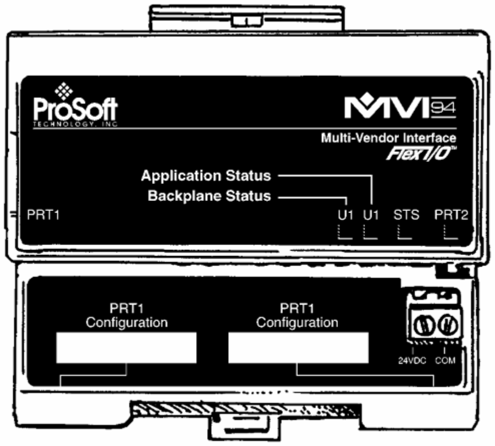
inRAx



MVI94-GSC

Flex Platform

Generic Serial Communication Module

User Manual

March 24, 2005

ProSoft
TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Allen-Bradley Flex Platform Generic Serial Communication Module hardware 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 assure that the information provided is accurate and a true reflection of the product's installation requirements. In order to assure a complete understanding of the operation of the product, the user should read all applicable Allen-Bradley documentation on the operation of the Allen-Bradley hardware.

Under no conditions will ProSoft Technology, Inc. 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, Inc. is prohibited.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology, Inc. 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, Inc.
1675 Chester Avenue, Second Floor
Bakersfield, CA 93301
(661) 716-5100
(661) 716-5101 (Fax)
http://www.prosoft-technology.com

MVI94-GSC User Manual
March 24, 2005

Contents

1 Product Specifications

In This Chapter

The MVI94-GSC ("Generic ASCII Communication Module") product allows Allen-Bradley Flex I/O compatible processors to easily interface with other serial communication devices. Compatible devices include barcode readers, scanners, scales, printers, terminals and any other serial communication device.

1.1 General Specifications

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

Some of the general specifications include:

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

Termination type	Stream mode, termination character(s), message timeout, intercharacter timeout and packet size limit
Baud Rate	110 to 115,200
Parity	None, Odd and Even
Data Bits	5 to 8
Stop Bits	1 or 2
RTS On and Off Timing	0 to 65535 milliseconds

Minimum Response Delay	0 to 65535 milliseconds
Hardware or Software Handshaking	RTS/CTS, DTR/DSR or XON/XOFF
Byte Swapping	Swap bytes received or to transmit

1.2 Physical

This module is designed by ProSoft Technology and incorporates licensed technology from Allen-Bradley (Flex backplane technology).

- Flex Form Factor - Single Slot

Connections:

- 1 – connector for user application of RS-232, RS-422 or RS-485 interface
- 1 – RS-232 Configuration/Debug Tool Connector
- TTL is only supported with a compatible RS-232C to TTL converter (not supplied)

1.3 Flex Interface

- Operation via simple ladder logic
- Complete setup and monitoring of module through Debug port and user configuration file (MVI94-GSC.TXT)
- Flex backplane interface via I/O access

Hardware Specifications

- The MVI94-GSC module is designed by ProSoft Technology and incorporates licensed technology from Allen-Bradley (Flex backplane technology).

Current Loads:	20 mA @ 5V (from backplane)
Operating Temperature:	0 to 60 Deg C
	32 to 140 Deg F
Storage Temperature:	-40 to 85 Deg C
	-40 to 185 Deg F
Relative Humidity:	5-95% (w/o condensation)

Application Port Connector: One Mini Din 8 Connector (mini din 8 to DB9 cable shipped with unit) supporting RS-232, RS-422 and RS-485 interfaces

Configuration Connector: Mini Din 8 RS-232 Connector (mini din 8 to DB9 cable and null modem cable shipped with unit)

2 Functional Overview

In This Chapter

This section gives the reader a functional overview of the MVI94-GSC module. A thorough understanding of the information contained in this document is required for successful implementation of the module in a user application. If you are not familiar with the data transfer, read this document before setting up the module.

2.1 General Concepts

The following discussion covers several concepts that are key to understanding the operation of the MVI94-GSC module.

2.1.1 Module Power Up

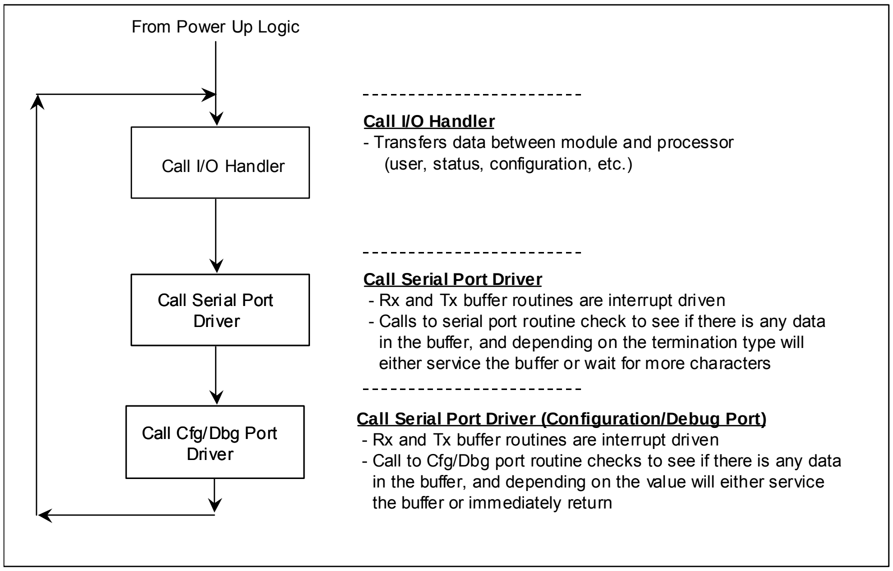
On power up the module begins performing the following logical functions:

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

Once 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 port.

2.1.2 Main Logic Loop

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


From Power Up Logic
Call I/O Handler
Call Serial Port Driver
Call Cfg/Dbg Port Driver
Call I/O Handler
- Transfers data between module and processor (user, status, configuration, etc.)
Call Serial Port Driver
- Rx and Tx buffer routines are interrupt driven
- Calls to serial port routine check to see if there is any data in the buffer, and depending on the termination type will either service the buffer or wait for more characters
Call Serial Port Driver (Configuration/Debug Port)
- Rx and Tx buffer routines are interrupt driven
- Call to Cfg/Dbg port routine checks to see if there is any data in the buffer, and depending on the value will either service the buffer or immediately return

2.1.3 Flex Processor Not in Run

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

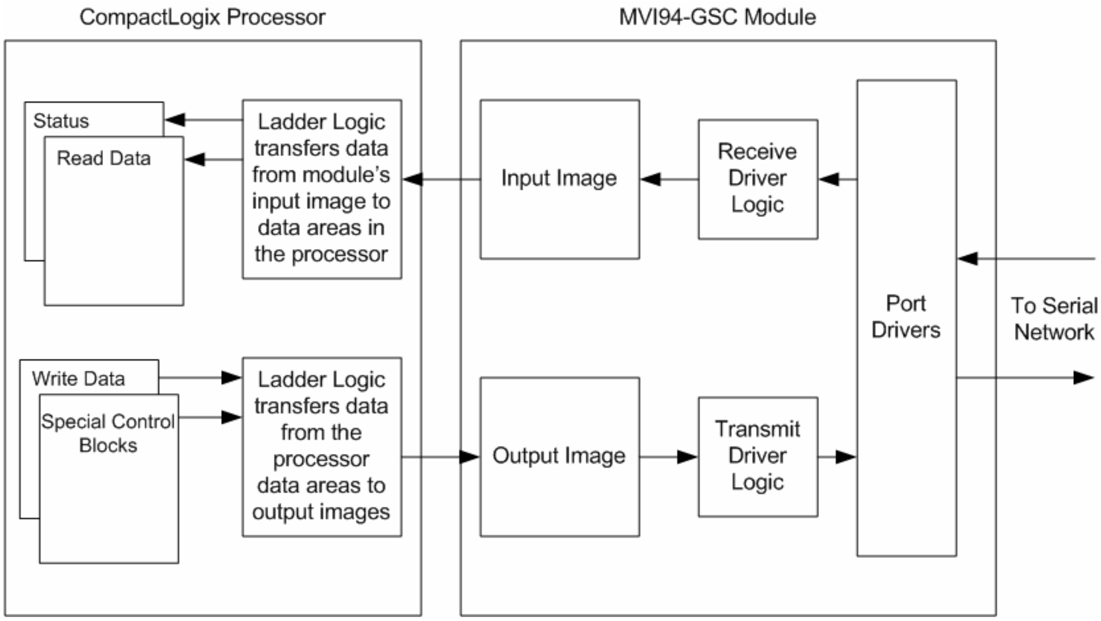
2.2 Backplane Data Transfer

The MVI94-GSC module is unique in the way that the Flex backplane is utilized. Data is sent between the module and the Flex processor across the backplane using the module's input and output images. The update frequency of the images is determined by the ladder logic scan rate and the backplane bus rate.

Data received on the application port is placed in the module’s input image. This data is processed by the ladder logic in the Flex processor. The input image for the module is set to 14 bytes.

The processor inserts data in the module's output image to be transferred to the module. The module's program extracts the data and transmits the data out the communication port. The output image allowed for data to the module is set at 10 bytes.

The diagram below displays the data transfer method used to move data between the Flex processor, the MVI94-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 input and output images. Ladder logic must be written in the Flex processor to interface the input and output image data with data defined in the processor. The user is responsible for handling and interpreting all data received on the application ports and transferred in the input image. Additionally, the user is responsible for constructing messages to be transferred out of the application ports by building the messages in the output image of the module.

Each block transferred between the module and the processor has a block number in byte zero of the input or output image. The table below lists the block identification codes used by the module:

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

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

2.3 Normal Data Transfer

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

2.3.1 Read Block

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

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

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

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

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

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

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

Member Name	Bit in Word	Description
Cfg_type	Bit 0	The termination type configured for the port is not valid. Values between 0 and 15 are the only ones valid. The module will use type 0 (stream mode) for the port.
Cfg_Baud	Bit 1	The baud rate entered for the port is not valid. The module will use 9600 baud for the port.
Cfg_Parity	Bit 2	The parity value entered is not valid. Expected values are None, Even, Odd, Mark and Space. Only the first letter is inspected to make the selection. The module will set the parity to a value of none if an invalid selection is entered.
Cfg_DataBits	Bit 3	The number of data bits for the protocol is not valid. Values between 5 and 8 are accepted. The module assumes a value of 8 data bits.
Cfg_StopBits	Bit 4	The number of stop bits for the protocol is not valid. Values of 1 or 2 are accepted. The module assumes a value of 1 stop bit.
Cfg_Handshake	Bit 5	The handshake code for the port is not valid. Expected values are None, RTS/CTS, DTR/DSR and XON/XOFF. Only the first letter of the selection is used. The module assumes a value of None (no handshaking) if the parameter is in error.
Cfg_Rtermcount	Bit 6	The number of termination characters is not valid. The value must be set between 1 and 12 when using the termination character string to end a receive buffer. The module will not terminate a buffer when using the termination character(s) when this bit is set.

Member Name	Bit in Word	Description
Cfg_RPacketLen	Bit 7	The number of characters for a packet is not valid. The value must be set between 1 and 4096 when the packet size termination option is used. The module will not use the packet length termination option when this bit is set.
Cfg_Rtimeout	Bit 8	The message timeout value is set to zero. The module will not use the message timeout termination option when this bit is set.
Cfg_Rdelay	Bit 9	The intercharacter delay value configured is set to zero. The module will not use the intercharacter delay option when this bit is set.
Cfg_Wtimeout	Bit 10	The write message timeout parameter is set to zero. The module assumes a value of 5000 milliseconds.
	Bit 11	
	Bit 12	
	Bit 13	
	Bit 14	
Err_ROverflow	Bit 15	Data is being received faster on the port than the ladder logic can process the read blocks. Alter the configuration of the module or the connected device. Receive data is being lost.

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

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

	Byte Offset	Description
	0	Block Sequence Number (Bumped each scan by module)
	1	This byte contains a value of 253 (0xFD)
Module Status	2 to 3	Program cycle counter
	4 to 5	Port 1 receive state
	6 to 7	Port 1 receive character count
	8 to 9	Port 1 receive block count
	10 to 11	Port 1 transmit character count
	12 to 13	Port 1 transmit block count
	14 to 15	Port 1 transmit state

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

2.3.2 Write Block

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

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

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

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

2.4 Control Blocks

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

2.4.1 Warm Boot

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

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

2.4.2 Cold Boot

This block is sent from the Flex processor to the module (output image) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The format of the output image to perform this task is as follows:

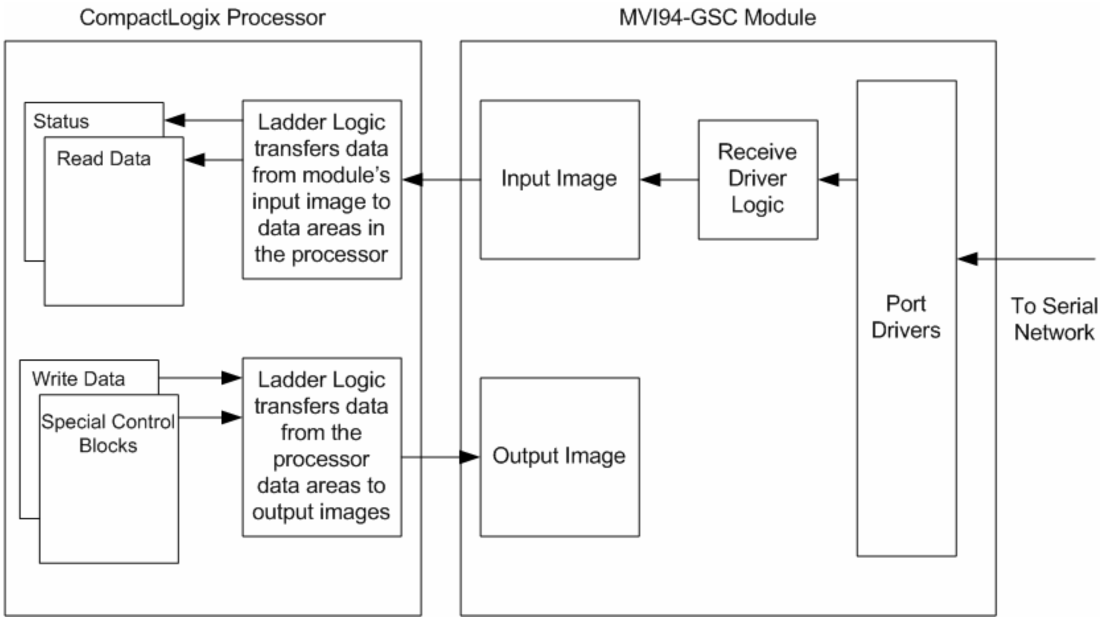
Byte Offset	**Description**
0	Block Sequence Number set to 255 (0xFF)
1 to 13	Not Used

2.5 Data Flow between MVI94-GSC Module and Flex Processor

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

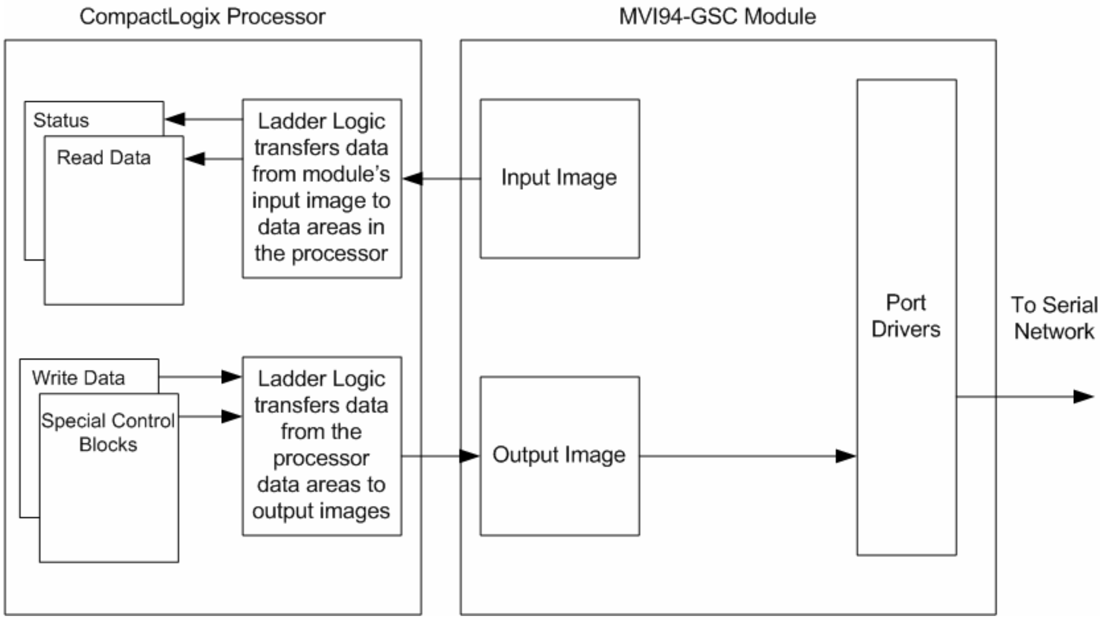
2.5.1 Write-Only Devices

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


CompactLogix Processor
MVI94-GSC Module
Status
Read Data
Ladder Logic transfers data from module's input image to data areas in the processor
Input Image
Receive Driver Logic
Port Drivers
To Serial Network
Write Data
Special Control Blocks
Ladder Logic transfers data from the processor data areas to output images
Output Image

2.5.2 Read-Only Devices

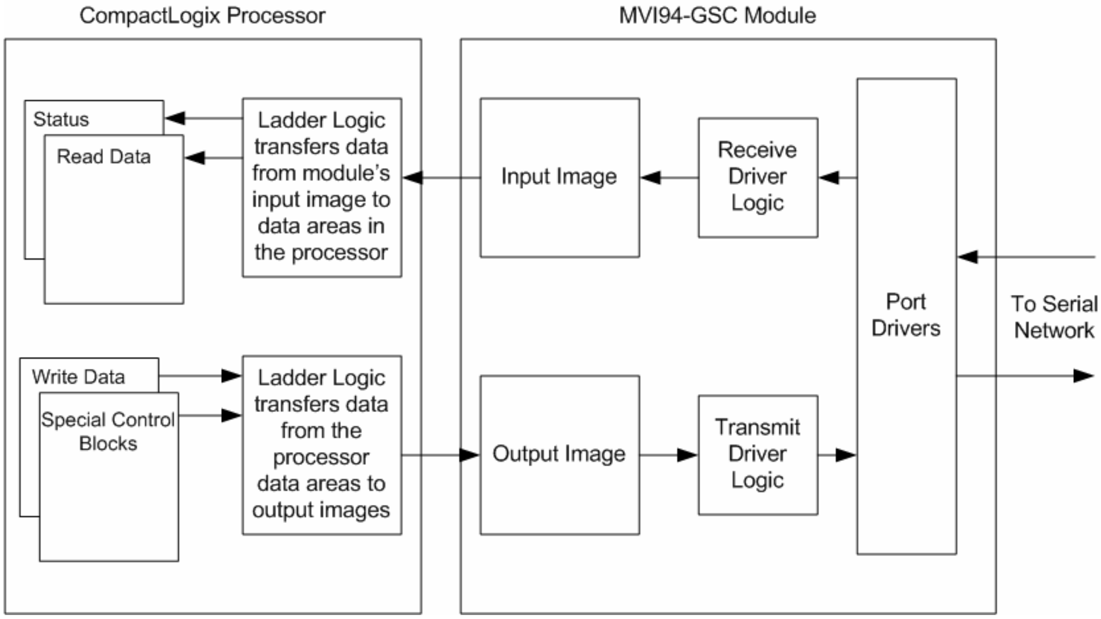
Read-Only devices are those that only receive data from the module. An example of this type of device is a printer. The printer will generate output or be controlled based on the data it receives on its communication port. Ladder logic is used to construct the write blocks to be sent to the module. When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


CompactLogix Processor
MVI94-GSC Module
Status
Read Data
Ladder Logic transfers data from module's input image to data areas in the processor
Input Image
Port Drivers
To Serial Network
Write Data
Special Control Blocks
Ladder Logic transfers data from the processor data areas to output images
Output Image

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

2.5.3 Read-Write Devices

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



2.6 Termination of Received Data

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

Termination Type Field

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

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

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

Termination character(s) used

Settings:

Count = 1 (RTermCnt=1)
Termination on 0x0d (carriage return character) (RTermChar = 0d 00 00 00 …)

Data Received on port:

A B C 0x0d D E

Comment:

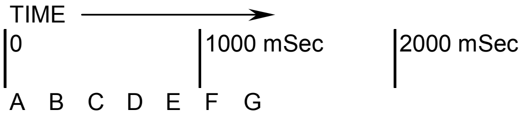
The characters "ABC" will be sent along with the 0x0d character to the controller after the 0x0d character is received. The characters "DE" will not be sent until the 0x0d character is received.

Message timeout used

Settings:

Message timeout = 1000 mSec (Rtimeout=1000)

Data Received on port:



Comment:

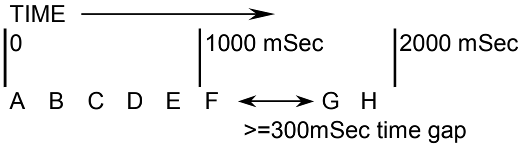
After the 'A' character is received on the port, the message timeout is started.
The characters "ABCDE" will be sent to the controller in one block. The characters "FG" will follow in the second block one second later.

Intercharacter delay timeout used

Settings:

Intercharacter delay timeout = 300 mSec (Rdelay=300)

Data Received on port:



Comment:

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

Packet size limit used

Settings:

Packet size = 4 (RPacketLen=4)

Data Received on port:

A B C D E F G H I J

Comment:

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

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

3 Module Configuration

In order for the MVI94-GSC module to function, a minimum amount of configuration data must be transferred to the module. Configuration information is transferred to the module using the PS_Term program with the ProSoft Protocol by attaching to the Configuration/Debug port on the module. This text file that is transferred from the remote personal computer to the module has a very simple format. Two sections are defined in the file to separate the configuration areas. These sections are as follows: [Module] and [ASCII Port 2]. Items are contained in each section that define the configuration of the module. The parameters that can be defined for the module are shown in the following table:

[Section]/Item	Value	Range	Description
[MODULE]			Module section header
Module Name:		0 to 80 characters	This parameter is used to assign a name to the module that can be viewed using the configuration/debug port. It can be used to identify the module and the configuration file.
Backplane Fail Count:		0 to 65535	This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to zero, the communication ports will continue to operate under all conditions. If the value is set larger than 0 (1-65535), communications will cease if the specified number of failures occur.

[Section]/Item	Value	Range	Description
[GSC Port 0]			GSC port definition header
Enabled:		Yes or No	This parameter is used to define if this port will be utilized. If the parameter is set to No, the port is disabled. A value of Yes will enable the port.
Type:		0 to 15	This parameter is used to specify the receive termination characteristics for the port. This value is bit mapped as follows: Bit 0 = Termination character(s) used, Bit1=Message timeout used, Bit2=Intercharacter delay timeout used and Bit3=Packet size limit used. If the parameter is set to zero, the port is placed in stream mode.
Baud Rate:		From selected list of codes	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entries for this field include: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 and 115.

[Section]/Item	Value	Range	Description
Parity:		None, Even, Odd, Mark or Space	This is the Parity code to be used for the port. The values are as follows: None, Odd, Even, Mark and Space.
Data Bits:		5 to 8	This parameter sets the number of data bits for each word used by the protocol. Valid entries for this field are 5, 6, 7 and 8.
Stop Bits:		1 or 2	This parameter sets the number of stop bits to be used with each data value sent. Valid entries for this field are 1 and 2.
RTS On:		0 to 65535	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Valid values are in the range of 0 to 65535.
RTS Off:		0 to 65535	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low. Valid values are in the range of 0 to 65535.
Handshaking:		NONE, RTS/CTS, DTR/DSR or XON/XOFF	This parameter is used to specify the handshaking used on the port. The values are as follows: None=No hardware or software handshaking, RTS/CTS hardware handshaking, DTR/DSR hardware handshaking and XON/XOFF software handshaking.
Rx Term Char Count:		0 to 12	This parameter is used if bit 0 of the Type parameter is set. This value (0 to 12) defines the number of termination characters used to define the end of received message.
Rx Term Characters:		List of up to 12 integer values	This array of 12 integer values representing the characters used to define the termination characters at the end of each received message. The number of characters to be used in the array is set in the RTermCnt parameter.
Rx Packet Length:		0 to 4096	This parameter is used if bit 3 is set in the Type parameter. The parameter sets the length of data required to be received on the port before transferring the data to the processor.
Rx Message Timeout:		0 to 65535	This parameter is used if bit 1 is set in the Type parameter. The parameter sets the number of milliseconds to wait after the first character is received on the port before automatically sending the data to the processor.
Rx Intercharacter Delay:		0 to 65535	This parameter is used if bit 2 is set in the Type parameter. The parameter sets the number of milliseconds to wait between each character received on the port before sending the data to the processor.
Rx Swap Bytes:		Yes or No	This parameter specifies if the data received should have its bytes swapped before sending over the backplane.

[Section]/Item	Value	Range	Description
Tx Message Timeout:		0 to 65535	This parameter specifies the timeout period to transmit a message out the port. A message must be transmitted out the port within the specified timeout period. Message transmission will be aborted if the timeout is exceeded.
Tx Minimum Delay:		0 to 65535	This parameter specifies the minimum number of milliseconds to delay before transmitting a message out the port. This pre-send delay is applied before the RTS on time. This may be required when communicating with slow devices.
Tx Swap Bytes:		Yes or No	This parameter specifies if the data to be transmitted out the port will have the bytes swapped from the data presented across the backplane.

3.1 Example Configuration File

An example configuration file for the module is as follows:

```
#MVI94-GSC CONFIGURATION EXAMPLE
#
#

[Module]
Module Name : Test of MVI94-GSC module

Backplane Fail Count : 0


[GSC Port 2]
Enabled                : Yes
Type                   : 15
Baud Rate              : 19200
Parity                 : None
Data Bits              : 8
Stop Bits              : 1

RTS On                 : 0
RTS Off                : 1
Handshaking            : None   #None, RTS/CTS, DTR/DSR, XON/XOFF

Rx Term Char Count     : 2      #0 to 12 characters to delimit end of
message
Rx Term Characters     : 13 10
Rx Packet Length       : 10
Rx Message Timeout     : 2000
Rx Intercharacter Delay : 500
Rx Swap Bytes          : No

Tx Message Timeout     : 5000
Tx Minimum Delay       : 0
Tx Swap Bytes          : No
```

4 Hardware Diagnostics & Troubleshooting

In This Chapter

The module provides diagnostic information in three forms to the user.

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

The following sections explain how to obtain the Status Data from the module and the meaning of the individual LED's on the module.

4.1 Reading Status Data from the Module

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

4.2 LED Status Indicators

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

4.2.1 PRT1

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

4.2.2 U1

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

4.2.3 U2

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

4.2.4 STS

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

4.2.5 PRT2

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

4.3 Using the Configuration/Debug Port

This section contains a description and lists the functionality provided on the configuration/ debugger port of the MVI94-GSC communication module. Use of this port can help in trouble-shooting problems that may exist with the module when interfacing with the Flex processor or the serial interface network.

Facilities on the port provide the following functionality:

- Full view of the module's configuration data
- View of all the module's status data
- Version information
- Control over the module (warm boot and cold boot)
- Data analyzer to view message processing on each port

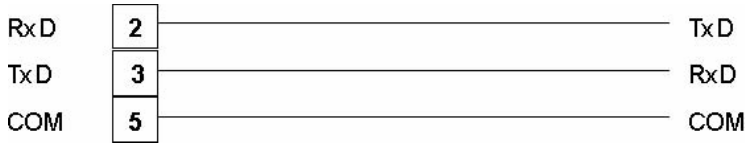
The sections below describe the required elements, connecting to the port and using the port. Following these sections is a detailed discussion of the menu options offered when using the port.

4.3.1 Required Hardware

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

ProSoft Technology, Inc. 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. The module's port has a DB-9 male connector at the end of a RJ-45 to DB-9 pigtail. The RJ-45 end of the cable is to be placed in the MVI94-GSC port 1 connector (top port). The cable required is displayed below:



4.3.2 Required Software

In order to send and receive data over the serial port (COM port) on your computer to the module, you need to 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 will need to obtain and install a compatible communication program. The following table lists communication programs that have been tested by ProSoft Technology, Inc.

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

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

4.3.3 Using the Port

The following steps are required to interface with the configuration/debugger port:

1 Connect your computer to the module's port using a null-modem cable.

2 Start the terminal emulation program on your computer and configure the communication parameters to those shown in the Required Software section (57600, N, 8, 1).

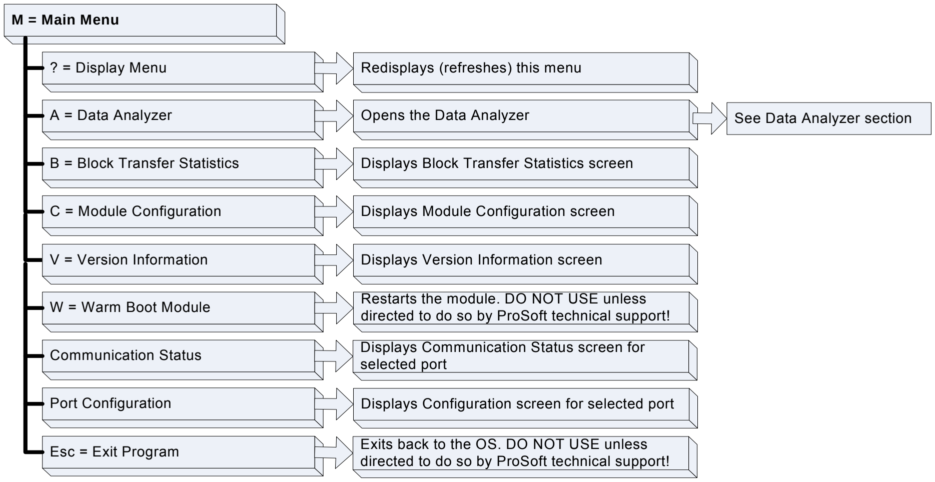
3 Enter the '?' character on your computer. If everything is set up correctly, the port's menu will be displayed.

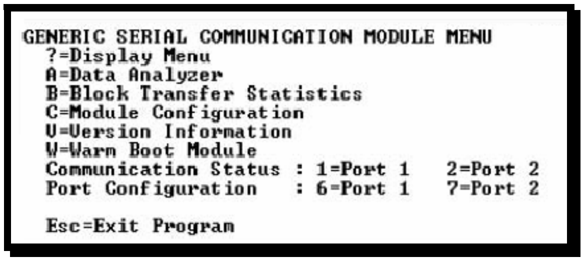
If there is no response from the module, check the communication setup and the cable. In addition, make sure you are connected to the correct port on your computer and the module.

4.3.4 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 [M] 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 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, Inc. 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.


M = Main Menu
? = Display Menu
Redisplays (refreshes) this menu
A = Data Analyzer
Opens the Data Analyzer
See Data Analyzer section
B = Block Transfer Statistics
Displays Block Transfer Statistics screen
C = Module Configuration
Displays Module Configuration screen
V = Version Information
Displays Version Information screen
W = Warm Boot Module
Restarts the module. DO NOT USE unless directed to do so by ProSoft technical support!
Communication Status
Displays Communication Status screen for selected port
Port Configuration
Displays Configuration screen for selected port
Esc = Exit Program
Exits back to the OS. DO NOT USE unless directed to do so by ProSoft technical support!



```
GENERIC SERIAL COMMUNICATION MODULE MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  V=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

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. The Data Analyzer section has more information about this menu.

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

Viewing Block Transfer Statistics

Press **[B]** from the Configuration/Debug Menu to view the Block Transfer Statistics screen.

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

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

Viewing Module Configuration

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

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

Viewing Version Information

Press [V] from the Main Menu 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.

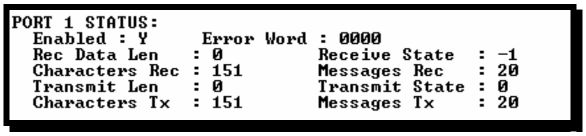
Warm Booting the Module

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 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, Inc. 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 are instructed to do so by the ProSoft Technology, Inc. Technical Support Group.

Viewing Port Communication Status

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

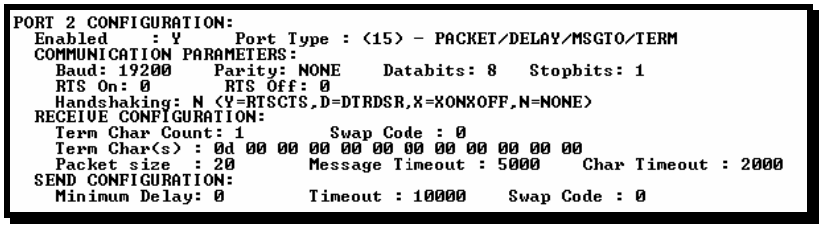


```
PORT 1 STATUS:
  Enabled : Y     Error Word : 0000
  Rec Data Len   : 0         Receive State  : -1
  Characters Rec : 151       Messages Rec   : 20
  Transmit Len   : 0         Transmit State : 0
  Characters Tx  : 151       Messages Tx    : 20
```

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

Viewing Port Configuration

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



```
PORT 2 CONFIGURATION:
  Enabled     : Y     Port Type : (15) - PACKET/DELAY/MSGTO/TERM
  COMMUNICATION PARAMETERS:
    Baud: 19200    Parity: NONE    Databits: 8   Stopbits: 1
    RTS On: 0       RTS Off: 0
    Handshaking: N (Y=RTSCTS,D=DTRDSR,X=XONXOFF,N=NONE)
  RECEIVE CONFIGURATION:
    Term Char Count: 1        Swap Code : 0
    Term Char(s) : 0d 00 00 00 00 00 00 00 00 00 00 00
    Packet size  : 20      Message Timeout : 5000    Char Timeout : 2000
  SEND CONFIGURATION:
    Minimum Delay: 0       Timeout : 10000    Swap Code : 0
```

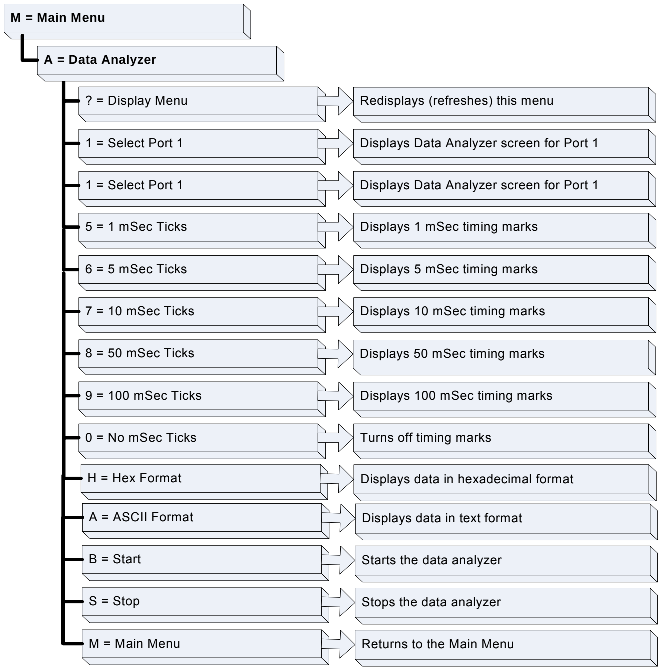
Use this command to display detailed configuration information for the selected port.

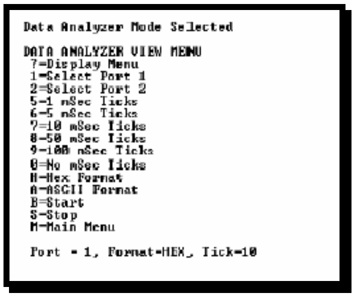
Esc = Exit Program

This option is used to exit the program and to display the operating system prompt. This option should only be selected if instructed by the ProSoft Technical Support Group. If you select the option, the module will cease operation. Data will no longer be transferred between the ports and the module and between the ControlLogix processor and the module. This might cause an upset to a currently running process.

4.3.5 Data Analyzer

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



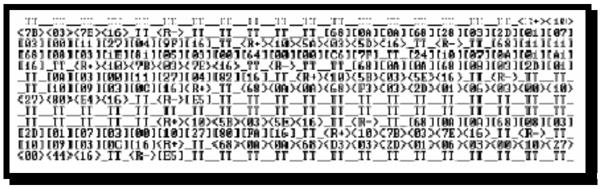


```
Data Analyzer Mode Selected

DATA ANALYZER VIEW MENU
 ?=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
```

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

Analyzing Data for Port 1

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



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

Analyzing Data for Port 2

Press [2] to display I/O data for Port 2 in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

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

Key	Interval
[5]	1 mSec ticks
[6]	5 mSec ticks
[7]	10 mSec ticks
[8]	50 mSec ticks
[9]	100 mSec ticks

Removing Timing Marks in the Data Analyzer

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

Viewing Data in Hexadecimal Format

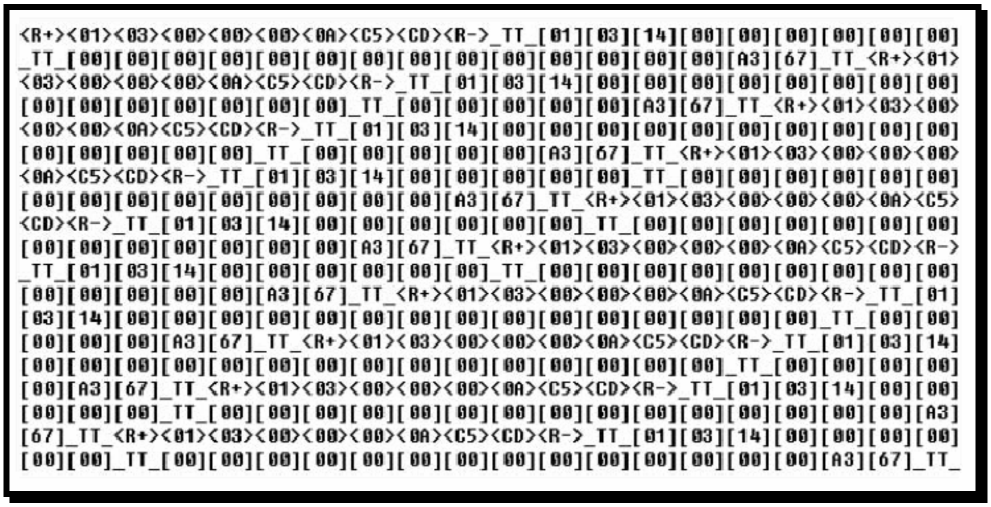
Press **[H]** from the Database View menu. Use this command to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

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



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

The Data Analyzer displays the following special characters:

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

Stopping the Data Analyzer

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

Important: When in analyzer mode, program execution will slow down. You should only use this tool during a trouble-shooting session. Before disconnecting from the Config/Debug port, please be sure to press [M] to return to the main menu and disable the data analyzer. This action will allow the module to resume its normal operating mode.

Returning to the Main Menu

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

5 Cable Connections to the Module

In This Chapter

The MVI94-GSC module has the following communication connections on the module:

- One application serial communication port
- One RS-232 Configuration/Debug port

5.1 Application Serial Communication Port

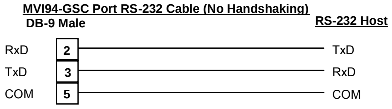
The MVI94-GSC module has one physical connector with a Mini Din 8 connection located on the front of the module for use as an application port (Mini Din 8 to DB9 cable shipped with module).

5.1.1 Connecting the Cable to the Connector

ProSoft provides one mini din 8 to male DB-9 pigtail to permit simpler interfacing to other devices. The module's serial port can be configured to operate in RS-232, RS-422 or RS-485 mode. The interface to be associated with a port is set with jumpers on the module. Additionally, the use of the modem control lines is user definable. Each interface is described below:

RS-232

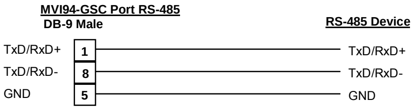
When the RS-232 interface is selected, the use of the modem control lines is user definable. If no modem control lines will be used, the cable to connect to the port is as shown below:



The RTS line is controlled by the RTS on and off parameters set for the port. If the CTS line is used (usually only required for half-duplex modems), the RTS and CTS lines must either be connected together or connected to the modem. The diagram below displays the cable required when connecting the port to a modem.

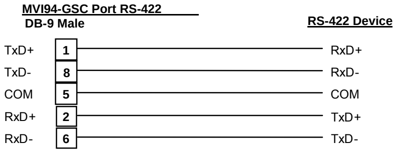
RS-485

When the RS-485 interface is used, a single two or three wire cable is required. The use of the ground is optional and dependent on the RS-485 network. The cable required for this interface is shown below:



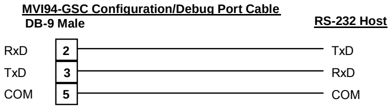
RS-422

When the RS-422 interface is used, a four or five wire cable is required. The use of the ground is optional and dependent on the RS-422 network. The cable required for this interface is shown below:



5.2 RS-232 Configuration/Debug Port

This port is physically a Mini Din 8 connection. A Mini Din 8 to DB-9 pigtail cable is shipped 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. Refer to the **MVI94-GSC Configuration/Debugger Guide** for instructions on how to use this port. The cable for communications on this port is shown below:



Support, Service & Warranty

ProSoft Technology, Inc. survives on its ability 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
Phone	(661) 716-5100 (661) 716-5101 (Fax)
Postal Mail	ProSoft Technology, Inc. 1675 Chester Avenue, Second Floor Bakersfield, CA 93301

Before calling for support, please prepare yourself for the call. In order to provide the best and quickest support possible, we will most likely ask for the following information (you may wish to fax it to us prior to calling):

1 Product Version Number

2 System hierarchy

3 Module Operation

- o Configuration/Debug status information
- o LED patterns

4 Information about the processor and user data files and LED patterns on the processor

5 Details about the serial devices interfaced

An after-hours answering system (on the Bakersfield number) allows pager access to one of our qualified technical and/or application support engineers at any time to answer the questions that are important to you.

Module Service and Repair

The MVI94-GSC device is an electronic product, designed and manufactured to function under somewhat adverse conditions. As with any product, through age, misapplication, or any one of many possible problems the device may require repair.

When purchased from ProSoft Technology, Inc., the device has a 1 year parts and labor warranty according to the limits specified in the warranty. Replacement and/or returns should be directed to the distributor from whom the product was purchased. If you need to return the device for repair, obtain an RMA (Returned Material Authorization) number from ProSoft Technology, Inc. Please call the factory for this number, and print the number prominently on the outside of the shipping carton used to return the device.

General Warranty Policy

ProSoft Technology, Inc. (Hereinafter referred to as ProSoft) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product.

This warranty is limited to the repair and/or replacement, at ProSoft's election, of defective or non-conforming Product, and ProSoft shall not be responsible for the failure of the Product to perform specified functions, or any other non-conformance caused by or attributable to: (a) any misapplication or misuse of the Product; (b) failure of Customer to adhere to any of ProSoft's specifications or instructions; (c) neglect of, abuse of, or accident to, the Product; or (d) any associated or complementary equipment or software not furnished by ProSoft.

Limited warranty service may be obtained by delivering the Product to ProSoft and providing proof of purchase or receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to ProSoft, and to use the original shipping container or equivalent. Contact ProSoft Customer Service for further information.

Limitation of Liability

EXCEPT AS EXPRESSLY PROVIDED HEREIN, PROSOFT MAKES NO WARRANT OF ANY KIND, EXPRESSED OR IMPLIED, WITH RESPECT TO ANY EQUIPMENT, PARTS OR SERVICES PROVIDED PURSUANT TO THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANT ABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER PROSOFT OR ITS DEALER SHALL BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION IN CONTRACT OR TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS RESULTING FROM, OR ARISING OUT OF, OR IN CONNECTION WITH THE USE OR FURNISHING OF EQUIPMENT, PARTS OR SERVICES HEREUNDER OR THE PERFORMANCE, USE OR INABILITY TO USE THE SAME, EVEN IF PROSOFT OR ITS DEALER'S TOTAL LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT.

Where directed by State Law, some of the above exclusions or limitations may not be applicable in some states. This warranty provides specific legal rights;

other rights that vary from state to state may also exist. This warranty shall not be applicable to the extent that any provisions of this warranty are prohibited by any Federal, State or Municipal Law that cannot be preempted.

Hardware Product Warranty Details

Warranty Period: ProSoft warranties hardware Product for a period of 1 year.

Warranty Procedure: Upon return of the hardware Product ProSoft will, at its option, repair or replace Product at no additional charge, freight prepaid, except as set forth below. Repair parts and replacement Product will be furnished on an exchange basis and will be either reconditioned or new. All replaced Product and parts become the property of ProSoft. If ProSoft determines that the Product is not under warranty, it will, at the Customer's option, repair the Product using current ProSoft standard rates for parts and labor, and return the Product freight collect.

Index