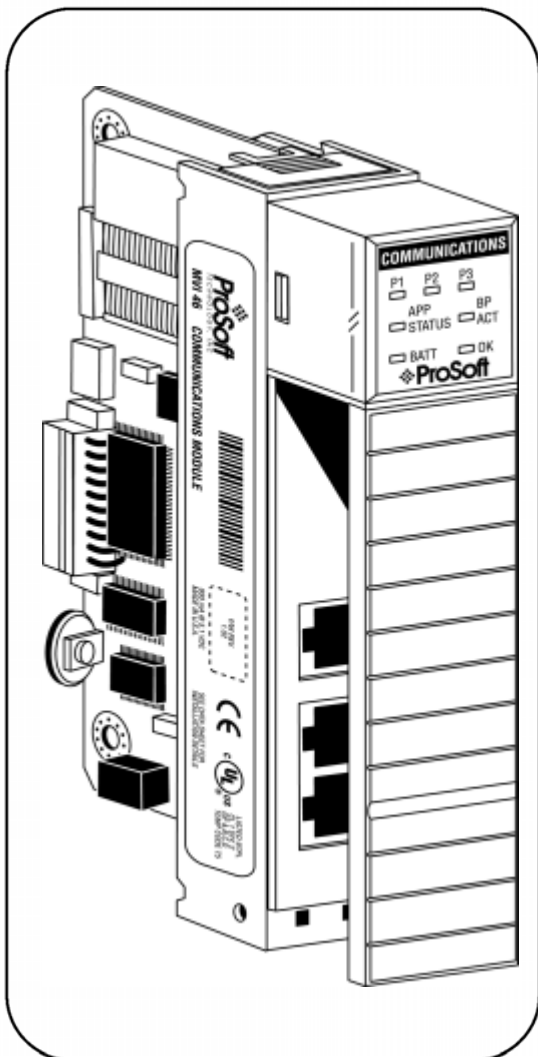


inRAx



MVI46-GSC

SLC Platform

Generic Serial Communication Module

User Manual

December 05, 2007


ProSoft
TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation SLC hardware, the MVI46-GSC Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

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

Your Feedback Please

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

ProSoft Technology

1675 Chester Avenue, Fourth Floor

Bakersfield, CA 93301

+1 (661) 716-5100

+1 (661) 716-5101 (Fax)

<http://www.prosoft-technology.com>

Copyright © ProSoft Technology, Inc. 2000 - 2007. All Rights Reserved.

MVI46-GSC User Manual

December 05, 2007

PSFT.GSC.MVI46.UM.07.12.05

ProSoft Technology®, ProLinX®, inRAX®, ProTalk® and RadioLinX® are Registered Trademarks of ProSoft Technology, Inc.

Contents

PLEASE READ THIS NOTICE	2
Your Feedback Please	2
GUIDE TO THE MVI46-GSC USER MANUAL	5
1 START HERE	7
1.1 System Requirements.....	7
1.2 Package Contents	8
1.3 Setting Jumpers.....	9
1.4 Install the Module in the Rack.....	9
1.5 Connect your PC to the Processor	11
1.6 Download the Sample Program to the Processor	12
1.6.1 Configuring RSLinx	13
1.7 Connect your PC to the Module.....	15
2 INSTALLING AND CONFIGURING THE MODULE	17
2.1 Module Configuration.....	19
2.2 Module Data	19
2.2.1 Configuration Data	19
2.2.2 Status Data	20
2.2.3 Error Word.....	21
2.2.4 Port Data	23
3 LADDER LOGIC	25
4 DIAGNOSTICS AND TROUBLESHOOTING	27
4.1 Reading Status Data from the module	27
4.1.1 The Configuration/Debug Menu	27
4.1.2 Required Hardware	28
4.1.3 Required Software.....	28
4.1.4 Using the Configuration/Debug Port.....	29
4.1.5 Main Menu	30
4.1.6 Data Analyzer.....	32
4.1.7 Data Analyzer Tips	35
4.2 LED Status Indicators	37
4.3 Clearing a Fault Condition	38
4.4 Troubleshooting	38
5 REFERENCE	39
5.1 Product Specifications	39
5.1.1 Features and Benefits	39
5.1.2 General Specifications	39
5.1.3 Hardware Specifications.....	40
5.1.4 Functional Specifications.....	40
5.2 Functional Overview	41
5.2.1 General Concepts	41
5.2.2 Data Flow between MVI46-GSC Module and SLC Processor.....	53
5.2.3 Termination of Received Data.....	57
5.3 Cable Connections	59
5.3.1 RS-232 Configuration/Debug Port.....	59
5.3.2 RS-232	61
5.3.3 RS-422	63
5.3.4 RS-485	64
5.3.5 DB9 to RJ45 Adaptor (Cable 14).....	64
5.4 MVI46-GSC I/O Images.....	65
5.4.1 Input Image	65

5.4.2	Output Image	65
5.4.3	M0: File Structure	65
5.4.4	M1: File Structure	66
6	SUPPORT, SERVICE & WARRANTY	67
6.1	How to Contact Us: Sales and Support	68
6.2	Return Material Authorization (RMA) Policies and Conditions	69
6.2.1	All Product Returns	69
6.3	Procedures for Return of Units Under Warranty	69
6.4	Procedures for Return of Units Out of Warranty	70
6.4.1	Un-repairable Units	70
6.4.2	Purchasing Warranty Extension	71
6.5	LIMITED WARRANTY	71
6.5.1	What Is Covered By This Warranty	71
6.5.2	What Is Not Covered By This Warranty	72
6.5.3	DISCLAIMER REGARDING HIGH RISK ACTIVITIES	73
6.5.4	DISCLAIMER OF ALL OTHER WARRANTIES	73
6.5.5	LIMITATION OF REMEDIES**	73
6.5.6	Time Limit for Bringing Suit	73
6.5.7	No Other Warranties	74
6.5.8	Intellectual Property	74
6.5.9	Additional Restrictions Relating To Software And Other Intellectual Property	74
6.5.10	Allocation of risks	75
6.5.11	Controlling Law and Severability	75
INDEX	77

Guide to the MVI46-GSC User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 7)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 27) Diagnostics and Troubleshooting (page 27)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 39) Functional Overview (page 41) Product Specifications (page 39)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 67)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- System Requirements 7
- Package Contents 8
- Setting Jumpers 9
- Install the Module in the Rack 9
- Connect your PC to the Processor 11
- Download the Sample Program to the Processor 12
- Connect your PC to the Module 15

Installing the MVI46-GSC module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI46-GSC Module and the application in which they will be used.



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

1.1 System Requirements

The MVI46-GSC module requires the following minimum hardware and software components:

- Rockwell Automation SLC 5/02 M0/M1 capable processors (or newer), with compatible power supply and one free slot in the rack, for the MVI46-GSC module. The module requires 800mA of available power.
- Rockwell Automation RSLogix 500 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® II 500 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
 - Microsoft® Windows 98
 - Windows NT® (version 4 with SP4 or higher)
 - Windows 2000
 - Windows XP

- 32 Mbytes of RAM minimum, 64 Mbytes of RAM recommended
- 50 Mbytes of free hard disk space (or more based on application requirements)
- 16-color VGA graphics adapter, 640 x 480 minimum resolution (256 Color 800 × 600 recommended)
- CD-ROM drive
- 3.5 inch floppy disk drive
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

1.2 Package Contents

The following components are included with your MVI46-GSC module, and are all required for installation and configuration.

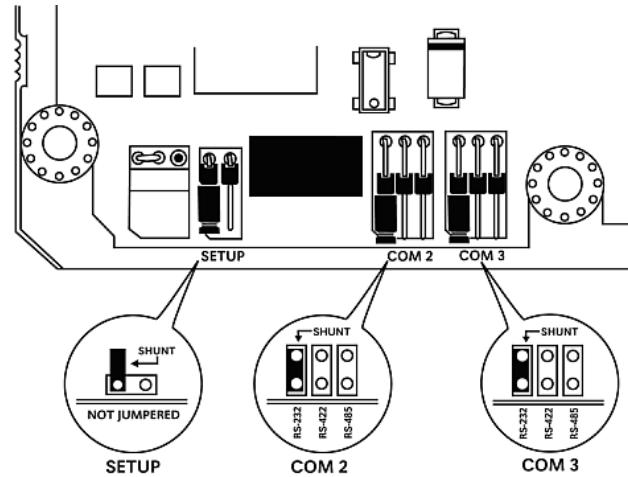
Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI46-GSC Module	MVI46-GSC	Generic Serial Communication Module
1	Cable	RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI46-GSC module.

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

1.3 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to match the interface. The following illustration shows the MVI46-GSC jumper configuration:



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

1.4 Install the Module in the Rack

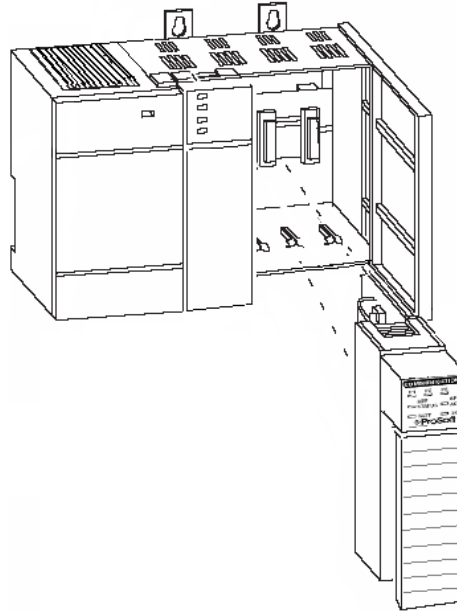
If you have not already installed and configured your SLC processor and power supply, please do so before installing the MVI46-GSC module. Refer to your Rockwell Automation product documentation for installation instructions.

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

After you have checked the placement of the jumpers, insert MVI46-GSC into the SLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install SLC™ modules.

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

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

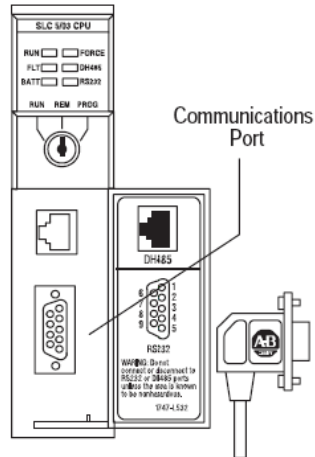


- 3 With a firm but steady push, snap the module into place.
- 4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
- 5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample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

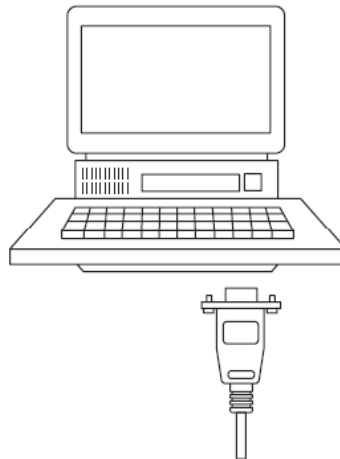
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.5 Connect your PC to the Processor

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



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

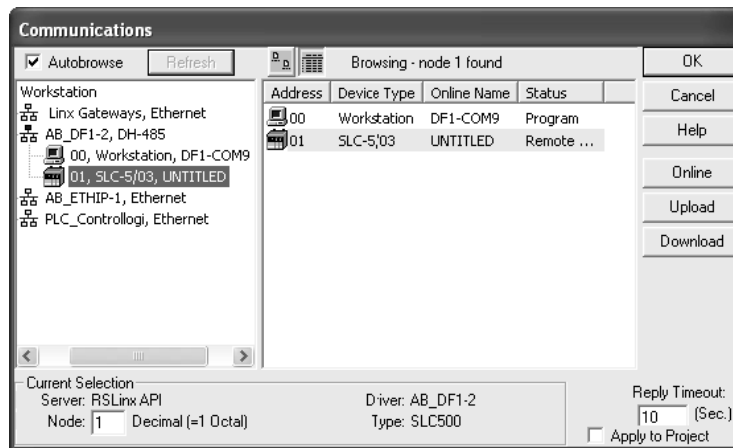


1.6 Download the Sample Program to the Processor

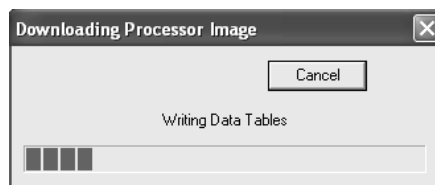
- **To download the sample program from RSLogix 500 to the SLC processor:**

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

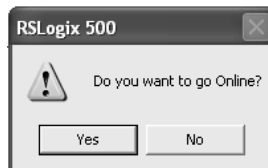
- 1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



- 2 Click the Download button to transfer the sample program to the processor.
- 3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



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

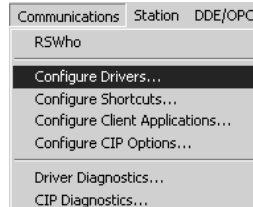


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

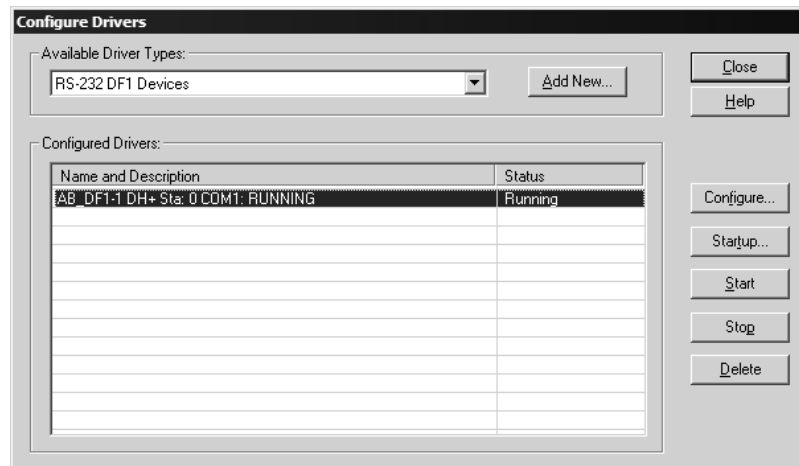
1.6.1 Configuring RSLinx

➤ **If RSLinx is unable to establish communication with the processor, follow these steps:**

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



This action opens the Configure Drivers dialog box.



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

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



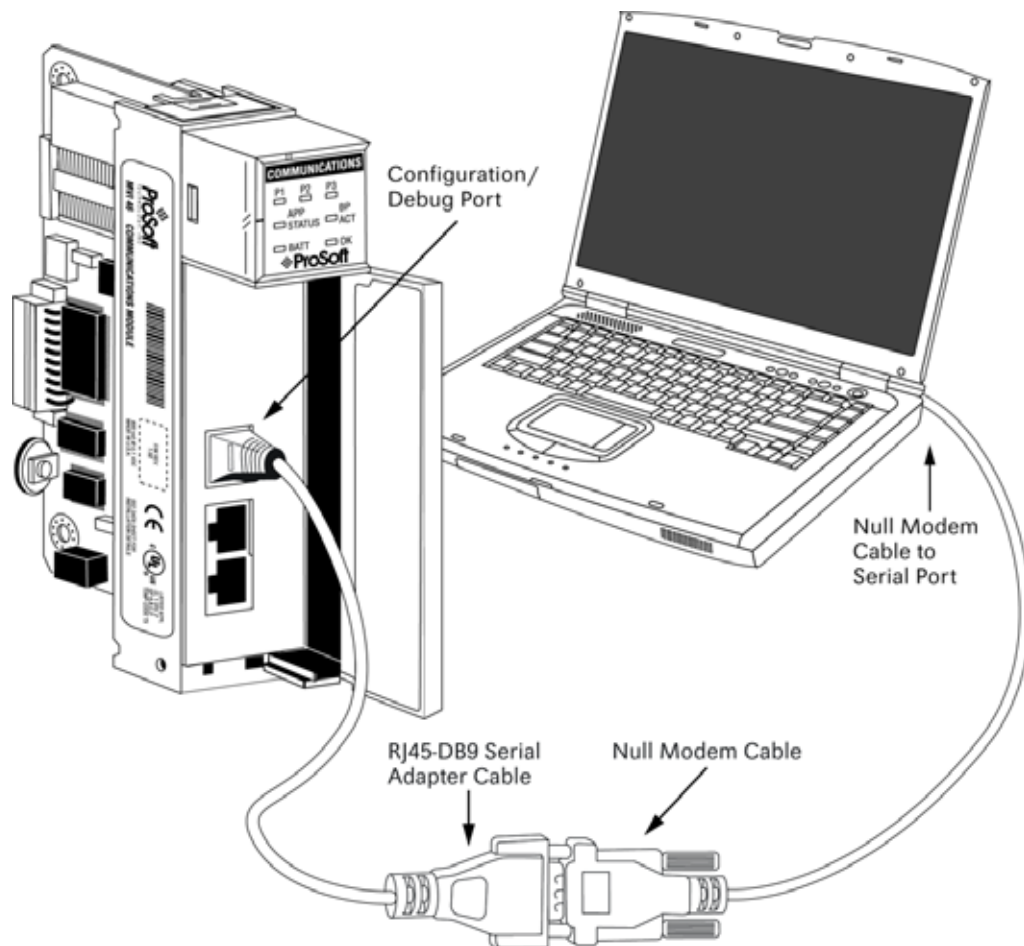
- 4 Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.
- 5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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

1.7 Connect your PC to the Module

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

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



2 Installing and Configuring the Module

In This Chapter

- Module Configuration 19
- Module Data..... 19

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

- 1 Use to identify the module to the processor and add the module to a project.

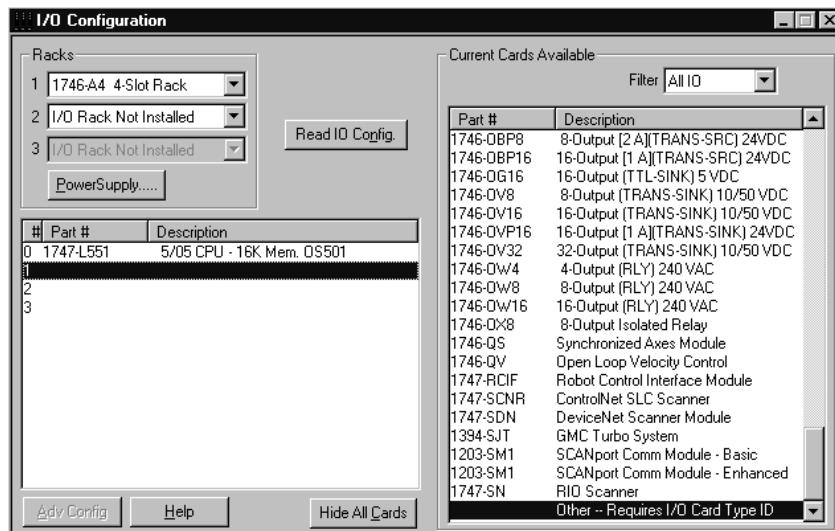
Note: The software must be in "offline" mode to add the module to a project.

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

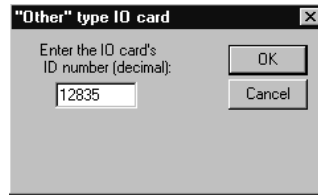
Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

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

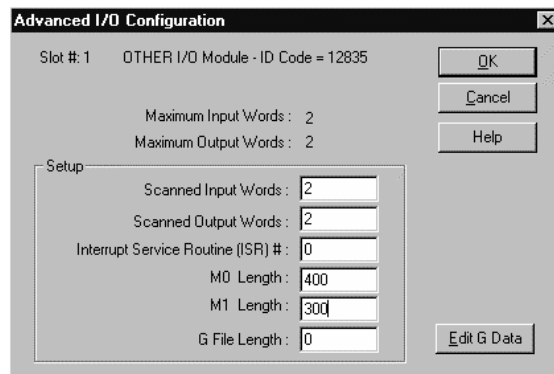
The first step in installing and configuring the module is to define the module to the system. Select the I/O Configuration option from the program screen. This will display the following dialog box:



Select the Other module from the list. This action opens the following dialog box.



Enter the module I/O card ID number as 12835, and then click OK. Double-click the mouse on the module just added to the rack. Fill in the dialog box as shown below:



Parameter	Value
Scanned Input Words	2
Scanned Output Words	2
Interrupt Service Routine (ISR#)	0
M0 Length	400
M1 Length	300
G File Length	0

Click OK to apply these settings to the module. Then, close the I/O Configuration dialog box.

The next step in the module's setup is to define the user defined data areas to hold the configuration, status and port transmit and receive data. In the example ladder logic, N10 contains the configuration data, N11 contains the status data, N12 contains the port 1 data and N13 contains the port 2 data.

At this point, take the time to fill in the configuration values in the GSC configuration data table. Refer to the Module Data section of this document for information on configuring the module.

The last step in the module setup is to add the ladder logic. If the example ladder logic is used, adjust the ladder to fit the application. When the ladder example is not used, copy the example ladder logic to your application and alter as necessary.

The module is now set up and ready to be used with your application. Turn off the power to the SLC rack and insert the module in the rack and attach the serial communication cables. Turn on the power to the rack. Download the new application to the controller and place the processor in run mode. If all the configuration parameters are set correctly and the module is attached to a serial network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink rapidly. Refer to the **Troubleshooting** (page 38) section if you encounter errors. Attach a computer or terminal to the Debug/ Configuration Port on the module and look at the status of the module using the Configuration/Debug Menu in the module.

2.1 Module Configuration

In order for the MVI46-GSC module to function, a minimum amount of configuration data must be transferred to the module. The following table provides an overview of the different types of configuration data that the module will require, depending on the operating modes to be supported.

Function	Name	Description
Data Transfer	General Module Configuration	This section of the configuration data contains the module configuration data that defines the data transfer between the module and the SLC processor.
Serial Port Drivers	Port Configuration	These sections define the characteristics of each of the serial communication ports on the module. These parameters must be set correctly for proper module operation.

The MVI46-GSC module must be configured at least once when the card is first powered, and any time thereafter when the parameters must be changed.

Changing parameters during operation

In order to change any of the module's configuration information, a warm- or cold-boot operation is required. This will force the module to request the configuration information from the processor. A warm- or cold-boot request can be generated by the processor through the ladder logic or through the Configuration/Debug port of the module.

2.2 Module Data

All data related to the MVI46-GSC is stored in user defined data areas. The user is responsible for setting up the data areas to match the specific application for which the module is used. Each data area is discussed below:

2.2.1 Configuration Data

Configuration of the module is performed by filling a user defined data table. In the example ladder logic, file N10 is used for this purpose. Each item in the file has an associated symbol and description to aid in filling in the data.

2.2.2 Status Data

This data area holds the status data transferred from the module. The following table lists the data contained in this data area.

Object in GSCInStat	M1: File Offset	Description
PassCnt	260	Program cycle counter
Product	261 to 262	Product name as ASCII string
Rev	263 to 264	Revision level as ASCII string
OP	265 to 266	Operating system level as ASCII string
Run	267 to 268	Run number as ASCII string
BlkErrs.Read	269	Number of blocks transferred from module to processor
BlkErrs.Write	270	Number of blocks transferred from processor to module
BlkErrs.Parse	271	Number of blocks parsed by module
BlkErrs.Err	272	Number of block errors in module
Port[0].RxState	273	Port 1 receive state: -1 = Listening for data 1 = Receiving Port Data 2 = Waiting for Backplane transfer
Port[0].RXCharCnt	274	Port 1 receive character count
Port[0].RxMsgCnt	275	Port 1 receive block count
Port[0].TxState	276	Port 1 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
Port[0].TxCharCnt	277	Port 1 transmit character count
Port[0].TxMsgCnt	278	Port 1 transmit block count
Port[0].ErrorWord	279	Port 1 error word
Port[1].RxState	280	Port 2 receive state: -1 = Listening for data 1 = Receiving Port Data 2 = Waiting for Backplane transfer
Port[1].RXCharCnt	281	Port 2 receive character count
Port[1].RxMsgCnt	282	Port 2 receive block count

Object in GSCInStat	M1: File Offset	Description
Port[1].TxState	283	Port 2 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
Port[1].TxCharCnt	284	Port 2 transmit character count
Port[1].TxMsgCnt	285	Port 2 transmit block count
Port[1].ErrorWord	286	Port 2 error word
Port[0].WriteCnt	287	Port 1 bytes written from last block
Port[1].WriteCnt	288	Port 2 bytes written from last block

2.2.3 Error Word

Each port contains an Error Word containing the configuration/error status flag word. These flags report configuration errors and receive overflow errors to the controller. If the module is not functioning as expected, inspect the values presented in this object. If a configuration error exists, the associated bit will be set. A value of zero for the bit indicates the configuration value is valid. This does not guarantee that the module is configured correctly for your application.

The Err_ROverflow bit is an important bit to monitor in the ladder logic. This bit is set anytime data is lost on the receive port. The module will buffer all received data until the termination condition programmed for the module exists or the receive buffer reaches a size of 4096 bytes. When the termination condition is met, the received data will be transferred from the module to the processor. Data will be transferred in blocks of 256 bytes per transfer. If data is received on the port and the termination condition is present and the last received data set has not been completely, the Err_ROverflow bit will be set for the port and the newly received data will be flushed from the receive buffer. Therefore, it will be lost. This error flag is important to monitor because it will indicate when data is lost on the receive port. The ladder logic must handle the data as fast as the module receives it.

The following table defines each bit in the error word:

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. Values between 0 and 4 are accepted. The module has set the parity to a value of none (0).
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.
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. The value entered must be in the range of 0 to 3. The module assumes a value of 0 (no handshaking).
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.
Cfg_Rswap	Bit 11	Invalid receive swap code (0 or 1)
Cfg_WSwap	Bit 12	Invalid transmit swap code (0 or 1)
	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.

2.2.4 Port Data

Each port should have a transmit and receive data file to hold the data to interface with the module. In the supplied example, a single file is used for both operations for each port. All data received on the port is echoed back on the port through the ladder logic.

3 Ladder Logic

Ladder logic is required for application of the MVI46-GSC module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

4 Diagnostics and Troubleshooting

In This Chapter

- Reading Status Data from the module 27
- LED Status Indicators 37
- Clearing a Fault Condition..... 38
- Troubleshooting 38

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

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

4.1 Reading Status Data from the module

The MVI46-GSC module returns four separate status data areas (page 20) to the SLC processor in each read block. This data is transferred to the SLC processor continuously with each read block.

4.1.1 The Configuration/Debug Menu

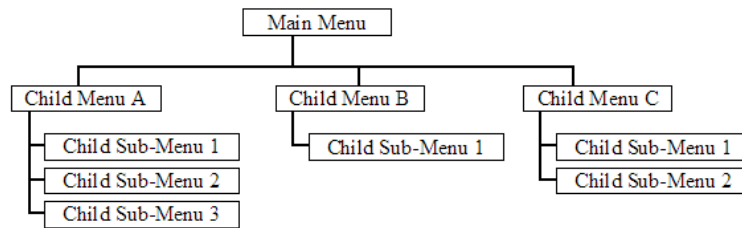
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the terminal application (for example, HyperTerminal). The module does not respond to mouse movements or clicks. The command executes as soon as you press the command letter — you do not need to press **[Enter]**. When you type a command letter, a new screen will be displayed in your terminal application.

Navigation

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

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



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

Keystrokes

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

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

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

4.1.2 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information and perform maintenance.

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 serial communications port available
- A null modem serial cable.

4.1.3 Required Software

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

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

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

4.1.4 Using the Configuration/Debug Port

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

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

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

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

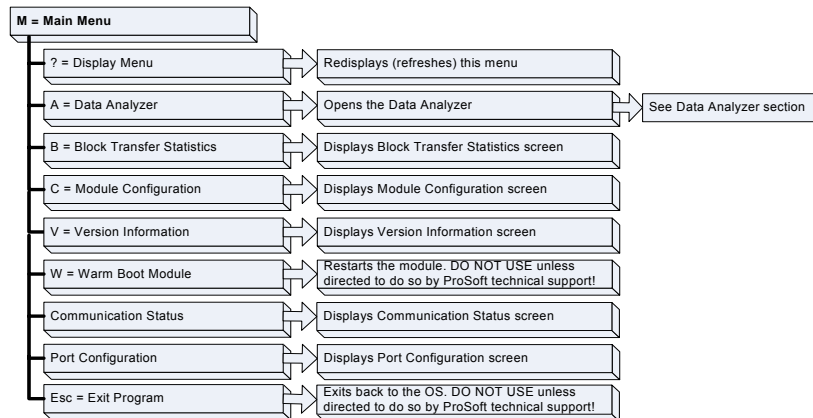
If there is no response from the module, follow these steps:

- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 Verify that RSLinx is not controlling the COM port. Refer to [Disabling the RSLinx Driver for the Com Port on the PC](#) (page 59).
- 3 Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
- 4 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

4.1.5 Main Menu

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



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 COMMUNICATION MODULE <MVI46-GSC> MENU
? = Display Menu
A = Data Analyzer
B = Block Transfer Statistics
C = Module Configuration
U = Version Information
W = Warm Boot Module
Communication Status : 1 = Port 1   2 = Port 2
Port Configuration  : 6 = Port 1   7 = Port 2

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]** or **[2]** from the Main Menu to view the port communication status for Ports 1 and 2.

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

Viewing Port Configuration

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

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

Exiting the Program

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

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

4.1.6 Data Analyzer

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

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

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

4.1.7 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

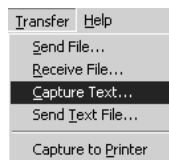
Port = 1, Format=HEX, Tick=10
```

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

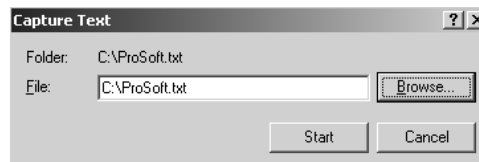
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).


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 your 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><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 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][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]
[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 TT
```

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

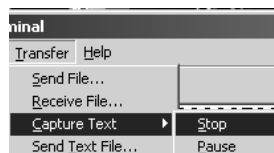
All characters shown in <> brackets are characters being sent out by the module.

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

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

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

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



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

4.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the serial network on its Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the serial network on its Port 2.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI46-GSC module program has recognized a communication error on one of its ports or the SLC is not in run mode.
		On	The MVI46-GSC is working normally.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Turn off the power to the rack, remove the card from the rack and re-insert the card to restart the module's program. Restore power to the rack.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact the factory, as this is not a user serviceable item.

4.3 Clearing a Fault Condition

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

To clear the condition, follow these steps:

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

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

4.4 Troubleshooting

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

Processor Errors

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

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: <ul style="list-style-type: none"> ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

5 Reference

In This Chapter

➤ Product Specifications.....	39
➤ Functional Overview.....	41
➤ Cable Connections.....	59
➤ MVI46-GSC I/O Images	65

5.1 Product Specifications

The MVI46 Generic ASCII Serial Interface Module is a fast and easy way to add two fully configurable ASCII communication ports to the SLC platform without consuming the front port of the processor, or using valuable processing time.

5.1.1 Features and Benefits

The MVI46-GSC module is a single slot, backplane compatible solution for the SLC platform. This module has two powerful and fully configurable ASCII communication ports, each supporting the sending and receiving of large ASCII character strings. With the implementation of some supporting ladder logic in the processor, the many different devices supporting ASCII communications can be integrated into the SLC platform.

5.1.2 General Specifications

- Single Slot – 1746 backplane compatible (Local or extended I/O rack only. Remote rack not supported)
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module using M0/M1 files
- Ladder Logic is used for data transfer between module and processor
- Configuration data obtained through user-defined ladder. Sample ladder file included

5.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 ma @ 5V (from backplane)
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
Relative Humidity	5% to 95% (non-condensing)
Vibration	5 g from 10150 Hz
Processor	Compatible with Rockwell Automation SLC 5/02 M0/M1 capable processors or newer
LED indicators	Module status, Backplane transfer status, Application status, Serial activity and error LED status
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Application Serial port (PRT1, PRT2) (Serial Modules)	Two RJ45 RS-232/422/485 Application ports

5.1.4 Functional Specifications

- ASCII Communication ports 1 & 2 (PRT1, PRT2)
 - Both ports are capable of transmitting and/or receiving ASCII character strings. Each port is individually configurable:
- Termination types: Stream mode, Termination character(s), Message timeout, Intercharacter timeout and/or Packet size limit
 - Baud rate: 110 to 115.2K baud
 - Parity: none, even, odd
 - Stop bits: 1 or 2
 - Data bits: 5 to 8
 - RTS on/off timing: 0 to 65535 milliseconds
 - Minimum response delay: 0 to 65535 milliseconds
- Handshaking (optional)
 - Hardware: RTS/CTS, DTR/DSR
 - Software: XON/XOFF
- Receive ASCII character strings up to 4096 characters in length
- Full hardware handshaking control provides radio, modem and multi-drop support
- Module error and status conditions returned to processor for diagnostic purposes
 - Module status
 - Port error status word (bit mapped)
 - Port receive state
 - Port receive character count
 - Port receive block count

- Port transmit state
- Port transmit character count
- Port transmit block count
- All data related to the module is contained in a single controller tag with defined objects to simplify configuration, monitoring and interfacing with the module
- Module configuration and communication configuration data is transferred to the module via a predefined user data type in the processor

5.2 Functional Overview

This section provides an overview of how the MVI46-GSC module transfers data using the GSC protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI46-GSC module.

Module Power Up

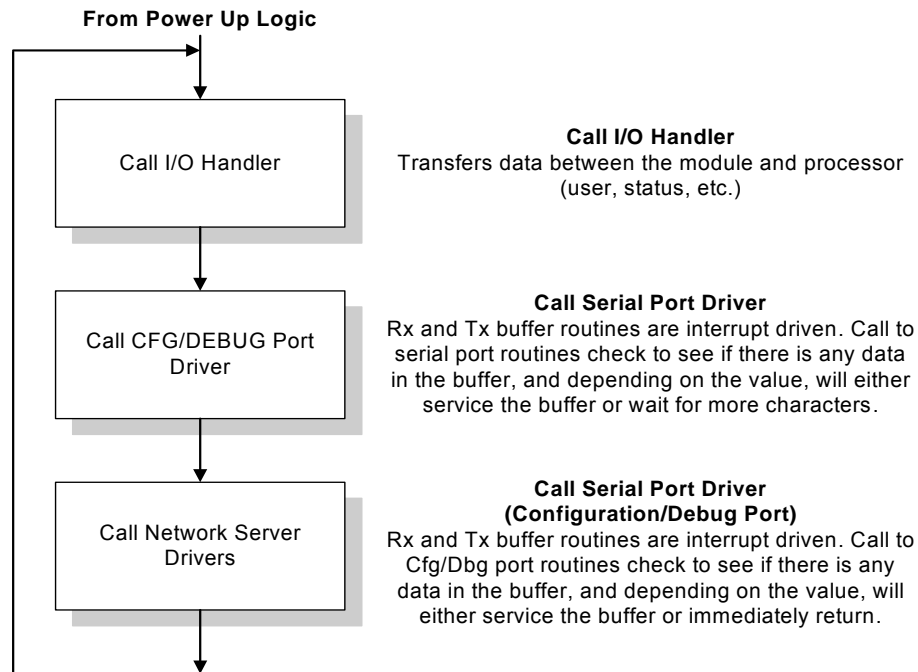
On power-up, the module performs the following logical functions:

- 1 Initialize hardware components
 - Initialize SLC backplane driver
 - Test and Clear all RAM
 - Initialize the serial communication ports
- 2 Wait for Module Configuration from SLC processor
- 3 Initialize Communication ports
- 4 Set up the serial communication interface driver on the selected ports

After the module has received the Module Configuration Block from the processor and the module is configured, the module will begin receiving and transmitting messages with devices on the serial ports.

Main Logic Loop

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



SLC Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (that is, Fault or PGM), the application ports can be shut down as prescribed in the user configuration. When the processor is returned to a running state, the module will resume communications on the serial networks. Data transfer will not occur across the backplane when the processor is not in run mode.

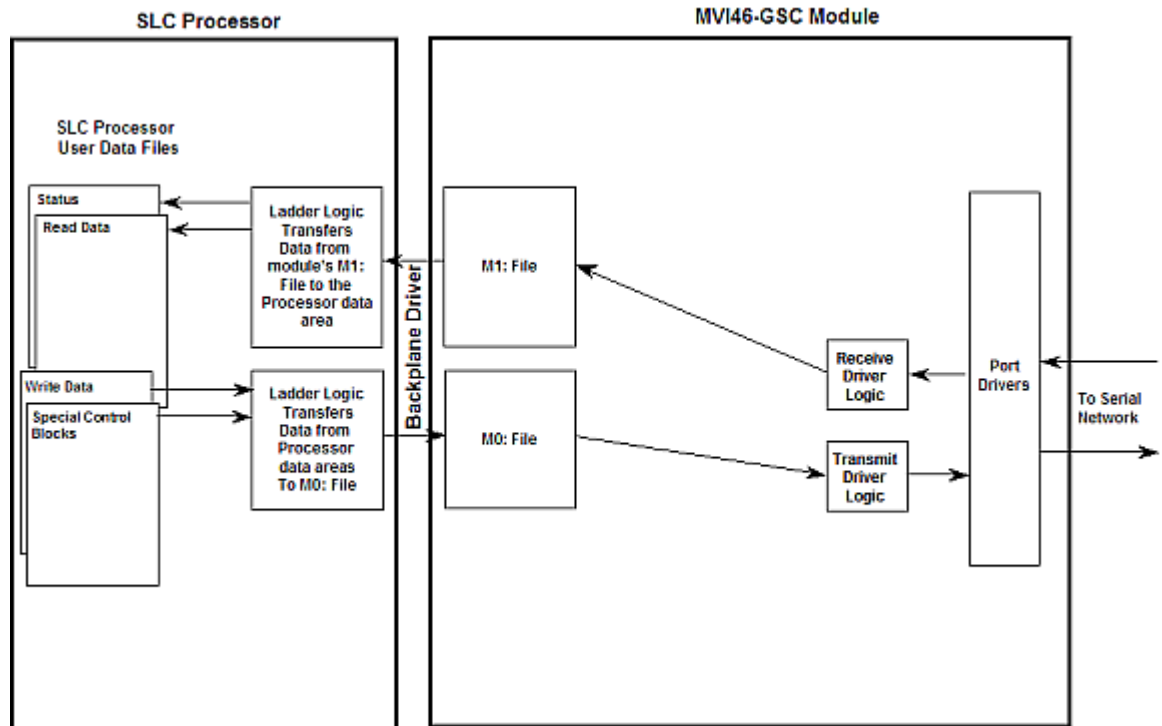
Backplane Data Transfer

The MVI46-GSC module communicates directly over the SLC backplane. Data travels between the module and the SLC processor across the backplane using the module's M-Files coordinated using bits in the input and output image of the module. The update frequency of the images is determined by the scan rate of the ladder logic.

Data received on the application ports is placed in the module's M1 file. This data is processed by the ladder logic in the SLC processor. Each port has a reserved data area of 128 words (256 bytes) to hold this data (4096 bytes maximum per message). This large data area permits fast throughput of data between the module and the processor.

The processor inserts data in the module's M0 file to transfer to the module. The module's program extracts the data and transmits the data out the communication ports. Each port has a data area of 128 words (256 bytes) set aside for the write operation. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the SLC processor, the MVI46-GSC module and the serial devices.



As shown in the diagram above, all data transferred between the module and the processor over the backplane is through the M0 and M1 files. Ladder logic must be written in the SLC processor to interface the M-Files data with data defined in the SLC user data files. The user is responsible for handling and interpreting all data received on the application ports and transferred in the M1 file. Additionally, the user is responsible for constructing messages to be transferred out of the application ports by building the messages in the M0 file of the module.

Normal Data Transfer

Normal data transfer includes the transferring of data received or to be transmitted on the ports and the status data. These data are transferred through the M1 file using read blocks and M0 file using write blocks. Each port has a reserved area in each of these files for the data transfer operation. In order to coordinate the data transfer between the module and the SLC, bit 4 in the input and output images is used. When the bits are equal, the module has control of the M-files. When the bits are not equal, the SLC has control of the M-files. The SLC's ladder logic must set the bits equal after interfacing with the data exchange areas and must not use the data in the files until the bits are not equal.

The structure and function of each block is discussed below:

Read Block

These blocks of data transfer information from the module to the SLC processor. The structure of the M1 file used to transfer this data is shown below:

Port 1

M1: File Offset	Description
0	Sequence block number for this data.
1	Number of characters (0 to 256) in Port 1 receive block (words 2 to 129). If the receive data in the module is larger than 256 bytes, multiple blocks will be transferred. Any block with a value of -1 in this field represents the first or continuation block and the block contains 256 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.
2 to 129	Port 1 data received (up to 128 words or 256 bytes)

Port 2

M1: File Offset	Description
130	Number of characters (0 to 256) in Port 2 receive block (words 2 to 129). If the receive data in the module is larger than 256 bytes, multiple blocks will be transferred. Any block with a value of -1 in this field represents the first or continuation block and the block contains 256 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.
131 to 258	Port 2 data received (up to 128 words or 256 bytes)
259	Reserved

Module Status

M1: File Offset	Description
260	Program cycle counter
261 to 262	Product name as ASCII string
263 to 264	Revision level as ASCII string
265 to 266	Operating system level as ASCII string
267 to 268	Run number as ASCII string
269	Number of blocks transferred from module to processor

M1: File Offset	Description
270	Number of blocks transferred from processor to module
271	Number of blocks parsed by module
272	Number of block errors in module

Port 1 Status

M1: File Offset	Description
273	Port 1 receive state: -1 = Listening for data 1 = Receiving Port Data 2 = Waiting for Backplane transfer
274	Port 1 receive character count
275	Port 1 receive block count
276	Port 1 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
277	Port 1 transmit character count
278	Port 1 transmit block count
279	Port 1 error word

Port 2 Status

M1: File Offset	Description
280	Port 2 receive state: -1 = Listening for data 1 = Receiving Port Data 2 = Waiting for Backplane transfer
281	Port 2 receive character count
282	Port 2 receive block count

M1: File Offset	Description
283	Port 2 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
284	Port 2 transmit character count
285	Port 2 transmit block count
286	Port 2 error word

Write Cnt

M1: File Offset	Description
287	Port 1 bytes written in last block
288	Port 2 bytes written in last block

The Sequence Block Number (word 0) is an index value used to signal to the SLC 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.

Important: The Sequence Block Number is generated by the module and sent in the input image. After the ladder has processed inputs and set up outputs, the module expects to see this number returned back in word offset 100 of the Output Image. When the module sees this value there, it assumes the ladder has completely processed the last Input Image and is ready for another. Until it sees the SBN it sent in the Input Image returned in the Output Image, then it will not set up a new Input Image.

The block contains the data received on each port and status data. The two word values 1 (port 1 receive length) and 130 (port 2 receive length) define the number of bytes to be processed by the ladder logic for each port. Receive data for the ports are found starting at words 2 and 131 for port 1 and 2, respectively. The ladder logic is responsible for handling all read data transferred from the communication ports to the processor. The simple example assumes the received data block is less than or equal to 128 words.

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 256 bytes is more complex. The module buffers the data in this buffer until one of the specified termination condition is recognized. The module will then transfer the received block of data to the controller. If the block of data received is larger than 256 bytes, multiple blocks will be used to transfer the data to the controller. The first block will contain a value of -1 in the Number of Characters Received data field. This indicates that there will be more blocks to follow and that the current block contains 256 bytes of data. As long as more than 256 bytes are present in the buffer being sent to the controller, the length field will be set to a value of -1. When 256 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.

The two word values 287 (port 1 transmit count) and 288 (port 2 transmit count) inform the processor of the number of bytes transferred in the last write block to the respective port transmit buffers. If a value of zero is returned in one of these words and data was sent in the last write block, the ladder logic must re-send the data in the next write block as the port is in a busy state and cannot buffer the new data. If a value is returned in one of these words, the value represents the number of bytes in the last write block moved into the port's transmit buffer.

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 for each port. This value contains the configuration error flags for each port and the receive buffer overflow error flag.

Write Block

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

The Block Sequence Number is that received on the last read block transfer through the M1 file image on the module. The ladder logic should copy this value from word 0 of the M1 file to word 100 of M0 file.

M0: File Offset	Description
100	Block Sequence Number (Read block number as set by module)
101	Intercharacter delay for this message (milliseconds between characters)

Port 1

M0: File Offset	Description
102	Number of characters to transmit on Port 1 (0 to 256)
103 to 230	Port 1 data to transmit
231	Intercharacter delay for this message (milliseconds between characters)

Port 2

M0: File Offset	Description
232	Number of characters to transmit on Port 2 (0 to 256)
233 to 360	Port 2 data to transmit

Important: The Sequence Block Number is generated by the module and sent in the input image. After the ladder has processed inputs and set up outputs, the module expects to see this number returned back in word offset 100 of the Output Image. When the module sees this value there, it assumes the ladder has completely processed the last Input Image and is ready for another. Until it sees the SBN it sent in the Input Image returned in the Output Image, then it will not set up a new Input Image.

If the number of characters to transmit in the write block is not set to zero (words 102 and 232 in the M0 file), data to transmit is present in the block. If the selected port is not busy transmitting data, the data in the block will be moved to the port's transmit buffer and sent out the port.

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 (words 101 or 231) to a value of 200 in the module's M0 file 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.

Configuration Data Transfer

When the module performs a restart operation, it will request configuration information from the SLC processor. This data is transferred to the module in the M0 file using registers 0 to 64. The module will indicate its request for the data by setting the Configuration Request Bit in the input image (bit 0). The ladder logic must recognize this bit and fill the M0 file with the module configuration information stored in a user file. After completing this task, the ladder logic must set the Configuration Set and Ready bit in the output image (bit 0). The module will recognize this bit and use the configuration in the M0 file to set up the module. After the module is completely configured, the Configuration Complete bit will be set in the input image (bit 1). This indicates to the ladder logic that normal data transfer can now be performed. The format of the configuration block is shown below:

Backplane Setup

Parameter	N-File Location	M0: Offset	Description
BPFail	N10:0	0	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.

Port 1 Setup

Parameter	N-File Location	M0: Offset	Description
Port[0].Enabled	N10:1	1	This parameter defines if this port will be utilized. If the parameter is set to 0, the port is disabled. A value of 1 will enable the port.
Port[0].Type	N10:2	2	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.
Port[0].Baudrate	N10:3	3	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, 300, 600, 1200, 2400, 4800, 9600, 19200 384 for 38400, 576 for 57600, and 115 for 115200.
Port[0].Parity	N10:4	4	This is the Parity code to be used for the port. The coded values are as follows: 0=None, 1=Odd, 2=Even, 3=Mark and 4=Space.
Port[0].DataBits	N10:5	5	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.
Port[0].StopBits	N10:6	6	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.
Port[0].RTSON	N10:7	7	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.
Port[0].RTSOFF	N10:8	8	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.
Port[0].Handshaking	N10:9	9	This parameter specifies the handshaking used on the port. The code values are as follows: 0=No hardware or software handshaking, 1=RTS/CTS hardware handshaking, 2=DTR/DSR hardware handshaking and 3=XON/XOFF software handshaking.

Parameter	N-File Location	M0: Offset	Description
Port[0].RTermCnt	N10:10	10	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.
Port[0].RTermChar	N10:11 to 16	11 to 16	This array of 12 characters define the termination characters at the end of each received message. Each character occupies one position in the array. The number of characters to be used in the array is set in the RTermCnt parameter.
Port[0].RPacketLen	N10:17	17	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.
Port[0].RTimeout	N10:18	18	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.
Port[0].RDelay	N10:19	19	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.
Port[0].RSwap	N10:20	20	This parameter specifies if the bytes received on the port must be swapped before sending them to the PLC processor. 0=No or 1=Swap bytes
Port[0].WTermCnt	N10:21	21	Not used in current release of product.
Port[0].WTermChar	N10:22 to 27	22 to 27	Not used in current release of product.
Port[0].WPacketLen	N10:28	28	Not used in current release of product.
Port[0].WTimeout	N10:29	29	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.
Port[0].WDelay	N10:30	30	Not used in current release of product.
Port[0].WMinDelay	N10:31	31	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.
Port[0].Wswap	N10:32	32	This parameter specifies if the bytes to transmit must be swapped. 0=No or 1=Swap bytes

Port 2 Setup

Parameter	N-File Location	M0: Offset	Description
Port[1].Enabled	N10:33	33	This parameter defines if this port will be utilized. If the parameter is set to 0, the port is disabled. A value of 1 will enable the port.
Port[1].Type	N10:34	34	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.
Port[1].Enabled	N10:33	33	This parameter defines if this port will be utilized. If the parameter is set to 0, the port is disabled. A value of 1 will enable the port.
Port[1].Type	N10:34	34	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.
Port[1].Baudrate	N10:35	35	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, 300, 600, 1200, 2400, 4800, 9600,19200 384 for 38400, 576 for 57600, and 115 for 115200.
Port[1].Parity	N10:36	36	This is the Parity code to be used for the port. The coded values are as follows: 0=None, 1=Odd, 2=Even, 3=Mark and 4=Space.
Port[1].DataBits	N10:37	37	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.
Port[1].StopBits	N10:38	38	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.
Port[1].RTSOn	N10:39	39	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.
Port[1].RTSOFF	N10:40	40	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.
Port[1].Handshaking	N10:41	41	This parameter specifies the handshaking used on the port. The code values are as follows: 0=No hardware or software handshaking, 1=RTS/CTS hardware handshaking, 2=DTR/DSR hardware handshaking and 3=XON/XOFF software handshaking.

Parameter	N-File Location	M0: Offset	Description
Port[1].RTermCnt	N10:42	42	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.
Port[1].RTermChar	N10:43 to 48	43 to 48	This array of 12 characters define the termination characters at the end of each received message. Each character occupies one position in the array. The number of characters to be used in the array is set in the RTermCnt parameter.
Port[1].RPacketLen	N10:49	49	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.
Port[1].RTimeout	N10:50	50	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.
Port[1].RDelay	N10:51	51	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.
Port[1].RSwap	N10:52	52	This parameter specifies if the bytes received on the port must be swapped before sending them to the PLC processor. 0=No or 1=Swap bytes
Port[1].WTermCnt	N10:53	53	Not used in current release of product.
Port[1].WTermChar	N10:54 to 59	54 to 59	Not used in current release of product.
Port[1].WPacketLen	N10:60	60	Not used in current release of product.
Port[1].WTimeout	N10:61	61	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.
Port[1].WDelay	N10:62	62	Not used in current release of product.
Port[1].WMinDelay	N10:63	63	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.
Port[1].Wswap	N10:64	64	This parameter specifies if the bytes to transmit must be swapped. 0=No or 1=Swap bytes

Command Control Blocks

Command control blocks are special blocks used to control the module or request special data from the module. The current version of the software supports two command control blocks: warm boot and cold boot.

Warm Boot

This block is sent from the SLC processor to the module when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made. This will force the module to read the new configuration information and to restart. To warm boot the module, place a value of 254 in word 100 of the M0 file.

Cold Boot

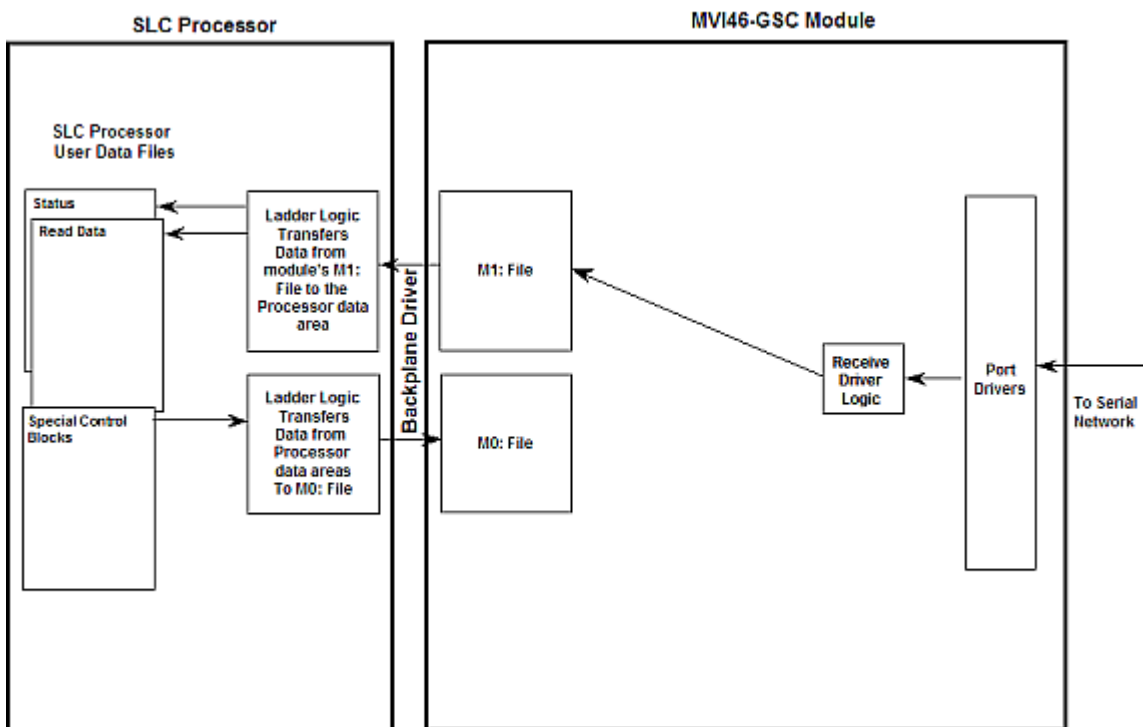
This block is sent from the SLC processor to the module when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. To cold boot the module, place a value of 253 in word 100 of the M0 file.

5.2.2 Data Flow between MVI46-GSC Module and SLC Processor

The following topics describe the flow of data between the two pieces of hardware (SLC processor and MVI46-GSC module) and other devices attached to the application ports. Each application port on the module is configured independently to interface with serial communication devices. The sections below show the three possible types of communication devices that can be attached to the application ports: write-only, read-only and read-write.

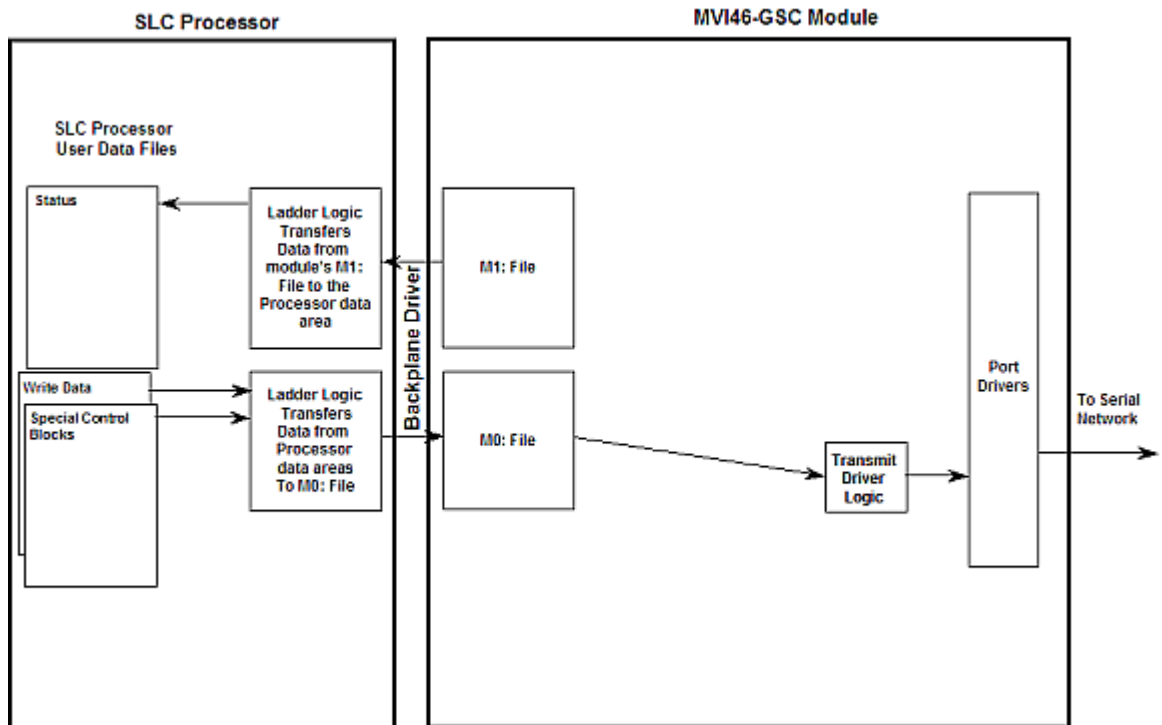
Write-Only Devices

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 MVI46-GSC module will never transmit data. All data received from the device will be passed from the module to the SLC processor through the module's M1 file. Ladder logic in the processor must handle the data received from the module. Word 100 of the M0 file will only be used to inform the module when the input image has been processed. This is accomplished by copying word 0 of the M1 file to word 100 of the M0 file. The data flow diagram for a write-only device is shown below:



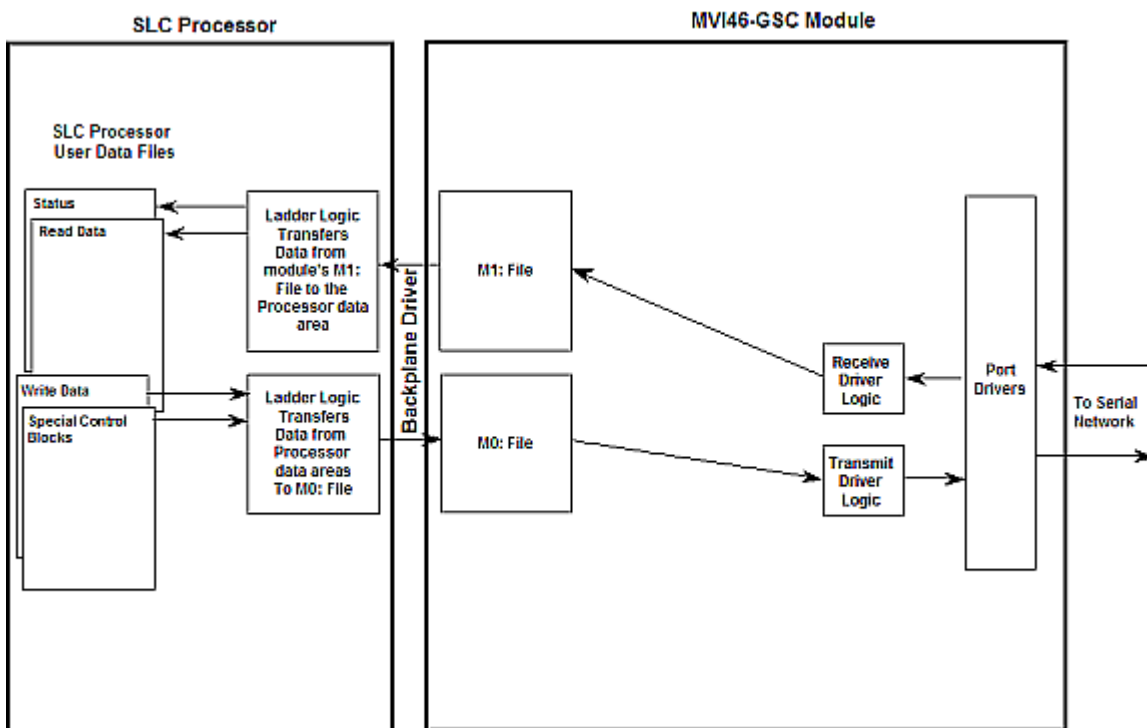
Read-Only Devices

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 should be the one received on the last read block. The data flow diagram for a read-only device is shown below:



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 SLC processor through the MVI46-GSC's M1 file. Data to be written to the terminal will be sent to the module using the M0 file. The module will send new data in the M0 file out the application port to the terminal. The example shipped with the module (MVI46-GSC.RSS) can be interfaced with a terminal to echo back all characters received. The data flow diagram for a read-write device is shown below:



5.2.3 Termination of Received Data

When data is received on either of the application ports, you must define in the configuration when this data will be transferred to the SLC 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's configuration. 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 M1 file 256 bytes at a time. The ladder logic must handle this condition if the configuration of the module would permit this situation.

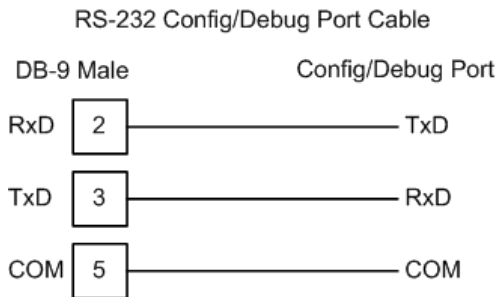
5.3 Cable Connections

The application ports on the MVI46-GSC module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

5.3.1 RS-232 Configuration/Debug Port

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

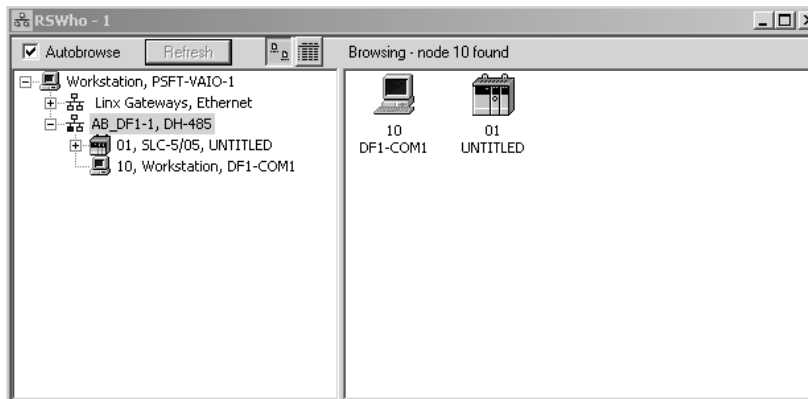


Disabling the RSLinx Driver for the Com Port on the PC

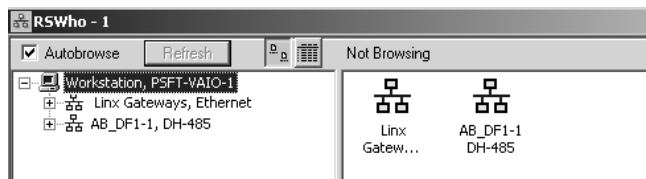
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using HyperTerminal or a similar terminal emulator, follow these steps to disable the RSLinx Driver.

- 1 Open RSLinx and go to Communications>RSWho

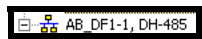
- 2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

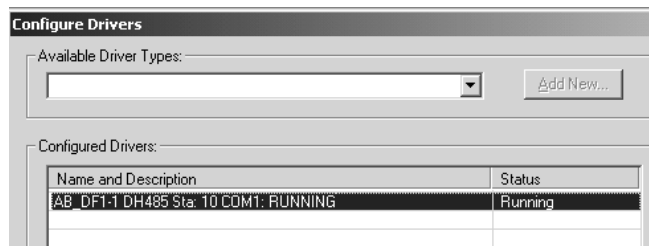


Branches are displayed or hidden by clicking on the  or the  icons.



- 4 When you have verified that the driver is not being browsed, go to **Communications>Configure Drivers**

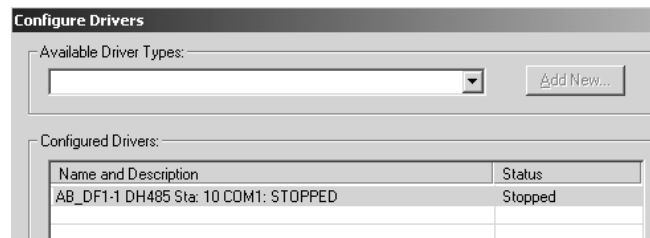
You may see something like this:



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



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

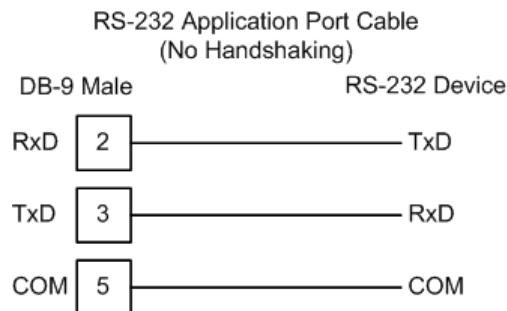


- 6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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

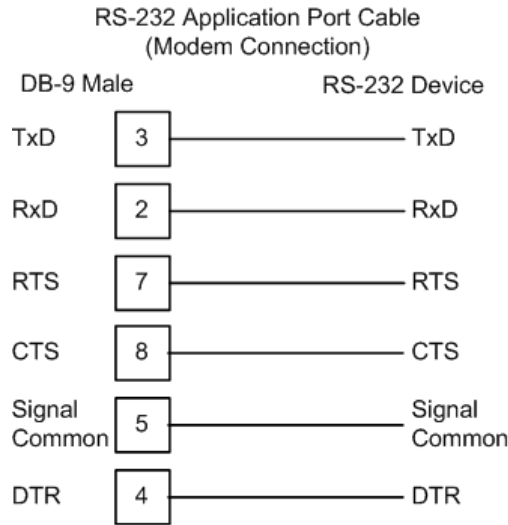
5.3.2 RS-232

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



RS-232: Modem Connection

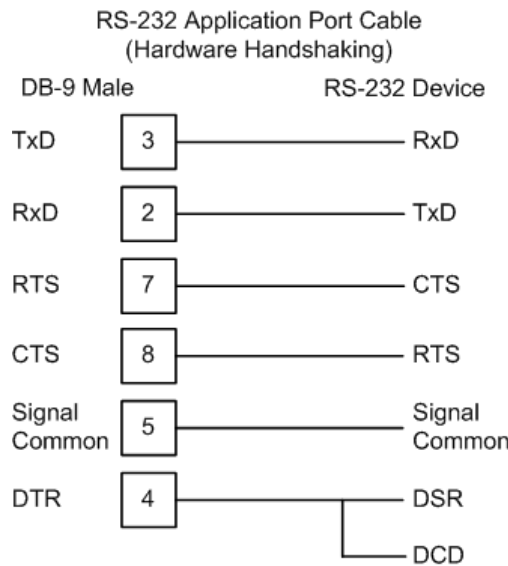
This type of connection is required between the module and a modem or other communication device.



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

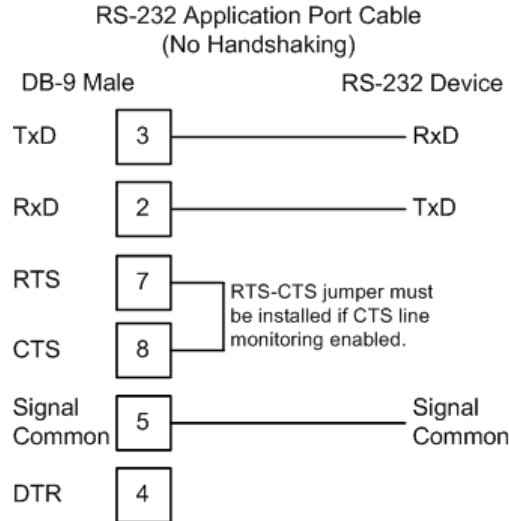
RS-232: Null Modem Connection (Hardware Handshaking)

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

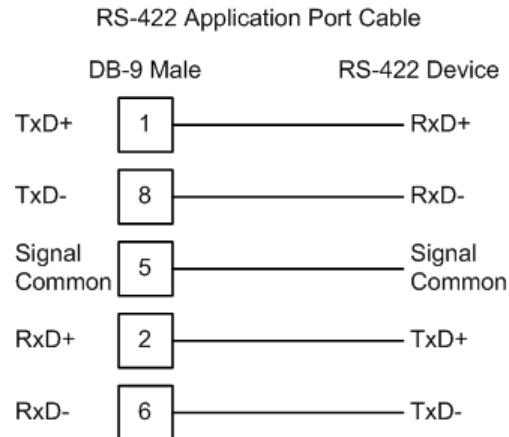


RS-232: Null Modem Connection (No Hardware Handshaking)

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

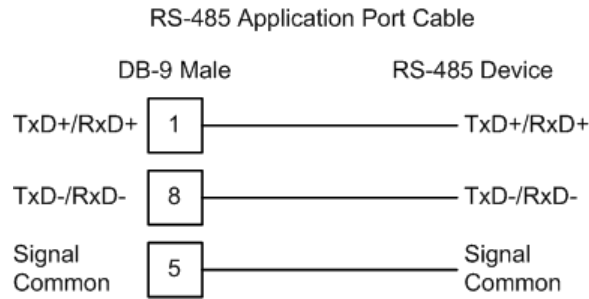


Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.3 RS-422

5.3.4 RS-485

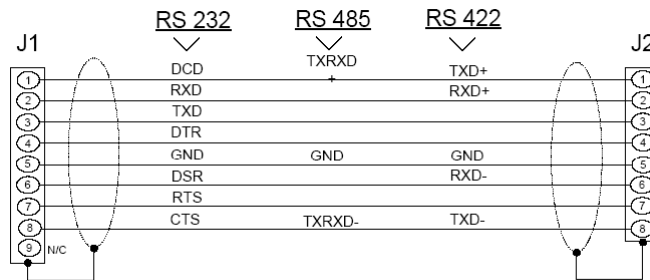
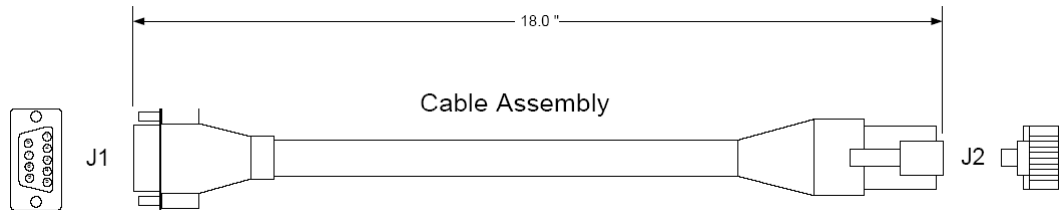
The RS-485 interface requires a single two or three wire cable. The Common connection is optional and dependent on the RS-485 network. The cable required for this interface is shown below:



RS-485 and RS-422 Tip

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

5.3.5 DB9 to RJ45 Adaptor (Cable 14)



Wiring Diagram

5.4 MVI46-GSC I/O Images

This section contains a description of the input/output and M-file structures.

5.4.1 Input Image

Bit	Description
0.0	Configuration Requested by Module
0.1	Configuration Complete
0.2	Reserved
0.3	Reserved
0.4	Block Handshake bit
0.5	Reserved
0.6	Reserved
0.7	Reserved
All others	Reserved

5.4.2 Output Image

Bit	Description
0.0	Configuration Set and Ready by SLC
0.1	Reserved
0.2	Reserved
0.3	Reserved
0.4	Block Handshake bit
0.5	Reserved
0.6	Reserved
0.7	Reserved
All others	Reserved

5.4.3 M0: File Structure

Offset	Description
0 to 64	Configuration
65 to 99	Reserved for Future
100	Write block sequence number
101	P1 intercharacter delay for this message
102	P1 write data length
103 to 230	P1 write data
231	P2 intercharacter delay for this message
232	P2 write data length
233 to 360	P2 write data
361 to 399	Reserved

5.4.4 M1: File Structure

Offset	Description
0	Block sequence number
1	P1 number of bytes received
2 to 129	P1 data received
130	P2 number of bytes received
131 to 258	P2 data received
259	Reserved
260 to 288	Status data
289 to 299	Reserved

6 Support, Service & Warranty

In This Chapter

- How to Contact Us: Sales and Support..... 68
- Return Material Authorization (RMA) Policies and Conditions 69
- Procedures for Return of Units Under Warranty..... 69
- Procedures for Return of Units Out of Warranty..... 70
- LIMITED WARRANTY 71

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

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

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

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

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

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

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

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

6.1 How to Contact Us: Sales and Support

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

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

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

Asia Pacific (office in Malaysia)

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

China Pacific (office in China)

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

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

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

Brasil (office in Sao Paulo)

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

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns

- 1 In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
- 2 In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above in Section A. A Technical Support Engineer will request several tests in an attempt to isolate the problem. If after these tests are completed, the Product is found to be the source of the problem, ProSoft will issue an RMA.
- 3 All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- 4 Out of warranty returns are not allowed on RadioLinx accessories such as antennas, cables, and brackets.

The following policy applies for Non-Warranty Credit Returns:

- A 10% Restocking Fee if Factory Seal is *not* broken
- B 20% Restocking Fee if Factory Seal is broken

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

6.3 Procedures for Return of Units Under Warranty

- 1 A Technical Support Engineer must pre-approve all product returns.
- 2 Module is repaired or replaced after a Return Material Authorization Number is entered and a replacement order is generated.
- 3 Credit for the warranted item is issued within 10 business days after receipt of product and evaluation of the defect has been performed by ProSoft. The credit will only be issued provided the product is returned with a valid Return Material Authorization Number and in accordance with ProSoft's shipping instructions.

- a) If no defect is found, a credit is issued.
- b) If a defect is found and is determined to be customer generated or if the defect is otherwise not covered by ProSoft's Warranty, or if the module is not repairable, a credit is not issued and payment of the replacement module is due.

6.4 Procedures for Return of Units Out of Warranty

- 1 Customer sends unit in for evaluation.
- 2 If no defect is found, Customer will be charged the equivalent of US \$100 plus shipping, duties and taxes that may apply. A new Purchase Order will be required for this evaluation fee.
If the unit is repaired the charge to the Customer will be 30%* of the list price plus any shipping, duties and taxes that may apply. A new Purchase Order will be required for a product repair.
- 3 For an immediate exchange, a new module may be purchased and sent to Customer while repair work is being performed. Credit for purchase of the new module will be issued when the new module is returned in accordance with ProSoft's shipping instructions and subject to ProSoft's policy on non-warranty returns. This is in addition to charges for repair of the old module and any associated charges to Customer.
- 4 If, upon contacting ProSoft Customer Service, the Customer is informed that unit is believed to be unrepairable, the Customer may choose to send unit in for evaluation to determine if the repair can be made. Customer will pay shipping, duties and taxes that may apply. If unit cannot be repaired, the Customer may purchase a new unit.

6.4.1 Un-repairable Units

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

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

6.4.2 Purchasing Warranty Extension

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

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

6.5 LIMITED WARRANTY

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

6.5.1 What Is Covered By This Warranty

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

- for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.
- C** The Warranty Period for RadioLinx accessories (such as antennas, cables, brackets, etc.) are the same as for RadioLinx modules, that is, three years from the date of shipment.

6.5.2 What Is Not Covered By This Warranty

- A** ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.
- B** With the exception of RadioLinx accessories referenced in paragraph 1(c) this Warranty does not cover any product, components, or parts not manufactured by ProSoft.
- C** This Warranty also does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3 programming languages, or "C") not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges outside of the limits indicated on the product specifications; or (viii) disasters such as fire, flood, earthquake, wind or lightning.
- D** The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guides included with your original product purchased by you from ProSoft, contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.5.3 *DISCLAIMER REGARDING HIGH RISK ACTIVITIES*

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

6.5.4 *DISCLAIMER OF ALL OTHER WARRANTIES*

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

6.5.5 *LIMITATION OF REMEDIES***

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

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

6.5.6 *Time Limit for Bringing Suit*

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

6.5.7 No Other Warranties

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

6.5.8 Intellectual Property

- A** Any documentation included with Product purchased from ProSoft is protected by copyright and may not be photocopied or reproduced in any form without prior written consent from ProSoft.
- B** ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
- C** Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
- D** Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.
- E** Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.

6.5.9 Additional Restrictions Relating To Software And Other Intellectual Property

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

6.5.10 Allocation of risks

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

6.5.11 Controlling Law and Severability

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

Index

A

Additional Restrictions Relating To Software
And Other Intellectual Property • 74
All Product Returns • 69
Allocation of risks • 75
Analyzing Data for the first application port •
33
Analyzing Data for the second application
port • 33

B

Backplane Data Transfer • 42

C

Cable Connections • 59
Clearing a Fault Condition • 38
Cold Boot • 53
Command Control Blocks • 53
Configuration Data • 19
Configuration Data Transfer • 48
Configuring RSLinx • 13
Connect your PC to the Module • 15
Connect your PC to the Processor • 11
Controlling Law and Severability • 75

D

Data Analyzer • 32
Data Analyzer Tips • 35
Data Flow between MVI46-GSC Module and
SLC Processor • 53
DB9 to RJ45 Adaptor (Cable 14) • 64
Diagnostics and Troubleshooting • 5, 27
Disabling the RSLinx Driver for the Com Port
on the PC • 29, 59
DISCLAIMER OF ALL OTHER
WARRANTIES • 73
DISCLAIMER REGARDING HIGH RISK
ACTIVITIES • 73
Displaying Timing Marks in the Data
Analyzer • 33
Download the Sample Program to the
Processor • 12

E

Error Word • 21
Exiting the Program • 32

F

Features and Benefits • 39
Functional Overview • 5, 41
Functional Specifications • 40

G

General Concepts • 41
General Specifications • 39
Guide to the MVI46-GSC User Manual • 5

H

Hardware Specifications • 40
How to Contact Us
Sales and Support • 68

I

Input Image • 65
Install the Module in the Rack • 9
Installing and Configuring the Module • 17
Intellectual Property • 74

K

Keystrokes • 28

L

Ladder Logic • 25
LED Status Indicators • 37
LIMITATION OF REMEDIES** • 73
LIMITED WARRANTY • 71

M

M0
File Structure • 65
M1
File Structure • 66
Main Logic Loop • 42
Main Menu • 30
Module Configuration • 19
Module Data • 19
Module Power Up • 41
MVI46-GSC I/O Images • 65

N

Navigation • 27
No Other Warranties • 74
Normal Data Transfer • 44

O

Opening the Data Analyzer Menu • 30

Output Image • 65

P

Package Contents • 8
Please Read This Notice • 2
Port Data • 23
Procedures for Return of Units Out of
Warranty • 70
Procedures for Return of Units Under
Warranty • 69
Product Specifications • 5, 39
Purchasing Warranty Extension • 71

R

Read Block • 44
Reading Status Data from the module • 5, 27
Read-Only Devices • 55
Read-Write Devices • 56
Reference • 5, 39
Removing Timing Marks in the Data
Analyzer • 33
Required Hardware • 28
Required Software • 28
Return Material Authorization (RMA) Policies
and Conditions • 69
Returning to the Main Menu • 34
RS-232 • 61
 Modem Connection • 62
 Null Modem Connection (Hardware
 Handshaking) • 62
 Null Modem Connection (No Hardware
 Handshaking) • 63
RS-232 Configuration/Debug Port • 59
RS-422 • 63
RS-485 • 64
RS-485 and RS-422 Tip • 64

S

Setting Jumpers • 9
SLC Processor Not in Run • 42
Start Here • 5, 7
Starting the Data Analyzer • 34
Status Data • 20, 27
Stopping the Data Analyzer • 34
Support, Service & Warranty • 5, 67
System Requirements • 7

T

Termination of Received Data • 57
The Configuration/Debug Menu • 27
Time Limit for Bringing Suit • 73
Troubleshooting • 19, 38

U

Un-repairable Units • 70
Using the Configuration/Debug Port • 29

V

Viewing Block Transfer Statistics • 31
Viewing Data in ASCII (Text) Format • 33
Viewing Data in Hexadecimal Format • 33
Viewing Module Configuration • 31
Viewing Port Communication Status • 32
Viewing Port Configuration • 32
Viewing Version Information • 31

W

Warm Boot • 53
Warm Booting the Module • 31
What Is Covered By This Warranty • 71
What Is Not Covered By This Warranty • 72
Write Block • 47
Write-Only Devices • 54

Y

Your Feedback Please • 2