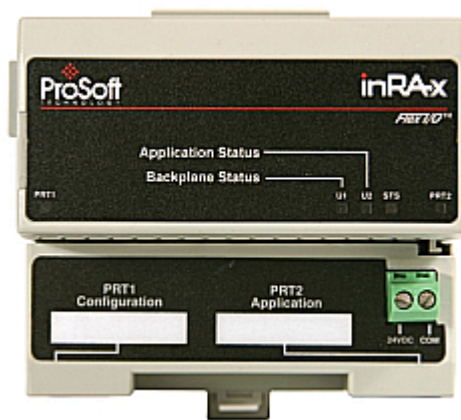




Where Automation Connects.



inRAX® **MVI94-GSC-E**

FLEX Platform

Generic ASCII Communication Module

July 06, 2009

USER MANUAL

Important Installation Instructions

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

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

MVI (Multi Vendor Interface) Modules

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

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

CL I Div 2 GPs A, B, C, D

Temp Code T5

II 3 G

Ex nA IIC T5 X

0° C ≤ Ta ≤ 60° C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

- A** Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
- B** Warning - Explosion Hazard - When in Hazardous Locations, turn off power before replacing or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
- C** Suitable for use in Class I, division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- 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 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings

cUL

C22.2 No. 213-1987



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

5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com

Copyright © ProSoft Technology, Inc. 2009. All Rights Reserved.

MVI94-GSC-E User Manual
July 06, 2009

ProSoft Technology®, ProLinx®, inRAX®, ProTalk®, and RadioLinx® are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

ProSoft Technology® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD, and are available at no charge from our web site: www.prosoft-technology.com

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33 (0) 5.3436.87.20

Latin America: +1.281.298.9109

North America: +1.661.716.5100

Contents

Important Installation Instructions	2
MVI (Multi Vendor Interface) Modules	2
Warnings	2
Your Feedback Please.....	3
ProSoft Technology® Product Documentation	3
1 Hardware Installation	7
1.1 Verify Package Contents.....	7
1.2 Mounting the MVI 94 Flex I/O Base	8
1.3 Setting Jumpers	9
1.4 Installing the Module onto the Base	10
1.5 Installing the Serial Adapter Cables	11
1.6 Wiring the Power Connections.....	11
2 Configuration	13
2.1 Configuration File	13
2.2 MVI94-GSC-E Configuration File	15
2.3 Uploading and Downloading the Configuration File.....	17
3 Diagnostics & Troubleshooting	23
3.1 Reading Status Data from the Module	23
3.2 LED Status Indicators.....	23
4 Reference	35
4.1 Product Specifications.....	35
4.2 Functional Overview.....	37
4.3 Cable Connections	48
5 Support, Service & Warranty	53
5.1 How to Contact Us: Technical Support	53
5.2 Return Material Authorization (RMA) Policies and Conditions.....	54
5.3 LIMITED WARRANTY.....	55
Index	59

1 Hardware Installation

In This Chapter

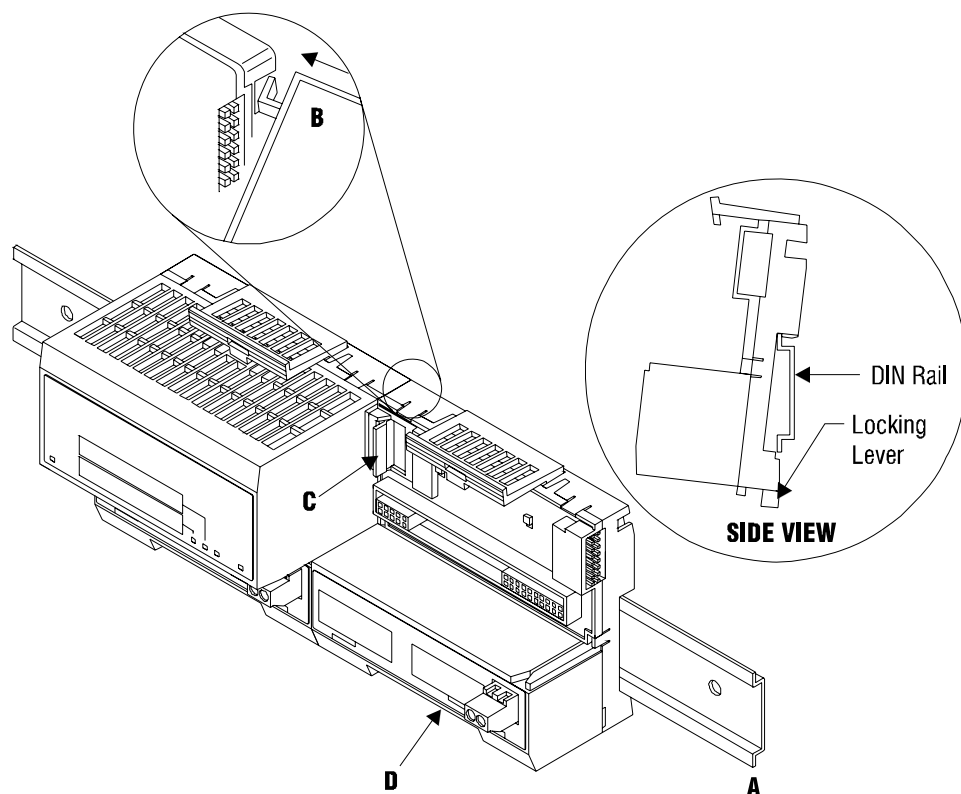
❖ Verify Package Contents	7
❖ Mounting the MVI 94 Flex I/O Base.....	8
❖ Setting Jumpers	9
❖ Installing the Module onto the Base	10
❖ Installing the Serial Adapter Cables	11
❖ Wiring the Power Connections	11

1.1 Verify Package Contents

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

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

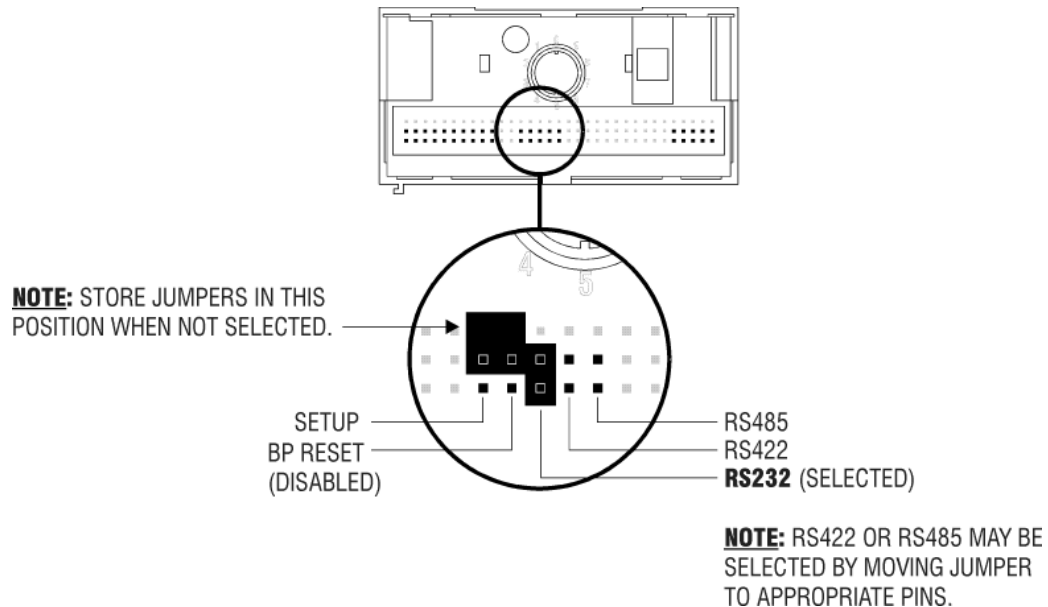
1.2 Mounting the MVI 94 Flex I/O Base



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

1.3 Setting Jumpers

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



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

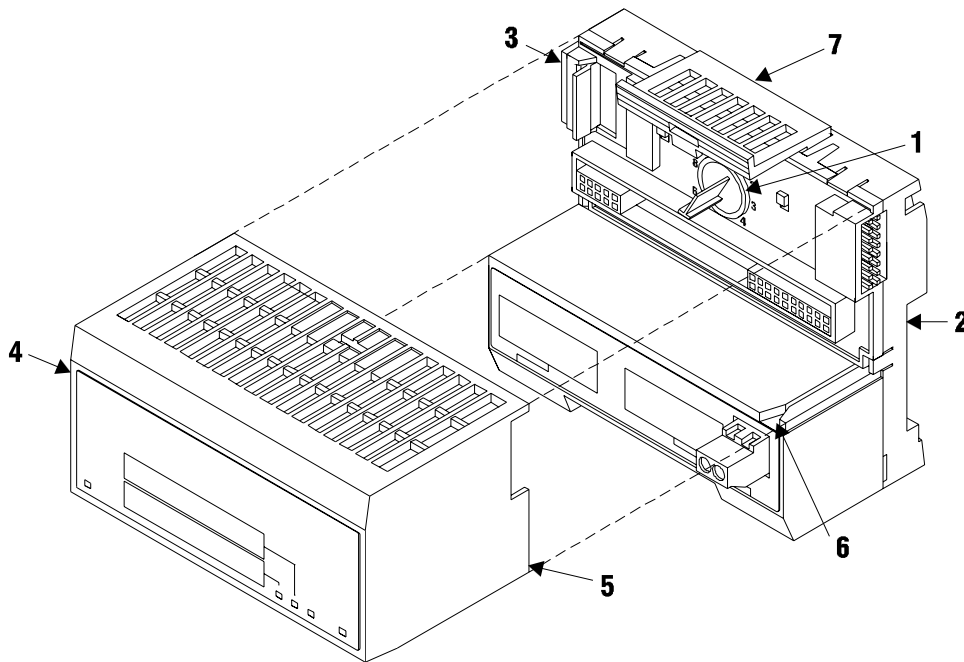
BP Reset: If the MVI94 module is to be reset when the Flex Bus is reset, install the BP RESET jumper in the Enabled position.

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

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

1.4 Installing the Module onto the Base

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

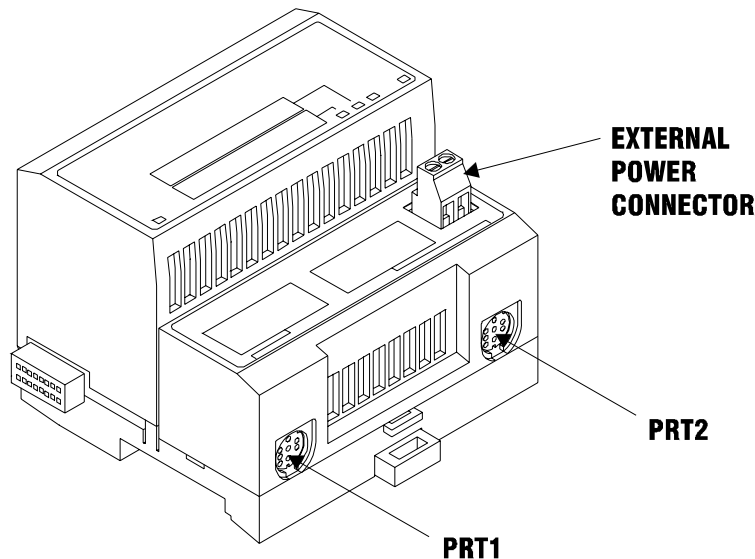


- 5 Press firmly and evenly to seat the Module in the Base. The Module is seated when the latch **7** on the Base is locked into the Module.

1.5 Installing the Serial Adapter Cables

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

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



1.6 Wiring the Power Connections

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

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

2 Configuration

In This Chapter

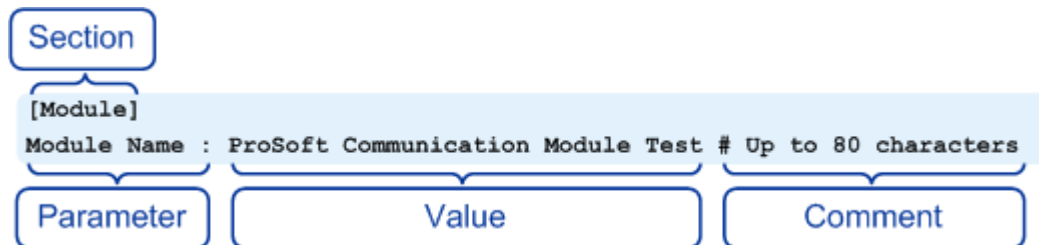
- ❖ Configuration File13
- ❖ MVI94-GSC-E Configuration File15
- ❖ Uploading and Downloading the Configuration File.....17

2.1 Configuration File

The MVI94-GSC-E module stores its configuration in a text file called configuration, located in the module's flash memory. When the module starts up, it reads the configuration file and uses the information to control how the GSC-E protocol interacts with the module's application port(s).

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

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



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

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

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

- Copy (page 17) the configuration from the module's flash memory to your PC
- Copy the configuration from the ProSoft Solutions CD-ROM supplied with the module
- Download the configuration from the ProSoft Technology web site at www.prosoft-technology.com

2.1.1 Editing the Configuration File

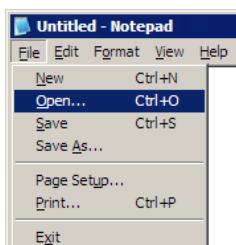
The configuration file is a plain ASCII text file. Use a text editor such as Notepad.exe (included with Microsoft Windows) to open and edit the file.

To open the configuration file in Notepad

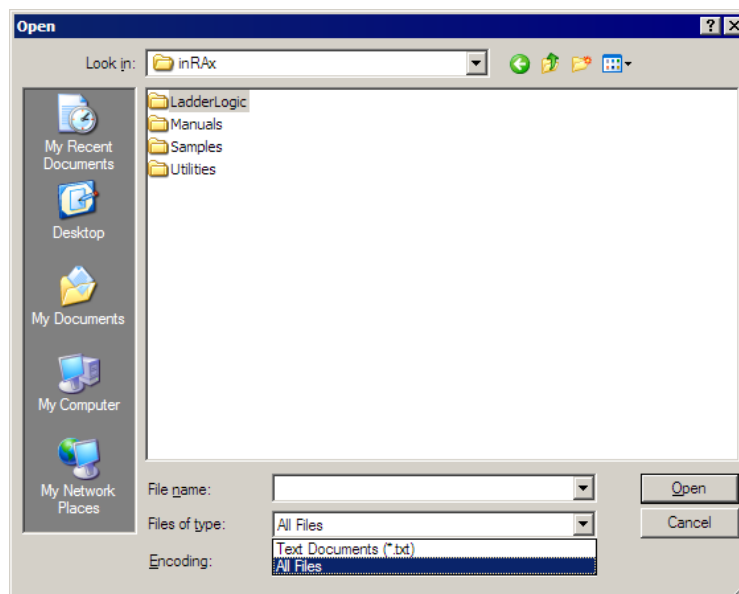
- 1 Click the Start button, and then choose Programs
- 2 Expand the Programs menu, and then choose Accessories.
- 3 On the Accessories menu, choose **Notepad**.



- 4 In Notepad, open the File menu, and then choose Open



- 5 In the Open dialog box, select "All Files" in the Files of Type: dropdown list.



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

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

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

2.2 MVI94-GSC-E Configuration File

The parameters that can be defined for the module are shown in the following table.

[Section]/Item	Value	Range	Description
[MODULE]			Module section header
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.
Backplane Fail 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 zero, 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.

[Section]/Item	Value	Range	Description
[GSC Port 0]			GSC port definition header
Enabled:		Yes or No	This parameter defines if this port will be utilized. If the parameter is set to No, the port is disabled. A value of Yes will enable the port.
Type:		0 to 15	This parameter specifies the receive termination characteristics for the port. This value is bit mapped as follows: Bit 0 = Termination character(s) used, Bit1=Message timeout used, Bit2=Intercharacter delay timeout used and Bit3=Packet size limit used. If the parameter is set to zero, the port is placed in stream mode.
Baud Rate:		From selected list of codes	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entries for this field include: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 576 and 115.
Parity:		None, Even, Odd, Mark or Space	This is the Parity code to be used for the port. The values are as follows: None, Odd, Even, Mark and Space.
Data Bits:		5 to 8	This parameter sets the number of data bits for each word used by the protocol. Valid entries for this field are 5, 6, 7 and 8.
Stop Bits:		1 or 2	This parameter sets the number of stop bits to be used with each data value sent. Valid entries for this field are 1 and 2.

[Section]/Item	Value	Range	Description
RTS On:		0 to 65535	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Valid values are in the range of 0 to 65535.
RTS Off:		0 to 65535	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low. Valid values are in the range of 0 to 65535.
Handshaking:	NONE, RTS/CTS, DTR/DSR or XON/XOFF		This parameter specifies the handshaking used on the port. The values are as follows: None=No hardware or software handshaking, RTS/CTS hardware handshaking, DTR/DSR hardware handshaking and XON/XOFF software handshaking.
Rx Term Char Count:		0 to 12	This parameter is used if bit 0 of the Type parameter is set. This value (0 to 12) defines the number of termination characters used to define the end of received message.
Rx Term Characters:	List of up to 12 integer values		This array of 12 integer values representing the characters used to define the termination characters at the end of each received message. The number of characters to be used in the array is set in the RTermCnt parameter.
Rx Packet Length:		0 to 4096	This parameter is used if bit 3 is set in the Type parameter. The parameter sets the length of data required to be received on the port before transferring the data to the processor.
Rx Message Timeout:		0 to 65535	This parameter is used if bit 1 is set in the Type parameter. The parameter sets the number of milliseconds to wait after the first character is received on the port before automatically sending the data to the processor.
Rx Intercharacter Delay:		0 to 65535	This parameter is used if bit 2 is set in the Type parameter. The parameter sets the number of milliseconds to wait between each character received on the port before sending the data to the processor.
Rx Swap Bytes:		Yes or No	This parameter specifies if the data received should have its bytes swapped before sending over the backplane.
Tx Message Timeout:		0 to 65535	This parameter specifies the timeout period to transmit a message out the port. A message must be transmitted out the port within the specified timeout period. Message transmission will be aborted if the timeout is exceeded.
Tx Minimum Delay:		0 to 65535	This parameter specifies the minimum number of milliseconds to delay before transmitting a message out the port. This pre-send delay is applied before the RTS on time. This may be required when communicating with slow devices.
Tx Swap Bytes:		Yes or No	This parameter specifies if the data to be transmitted out the port will have the bytes swapped from the data presented across the backplane.

2.3 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 24).

2.3.1 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.

2.3.2 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).

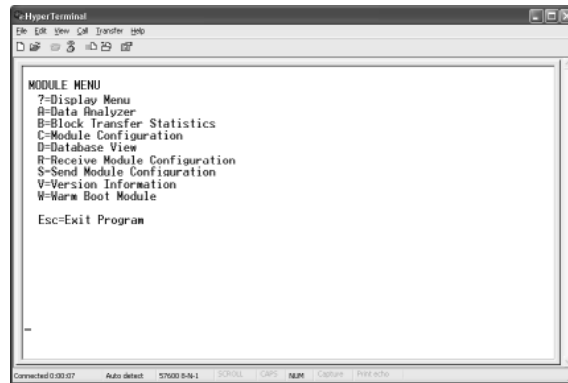
A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have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

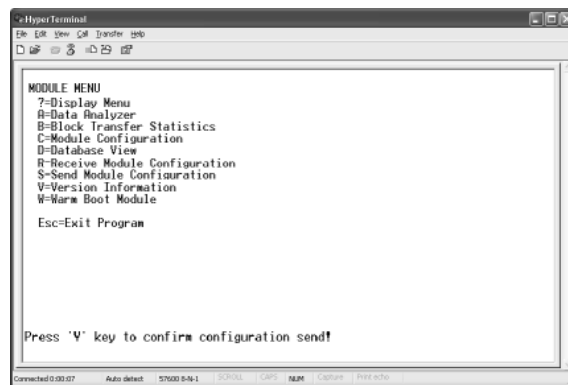
The module uses the Ymodem file transfer protocol to send (upload) and receive (download) configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

2.3.3 Transferring the Configuration File to Your PC

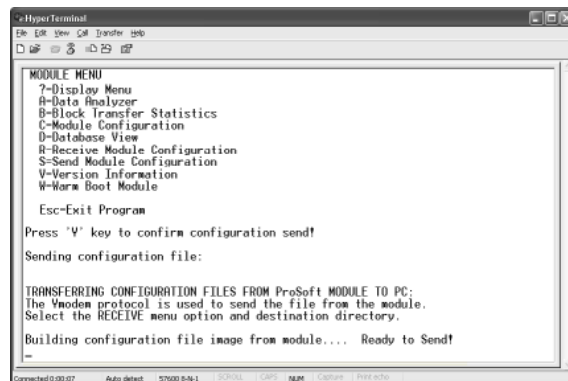
- 1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press [?] to display the main menu.



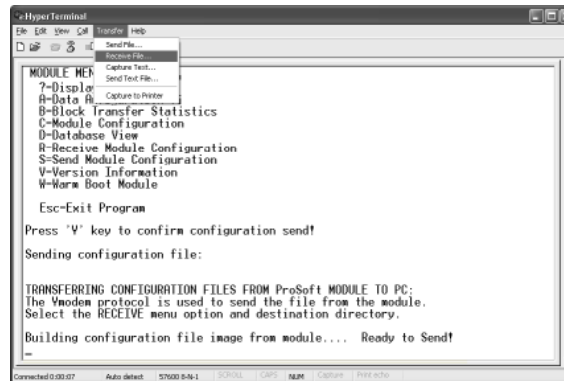
- 2 Press [S] (Send Module Configuration). The message "Press Y key to confirm configuration send!" is displayed at the bottom of the screen.



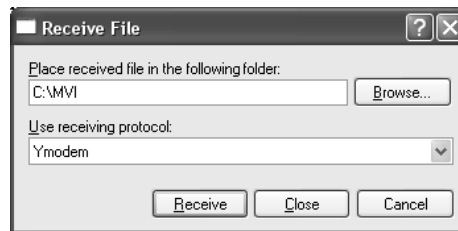
- 3 Press [Y]. The screen now indicates that the module is ready to send.



- From the **Transfer** menu in HyperTerminal, select **Receive File**. This action opens the Receive File dialog box.

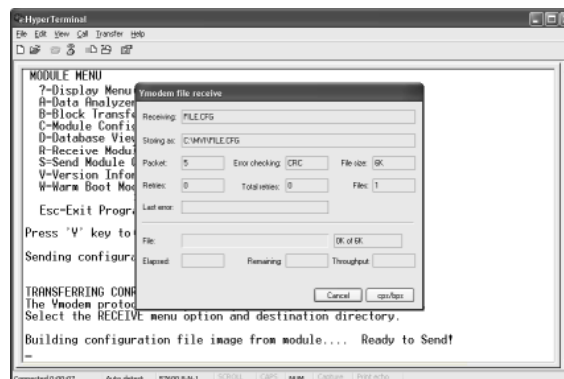


- Use the Browse button to choose a folder on your computer to save the file.

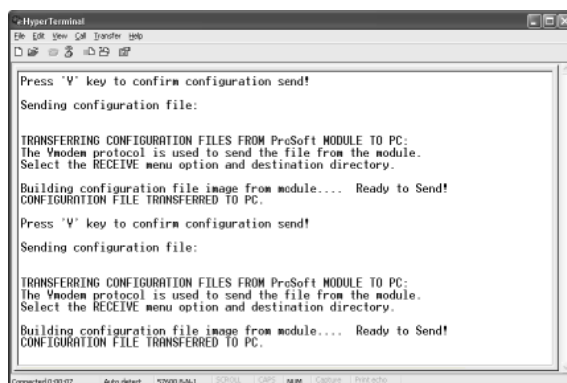


- Note: ProSoft Technology suggests that you upload the configuration file pre-loaded on your module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at www.prosoft-technology.com.

- Select Ymodem as the receiving protocol.
- Click the Receive button. This action opens the Ymodem File Receive dialog box, showing the progress of your file transfer.



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



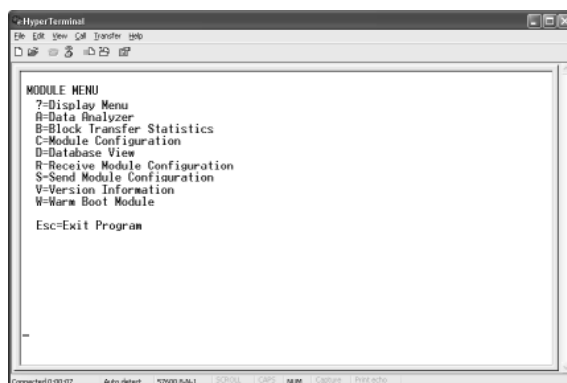
The configuration file is now on your PC at the location you specified.

- 8 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.

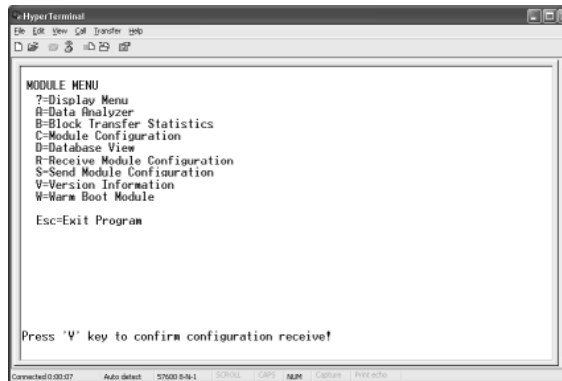
2.3.4 Transferring the Configuration File to the Module

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

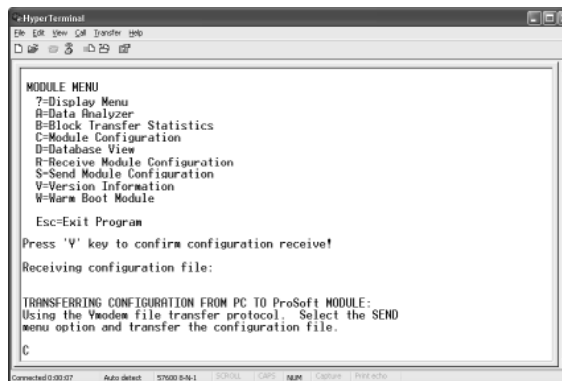
- 1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press [?] to display the main menu.



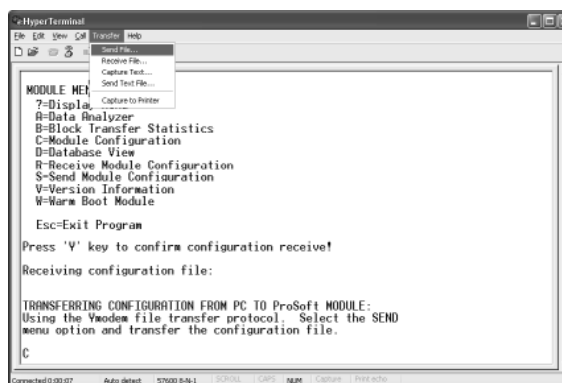
- 2 Press **[R]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



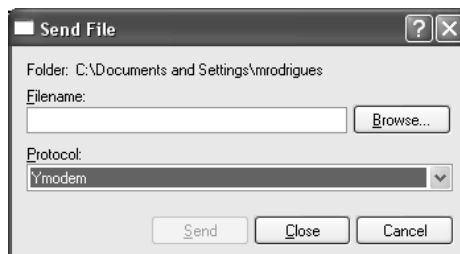
- 3 Press **[Y]**. The screen now indicates that the PC is ready to send.



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



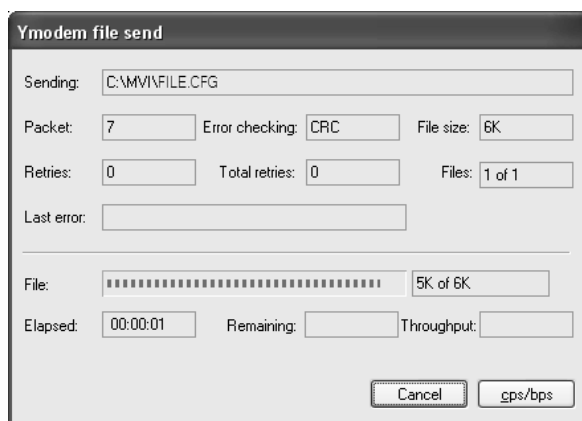
The Send File dialog appears.



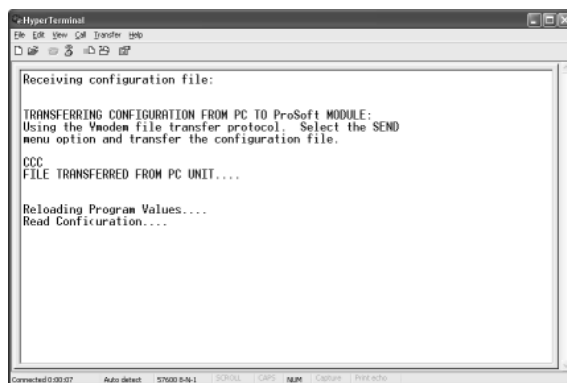
- 5 Use the Browse button to locate the configuration file your computer.

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

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



When the file transfer is complete, the module's configuration/debug screen indicates that the module has reloaded program values, and displays information about the module.



- 8 Your module now contains the new configuration.

3 Diagnostics & Troubleshooting

In This Chapter

- ❖ Reading Status Data from the Module 23
- ❖ LED Status Indicators..... 23

The module provides diagnostic information in the following forms.

- Status Data values are transferred from the module to the Flex processor.
- All data contained in the module can be viewed through the configuration/debug port on an attached terminal emulator.
- LED status indicators on the front of the module yield information on the modules status.

The following topics explain how to obtain the Status Data from the module and the meaning of the individual LEDs on the module.

3.1 Reading Status Data from the Module

The MVI94-GSC-E module returns two separate status data areas to the Flex processor when no receive data is available to transfer. For a complete listing of the status data objects, refer to the Backplane Data Transfer section of this document.

3.2 LED Status Indicators

This section defines the indications provided on the MVI94-MCM module through LEDs. A description of each LED is provided in the following topics.

3.2.1 PRT1

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

3.2.2 U1

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

3.2.3 U2

This LED indicates communication errors on the Modbus master port. The LED is illuminated (amber) when no error exists on the port. If a communication error is recognized on the port, the LED will turn off. If the LED is off, check for errors in the command list to determine the error condition recognized by the module.

3.2.4 STS

This LED indicates the "health" of the module. When power is applied to the module, the LED is illuminated. If the LED is green, the program is working correctly and the user configuration is being used. If the LED is red, the program is halted. Try restarting the module by cycling power.

3.2.5 PRT2

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

3.2.6 The Configuration/Debug Menu

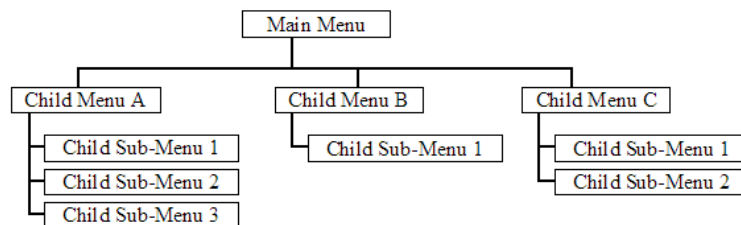
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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.

3.2.7 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.

3.2.8 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.

3.2.9 Using the Configuration/Debug Port

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

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

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

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

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

- 1** Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2** Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC.
- 3** Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
- 4** On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

3.2.10 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:

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.

```
GENERIC SERIAL ASCII MODULE (MVI94-GSC-E) MENU
?=Display Menu
A=Data Analyzer
B=Block Transfer Statistics
C=Module Configuration
V=Version Information
R=Receive Configuration from Remote Unit
S=Send Configuration to Remote Unit
W=Warm Boot Module
1=Communication Status
6=Port Configuration

Esc=Exit Program
```

Opening the Data Analyzer Menu

Press **[A]** to open the Data Analyzer Menu. Use this command to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Refer to Data Analyzer for more information about this menu.

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.

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 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 Port Communication Status

Press **[1]** from the Main Menu to view the port communication status for the application port.

Use this command to view communication status and statistics for the selected port. This information can be informative when trouble-shooting communication problems.

Viewing Port Configuration

Press **[6]** from the Main Menu to view configuration information for the application port.

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

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 memory to configure the module.

3.2.11 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.

```

Data Analyzer Mode Selected

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

Port = 1, Format=HEX, Tick=10

```

Important: When in analyzer mode, program execution will slow down. Only use this tool during a trouble-shooting session. Before disconnecting from the Config/Debug port, please be sure to press [S] to stop the Data Analyzer before returning to the main menu or disconnecting from the port. This action will allow the module to resume its normal operating mode.

Analyzing Data for Application Port

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

[illegible]

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. The following illustration shows an example.



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.

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

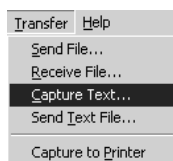
Port = 1, Format=HEX, Tick=10
```

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

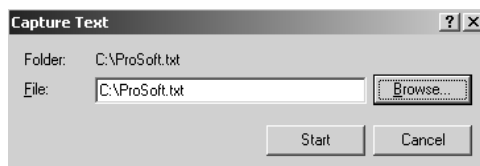
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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:



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  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><0A><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_[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><0A><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][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]
[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><0A><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.

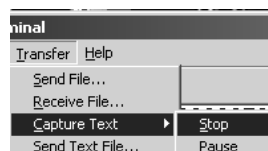
All characters shown in <> brackets are characters being sent out by the module.

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 Reference

In This Chapter

- ❖ Product Specifications 35
- ❖ Functional Overview 37
- ❖ Cable Connections 48

4.1 Product Specifications

The ProSoft Technology Generic ASCII Serial Single Port Interface Module is a fast and easy way to add a fully configurable ASCII communication port to the FLEX platform without consuming the front port of the processor.

The MVI94-GSC-E allows Rockwell Automation Flex I/O compatible processors to interface easily with other serial communication devices. Compatible devices include bar code readers, scanners, scales, printers, terminals and any other serial communication devices.

4.1.1 General Specifications

Some of the general specifications include:

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

4.1.2 FLEX I/O Interfaces

Specification	Description
Form Factor	Single Slot 1794 Backplane compatible Locate in any slot of Backplane
Backplane current load	20 mA @ 5 V
External power supply	12V to 24VDC 340 mA to 170 mA
Operating temperature	0 to 55°C (32 to 131°F)
Storage temperature	-40 to 85°C (-40 to 185°F)
Shock	30g operational 50g non-operational 5g from 10150 Hz
Relative humidity	5% to 95% (non-condensing)

Specification	Description
LED indicators	Module status Backplane transfer status Application status Serial activity and error LED status
Configuration Serial port (PRT1)	Mini-DIN RS-232 Hardware handshaking
Application serial Port (PRT2)	Mini-DIN RS-232/422/485 jumper selectable 500V optical isolation from backplane
Dimensions (with Module installed in Base)	3.7H x 3.7W x 2.7D inches 94H x 94W x 69D mm

4.1.3 Functional Specifications

The MVI94-GSC-E module interfaces a serial communication device with the Rockwell Automation Flex processor. One application port is present on the module to interface with devices such as bar code readers, scales, printers and terminals. The communication port permits both the reception and transmission of data between the Rockwell Automation processor and attached devices. The module is designed primarily to receive data (up to 4096 characters in consecutive blocks), but can also transmit up to 10 characters on a block transfer.

- One port to receive and/or transmit data
- Receive up to 4096 characters
- Transmit up to 10 characters per block

Configurable parameters include:

Parameter	Description
Termination type	Stream mode, termination character(s), message timeout, intercharacter timeout and packet size limit
Baud Rate	110 to 115,200
Parity	None, Odd and Even
Data Bits	5 to 8
Stop Bits	1 or 2
RTS On and Off Timing	0 to 65535 milliseconds
Minimum Response Delay	0 to 65535 milliseconds
Hardware or Software Handshaking	RTS/CTS, DTR/DSR or XON/XOFF
Byte Swapping	Swap bytes received or to transmit

4.2 Functional Overview

This section describes how the MVI94-GSC-E module transfers data between itself and the processor, and how it implements the GSC-E protocol.

4.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI94-GSC-E module.

Module Power Up

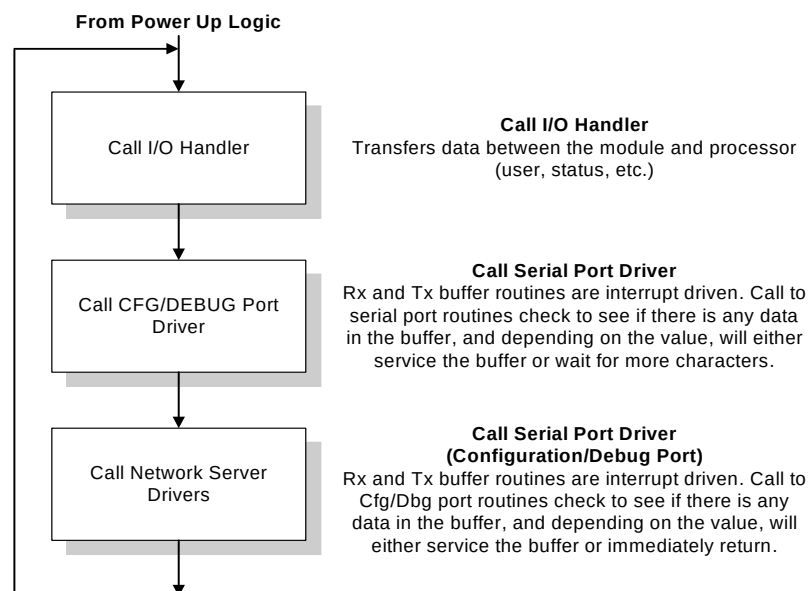
On power up the module begins performing the following logical functions:

- 1 Initialize hardware components
- 2 Initialize Flex backplane driver
 - o Test and Clear all RAM
 - o Initialize the serial communication port
 - o Use configuration stored on module's flash RAM to configure module
- 3 Initialize Communication port
- 4 Set up the serial communication interface driver on the port

After the module has loaded the configuration information stored in EEPROM and the module is configured, the module will begin receiving and transmitting messages with devices on the serial port.

Main Logic Loop

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



Flex Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (for example, Fault or PGM), the application port 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 serial network.

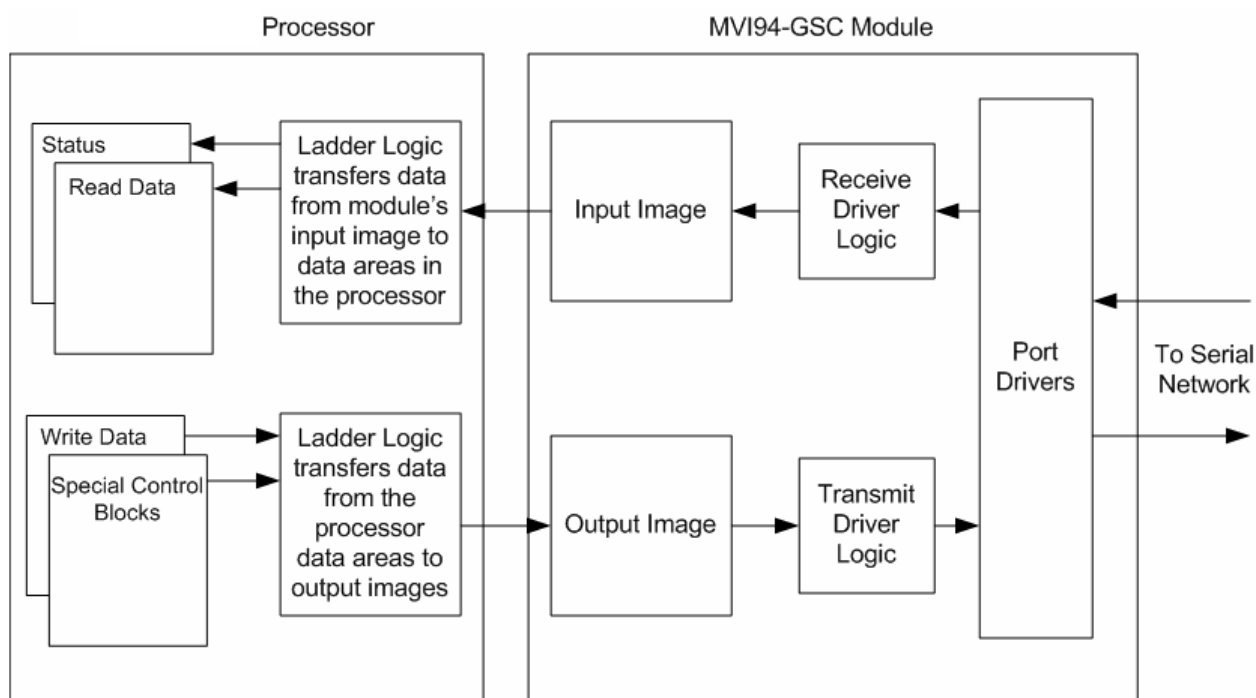
4.2.2 Backplane Data Transfer

The MVI94-GSC-E module communicates directly over the FLEX backplane. Data travels between the module and the Flex processor across the backplane using the module's input and output images. The update frequency of the images is determined by the ladder logic scan rate and the backplane bus rate.

Data received on the application port is placed in the module's input image. This data is processed by the ladder logic in the Flex processor. The input image for the module is set to 16 bytes.

The processor inserts data in the module's output image to transfer to the module. The module's program extracts the data and transmits the data out the communication port. The output image allowed for data to the module is set at 14 bytes.

The following illustration shows the data transfer method used to move data between the Flex processor, the MVI94-GSC-E module and the serial devices.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the Flex processor to interface the input and output image data with data defined in the processor. The user is responsible for handling and interpreting all data received on the application port and transferred in the input image. Additionally, the user is responsible for constructing messages to be transferred out of the application port by building the messages in the output image of the module.

Each block transferred between the module and the processor has a block number in byte zero of the input or output image.

Block ID	Description
0 to 127	Normal data blocks for read and write
254	Warm boot module
255	Cold boot module
128 to 253	Reserved for future use

Block identification codes 0 to 127 are used for normal data transfer. Blocks 254 and 255 are used as control blocks to control the module from the ladder logic. Blocks 128 to 253 are reserved and are intended for future use.

4.2.3 Normal Data Transfer

Normal data transfer includes the transferring of data received or to be transmitted on the port and the status data. These data are transferred through read (input image) and write (output image) blocks. The following topics describe the function and structure of each block.

Read Block

These blocks of data transfer information from the module to the Flex processor. When data is received on the port, blocks of the following structure are constructed by the module in the input image:

Byte Offset	Description
0	Block Sequence Number (Bumped each scan by module)
1	Number of characters (0 to 14) in Application Port receive block (2 to 15). If the receive data in the module is larger than 14 bytes, multiple blocks will be transferred. Any block with a value of 255 in this field represents the first or continuation block and the block contains 14 bytes of data. The last block of data will contain a positive number in this field that represents the number of characters in the last block. If this block contains a value of 252 or 253 the block contains status data.
2 to 15	Application Port data received

The Block Sequence Number (byte 0) is an index value used to signal to the Flex processor that a new block is ready for processing. The ladder logic must recognize a change in this value and process the data encapsulated in the input image. Byte 1 of the block contains a count of the number of bytes to process in the current block. Bytes 2 to 15 contain the data received on the port. If the receive byte swapping option is enabled, the bytes received will be swapped before being placed in the input image.

The receive buffer in the module is set to 4096 bytes. This large size permits the buffering of a large amount of data before a transfer of the data to the controller is required. The ladder logic to use a buffer larger than 14 bytes is more complex. The module buffers the data in this buffer until one of the specified termination conditions is recognized. The module will then transfer the received block of data to the controller. If the block of data received is larger than 14 bytes, multiple blocks will be used to transfer the data to the controller. The first block will contain a value of 255 in the Number of Characters Received data field (byte 1). This indicates that there will be more blocks to follow and that the current block contains 14 bytes of data. As long as more than 14 bytes are present in the buffer being sent to the controller, the length field will be set to a value of 255. When 14 or fewer bytes remain in the buffer, the module will send the last block with a positive number in the length field. The value passed represents the number of bytes present in the data area. The ladder logic must recognize the presence of the positive number and end the packet received.

When the module does not have any receive data to send to the processor, it will send status blocks from the module. The status information transferred in the read block can be used by the processor to determine the state and "health" of the module and the device(s) attached to each application port. An important member of the value in the status object is error word. This value contains the configuration error flags for each port and the receive buffer overflow error flag. The module indicates the presence of the status data by placing a value of 252 (0xFC) or 253 (0xFD) in the length field of the input image. The format of the block with a length value of 252 is as follows:

	Byte Offset	Description
	0	Block Sequence Number (Bumped each scan by module)
	1	This byte contains a value of 252 (0xFC)
Module Status	2 to 3	Program cycle counter
	4 to 5	Number of blocks transferred from module to processor
	6 to 7	Number of blocks transferred from processor to module
	8 to 9	Number of blocks parsed by module
	10 to 11	Number of block errors in module
	12 to 13	Application Port error word
	14 to 15	Application Port transmit state

The definition of the bits in the Port 2 Error Word is displayed in the following table.

Member Name	Bit in Word	Description
Cfg_type	Bit 0	The termination type configured for the port is not valid. Values between 0 and 15 are the only ones valid. The module will use type 0 (stream mode) for the port.
Cfg_Baud	Bit 1	The baud rate entered for the port is not valid. The module will use 9600 baud for the port.
Cfg_Parity	Bit 2	The parity value entered is not valid. Expected values are None, Even, Odd, Mark and Space. Only the first letter is inspected to make the selection. The module will set the parity to a value of none if an invalid selection is entered.
Cfg_DataBits	Bit 3	The number of data bits for the protocol is not valid. Values between 5 and 8 are accepted. The module assumes a value of 8 data bits.

Member Name	Bit in Word	Description
Cfg_StopBits	Bit 4	The number of stop bits for the protocol is not valid. Values of 1 or 2 are accepted. The module assumes a value of 1 stop bit.
Cfg_Handshake	Bit 5	The handshake code for the port is not valid. Expected values are None, RTS/CTS, DTR/DSR and XON/XOFF. Only the first letter of the selection is used. The module assumes a value of None (no handshaking) if the parameter is in error.
Cfg_Rtermcount	Bit 6	The number of termination characters is not valid. The value must be set between 1 and 12 when using the termination character string to end a receive buffer. The module will not terminate a buffer when using the termination character(s) when this bit is set.
Cfg_RPacketLen	Bit 7	The number of characters for a packet is not valid. The value must be set between 1 and 4096 when the packet size termination option is used. The module will not use the packet length termination option when this bit is set.
Cfg_Rtimeout	Bit 8	The message timeout value is set to zero. The module will not use the message timeout termination option when this bit is set.
Cfg_Rdelay	Bit 9	The intercharacter delay value configured is set to zero. The module will not use the intercharacter delay option when this bit is set.
Cfg_Wtimeout	Bit 10	The write message timeout parameter is set to zero. The module assumes a value of 5000 milliseconds.
	Bit 11	
	Bit 12	
	Bit 13	
	Bit 14	
Err_ROverflow	Bit 15	Data is being received faster on the port than the ladder logic can process the read blocks. Alter the configuration of the module or the connected device. Receive data is being lost.

If no configuration errors are present, the word will have a value of zero. Any errors indicated by the bits in this word should be addressed and corrected for proper module operation.

The format of the block with a length value of 253 is as follows:

	Byte Offset	Description
	0	Block Sequence Number (Bumped each scan by module)
	1	This byte contains a value of 253 (0xFD)
Module Status	2 to 3	Program cycle counter
	4 to 5	Application Port receive state: -1 = Listening for data 1 = Receiving Port Data 2 = Waiting for Backplane transfer
	6 to 7	Application Port receive character count
	8 to 9	Application Port receive block count
	10 to 11	Application Port transmit character count
	12 to 13	Application Port transmit block count

Byte Offset	Description
14 to 15	Application Port transmit state: 0 = Waiting for Data to Send 1 = RTS On 2 = RTS Timeout 3 = Sending data 4 = Waiting for RTS Off 5 = RTS turned off 30 = Intercharacter Delay 31 = Intercharacter Delay 32 = Intercharacter Delay 100 = Message Delay before Transmit 101 = Message Delay before Transmit

Notice that bytes 14 to 15 (word 7) in both of the status blocks contain the Port 2 transmit state. This value can be used to determine when the port is able to accept more characters to transmit. Each time a write block is transmitted with data to send to the port, the module will immediately send the data according to the timing characteristics set for the port and the current message. While the message is being sent, the Port 2 transmit state will contain a value other than zero. When the port has completed the full transmission of the write request, the Port 2 transmit state will have a value of 0.

Write Block

These blocks of data transfer information from the Flex processor to the module. The structure of the output image used to transfer this data is shown below:

	Byte Offset	Description
	0	Block Sequence Number (Read block number as set by module)
Application Port	1	Intercharacter delay for this message (milliseconds between characters with range of 0 to 255).
	2	Number of characters to transmit on Application Port (0 to 10)
	3	Reserved for future use
	4 to 13	Application Port data to transmit

The Block Sequence Number is that received on the last read block transfer through the input image on the module. The ladder logic should copy this value from byte 0 of the input image to byte 0 of output image in the ladder logic. This is the last operation performed when constructing the write block. The module's program will trigger the process write block function when a new value is recognized in byte 0 of the output image. If the number of characters to transmit in the write block is not set to zero (value in word at byte 2), data to transmit is present in the block. If the selected port is not busy transmitting data (Port 2 transmit state is zero), the data in the block will be moved to the port's transmit buffer and sent out the port. The data to transmit is held in bytes 4 to 13 of the block.

In order to pace the characters for the write operation, an inter-character delay value is associated with each write message. For devices that do not buffer received data, when interfacing with a modem in command mode or when simulating keyboard or keypad entry, inter-character delays may be required. For example, if the port is tied to a device that expects input with delays of 200 milliseconds between each character, place the data to send to the buffer along with the length and set the inter-character word (byte 1) to a value of 200 in the module's output image in the processor's ladder logic program. The message will be transmitted with a 200-millisecond wait period between each character. Because this delay value is sent from the processor for each write message, the inter-character delay can be set independently for each message. For example, when writing AT commands to a dial-up modem, an inter-character delay of 100 may be required. But when the modem is in data mode, the inter-character delay can be set to 0. When the delay is set to 0, the whole packet of data will be placed in the module's transmit buffer at one time.

4.2.4 Control Blocks

The module recognizes control block codes of 254 (0xFE) and 255 (0xFF) from the ladder logic for module control. The format and definition of these two blocks are provided in the following topics.

Warm Boot

This block is sent from the Flex processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. This operation will force the module to restart by reading in the configuration information and resetting all program status data. The format of the output image to perform this task is as follows:

Byte Offset	Description
0	Block Sequence Number set to 254 (0xFE)
1 to 13	Not Used

Cold Boot

This block is sent from the Flex processor to the module (output image) 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 format of the output image to perform this task is as follows:

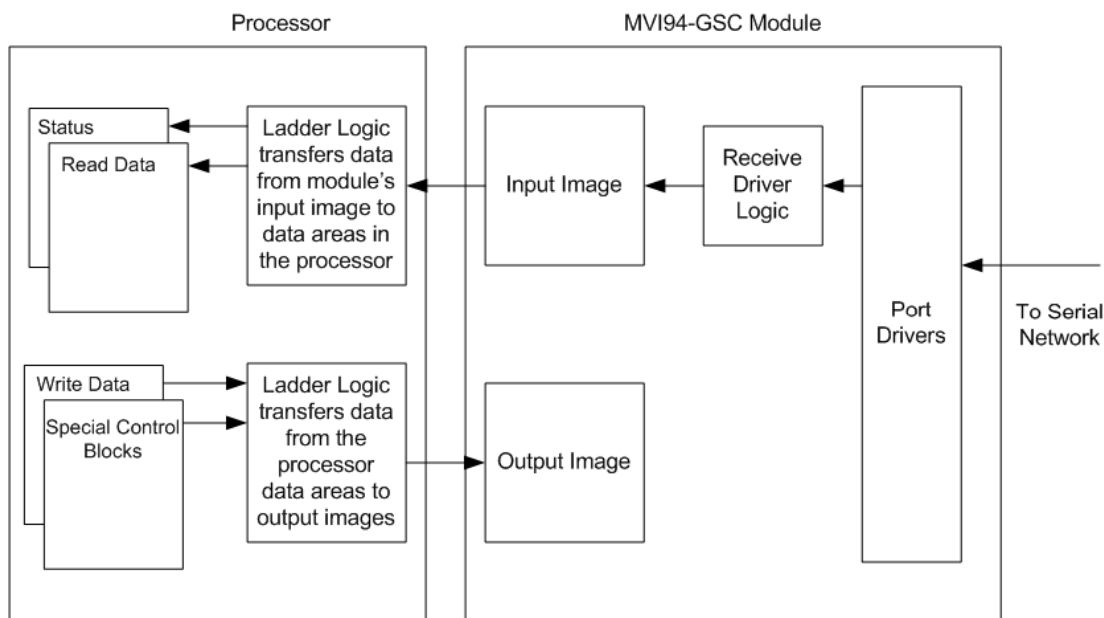
Byte Offset	Description
0	Block Sequence Number set to 255 (0xFF)
1 to 13	Not Used

4.2.5 Data Flow between MVI94-GSC-E Module and Flex Processor

The following topics describe the flow of data between the two pieces of hardware (Flex processor and MVI94-GSC-E module) and other devices attached to the application port. The sections below show the three possible types of communication devices that can be attached to the application port: write-only, read-only and read-write.

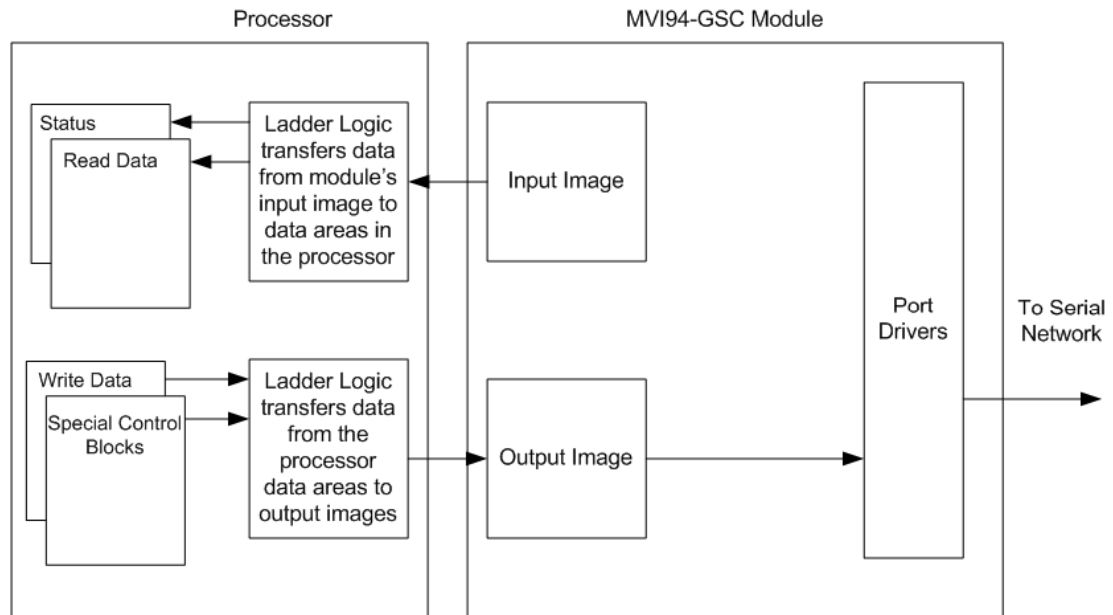
Write-Only Devices

Write-only devices are those that only send data to the module. An example of this type of device is a barcode reader. The reader is programmed to only send data and is not expected to receive data. In this configuration, the application port on the MVI94-GSC-E module will never transmit data. All data received from the device will be passed from the module to the Flex processor through the module's input image. Ladder logic in the processor must handle the data received from the module. The output image on the module will only be used to inform the module when the input image (read data) has been processed. This is accomplished by copying the byte 0 in the input image to byte 0 of the output image. The data flow diagram for a write-only device is shown below:



Read-Only Devices

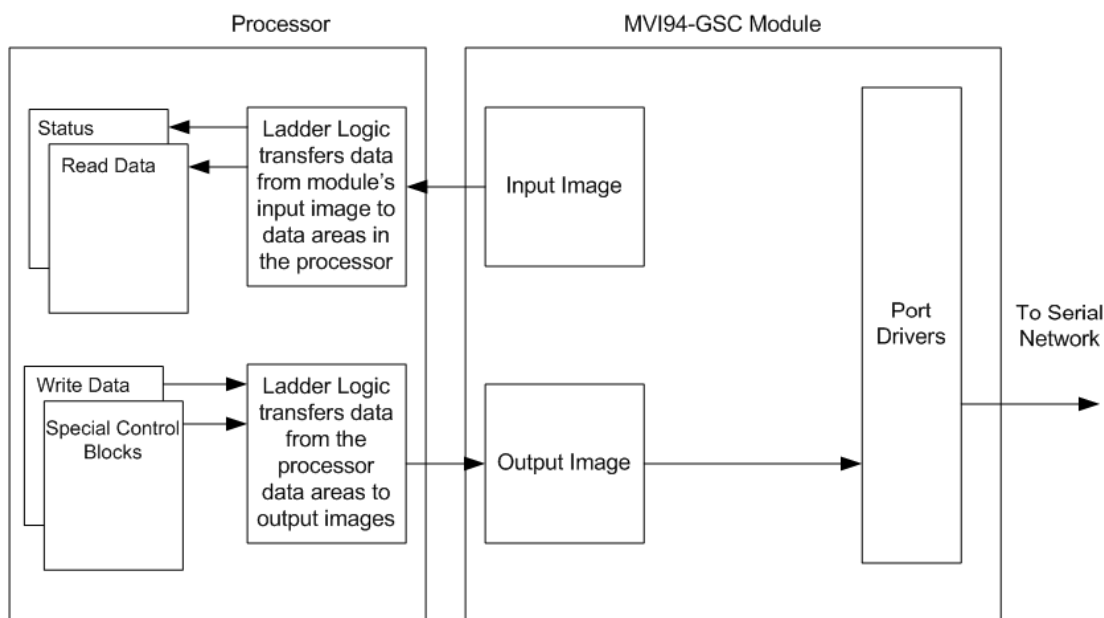
Read-Only devices are those that only receive data from the module. An example of this type of device is a printer. The printer will generate output or be controlled based on the data it receives on its communication port. Ladder logic is used to construct the write blocks to be sent to the module. When the module receives a new write block containing data, it will transmit the data out the port. The Block Sequence Number used in the write block (output image) should be the one received on the last read block (input image). The data flow diagram for a read-only device is shown below:



Before data is transmitted in the write block, the ladder logic should insure that the port is not busy sending the previous write message. The Port 2 transmit state returned in each status block should be monitored to determine the state of the port. If the Port 2 transmit state has a value other than zero, the port is busy. If the Port 2 transmit state has a value of zero, the port is ready for a new message to transmit.

Read-Write Devices

Read-write devices are those that both send and receive data. An example of this type of device is a terminal. A terminal will send data entered on the keyboard out its serial port and display any data received on its port on the monitor. All data received from the terminal will be routed to the Flex processor through the MVI94-GSC-E's input image. Data to be written to the terminal will be sent to the module using the output image. The module will send new data in the output image out the application port to the terminal. The data flow diagram for a read-write device is shown below:



4.2.6 Termination of Received Data

When data is received on the application port, you must define in the configuration when this data will be transferred to the Flex processor. Within the module, this is known as the termination type for port. When the termination condition is met, the data will be sent from the port's receive buffer (data area of 4096 bytes) to the processor using the input image. This termination type is set in the bit mapped, Type field of the module object. The following illustration shows the bit map used for this parameter.

Termination Type Field

Bit(s)	4 to 7	3	2	1	0
Bit Value	-	8	4	2	1
Definition	Reserved	Packet size limit used	Intercharacter delay timeout used	Message timeout used	Termination character(s) used

If none of the bits are set (Type=0), the port will be configured for stream mode. Any characters received on the port are immediately sent to the processor. The processor must buffer and assemble a packet of information if this mode is selected as required by the application. If the data can be handled by the processor in this mode and it is appropriate for your application, this is the fastest method of communication between the device and the processor.

Any combination of bits is acceptable to the module and should be set to match the device on the specific port. An example of each termination type is given below.

Termination character(s) used

Settings:

Count = 1 (RTermCnt=1)

Termination on 0x0d (carriage return character) (RTermChar = 0d 00 00 00 ...)

Data Received on port:

A B C 0x0d D E

Comment:

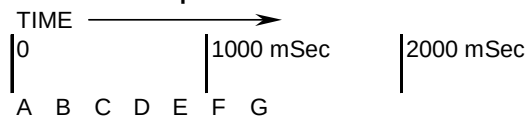
The characters "ABC" will be sent along with the 0x0d character to the controller after the 0x0d character is received. The characters "DE" will not be sent until the 0x0d character is received.

Message timeout used

Settings:

Message timeout = 1000 mSec (Rtimeout=1000)

Data Received on port:



Comment:

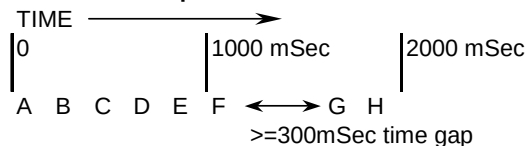
After the 'A' character is received on the port, the message timeout is started. The characters "ABCDE" will be sent to the controller in one block. The characters "FG" will follow in the second block one second later.

Intercharacter delay timeout used

Settings:

Intercharacter delay timeout = 300 mSec (Rdelay=300)

Data Received on port:



Comment:

After each character is received, the intercharacter delay timer is reset. The characters "ABCDEF" will be sent to the controller in one block because the delay timer expires. The characters "GH" will follow in the second block when the next time gap is recognized.

Packet size limit used

Settings:

Packet size = 4 (RPacketLen=4)

Data Received on port:

A B C D E F G H I J

Comment:

The first block sent to the controller will contain the characters "ABCD", and the second block will contain the characters "EFGH". The characters "IJ" will not be sent until two more characters are received on the port.

The maximum size of a message that can be sent using the input image is 4096 bytes. This is the size of the receive buffer in the module. If this buffer is filled before the termination condition is present, the module will automatically transfer the received data to the controller. This will prevent the loss of data. The data will be paged using the input image 14 bytes at a time. The ladder logic must handle this condition if the configuration of the module would permit this situation.

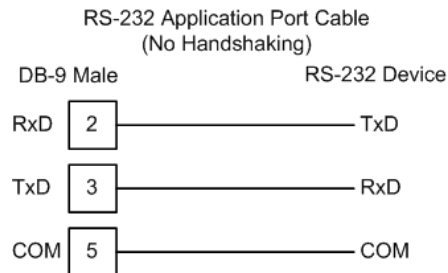
4.3 Cable Connections

The application ports on the MVI94-GSC-E 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.

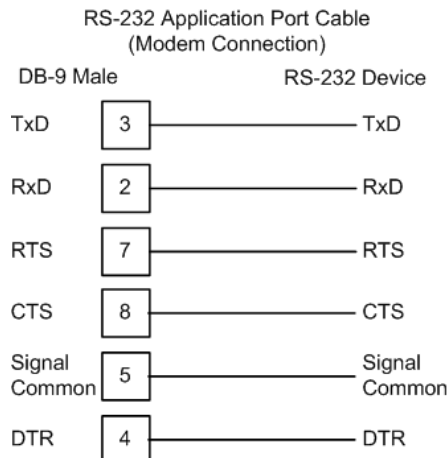
4.3.1 RS-232

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

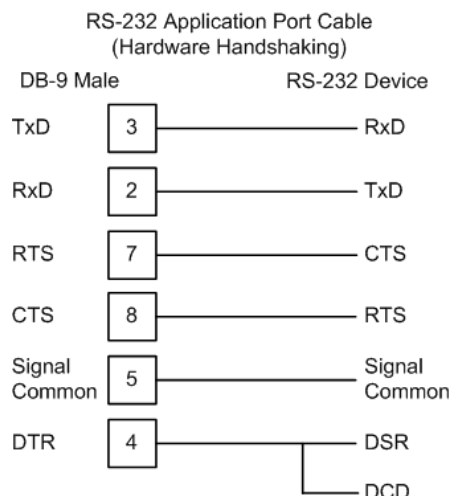
This type of connection is required between the module and a modem or other communication device.



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

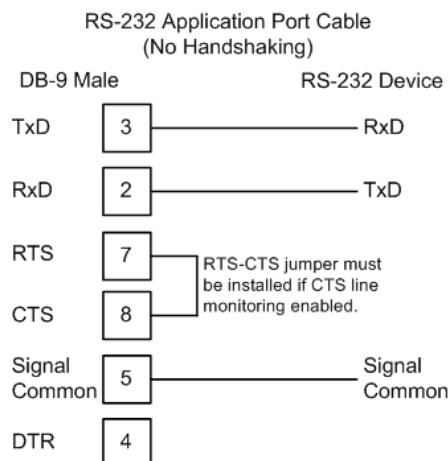
RS-232: Null Modem Connection (Hardware Handshaking)

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)

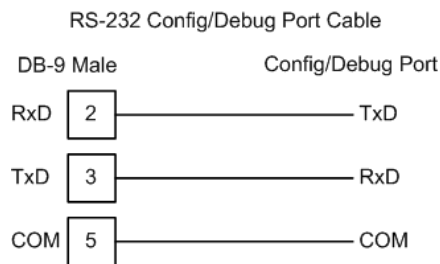
This type of connection can be used to connect the module to a computer or field device communication port.



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.

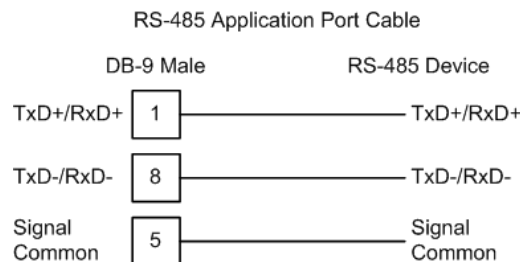
4.3.2 RS-232 Configuration/Debug Port

This port is physically a Mini-DIN connection. A Mini-DIN 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:



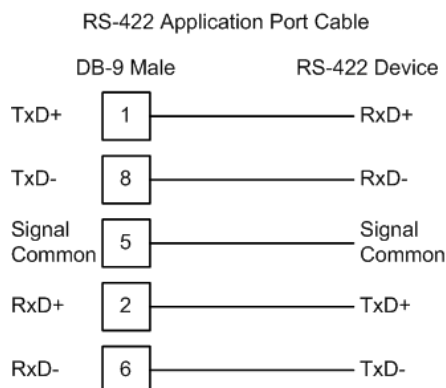
4.3.3 RS-485

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:



Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120-ohm terminating resistor on the RS-485 line.

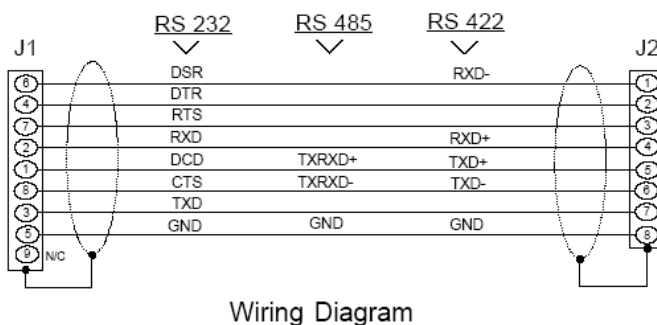
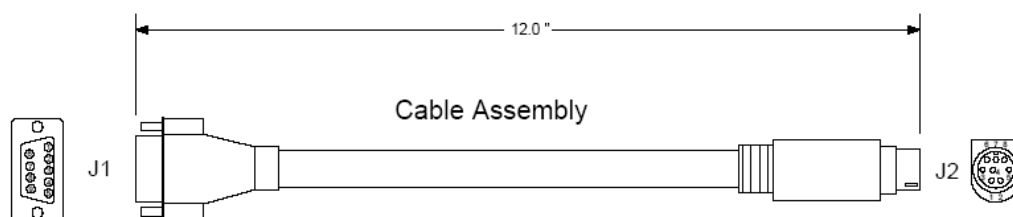
4.3.4 RS-422



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.

4.3.5 DB9 to Mini-DIN Adaptor (Cable 09)



5 Support, Service & Warranty

In This Chapter

- ❖ How to Contact Us: Technical Support..... 53
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 54
- ❖ LIMITED WARRANTY 55

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

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

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

- 1 Module configuration and contents of file
 - o Module Operation
 - o Configuration/Debug status information
 - o LED patterns
- 2 Information about the processor and user data files as viewed through and LED patterns on the processor.
- 3 Details about the serial devices interfaced, if any.

5.1 How to Contact Us: Technical Support

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

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com
Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com
Languages spoken include: French, English

North America/Latin America (excluding Brasil) (location in California)

+1.661.716.5100, support@prosoft-technology.com
Languages spoken include: English, Spanish

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.

Brasil (location in Sao Paulo)

+55-11-5084-5178, eduardo@prosoft-technology.com
Languages spoken include: Portuguese, English

5.2 Return Material Authorization (RMA) Policies and Conditions

The following RMA Policies and Conditions (collectively, "RMA Policies") apply to any returned Product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see "Limited Warranty". In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

5.2.1 All Product Returns:

- a) 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.
- b) 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 (page 53). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
- c) 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 and receipt date. 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 using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, and so on.

5.2.2 Procedures for Return of Units Under Warranty:

A Technical Support Engineer must approve the return of Product under ProSoft's Warranty:

- a) A replacement module will be shipped and invoiced. A purchase order will be required.
- b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

5.2.3 Procedures for Return of Units Out of Warranty:

- a) Customer sends unit in for evaluation
- b) If no defect is found, Customer will be charged the equivalent of \$100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.

- c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

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

5.3 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.

5.3.1 *What Is Covered By This Warranty*

- a) *Warranty On New Products:* ProSoft warrants, to the original purchaser, 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 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 39 months. 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.

- b) *Warranty On Services:* Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranted in 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.

5.3.2 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) This Warranty 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, "C" or any variant of "C" programming languages) 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; or (viii) disasters such as fire, flood, earthquake, wind and lightning.
- c) 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 guide included with your original product purchase 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.

5.3.3 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 and without limitation: the operation of nuclear facilities, aircraft navigation or 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.

5.3.4 Intellectual Property Indemnity

Buyer shall indemnify and hold harmless ProSoft and its employees from and against all liabilities, losses, claims, costs and expenses (including attorney's fees and expenses) related to any claim, investigation, litigation or proceeding (whether or not ProSoft is a party) which arises or is alleged to arise from Buyer's acts or omissions under these Terms or in any way with respect to the Products. Without limiting the foregoing, Buyer (at its own expense) shall indemnify and hold harmless ProSoft and defend or settle any action brought against such Companies to the extent based on a claim that any Product made to Buyer specifications infringed intellectual property rights of another party. ProSoft makes no warranty that the product is or will be delivered free of any person's claiming of patent, trademark, or similar infringement. The Buyer assumes all risks (including the risk of suit) that the product or any use of the product will infringe existing or subsequently issued patents, trademarks, or copyrights.

- a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated 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.
- f) **Additional Restrictions Relating To Software And Other Intellectual Property**

In addition to compliance 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.

5.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 55) 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.

5.3.6 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 or its Dealer will not be responsible for included, 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.

5.3.7 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 39 months following shipment of the Product.

5.3.8 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.

5.3.9 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.

5.3.10 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

A

All Product Returns • 54
Allocation of Risks • 58
Analyzing Data for Application Port • 29

B

Backplane Data Transfer • 38

C

Cable Connections • 48
Cold Boot • 43
Configuration • 13
Configuration File • 13
Control Blocks • 43
Controlling Law and Severability • 58

D

Data Analyzer • 29
Data Analyzer Tips • 31
Data Flow between MVI94-GSC-E Module and Flex
Processor • 44
DB9 to Mini-DIN Adaptor (Cable 09) • 52
Diagnostics & Troubleshooting • 23
Disclaimer of all Other Warranties • 57
Disclaimer Regarding High Risk Activities • 56
Displaying Timing Marks in the Data Analyzer • 30

E

Editing the Configuration File • 14
Exiting the Program • 29

F

FLEX I/O Interfaces • 35
Flex Processor Not in Run • 38
Functional Overview • 37
Functional Specifications • 36

G

General Concepts • 37
General Specifications • 35

H

Hardware Installation • 7
How to Contact Us
Technical Support • 53, 54

I

Important Installation Instructions • 2
Installing the Module onto the Base • 10

Installing the Serial Adapter Cables • 11
Intellectual Property Indemnity • 57

K

Keystrokes • 25

L

LED Status Indicators • 23
Limitation of Remedies ** • 58
LIMITED WARRANTY • 55

M

Main Logic Loop • 37
Main Menu • 27
Module Power Up • 37
Mounting the MVI 94 Flex I/O Base • 8
MVI (Multi Vendor Interface) Modules • 2
MVI94-GSC-E Configuration File • 15

N

Navigation • 24
No Other Warranties • 58
Normal Data Transfer • 39

O

Opening the Data Analyzer Menu • 27

P

Pinouts • 11, 48, 52
Procedures for Return of Units Out of Warranty: • 54
Procedures for Return of Units Under Warranty: • 54
Product Specifications • 35
ProSoft Technology® Product Documentation • 3
PRT1 • 23
PRT2 • 24

R

Read Block • 39
Reading Status Data from the Module • 23
Read-Only Devices • 45
Read-Write Devices • 46
Reference • 35
Removing Timing Marks in the Data Analyzer • 30
Required Hardware • 17, 25
Required Software • 17, 25
Return Material Authorization (RMA) Policies and
Conditions • 54
Returning to the Main Menu • 31
RS-232 • 49
Modem Connection • 49
Null Modem Connection (Hardware Handshaking)
• 50
Null Modem Connection (No Hardware
Handshaking) • 50
RS-232 Configuration/Debug Port • 51
RS-422 • 52
RS-485 • 51

RS-485 and RS-422 Tip • 52

S

Setting Jumpers • 9

Starting the Data Analyzer • 30

Stopping the Data Analyzer • 31

STS • 24

Support, Service & Warranty • 53

T

Termination of Received Data • 46

The Configuration/Debug Menu • 17, 24

Time Limit for Bringing Suit • 58

Transferring the Configuration File to the Module • 20

Transferring the Configuration File to Your PC • 18

U

U1 • 23

U2 • 24

Uploading and Downloading the Configuration File •
13, 17

Using the Configuration/Debug Port • 26

V

Verify Package Contents • 7

Viewing Block Transfer Statistics • 27

Viewing Data in ASCII (Text) Format • 30

Viewing Data in Hexadecimal Format • 30

Viewing Module Configuration • 28

Viewing Port Communication Status • 28

Viewing Port Configuration • 28

Viewing Version Information • 28

W

Warm Boot • 43

Warm Booting the Module • 28

Warnings • 2

What Is Covered By This Warranty • 55, 57

What Is Not Covered By This Warranty • 56

Wiring the Power Connections • 11

Write Block • 42

Write-Only Devices • 44

Y

Your Feedback Please • 3