

Transmission speed (monitor direction)

Unbalanced interchange Unbalanced interchange Balanced interchange

circuit V.24/V.28 circuit V.24/V.28 circuit X.24/X.27

Standard Recommended if >1 200 bit/s

☐ 100 bit/s ☒ 2400 bit/s ☒ 2400 bit/s ☐
56000 bit/s

☐ 200 bit/s ☒ 4800 bit/s ☒ 4800 bit/s ☐
64000 bit/s

☒ 300 bit/s ☒ 9600 bit/s ☒ 9600 bit/s

☒ 600 bit/s ☒ 19200 bit/s

☒ 1200 bit/s ☒ 38400 bit/s

Link Layer

(Network-specific parameter)

Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.

Link transmission procedure Address field of link

☒ Balanced transmission ☒ Not present (balanced transmission only)

☒ Unbalanced transmission ☒ One octet

☒ Two octets

☒ Structured

Frame length

☒ Unstructured

255 Maximum length L (number of octets) See Note 3.

Application Layer

Transmission mode for application data

Mode 1 (Least significant octet first), as defined in clause 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common address of ASDU

(System-specific parameter)

☒ One octet ☒ Two octets

Information object address

(System-specific parameter)

One octet ☒ structured

☒ Two octets ☒ unstructured

☒ Three octets

Cause of transmission

(System-specific parameter)

☒ One octet S Two octets (with originator address)

Selection of standard ASDUs

Process information in monitor direction

(Station-specific parameter)

☒	<1>	:= Single-point information	M_SP_NA_1
☒	<2>	:= Single-point information with time tag	M_SP_TA_1
☒	<3>	:= Double-point information	M_DP_NA_1
☒	<4>	:= Double-point information with time tag	M_DP_TA_1
☒	<5>	:= Step position information	M_ST_NA_1
☒	<6>	:= Step position information with time tag	M_ST_TA_1
□	<7>	:= Bitstring of 32 bit	M_BO_NA_1
□	<8>	:= Bitstring of 32 bit with time tag	M_BO_TA_1
☒	<9>	:= Measured value, normalized value	M_ME_NA_1
☒	<10>	:= Measured value, normalized value with time tag	M_ME_TA_1
☒	<11>	:= Measured value, scaled value	M_ME_NB_1
☒	<12>	:= Measured value, scaled value with time tag	M_ME_TB_1

☒ <13> := Measured value, short floating point value
M_ME_NC_I

☒ <14> := Measured value, short floating point value with time tag
M_ME_TC_1

☒ <15> := Integrated totals M_IT_NA_1

☒ <16> := Integrated totals with time tag M_IT_TA_1

☐ <17> := Event of protection equipment with time tag M_EP_TA_1

☐ <18> := Packed start events of protection equipment with time tag
M_EP_TB_1

☐ <19> := Packed output circuit information of protection equipment with
time tag M_EP_TC_1

☐ <20> := Packed single-point information with status change detection
M_PS_NA_1

☐ <21> := Measured value, normalized value without quality description
M_ME_ND_1

☒ <30> := Single-point information with time tag CP56Time2a
M_SP_TB_1

☒ <31> := Double-point information with time tag CP56Time2A
M_DP_TB_1

☒⑨<32> := Step position information with time tag CP56Time2A
M_ST_TB_1

☐ <33> := Bitstring of 32 bit with time tag CP56Time2A M_BO_TB_1

☒ <34> := Measured value, normalized value with time tag CP56Time2A
M_ME_TD_1

☒ <35> := Measured value, scaled value with time tag CP56Time2A
M_ME_TE_1

☐ <36> := Measured value, short floating point value with time tag
CP56Time2A M_ME_TF_1

☒ <37> := Integrated totals with time tag CP56Time2A M_IT_TB_1

☐ <38> := Event of protection equipment with time tag CP56Time2A
M_EP_TD_1

☐ <39> := Packed start events of protection equipment with time tag
CP56time2A M_EP_TE_1

☐ <40> := Packed output circuit information of protection equipment with
time tag CP56Time2a M_EP_TF_1

Process information in control direction

(Station-specific parameter)

☒ <45> := Single command C_SC_NA_1

☒ <46> := Double command C_DC_NA_1

☒ <47> := Regulating step command C_RC_NA_1

☒ <48> := Set point command, normalized value C_SE_NA_1

☒ <49> := Set point command, scaled value C_SE_NB_1

☒ <50> := Set point command, short floating point value C_SE_NC_1

□ <51> := Bitstring of 32 bit C_BO_NA_1

System information in monitor direction

(Station-specific parameter)

☒ <70> := End of initialization M_EI_NA_1

System information in control direction

(Station-specific parameter)

☒ <100> := Interrogation command C_IC_NA_1

☒ <101> := Counter interrogation command C_CI_NA_1

☒ <102> := Read command C_RD_NA_1

☒ <103> := Clock synchronization command C_CS_NA_1

☒ <104> := Test command C_TS_NB_1

☒ <105> := Reset process command C_RP_NC_1

☒ <106> := Delay acquisition command Note 1
C_CD_NA_1

Parameter in control direction

(Station-specific parameter)

☒ <110> := Parameter of measured value, normalized value P_ME_NA_1

☒ <111> := Parameter of measured value, scaled value P_ME_NB_1

□ <112> := Parameter of measured value, short floating point value
P_ME_NC_1

□ <113> := Parameter activation P_AC_NA_1

File transfer

(Station-specific parameter)

□ <120> := File ready F_FR_NA_1

□ <121> := Section ready F_SR_NA_1

□ <122> := Call directory, select file, call file, call section F_SC_NA_1

□ <123> := Last section, last segment F_LS_NA_1

□ <124> := Ack file, ack section F_AF_NA_1

□ <125> := Segment F_SG_NA_1

□ <126> := Directory F_DR_TA_1

Basic Application Functions

Station initialization

(Station-specific parameter)

☒ Remote initialization

General Interrogation

(System- or station-specific parameter)

☒ global

☒ group 1

☒ group 2

☒ group 3

☒ group 4

☒ group 5

☒ group 6

☒ group 7

☒ group 8

☒ group 9

☒ group 10

☒ group 11

☒ group 12

☒ group 13

☒ group I4

☒ group 15

☒ group 16

Addresses per group have to be defined

Clock synchronization

(Station-specific parameter)

☒ Clock synchronization

Command transmission

(Object-specific parameter)

☒ Direct command transmission ☒ Select and execute command

☒ Direct set point command transmission ☒ Select and execute set point command

☒ C_SE_ACTTERM used [Note 2]

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation) [Note 4]

☒ Long pulse duration (duration determined by a system parameter in the outstation) [Note 4]

☒ Persistent output

Transmission of Integrated totals

(Station- or object-specific parameter)

☒ Counter request ☒ General request counter

☒ Counter freeze without reset ☒ Request counter group 1

□ Counter freeze with reset ☒ Request counter group 2

□ Counter reset ☒ Request counter group 3

☒ Request counter group 4

Addresses per group have to be defined

Parameter loading

(Object-specific parameter)

☒ Threshold value

□ Smoothing factor

☒ Low limit for transmission of measured value

☒ High limit for transmission of measured value

Parameter activation

(Object-specific parameter)

□ Act/deact of persistent cyclic or periodic transmission of the addressed object

File transfer

(Station-specific parameter)

□ File transfer in monitor direction

□ File transfer in control direction

Note 1: Delay acquisition command supports Load Delay only.

Note 2: C_SE_ACTTERM may be enabled or disabled at time of installation.

Note 3: The ASDU length can be configured by the user (between 25 and 252).

Note 4: Only applies to C_SC_NA_1 and C_DC_NA_1 commands.

Note 5: The low limits and high limit values are calculated based on the Deadband values as follows:

Low Limit: Last reported event value - threshold

High Limit: Last reported event value + threshold

6 Support, Service & Warranty

In This Chapter

Be sure and read the full Warranty that can be found on our web site at www.prosoft-technology.com *for details and other terms and conditions. The content in this summary is subject to change without notice. The content is current at date of publication.*

ProSoft Technology, Inc. strives to provide meaningful support to its customers. Should any questions or problems arise, please feel free to contact us at:

Internet	Web Site: http://www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com

Those of us at ProSoft Technology, Inc. want to provide the best and quickest support possible, so before calling please have the following information available. You may wish to fax this information to us prior to calling.

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

In the case of hardware, we will also need the following information:

1. Module configuration and contents of file
2. Module Operation
3. Configuration/Debug status information
4. LED patterns
5. Information about the processor and user data files as viewed through the development software and LED patterns on the processor
6. Details about the networked devices interfaced, if any

For technical support calls within the United States, an after-hours answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer your questions.

6.1 How to Contact Us: Sales and Support

All ProSoft Technology Products are backed with full technical support. Contact our worldwide Technical Support team and Customer Service representatives directly by phone or email:

USA / Latin America (excluding Brasil) (Office in California)

+1(661) 716-5100
+1(661) 716-5101 (Fax)
1675 Chester Avenue, 4th Floor
Bakersfield, California 93301
U.S.A.
+1.661.716.5100, support@prosoft-technology.com
Languages spoken include: English, Spanish

Asia Pacific Sales (office in Malaysia)

+603.7724.2080
+603.7724.2090 (Fax)
C210, Damansara Intan,
1 Jalan SS20/27, 47400 Petaling Jaya
Selangor, Malaysia
+603.7724.2080, asiapc@prosoft-technology.com
Languages spoken include: Chinese, Japanese, English

Asia Pacific Support (office in China)

+86.21.64518356 x 8011
+86.21.64756957 (Fax)
4/F, No. 16 Hongcao Road
Shanghai, China 200233
China
+86.21.64518356 x 8011, zhang@prosoft-technology.com
Languages spoken include: Chinese, English

Europe / Middle East / Africa (office in Toulouse, France)

+33 (0) 5.34.36.87.20
+33 (0) 5.61.78.40.52 (Fax)
Zone d'activité de Font Grasse
17, rue des Briquetiers
F-31700 Blagnac
France
+33 (0) 5.34.36.87.20. support. EMEA@prosoft-technology.com
Languages spoken include: French, English

Brasil (office in Sao Paulo)

+55-11-5084-5178
+55-11-5083-3776 (Fax)
Rua Vergueiro, 2949 - sala 182 - Edifício Vergueiro Work Center
Vila Mariana - São Paulo
Cep: 04101-300 - Brasil
+55-11-5084-5178, eduardo@prosoft-technology.com
Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

The following RMA Policies and Conditions apply to any returned product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see Section C below entitled "Limited Warranty". In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

6.2.1 All Product Returns

1 In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.

2 In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above in Section A. A Technical Support Engineer will request several tests in an attempt to isolate the problem. If after these tests are completed, the Product is found to be the source of the problem, ProSoft will issue an RMA.

3 All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.

4 Out of warranty returns are not allowed on RadioLinx accessories such as antennas, cables, and brackets.

The following policy applies for Non-Warranty Credit Returns:

A 10% Restocking Fee if Factory Seal is *not* broken

B 20% Restocking Fee if Factory Seal is broken

ProSoft retains the right, in its absolute and sole discretion, to reject any non-warranty returns for credit if the return is not requested within three (3) months after shipment of the Product to Customer, if the Customer fails to comply with ProSoft's shipping instructions, or if the Customer fails to return the Product to ProSoft within six (6) months after Product was originally shipped.

6.3 Procedures for Return of Units Under Warranty

1 A Technical Support Engineer must pre-approve all product returns.

2 Module is repaired or replaced after a Return Material Authorization Number is entered and a replacement order is generated.

3 Credit for the warranted item is issued within 10 business days after receipt of product and evaluation of the defect has been performed by ProSoft. The credit will only be issued provided the product is returned with a valid Return Material Authorization Number and in accordance with ProSoft's shipping instructions.

a) If no defect is found, a credit is issued.

b) If a defect is found and is determined to be customer generated or if the defect is otherwise not covered by ProSoft's Warranty, or if the module is not repairable, a credit is not issued and payment of the replacement module is due.

6.4 Procedures for Return of Units Out of Warranty

1 Customer sends unit in for evaluation.

2 If no defect is found, Customer will be charged the equivalent of US $100 plus shipping, duties and taxes that may apply. A new Purchase Order will be required for this evaluation fee.

If the unit is repaired the charge to the Customer will be 30%* of the list price plus any shipping, duties and taxes that may apply. A new Purchase Order will be required for a product repair.

3 For an immediate exchange, a new module may be purchased and sent to Customer while repair work is being performed. Credit for purchase of the new module will be issued when the new module is returned in accordance with ProSoft's shipping instructions and subject to ProSoft's policy on non-warranty returns. This is in addition to charges for repair of the old module and any associated charges to Customer.

4 If, upon contacting ProSoft Customer Service, the Customer is informed that unit is believed to be unrepairable, the Customer may choose to send unit in for evaluation to determine if the repair can be made. Customer will pay shipping, duties and taxes that may apply. If unit cannot be repaired, the Customer may purchase a new unit.

6.4.1 Un-repairable Units

- 3150-All
- 3750
- 3600-All
- 3700
- 3170-All
- 3250
- 1560 can be repaired, if defect is the power supply
- 1550 can be repaired, if defect is the power supply
- 3350
- 3300
- 1500-All

*** 30% of list price is an estimated repair cost only. The actual cost of repairs will be determined when the module is received by ProSoft and evaluated for needed repairs.**

6.4.2 Purchasing Warranty Extension

As detailed below in ProSoft's Warranty, the standard Warranty Period is one year (or in the case of RadioLinx modules, three years) from the date of delivery. The Warranty Period may be extended for an additional charge, as follows:

- Additional 1 year = 10% of list price
- Additional 2 years = 20% of list price
- Additional 3 years = 30% of list price

6.5 LIMITED WARRANTY

This Limited Warranty ("Warranty") governs all sales of hardware, software and other products (collectively, "Product") manufactured and/or offered for sale by ProSoft, and all related services provided by ProSoft, including maintenance, repair, warranty exchange, and service programs (collectively, "Services"). By purchasing or using the Product or Services, the individual or entity purchasing or using the Product or Services ("Customer") agrees to all of the terms and provisions (collectively, the "Terms") of this Limited Warranty. All sales of software or other intellectual property are, in addition, subject to any license agreement accompanying such software or other intellectual property.

6.5.1 What Is Covered By This Warranty

A *Warranty On New Products*: ProSoft warrants, to the original purchaser only, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved, and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires one year (or in the case of RadioLinx modules, three years) from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 15 months (or in the case of RadioLinx modules, 39 months) from the date of delivery. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft. If ProSoft is unable to repair the Product to conform to this Warranty after a reasonable number of attempts, ProSoft will provide, at its option, one of the following: a replacement product, a full refund of the purchase price or a credit in the amount of the purchase price. All replaced product and parts become the property of ProSoft. These remedies are the Customer's only remedies for breach of warranty.

B *Warranty On Services*: Material and labor used by ProSoft to repair a verified malfunction or defect are warranted on the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

C The Warranty Period for RadioLinx accessories (such as antennas, cables, brackets, etc.) are the same as for RadioLinx modules, that is, three years from the date of shipment.

6.5.2 What Is Not Covered By This Warranty

A ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.

B With the exception of RadioLinx accessories referenced in paragraph 1(c) this Warranty does not cover any product, components, or parts not manufactured by ProSoft.

C This Warranty also does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3 programming languages, or "C") not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges outside of the limits indicated on the product specifications; or (viii) disasters such as fire, flood, earthquake, wind or lightning.

D The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guides included with your original product purchased by you from ProSoft, contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.5.3 DISCLAIMER REGARDING HIGH RISK ACTIVITIES

PRODUCT MANUFACTURED OR SUPPLIED BY PROSOFT IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE (INCLUDING, WITHOUT LIMITATION, THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OF COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES OR WEAPONS SYSTEMS), IN WHICH THE FAILURE OF THE PRODUCT COULD LEAD DIRECTLY OR INDIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE (COLLECTIVELY, "HIGH RISK ACTIVITIES"). PROSOFT SPECIFICALLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

6.5.4 DISCLAIMER OF ALL OTHER WARRANTIES

THE WARRANTIES SET FORTH IN PARAGRAPH 1 ABOVE ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

6.5.5 LIMITATION OF REMEDIES**

IN NO EVENT WILL PROSOFT (OR ITS DEALER) BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES BASED ON BREACH OF WARRANTY, BREACH OF CONTRACT, NEGLIGENCE, STRICT TORT, OR ANY OTHER LEGAL THEORY. DAMAGES THAT PROSOFT AND ITS DEALER WILL NOT BE RESPONSIBLE FOR INCLUDE, BUT ARE NOT LIMITED TO: LOSS OF PROFITS; LOSS OF SAVINGS OR REVENUE; LOSS OF USE OF THE PRODUCT OR ANY ASSOCIATED EQUIPMENT; LOSS OF DATA; COST OF CAPITAL; COST OF ANY SUBSTITUTE EQUIPMENT, FACILITIES, OR SERVICES; DOWNTIME; THE CLAIMS OF THIRD PARTIES, INCLUDING CUSTOMERS OF THE PURCHASER; AND INJURY TO PROPERTY.

** Some areas do not allow time limitations on an implied warranty, or allow the exclusion or limitation of incidental or consequential damages. In such areas the above limitations may not apply. This Warranty gives you specific legal rights, and you may also have other rights which vary from place to place.

6.5.6 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 15 months (or in the case of RadioLinx modules, 39 months) following shipment of the Product.

6.5.7 No Other Warranties

Unless modified in writing and signed by both parties, this Warranty is understood to be the complete and exclusive agreement between the parties, suspending all oral or written prior agreements and all other communications between the parties relating to the subject matter of this Warranty, including statements made by salesperson. No employee of ProSoft or any other party is authorized to make any warranty in addition to those made in this Warranty. The Customer is warned, therefore, to check this Warranty carefully to see that it correctly reflects those terms that are important to the Customer.

6.5.8 Intellectual Property

A Any documentation included with Product purchased from ProSoft is protected by copyright and may not be photocopied or reproduced in any form without prior written consent from ProSoft.

B ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.

C Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.

D Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.

E Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.

6.5.9 Additional Restrictions Relating To Software And Other Intellectual Property

In addition to complying with the Terms of this Warranty, Customers purchasing software or other intellectual property shall comply with any license agreement accompanying such software or other intellectual property. Failure to do so may void this Warranty with respect to such software and/or other intellectual property.

6.5.10 Allocation of risks

This Warranty allocates the risk of product failure between ProSoft and the Customer. This allocation is recognized by both parties and is reflected in the price of the goods. The Customer acknowledges that it has read this Warranty, understands it, and is bound by its Terms.

6.5.11 Controlling Law and Severability

This Warranty shall be governed by and construed in accordance with the laws of the United States and the domestic laws of the State of California, without reference to its conflicts of law provisions. If for any reason a court of competent jurisdiction finds any provisions of this Warranty, or a portion thereof, to be unenforceable, that provision shall be enforced to the maximum extent permissible and the remainder of this Warranty shall remain in full force and effect. Any cause of action with respect to the Product or Services must be instituted in a court of competent jurisdiction in the State of California.

Index