ProSoft TECHNOLOGY

Where Automation Connects.

inRAx®

MVI69-103M

CompactLogix or MicroLogix Platform

IEC 60870-5-103 Master Communication Module

11/3/2008

USER MANUAL

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation CompactLogix or MicroLogix hardware, the MVI69-103M 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.

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.

Battery Life Advisory

All modules in the MVI series use a rechargeable Lithium Vanadium Pentoxide battery to backup the 512K SRAM memory, real-time clock, and CMOS. The battery should last for the life of the module.
The module must be powered for approximately twenty hours before it becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and configuration data, the real-time clock, and the 512K SRAM memory for approximately 21 days.
Before you remove a module from its power source, ensure that the battery within the module is fully charged. A fully charged battery will hold the BIOS settings (after being removed from its power source) for a limited number of days. When the battery is fully discharged, the module will revert to the default BIOS settings.

Note: The battery is not user replaceable.

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

MVI69-103M User Manual
11/3/2008

ProSoft® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD and are available at no charge from our web site: http://www.prosoft-technology.com
Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.
Asia Pacific: +603.7724.2080
Europe, Middle East, Africa: +33.5.34.36.87.20
Latin America: +1.281.298.9109
North America: +1.661.716.5100

Contents

Guide to the MVI69-103M User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 69) Diagnostics and Troubleshooting (page 51)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 71) Functional Overview (page 73) Product Specifications (page 71)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 121)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI69-103M module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-103M 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 MVI69-103M module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix processor, with compatible power supply and one free slot in the rack, for the MVI69-103M module. The module requires 800mA of available power.

Important: The MVI69-103M module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).
Important: For 1769-L23x processors, please make note of the following limitations.

- 1769-L23-QBFC1B = 800mA at 5Vdc (1 MVI69-103M will use all 800mA of available power. No other modules can be used with an MVI69 module connected to this processor).
- 1769-L23E-QB1B = 1000mA at 5Vdc (1 MVI69-103M will use 800mA of available power. One other module can be used on this rack provided it consumes less than 200mA at 5Vdc.
- 1769-L23E-QBFC1B = 450mA at 5Vdc (no MVI69 module can be used with this processor)

- Rockwell Automation RSLogix 5000 (CompactLogix) or RSLogix 500 (MicroLogix) programming software
- Rockwell Automation RSLinx communication software

- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP Professional with Service Pack 1 or 2
  - Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3
  - Microsoft Windows Server 2003
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- CD-ROM drive
- HyperTerminal or other terminal emulator program capable of file transfers using Zmodem protocol.

1.2 Package Contents

The following components are included with your MVI69-103M module, and are all required for installation and configuration.

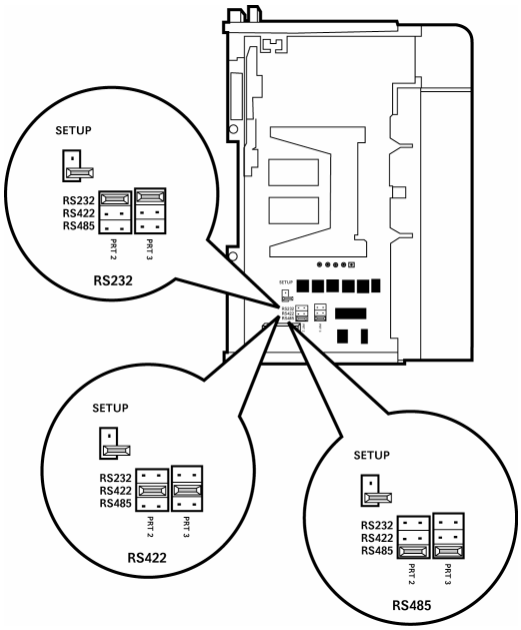
Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI69-103M Module	MVI69-103M	IEC 60870-5-103 Master Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	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 MVI69-103M module.

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

1.3 Setting Jumpers

When the module is manufactured, the port selection jumpers are set to RS-232. To use RS-422 or RS-485, you must set the jumpers to the correct position. The following diagram describes the jumper settings.



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.

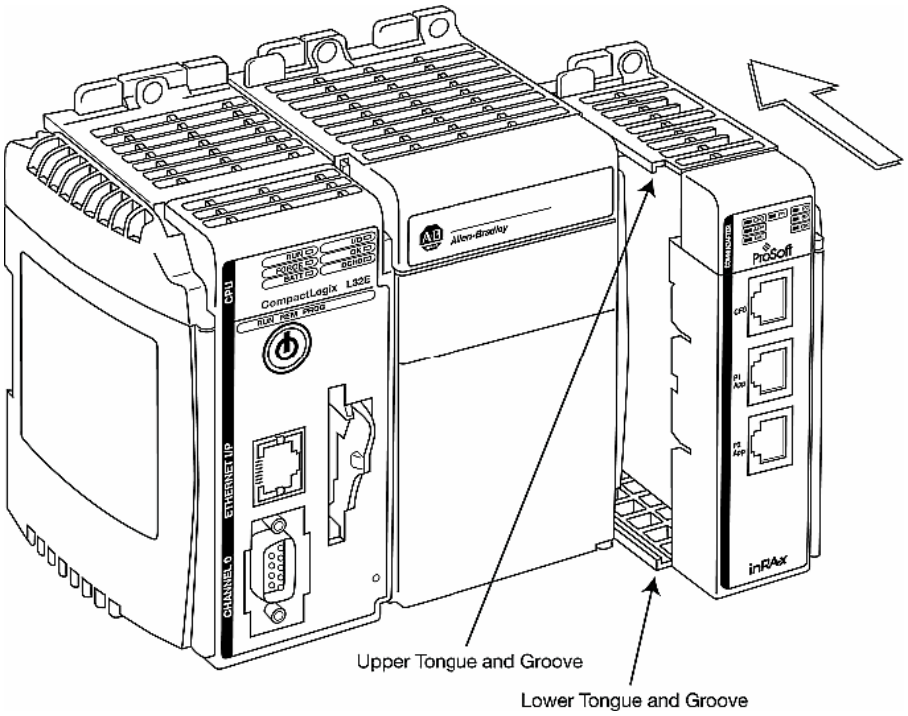
1.4 Install the Module in the Rack

This section describes how to install the module into a CompactLogix or MicroLogix rack

Before you attempt to install the module, make sure that the bus lever of the adjacent module is in the unlocked (fully right) position.

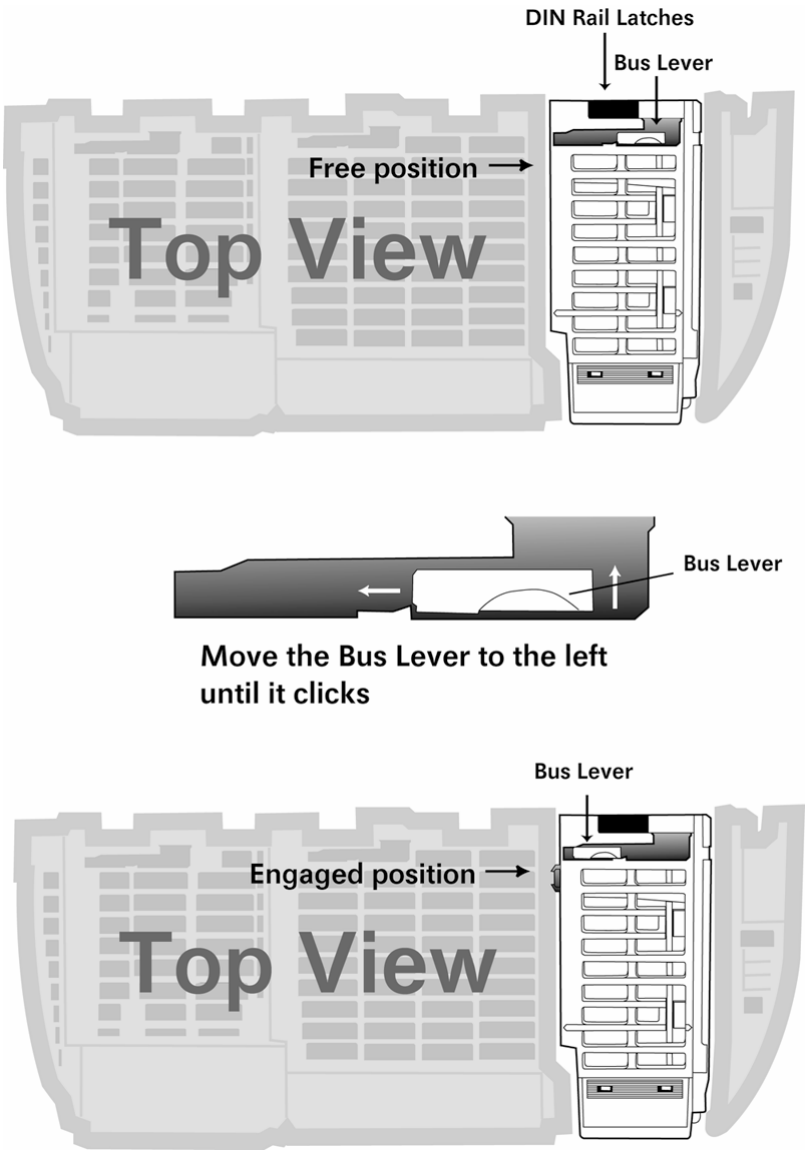
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1 Align the module using the upper and lower tongue-and-groove slots with the adjacent module and slide forward in the direction of the arrow.



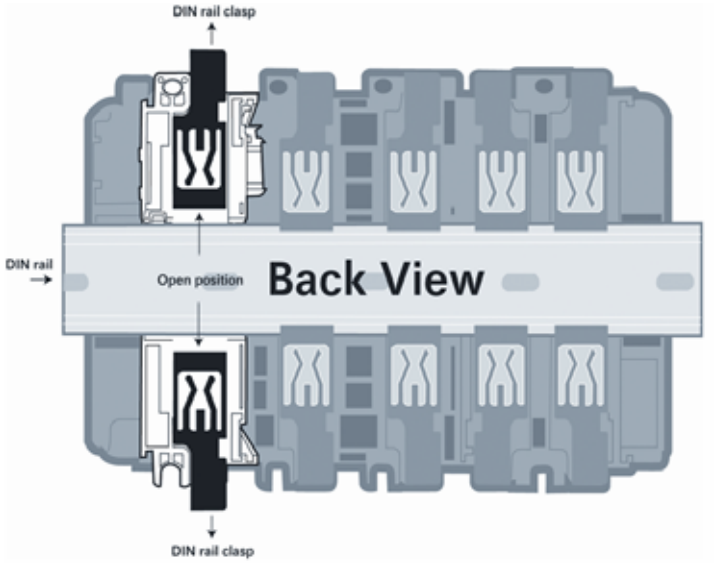
2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

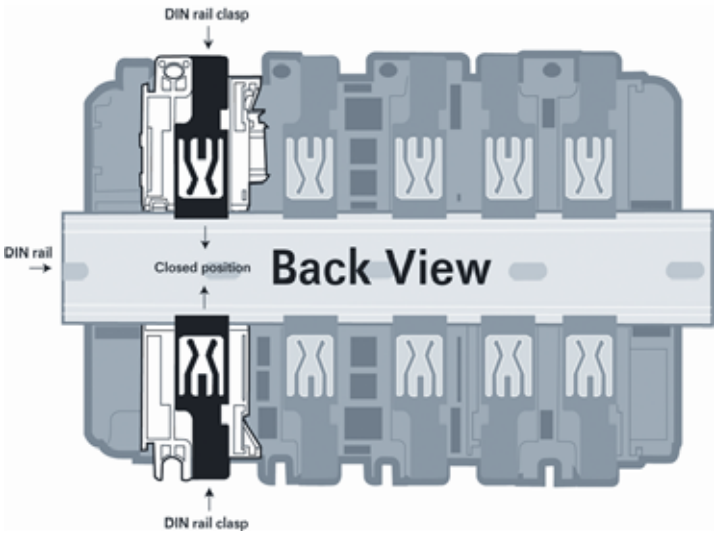
3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



4 Close all DIN rail latches.

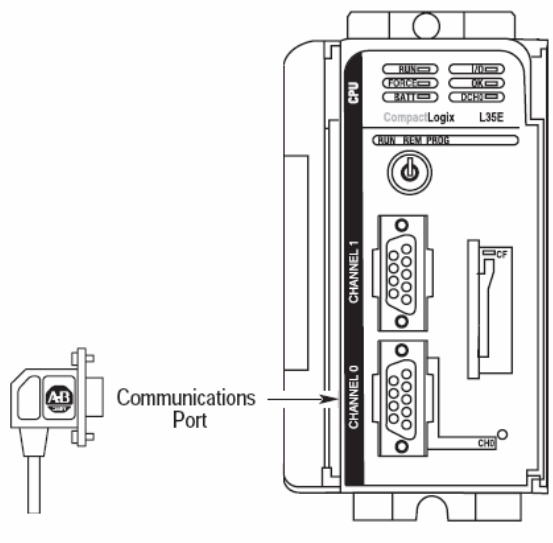
5 Press the DIN rail mounting area of the controller against the DIN rail. The latches will momentarily open and lock into place.



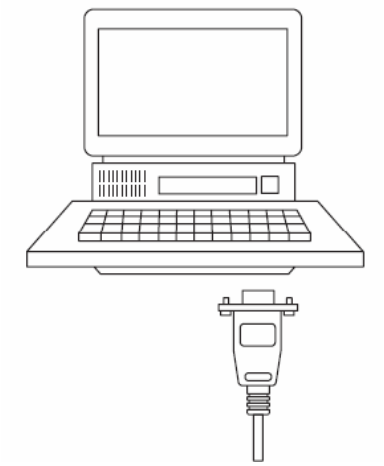


1.5 Connect your PC to the Processor

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



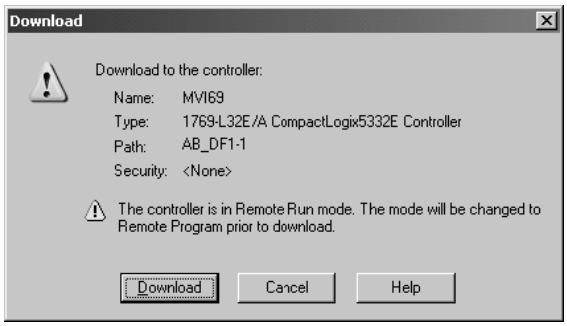
2 Connect the straight connector end of the cable to the serial port on your computer.



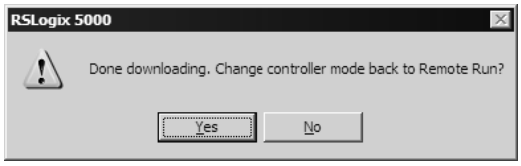
1.6 Download the Sample Program to the Processor

Important: For most applications, the sample program will work without modification.
Note: The key switch on the front of the CompactLogix 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. When communication is established, RSLogix will open a confirmation dialog box. 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 OK to switch the processor from Program mode to Run mode.

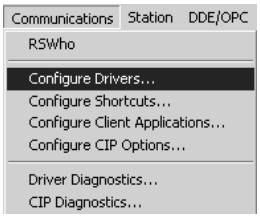


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

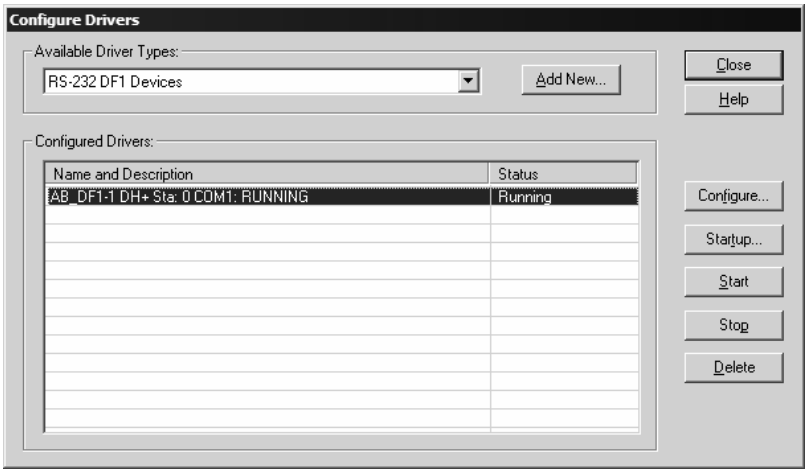
1.6.1 Configuring RSLinx

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

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



This action opens the Configure Drivers dialog box.



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.

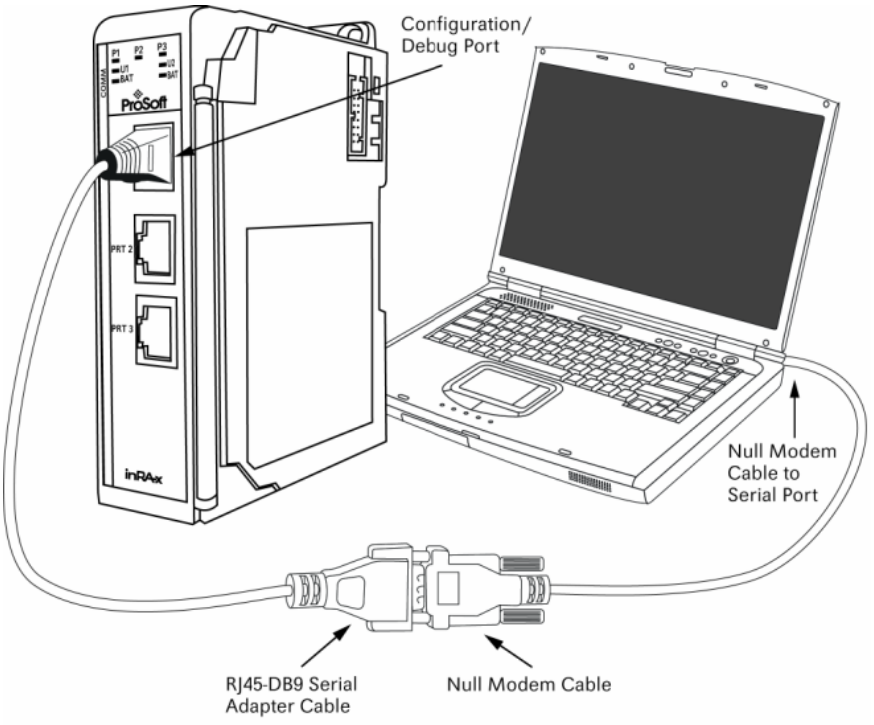
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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 Configuring the MVI69-103M Module

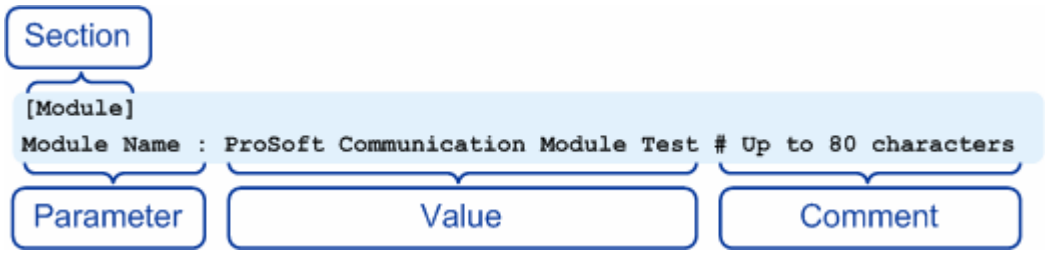
In This Chapter

2.1 Configuration File

The MVI69-103M module stores its configuration in a text file called IEC103M.CFG, located in the module's flash memory. When the module starts up, it reads the configuration file and uses the information to control how the 103 protocol interacts with the module's application port(s).

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

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



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

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

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

- Copy (page 31) the IEC103M.CFG from the module's flash memory to your PC

- Copy the IEC103M.CFG from the ProSoft Solutions CD-ROM supplied with the module
- Download the IEC103M.CFG from the ProSoft Technology web site at http://www.prosoft-technology.com

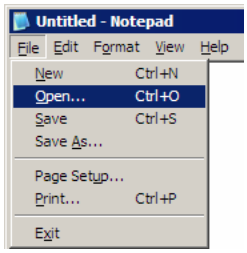
2.1.1 Editing the Configuration File

The IEC103M.CFG file is a plain ASCII text file. Use a text editor such as Notepad.exe (included with Microsoft Windows) to open and edit the file.

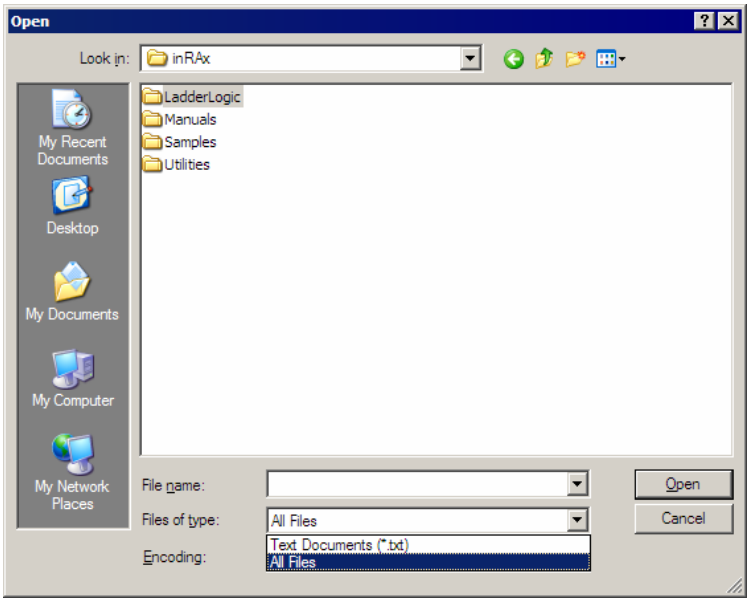
To open the configuration file in Notepad

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

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



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



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

6 Navigate to the folder containing the configuration file, and then select the file to edit.
7 Click Open to open the file.

8 When you have finished editing, save the file and close Notepad.

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

2.2 [Backplane Configuration]

2.2.1 Module Name

0 to 80 characters

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

2.2.2 Read Register Start

Range 0 to 3999

This parameter specifies the starting register in the module where data will be transferred from the module to the processor. Valid range for this parameter is 0 to 3999.

2.2.3 Read Register Count

Range 0 to 3999

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 3999.

2.2.4 Write Register Start

0 to 3999

This parameter specifies the starting register in the module where the data will be transferred from the processor to the module.

2.2.5 Write Register Count

Range 0 to 3999

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 3999

2.2.6 Failure Flag Count

0 through 65535

This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to 0, the communication ports will continue to operate under all conditions. If the value is set larger than 0 (1 to 65535), communications will cease if the specified number of failures occur.

2.2.7 Pass-Through Events

Y or N (N = Default)

This parameter specifies if event messages received on the master ports will be passed to the processor. If the parameter is set to N, event messages will not be passed to the processor. If the parameter is set to Y, the module will pass all events received to the processor using block identifier 9903.

2.2.8 Block Transfer Size

60, 120 or 240

This read-only parameter specifies the number of words in each block transferred between the module and processor. Valid values for this parameter are 60, 120 and 240.

2.3 [IEC-870-5-103 Master]

2.3.1 Session Count

1 to 16

This parameter specifies the maximum number of sessions to establish on the module. This corresponds to the number of slaves to be interfaced with the module. This value represents the total number of slaves on both ports combined.

2.4 [IEC-870-5-103 Master Port x]

2.4.1 Baud Rate

Value for baud rate

This parameter specifies the baud rate to be used on the communication channel (port). Two values are valid; 19200 or 9600.

2.4.2 Parity

N, O, E, M, or S

This parameter sets the parity to be used on the port. The values correspond to the following settings: N=None, O=Odd, E=Even, M=Mark and S=Space.

Note: the specification only utilizes Even Parity.

2.4.3 RTS On

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted.

2.4.4 RTS Off

0 to 65535 milliseconds

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.

2.4.5 Minimum Delay

1 to 65535

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

2.4.6 Receive Timeout

1 to 65535

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

2.4.7 Single char ACK F0, 1, or 3

Yes or No

This parameter specifies if the signal E5 character will be used for ACK messages.

2.5 [IEC-101 Master Session x]

This section is used to define session x which runs on Port x. The session sections of the configuration file are determined by the number of sessions set in the configuration file. The sessions are referenced by a zero based index value. For example, if the module is configured for four sessions, the configuration file should contain sections for sessions 0 to 3 (that is, [IEC-101 Master Session 0] to [IEC-101 Master Session 3]. Each of these sections will define the characteristics of the specific controlled device to be interfaced.

2.5.1 Communication Port

0 or 1

This parameter sets the port to which the controlled device is connected. On this module, values of 0 and 1 are permitted.

2.5.2 Sector Count

1 to 3

This parameter sets the number of sectors contained in this controlled device. This version of the application from 1 to 3 sectors (separate databases) for each session.

2.5.3 Data Link Address

0 to 254

This parameter uniquely defines the data link address for this unit on the communication channel. The ranges of values are from 0 to 254. Address 255 is the broadcast address.

2.5.4 Failure Delay

0 to 2000

This parameter sets the minimum number of seconds to delay before polling this session when it is not online. This parameter is only used in unbalanced mode.

2.5.5 Confirm Timeout

0 to 2^32-1

This parameter sets the number of milliseconds to wait for a confirm response from the controlled device.

2.5.6 Retry Count

0 to 255

This parameter sets the number of retries to be performed on the controlled device when a communication occurs.

2.5.7 C1/C2 Poll Count Pend

0 to 65535

This parameter sets the maximum number of class 1 and class 2 polls performed on this session before trying the next session. This parameter prevents a session from monopolizing the communication port.

2.5.8 Class 1 Polls

0 to 100

This parameter sets the maximum number of class 1 polls performed on this session before switching to another session. This parameter prevents a session from monopolizing the communication port.

2.5.9 Class 1 Pend Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between class 1 polls for pending data.

2.5.10 Class 2 Pend Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between class 2 polls for pending data.

2.5.11 Class 1 Poll Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between each class 1 poll.

2.5.12 Class 2 Poll Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between each class 2 poll.

2.5.13 Auto Clock Req Mode

0=Sync Only, 1=Load delay/sync, 2=Acquire delay/load delay/sync

This parameter specifies the method used to perform automatic clock synchronization. 0 performs a synchronization without delay, 1 performs synchronization using the fixed Propagation Delay and 2 computes the delay and use this value when synchronization takes place.

2.5.14 Propagation Delay

0 to 65535

This parameter sets the fixed propagation delay to be utilized if the Auto Clock Req Mode parameter is set to a value of 1.

2.5.15 Response Timeout

0 to 2^32-1 milliseconds

This parameter sets the maximum number of milliseconds to wait for a confirmation from the controlled station to a request from this module.

2.5.16 ACTTERM with setpoint

Yes or No

This parameter determines if an ACTTERM will be sent. If the parameter is set to Yes, then setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to No, ACTCON is the last response to a setpoint command.

2.6 [IEC-103 Master Session x Sector y]

Within each session definition, is a parameter that specifies the number of sectors for the session. For each sector defined for a session, there must exist a [IEC-103 Master Session x Sector y] section. Where the x value represents the session index and the y value represents sector index. For example if session 0 contains 1 sector, there must be a section with the following name in the configuration file: [IEC-103 Master Session 0 Sector 0].

2.6.1 Common ASDU Address

0 to 255

This parameter sets the common ASDU address to association with this sector of the specified session. This parameter is usually set the same as the data link address when only one sector is used.

2.6.2 Online Time Sync.

Yes or No

This parameter specifies if the sector in the controlled device will be sent a time synchronization command when the unit is first recognized as being online. This should only be used for devices that do not send an EOI message after initializing.

2.6.3 Online General Int

Yes or No

This parameter specifies if the sector in the controlled device will be sent a general interrogation command when the unit is first recognized as being online. This should only be used for devices that do not send an EOI message after initializing.

2.6.4 EOI Time Sync.

Yes or No

This parameter specifies if the sector in the controlled device will be sent a time synchronization command after this module received an EOI message from the controlled unit.

2.6.5 EOI General Int

Yes or No

This parameter specifies if the sector in the controlled device will be sent a general interrogation command after this module received an EOI message from the controlled unit.

2.7 [IEC-103 Master Commands]

This section can contain up to 1000 user defined commands to be executed by the module and sent to the controlled devices. There is no need to place Class 1 or Class 2 polls in the this list for the controlled devices as the master driver for each port will execute these automatically when the port is idle. In order for the port to be idle, make sure that there is idle time available, and that the commands do not constantly utilize the ports. The command list section starts with a reserved label **START** and ends with the label **END**. Each row in the file corresponds to an individual command with the first character position in each row left blank (white space).

As an alternative to using a command list, blocks with an identification code of 9901 can be used to issue commands from the ladder logic.

2.7.1 Enable Code

0 = Disabled

1 = Enabled with Poll Interval (seconds) utilized

2 = Conditional (executed when point in database changes)

This field defines whether or not the command is to be executed and under what conditions. If the parameter is set to 0, the command is disabled and will not be executed in the normal polling sequence. The command can be executed under the control of the PLC processor through the use of a Command Control block. Setting the parameter to a value of 1 for the command causes the command to be executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command will be executed, when the interval timer expires. If the parameter is set to 2, the command will execute only if the internal data associated with the command changes. This value is valid only for write commands.

2.7.2 Database Index

Database Index is the location in the module's database to use as the source for the data in the command. The data type field determines the meaning of the index as follows:

Type	Description	DB Index type
6	Clock synchronization	NA
7	General interrogation	NA
20	General Command	Bit address

2.7.3 Poll Interval

This parameter specifies the minimum frequency at which the module should execute the command. The value is entered in units of seconds. For example, to execute a command every 10 seconds, enter a value of 10 in the field. A value of 0 for the parameter implies that the command should be executed every scan of the list.

2.7.4 Session Index

Session Index represents the session index in the module to associate with the command. This index is set when the session is read in from this file. The range of values for this field is 0 to 31.

2.7.5 Sector Index

Sector Index represents the sector index for the specific session. The range of values for this field is 0 to 2.

2.7.6 Data Type

Data type file represents the ASDU type as follows:

6 = Time synchronization

7 = General Interrogation

20 = General Command

2.7.7 Function Code

Code Definition

128 Distance protection

160 Overcurrent protection

176 Transformer differential protection

192 Line Differential protection

255 Global function type

2.7.8 Point Index

Point Index field specifies the address in the remote slave device of the point to interact with.

2.8 Uploading and Downloading the Configuration File

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

This section describes these procedures.

Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module. For specific information about the configuration/debug menus in your module, refer to The Configuration/Debug Menu (page 51).

2.8.1 Required Hardware

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

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

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

2.8.2 Required Software

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

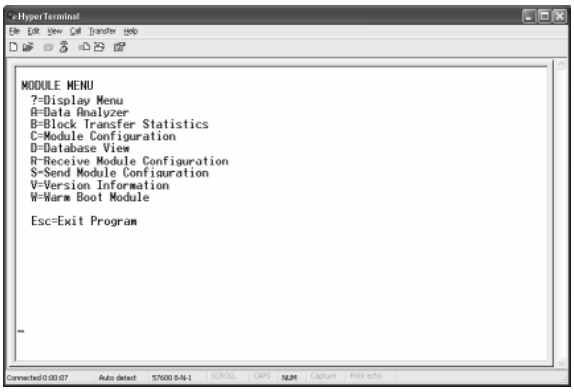
A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have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

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

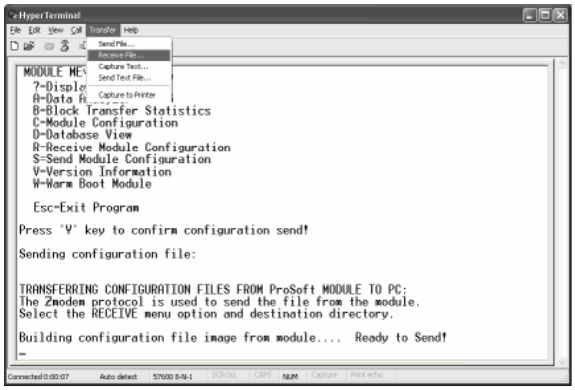
The module uses the Zmodem file transfer protocol to send (upload) and receive (download) configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Zmodem file transfers.

2.8.3 Transferring the Configuration File to Your PC

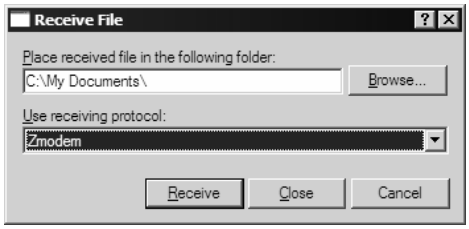
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



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

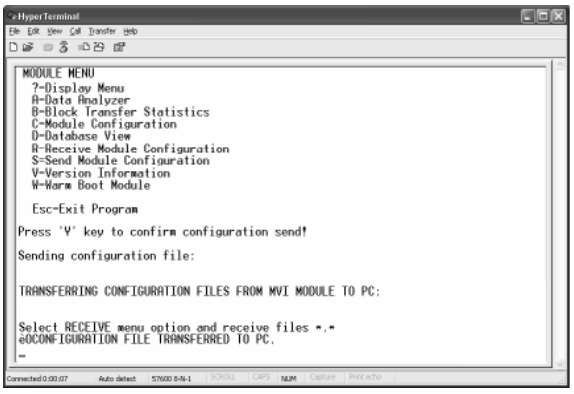


3 In the Receive File dialog box, browse to the location on your PC where the configuration file should be stored, and select Zmodem (or Zmodem with Crash Recovery) as the receiving protocol.

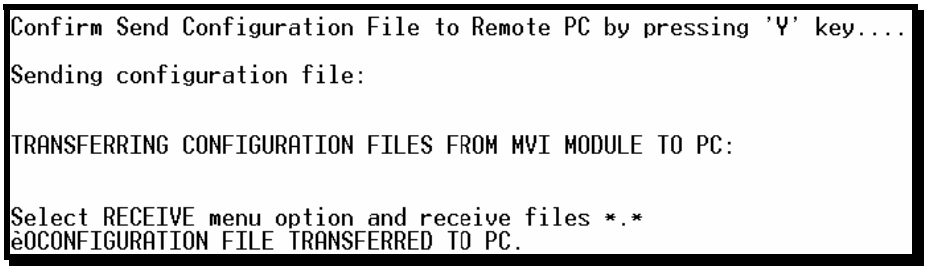


When you have completed your selections, click Close.

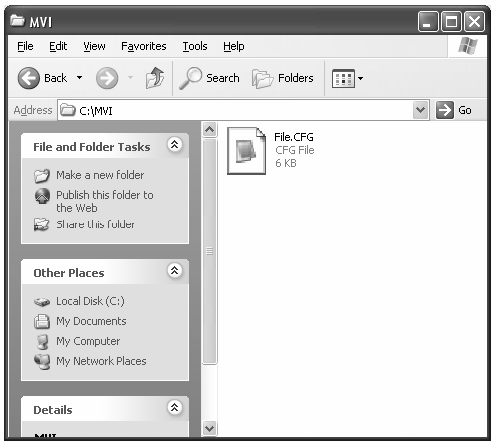
4 Press **[S]** (Send Module Configuration), and then press **[Y]** to confirm the transfer.



The file transfer will then begin automatically, using the protocol and location you specified in Step 3.
When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.



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

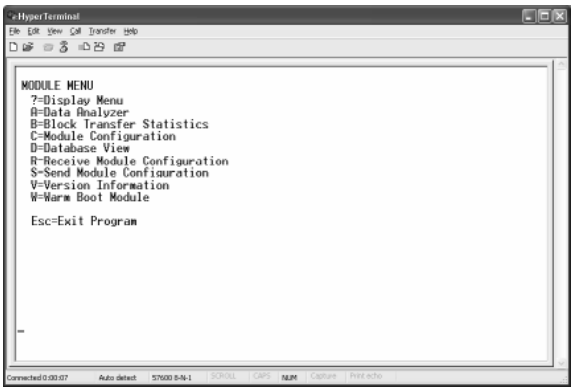


5 You can now open and edit the file in a text editor such as Notepad. (page 22) When you have finished editing the file, save it and close Notepad.

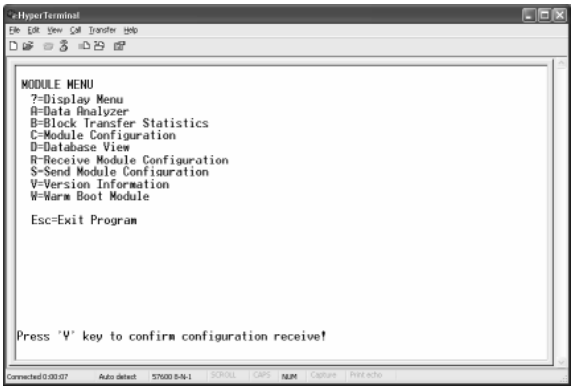
2.8.4 Transferring the Configuration File to the Module

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

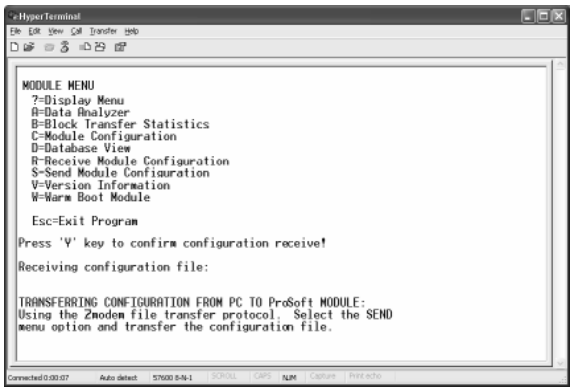
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



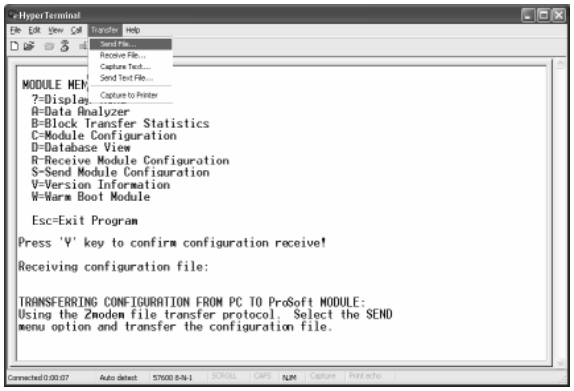
2 Press **[R]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



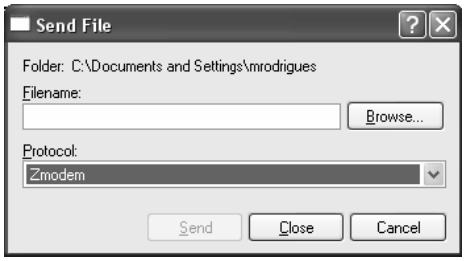
3 Press **[Y]**. The screen now indicates that the PC is ready to send.



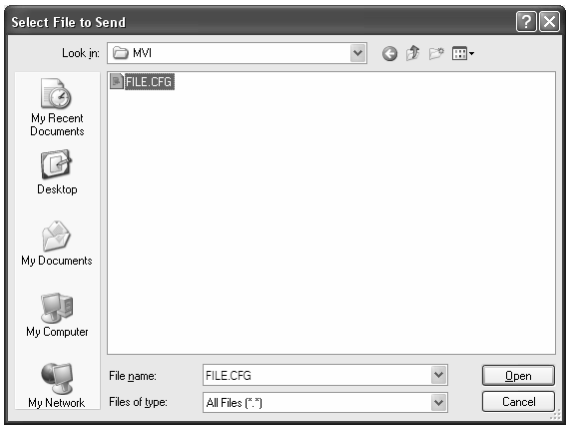
4 From the **Transfer** menu in HyperTerminal, select **Send File**.



The Send File dialog appears.



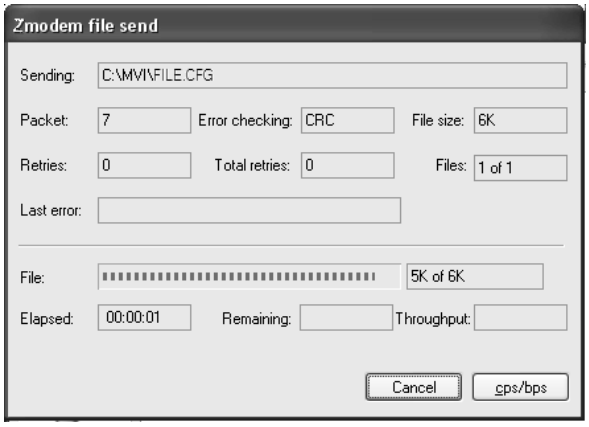
5 Use the Browse button to locate the configuration file your computer.



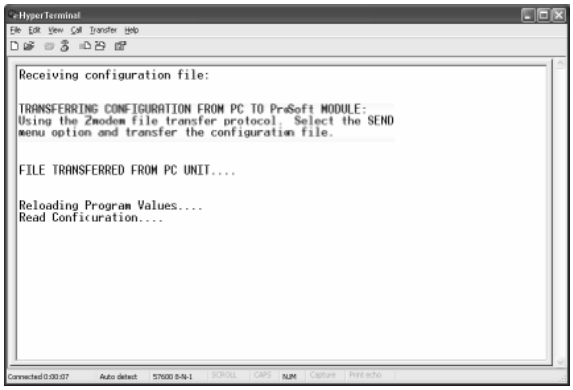
Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site.

6 Select Zmodem as the protocol.

7 Click the Send button. This action opens the Zmodem File Send dialog box.



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



8 Your module now contains the new configuration.

3 Ladder Logic

In This Chapter

Ladder logic is required for application of the MVI69-103M 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.

3.1 Module Data

This section describes the controller tags that are defined in the example logic to interface with the module. The user can extend these tags to meet the specifications required for their application.

3.1.1 Module Status Data and Variables (MVI69103M_ModuleDef)

All status and variable data related to the MVI69-103M is stored in a user-defined data type. An instance of the data type is required before the module can be used. This is done by declaring a variable of the data type in the Controller Tags Edit Tags dialog box. The structure of the object is displayed in the following example:

Tag Type	Data Type	Description
DATA	MV69103M_DATA	Stores data point values
CONTROL	MVI69103M_CONTROL	Optional block handling
STATUS	MVI69103M_STATUS	Stores module status
UTIL	MVI69103M_UTIL	Stores convenience variables for ladder logic usage

This object contains objects that define variables to be used with the module and status data related to the module. Each of these object types is discussed in the following topics of the document.

Status Object (MVI69103M_Status)

This object stores the status data of the module. The MVI69103M_Status object shown below is updated each time a read block is received by the processor. Use this data to monitor the state of the module at a "real-time rate".

Data Type	Description
Scan_Cnt	Program Scan Counter
Product_Name	Product Code
Revision_Level	Revision
Operating_System	Operating System revision
Run_Number	Run number
Block_Read_Count	Number of block read transfers
Block_Write_Count	Number of block write transfers
Block_Parse_Count	Number of blocks parsed by module
Block_Error	Number of block errors
Event_Count	Number of event messages in buffer
Event_Overflow	Flag to indicate event message buffer overflow (1=overflow)
SessionCount	Number of sessions configured
CurentCommand	Index of command executing
CommanddBusy	Command busy flag
CommandMax	Maximum number of commands configured
CommandDelay	Command delay counter
CommandQueue	Command Queue Flag
CommandQueueCount	Number of commands in command queue
Online	Online status bits for each session
ChStat	Channel Status Data
BlockTransferSize	60, 120 or 240

Within the MVI69103M_Statust objects are objects containing the status information for each application port (MVI69103M_ChannelStatus). Refer to 103M Status Data Area for a complete listing of the data stored in this object.

Channel Status Object (MVI69103M_ChannelStatus)

The MVI69103M_ChannelStatus object holds the status data related to a single IEC 60870-5-103 Master port. The structure of this object is shown in the following example:

Data Type	Description
State	State machine value
CmdReq	Number of command requests
CmdResp	Number of command responses
CmdErr	Number of command errors
Req	Number of request messages
Resp	Number of responses
ErrSent	Number of errors sent
ErrRec	Number of errors received

Data Type	Description
CfgErr	Configuration Error Word for channel
CurErr	Current error code for channel
LastErr	Last error for channel

This information is passed to the controller from the module with each normal read block image.

Configuration/Error Status Flags (CfgErr)

The CfgErr word member of the MVI69103M_ChannelStatus reports configuration errors for the respective server. If the module is not functioning as expected, inspect the value 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 bits used by this member are shown in the following table.

Bit	Code	Description
0	0x0001	Invalid baud rate selected
1	0x0002	Invalid parity selected
2	0x0004	Received timeout set to 0
3	0x0008	Invalid Port selected for a session
4	0x0010	Invalid sector count for session
5	0x0020	Could not allocate memory for sector of a session.
6	0x0040	
7	0x0080	Invalid failure delay or confirm timeout for session.
8	0x0100	
9	0x0200	
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

3.1.2 Backplane Object (MVI69103M_UTIL)

The MVI69103M_UTIL object stores all the variables required for the data transfer operation between the module and the controller. The LastRead data member is used as the handshaking byte to indicate the arrival of new data from the module. The structure of this object is shown in the following illustration:

Data Type	Description
Warmboot	Requests warmboot of the module
ColdBoot	Requests coldboot of the module
BPLastRead	Index of last read
BPLastWrite	Index of last write
BlockIndex	Computed block offset for data table

Data Type	Description
BootTimer	Used to clear output block ID after power up
LoopIndex	Used as index during Loops
TempByte	Used for INT/SINT conversion
TempINT	Used for SINT/INT Conversion
TempDINT	Used for INT/SINT conversion

The other members of the object are be utilized in the ladder logic to assist in the data transfer operation.

3.1.3 Data Object (MV69103M_DATA)

Data for the module is stored in two controller tags for the example ladder logic. The read data (data transferred from the module to the processor) is stored in the controller tag MVI69103M.DATA.ReadData[ ]. The write data (data transferred from the processor to the module) is stored in the controller tag MVI69103M.DATA.WriteData[ ]. Separate tags can be constructed for each data type utilized by the controlled devices and for each device.

3.1.4 User Command Data Object (MVI69103M_UserCommand)

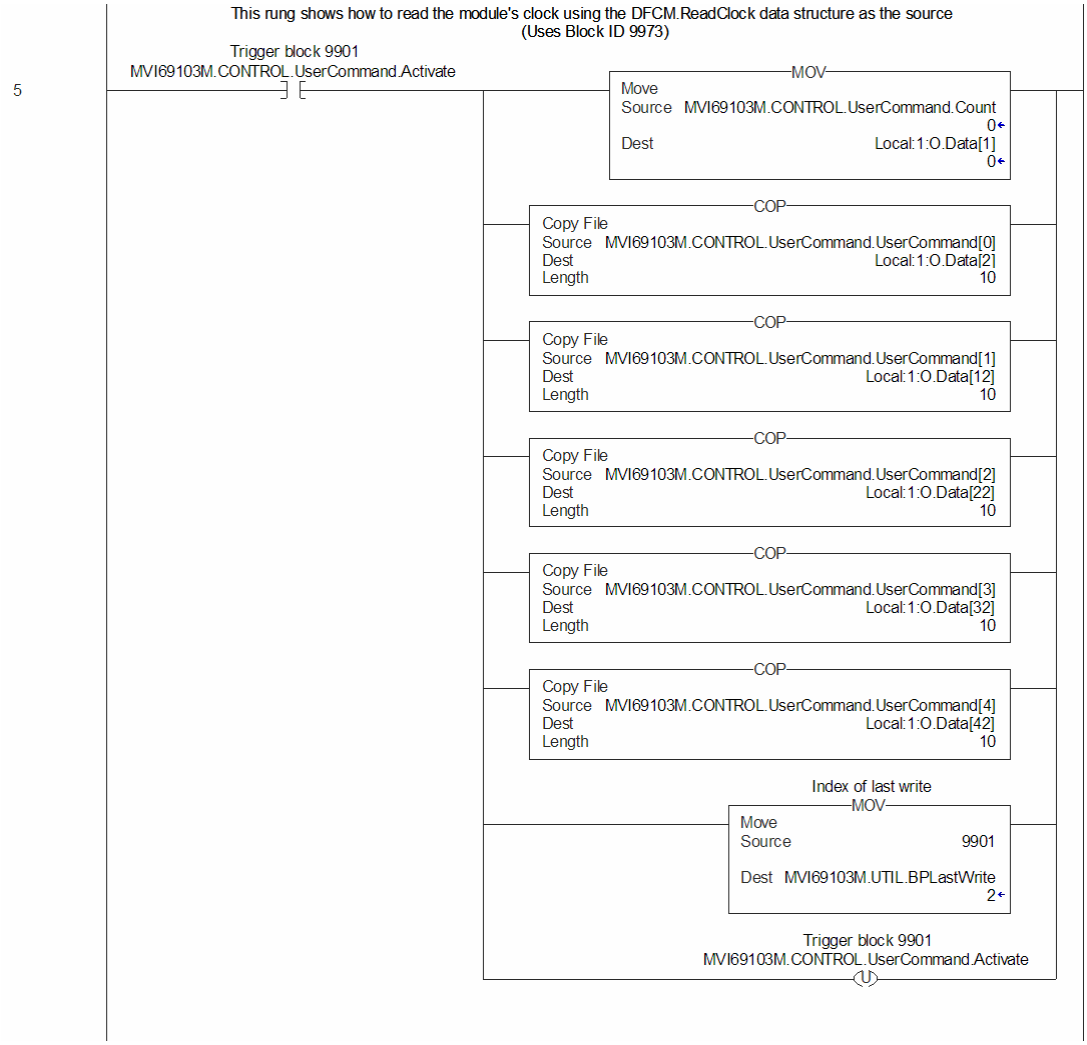
User Command is an optional feature that allows the processor to dynamically build commands, instead of using the configured commands from the configuration file.

In order to support the user command control of the module, refer to MVI69103M.CONTROL.UserCommand.Activate to trigger the block. The tag MVI69103M.CONTROL.UserCommand.Count sets the number of commands to be sent (maximum of 5 per block). Use the tag MVI69103M.CONTROL.UserCommand.UserCommand to set the commands. The structure of this object is shown in the following figure:

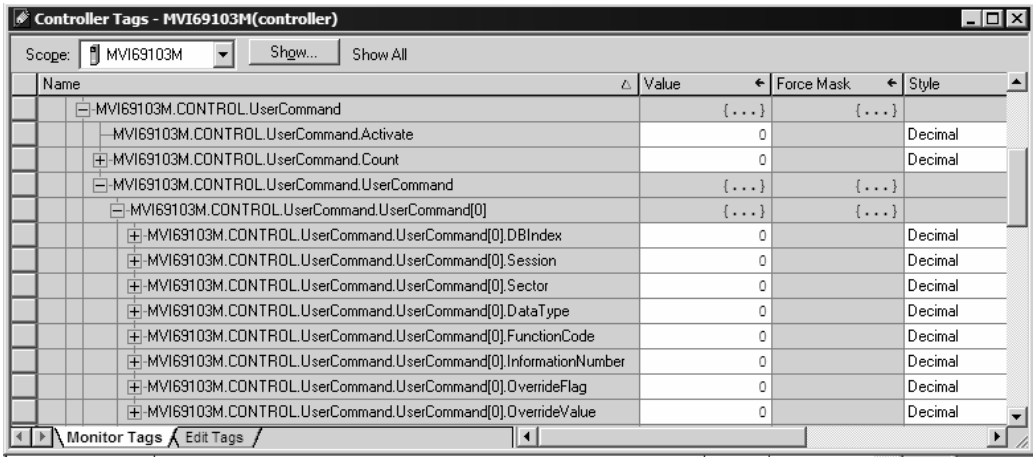
Data Type	Description
DBIndex	Address in module's database to associate with command
Session	Session Index
Sector	Sector Index
DataType	ASDU data type for command
FunctionCode	Function code
InformationNumber	Information object address of the point
OverrideFlag	Override flag used with ASDU 20 (0=use db value, 1=use override)
OverrideValue	Override value to use if override flag set

Refer to the Command List section of this document for the definition of each of the parameters. This option permits execution of user-generated commands from the ladder logic directly to controlled devices. This feature is especially important in generating general commands (ASDU 20 object) request.

Example ladder logic to use this feature is shown in the following rung:

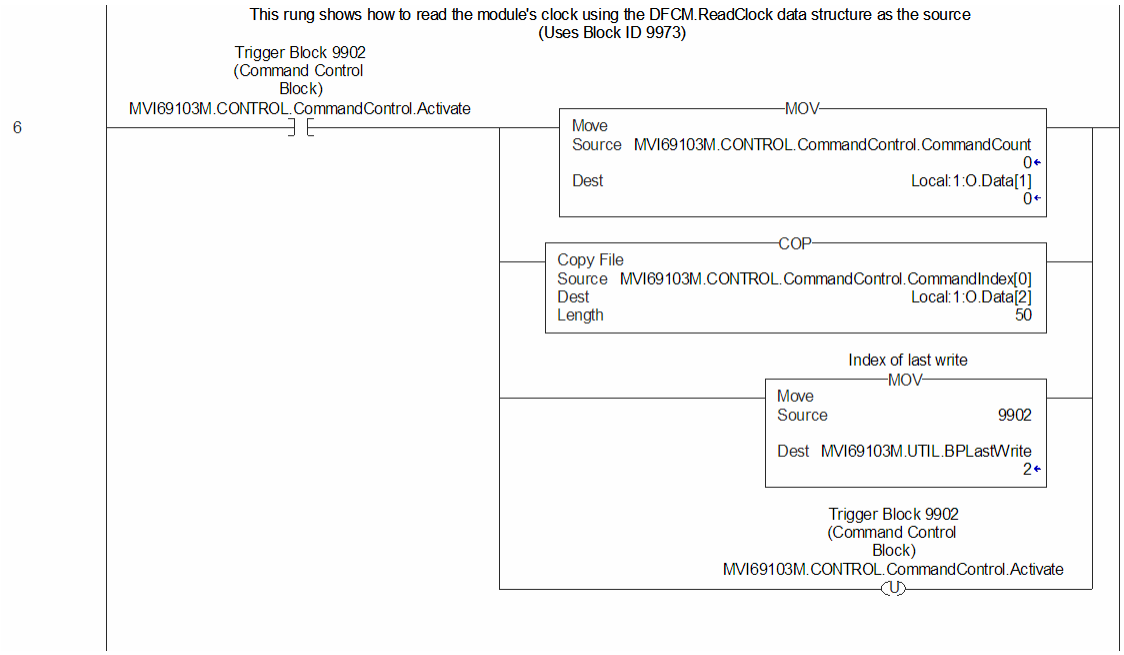


When the command bit (MVI69103M.CONTROL.UserCommand.Activate[0]) is set, the module will build a block 9901 with the command contained in the first command of the MVI69103M.CONTROL.UserCommand.UserCommand[ ]. The module receives this block and builds and sends the command to the specified control device. The data for the command element in the MVI69103M.CONTROL.UserCommand.UserCommand array is shown in the following figure:



3.1.5 Command Control Data Object (MVI69103M_UserCommand)

The following rung displays ladder logic to build a 9902 block to place commands 2 and 3 of the user command list configured in the module's command queue:



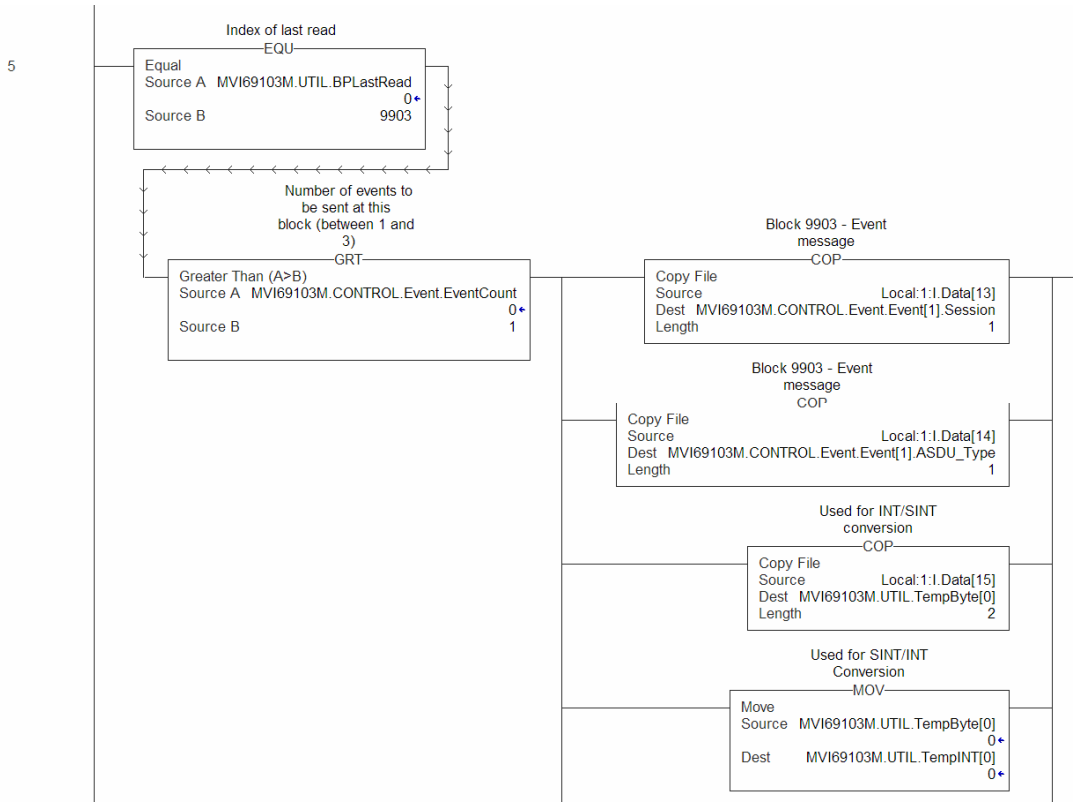
When the command bit (MVI69103M.CONTROL.UserCommand.Activate) is set, the module will build the block 9902 in the output image. The module will receive the new block and place the commands into the command queue for execution.

3.1.6 Event Message Data Object (MVI69103M_Event)

The module can be configured to pass event messages received from the controlled devices to the processor. Refer to the Pass-Through Events (page 24) parameter for further information about this feature. The module sends this information to the processor in read blocks with identification codes of 9903. The example ladder logic has a data structure that conforms to the data structure sent for each element by the module. The following figure displays the object:

Data Type	Description
Session	Session values
Sector	Sector values
ASDU_Type	ASDU type for event message
PointIndex	Point index
Function	Function index
FaultNumber	Fault Number for event (ASDU type 2 and 4 only)
Milliseconds	Milliseconds
Seconds	Seconds
Minutes	Minutes
Hour	Hour
Invalid	Valid time flag
DST	Daylight savings time flag
RelativeTime	Relative Time (ASDU type 2 and 4 only)
Value	Value for event received (may want to set float type for ASDU 4)

The example ladder logic defines a 5-element array of the MVI69103M_Event objects to hold the data received in a single 9903 block. Ladder logic to handle a 9903 block is shown in the following figure:



This simple logic does not use the message count parameter in the block and will not work for most applications. It is only provided as an example to display how to move the 9903 data to a user controller tag. If the message pass-through feature is utilized, more logic is required to store the messages received. This feature is most commonly utilized to pass events from the controlled device to an event logger connected to the processor.

3.1.7 Data Object (MVI69103M_Clock)

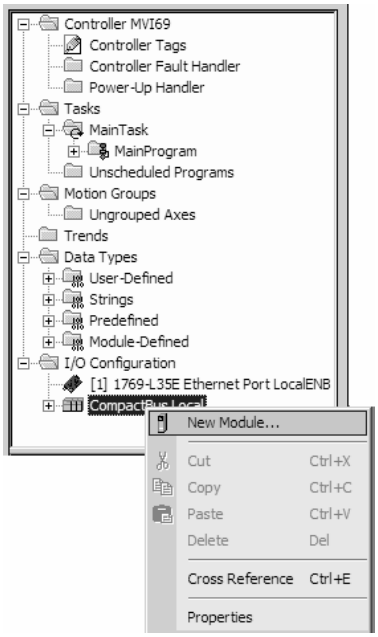
This type is used to copy the date and time information between the MVI and the processor through optional block 9970 and 9971. Please refer to the sample ladder logic (on the ProSoft Solutions CD) for further information about this feature.

3.2 Adding the Module to an Existing CompactLogix Project

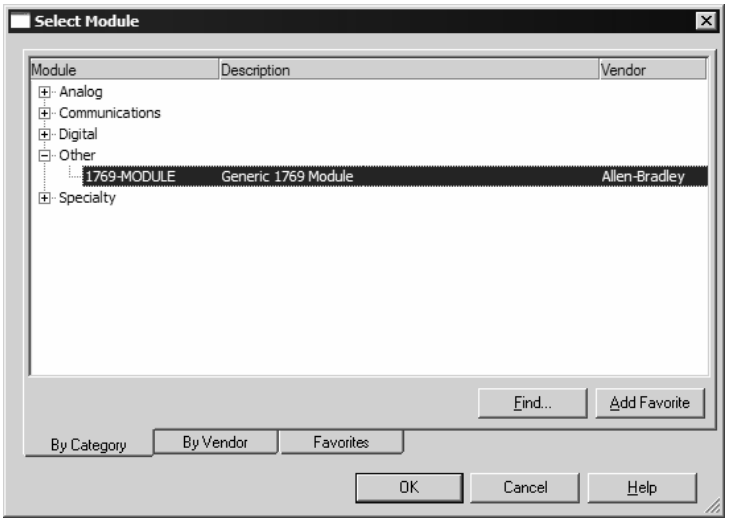
Important: The MVI69-103M module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

If you are installing and configuring the module with a CompactLogix processor, follow these steps. If you are using a MicroLogix processor, refer to the next section.

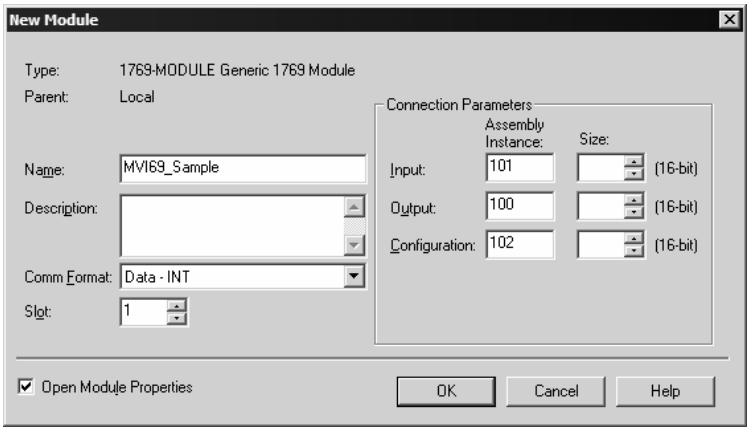
1 **Add the** MVI69-103M **module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



This action opens the following dialog box:



2 Select the 1769-Module (Generic 1769 Module) from the list and click OK.



3 Enter the Name, Description and Slot options for your application, using the values in the illustration above. You must select the **Comm Format** as **Data - INT** in the dialog box, otherwise the module will not communicate over the backplane of the CompactLogix rack.

4 Configure the Connection Parameters to match to the Block Transfer Size parameter in the configuration file. Use the values in the table corresponding with the block transfer size you configured.

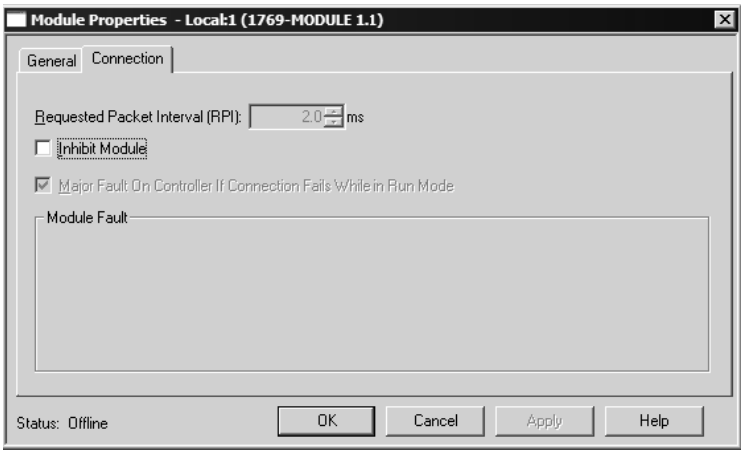
Block Transfer Size = 60	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT

Block Transfer Size = 60	
Field	**Recommended Value**
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	62
Output Assembly Instance	100
Output Size	61
Configuration Assembly Instance	102
Configuration Size	0

Block Transfer Size = 120	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	122
Output Assembly Instance	100
Output Size	121
Configuration Assembly Instance	102
Configuration Size	0

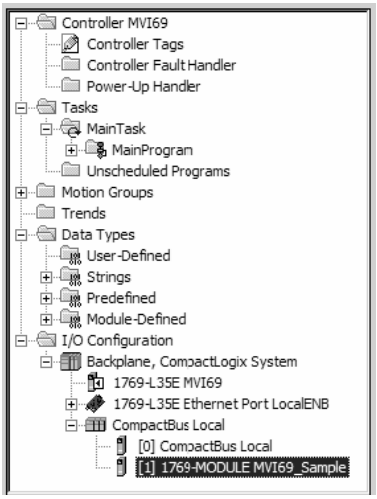
Block Transfer Size = 240	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	242
Output Assembly Instance	100
Output Size	241
Configuration Assembly Instance	102
Configuration Size	0

5 Click **Next** to continue.



6 Select the Request Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications.

7 Save the module. Click OK to dismiss the dialog box. The Controller Organization window now displays the module's presence. The following illustration shows the Controller Organization window:



8 Copy the Controller Tags from the sample program.

9 Copy the User Defined Data Types from the sample program.

10 Copy the Ladder Rungs from the sample program.

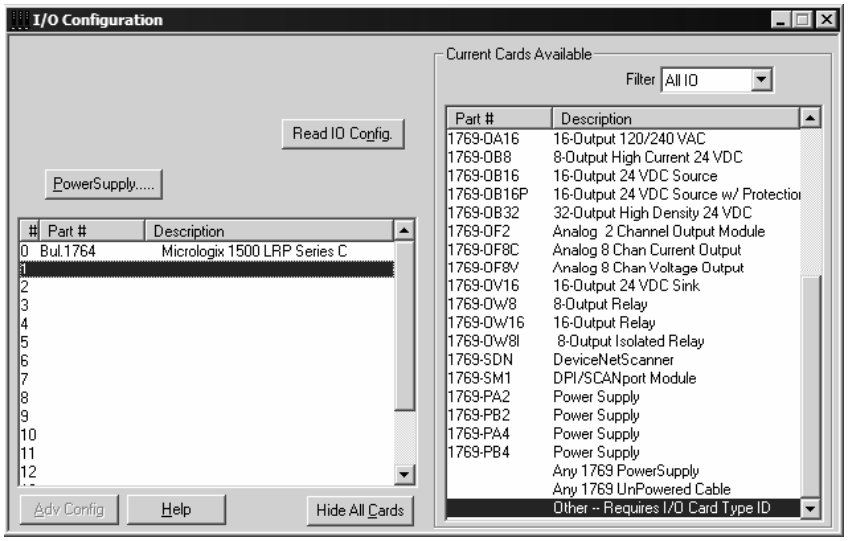
11 Save and Download the new application to the controller and place the processor in run mode.

3.3 Adding the Module to an Existing MicroLogix Project

If you are installing and configuring the module with a MicroLogix processor, follow these steps. If you are using a CompactLogix processor, refer to the previous section.

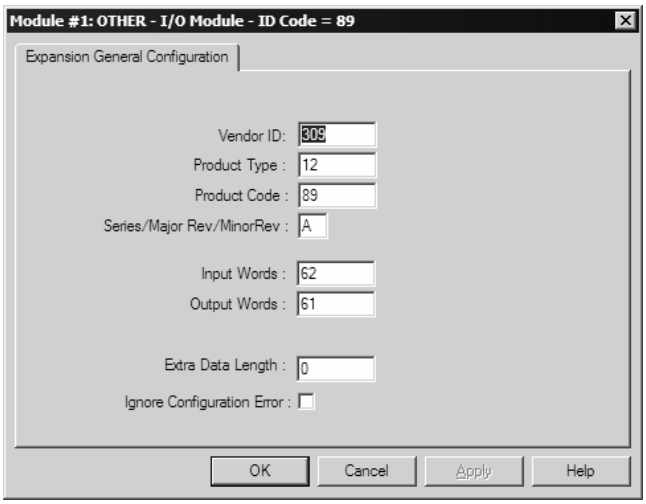
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

1. In RSLogix, open your existing application, or start a new application, depending on your requirements.
2. Double-click the I/O Configuration icon located in the Controller folder in the project tree. This action opens the I/O Configuration dialog box.



3. On the I/O Configuration dialog box, select "Other - Requires I/O Card Type ID" at the bottom of the list in the right pane, and then double-click to open the Module dialog box.

4 Enter the values shown in the following illustration to define the module correctly for the MicroLogix processor, and then click OK to save your configuration.



The input words and output words parameter will depend on the Block Transfer Size parameter you specify in the configuration file. Use the values from the following table.

Block Transfer Size	Input Words	Output Words
60	62	61
120	122	121
240	242	241

5 Click **Next** to continue.

6 After completing the module setup, the I/O configuration dialog box will display the module's presence.

The last step is to add the ladder logic. If you are using the example ladder logic, adjust the ladder to fit your application. Refer to the example Ladder Logic section in this manual.

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 51) for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

4 Diagnostics and Troubleshooting

In This Chapter

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

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

4.1 Reading Status Data from the Module

The MVI69-103M module returns a status data set to the CompactLogix or MicroLogix processor in each read block. This data is transferred to the CompactLogix or MicroLogix processor continuously with each read block.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Version Information
- Control over the module (warm boot and cold boot)
- Facility to upload and download the module's configuration file

4.1.1 The Configuration/Debug Menu

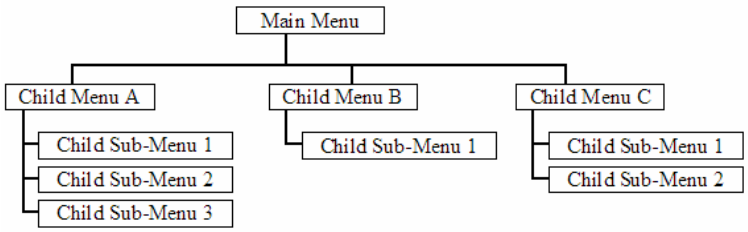
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

Navigation

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

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



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

Keystrokes

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

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

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

4.1.2 Required Hardware

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

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

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

4.1.3 Required Software

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

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

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

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

4.1.4 Using the Configuration/Debug Port

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

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

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

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

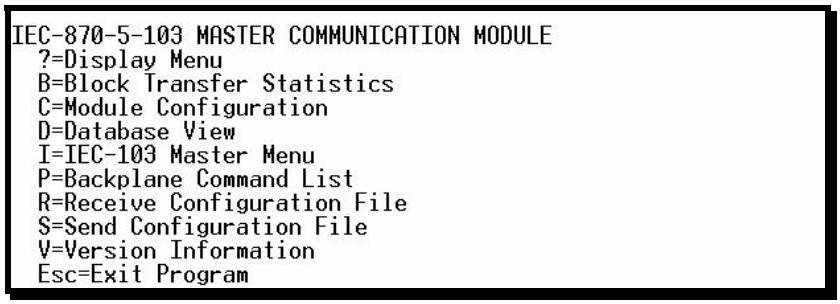
If there is no response from the module, follow these steps:

1. Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2. Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC (page 90).
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.

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.

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

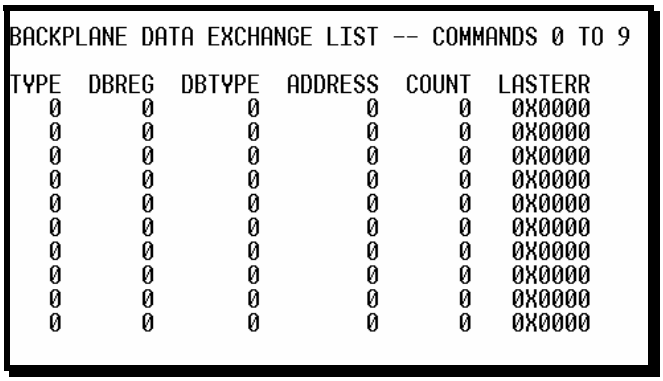
Opening the IEC-103 Master Menu

Press **[I]** from the Main Menu to open the IEC-870-5-103 Master Driver Menu. Use this menu command to view detailed configuration information for the module.

For more information about the commands on this menu, refer to IEC-103 Master Driver Menu (page 58).

Viewing the Backplane Command List

Press **[P]** from the Main Menu to view the Backplane Data Exchange List. Use this command to display the configuration and statistics of the backplane data transfer operations.



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

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File (page 31).

Sending the Configuration File

Press **[S]** to upload (send) an updated configuration file to the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File (page 31).

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.

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.

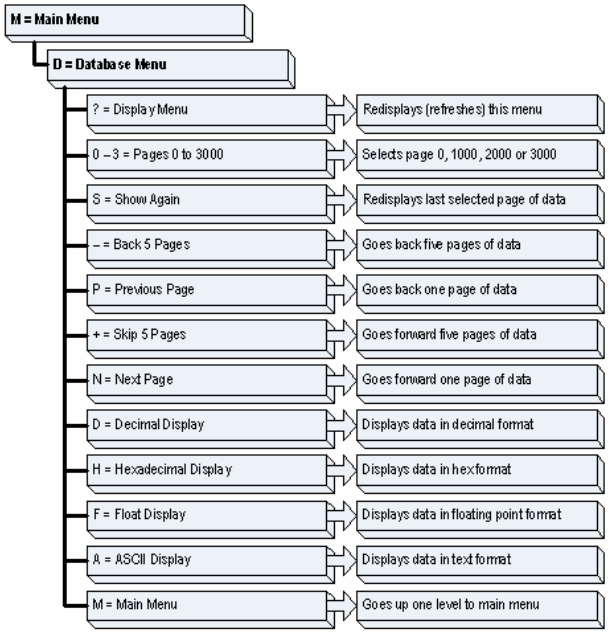
Warm Booting the Module

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.

4.1.6 Database View Menu

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



Viewing Database Pages 0 to 3000

The database is divided into pages that correspond with a specific number of registers. The total number of database pages and registers depends on the memory capacity and configuration of the module.

Use the keyboard commands **[0]** through **[3]** to display database contents starting from 0 (zero), 1000, 2000 and 3000 respectively.

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

Press **[F]** from the Database View menu. Use this command to display the data on the current page in floating point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

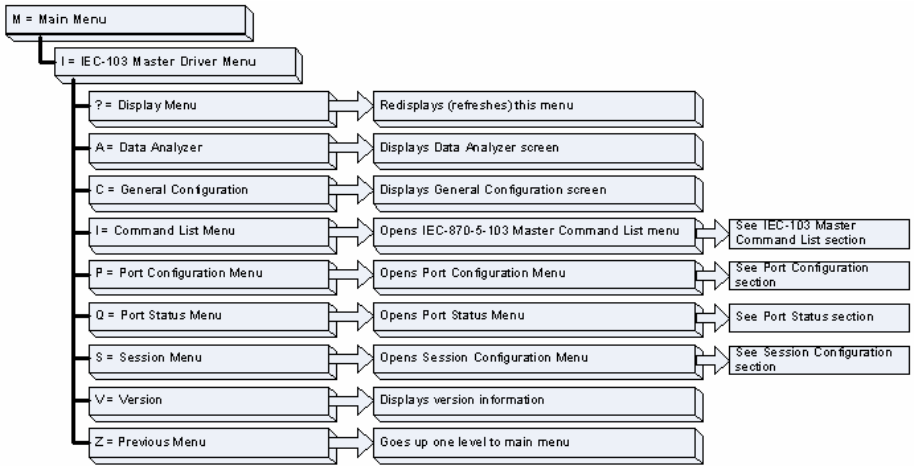
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.1.7 IEC-103 Master Driver Menu

Press **[I]** from the Main Menu to open the IEC-870-5-103 Master Driver Menu. Use this menu command to view detailed configuration information for the module.



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 General Configuration

Press **[C]** from the IEC-103 Master Driver Menu to display the general configuration for the protocol. The following illustration shows an example of the Module Configuration screen:

```
MODULE CONFIGURATION
  Max Sessions    = 0 (of maximum 32)
  Online States   = 0x00000000 (bit mapped in hex format)
  Command Count   =    0 (of maximum 1000)
  Current Command =    0 (IDLE)
 MEMORY COUNTERS:
  applRec   = 0
  applTrans = 0
  anything  = 0
```

The Busy/Idle message indicates the current activity state of the module. "Idle" means it is waiting to execute a command. "Busy" means it is executing a command and is waiting for the response to the request. This does not include the normal class 1 and 2 polls as these are automatically generated.

The counter data displays the number of memory areas allocated for the application layer. When no packets are pending, the counts should all be 0. If messages are waiting to be sent, the applRec count will indicate the number waiting to be sent. If many messages are received at the same time, the applRec count will indicate the number of packets that must be processed. The "anything" count indicates any other buffer area that is allocated and must be processed by the application.

Opening the IEC-870-Master Command List Menu

Press **[I]** from the IEC-103 Master Driver Menu to open the ICE-870 Master Command List menu. Use this command to view the configured command list for the module.

```
IEC 870 5 103 MASTER COMMAND LIST, COMMANDS 0 TO 9

CMD DDIDX POLLI SES SEC ASDU FUNC POINT LAST POLL LENROR OVR VALUE
  1     0    10   0   0   20  128    16          7      0   1     2
  0     0    11   0   0   20  128    16      14932      0   1     1
  0  1680    41   0   0   20  128    17      14932      0   0     0
  0  1682    41   0   0   20  128    18      14932      0   0     0
  1     0     5   0   0    7  255    40          4      0

***** END OF COMMAND LIST REACHED *****
```

For more information about the commands on this menu, refer to IEC-870-Master Command List Menu (page 61).

Opening the Port Configuration Menu

Press **[P]** from the IEC-103 Master Driver Menu to open the Port Configuration menu. Use this command to view the port configuration information for each of the application ports.

```
Port Configuration Menu Selected

IEC-870-5-103 MASTER CHANNEL 0 CONFIGURATION
  Baudrate          = 19200
  Parity            = EVEN
  RTS On            = 1
  RTS Off           = 0
  Mimimum Delay     = 30
  Receive Timeout   = 2000
  Single ACK        = YES
  Data Link Length  = 1
  Use Balanced Mode = NO
```

The *Port Configuration Menu* section has more information about the commands on this menu.

Opening the Port Status Menu

Press **[Q]** from the IEC-103 Master Driver Menu to open the Port Status menu. Use this command to verify the status of the master commands sent through the port. If the display indicates a communication error, you should compare the generated error code with the command error codes listed in the Appendices of this manual.

```
Port Status Menu Selected

IEC-870-5-103 MASTER PORT 0 STATUS
  Commands Executed  = 0
  Command Responses  = 0
  Command Err Count  = 0
  Request Count      = 0
  Response Count     = 0
  Error Sent Count   = 0
  Error Rec Count    = 0
  Cfg Error Word     = 0x0000
  Current Error Code = 0
  Last Error Code    = 0
```

The *Port Status Menu* section has more information about the commands on this menu.

Opening the Session Configuration Menu

Press **[S]** to open the Session Configuration menu. Use this command to view the session configuration data.

Refer to *Session Configuration Menu* (page 66) for more information about the commands on this menu.

Opening the Sector Menu

Press **[1]** from the IEC-103 Master Driver Menu to open the Sector Configuration menu. Use this command to view the Sector Configuration data.

```
SECTOR CONFIGURATION MENU
 ?=Display Menu
 D=Sector Database Menu
 S=Show Again
 P=Previous Page
 N=Next Page
 M=Return to Session Menu
```

The *Sector Configuration Menu* section has more information about the commands on this menu.

Viewing Master Driver Version Information

Press **[V]** from the IEC-103 Master Driver Menu to view the master driver version information.

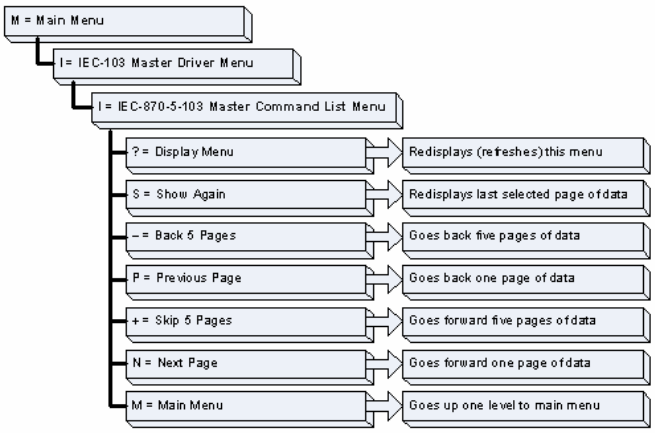
```
DRIVER VERSION INFORMATION:

  (c) 1999-2003, ProLinx Communication Gateways, Inc.

  PRODUCT NAME CODE       : IEC-870-5-103 Master Driver
  SOFTWARE REVISION LEVEL : 2.14
  SOFTWARE REVISION DATE  : 04/17/2003
  FAR CORE LEFT           : 322368
```

Returning to the Main Menu

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

4.1.8 IEC-870-Master Command List Menu

Press **[I]** from the IEC-103 Master Driver Menu to open the ICE-870 Master Command List menu. Use this command to view the configured command list for the module.



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

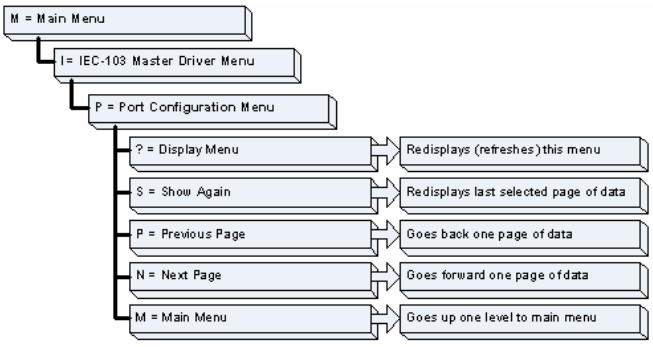
Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Returning to the Main Menu

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

4.1.9 Port Configuration Menu

Press **[P]** from the IEC-103 Master Driver Menu to open the Port Configuration menu. Use this command to view the port configuration information for each of the application ports.



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Next Page

Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Displaying the Previous Page

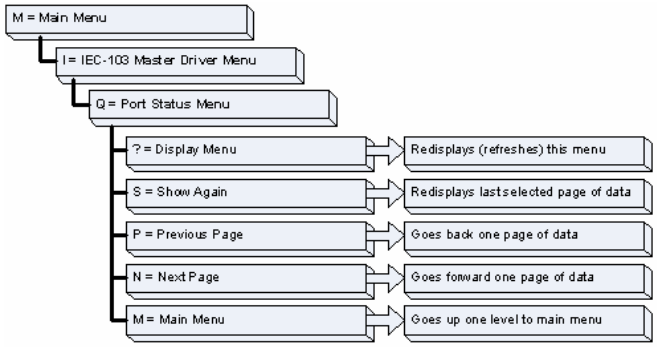
Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Returning to the Main Menu

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

4.1.10 Port Status Menu

Press **[Q]** from the IEC-103 Master Driver Menu to open the Port Status menu. Use this command to view the communication status information for each application port.



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Previous Page

Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Displaying the Next Page

Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Returning to the Main Menu

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

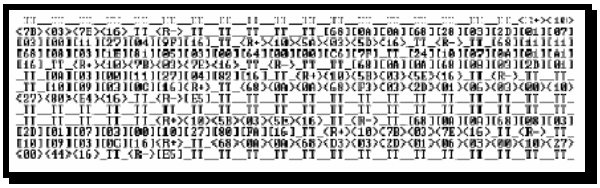
4.1.11 Data Analyzer

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

Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.
Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

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



Analyzing Data for the second application port

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

Displaying Timing Marks in the Data Analyzer

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

Key	Interval
[5]	1 milliseconds ticks

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

Removing Timing Marks in the Data Analyzer

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

Viewing Data in Hexadecimal Format

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

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

Press **[B]** to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port will be displayed. The following illustration shows an example.

```
<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][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]_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]_TT_[00][00][00][00][00][A3][67]_TT_<R+><01><03><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][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][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][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]_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]_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][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][A3][67]_TT_
```

The Data Analyzer displays the following special characters:

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

Stopping the Data Analyzer

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

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

Returning to the Main Menu

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

4.1.12 Session Configuration Menu

Press **[S]** from the IEC-103 Master Driver Menu to open the Session Configuration menu. Use this command to view the session configuration for each controlled device.

```
IEC-870-5-103 MASTER SESSION 0 CONFIGURATION
  Online State           = 0
  Session State          = 2  (0)
  Communication Port     = 0
  Sector Count           = 3
  Data Link Address      = 0
  Failure Delay          = 10
  Confirm Timeout        = 1000
  Retry Count            = 0
  C1/C2 Poll Count Pend  = 5
  Class 1 Polls          = 0
  Class 1 Pend Delay     = 0
  Class 2 Pend Delay     = 0
  Class 1 Poll Delay     = 0
  Class 2 Poll Delay     = 0
  Response Timeout       = 20000
```

Online State

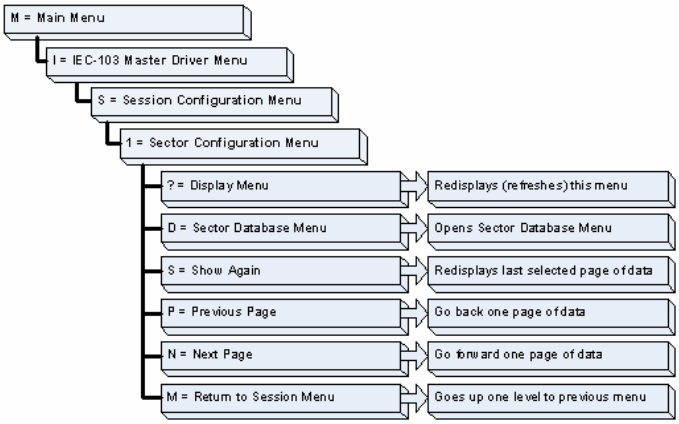
The Online State indicator displays 0 if the module is not online, 1 if the module is online.

Session State

The Session State indicator displays 1 if there is a configuration error, or 2 if the module is ready for communication. If the session is not in use, the Session State indicator displays 0.

4.1.13 Sector Configuration Menu

Press **[1]** from the IEC-103 Master Driver Menu to open the Sector Configuration menu. Use this command to view the contents of the Sector Configuration Databases for each session (controlled device). The module supports up to three sectors (databases) per session.



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Opening the Sector Database Menu

Press **[D]** from the Sector Configuration menu to open the Sector Database menu. Use this command to look at the configuration and current value for each point.

The *IEC-870-Master Command List Menu* section has more information about the commands on this menu.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Next Page

Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Displaying the Previous Page

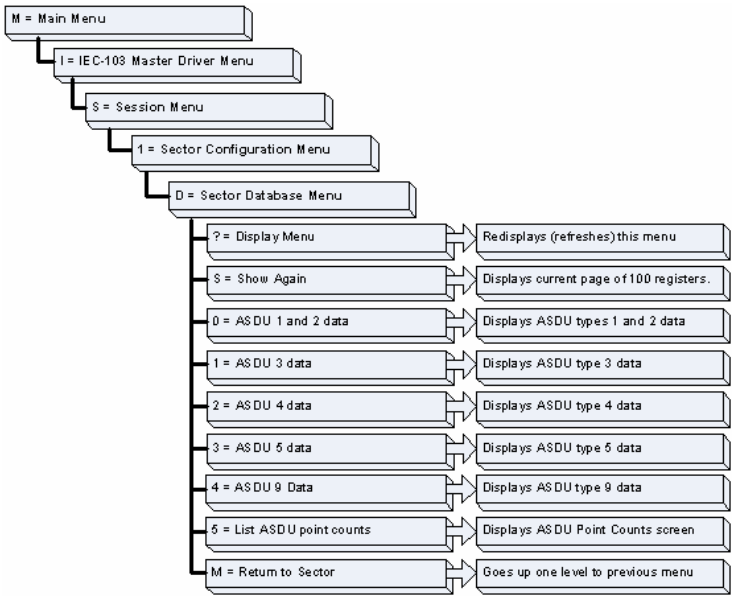
Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Returning to the Main Menu

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

4.1.14 Sector Database Menu

Press **[D]** from the Sector Configuration menu to open the Sector Database menu. Use this command to display the sector database values. Each session (controlled device) contains one or more data sets (sectors) that are defined by the vendor of the device.



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing ASDU n Data

Press keys **[0]** (zero) through **[4]** to display ASDU (Application Data Service Unit) data for each of the supported data types. Refer to *[IEC-103 Master Session x Sector x]* for a list of ASDU types.

Listing ASDU point counts

Press **[5]** to display the ASDU point counts for each ASDU type.

```
SESSION 0 SECTOR 0 DATABASE SIZES:
  ASDU    COUNT
   1&2       10
     3        0
     4        0
     5        0
     9        0
```

Returning to the Main Menu

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

4.2 LED Status Indicators

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

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 being transferred on the first application port
		Off	No data being transferred on port
P2	Green	On	Data being transferred on the second application port
		Off	No data being transferred on port
APP Status	Amber	Off	The MVI69-103M is working normally.
		On	The MVI69-103M module program has recognized a communication error.
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. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

4.2.1 Clearing a Fault Condition

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

To clear the condition, follow these steps:

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

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

4.2.2 Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic. Module has a power supply distance rating of 2 on Compact Logix. The module must be within 2 slots of the power supply on Compact Logix, or that the MicroLogix backplane can supply the 800ma required for the module.

Module Errors

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI69 IEC 60870-5-103 Master Communication Module allows CompactLogix I/O compatible processors to interface easily with IEC 60870-5-103 slave (controlled unit) protection devices. Devices commonly supporting the protocol include relays, breakers, sub-station communication modules and other serial communication devices used in power monitoring.

The MVI69-103M supports up to 16 total field devices between the module's two IEC 60870-5-103 Master ports. The module's communication ports can be independently configured, allowing two separate field networks to be implemented. The field device data is exchanged between the MVI module and the CompactLogix processor over the backplane.

The MVI69-103M module is a powerful communication interface for CompactLogix processors. Developed under license from Rockwell Automation, the module incorporates proprietary backplane technology that enables powerful data access to the CompactLogix processor.

5.1.1 General Specifications

- Single Slot - 1769 backplane compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module
- Ladder Logic is used for data transfer between module and processor. Sample ladder file included.
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included.
- Supports all CompactLogix processors: L20/L23/L30/L31/L32/L35, L43 and L45 (L43 and L45 supported with RSLogix 5000 v16.03 or later)
- Also supports MicroLogix 1500 LRP

5.1.2 Hardware Specifications

Specification	Description
Dimensions	Standard 1769 Single-slot module
Current Load	800 mA max@ 5 VDC
	Power supply distance rating of 2
Operating Temp.	0 to 60°C (32 to 140°F)
Storage Temp.	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Power and Module Status
	Application Status
	Serial Port Activity
	Error Status
CFG Port (CFG)	RJ45 (DB-9F with supplied cable)
	RS-232 only
	No hardware handshaking
App Ports (P1,P2) (Serial modules)	RJ45 (DB-9F with supplied cable)
	RS-232, RS-422 and RS-485 modes supported.
	Both ports operate as individual IEC 60870-3-103 controlling units (masters)
	500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port
	6-foot RS-232 configuration Cable

5.1.3 Functional Specifications

The MVI69-103M module supports the IEC 60870-5-103 protocol to the following specifications:

- The IEC 60870-5-103 communication driver is built in accordance to the approved IEC specification
- User-definable module memory usage
- The module has two independent master ports, each configurable via a simple configuration file
- Supports up to 16 sessions (controlled devices) between the two ports
- Supports up to three sectors (separate databases) for each session, with individual database definition for each sector
- Total of 1000 user configurable commands to control data transfer to/from devices (controlled devices)
- Supports clock synchronization from/to the processor
- Event data received from the Control Devices updates the module database (Date and Time stamping is not stored or used by module)
- Class 1 and Class 2 delay parameters are configurable for each session
- An IEC Interoperability Document for the module is available from the web site, which fully documents data types supported by the module

5.2 Functional Overview

This section provides an overview of how the MVI69-103M module transfers data using the 103M protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

The standards used to build the module are listed in the following table.

Publication	Title
IEC 60870-5-103	Companion Standard for the informative interface of protection equipment.
IEC 60870-5-103 Annex A	Generic functions --Examples of constructing a directory
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

These documents should be obtained, reviewed, and understood in order to fully appreciate the protocol implementation. Most of the complexity of the protocol is hidden from the user and simplified in the application of the module. Detailed questions of about the protocol can be answered by reading these documents. In addition to calling our technical support group, there is also help available for the protocol using the following mail list Web Site: www.TriangleMicroWorks.com/iec870-5 (http://www.trianglemicroworks.com/iec870-5). Go to this site to join the mail list and to review questions and answers from mail list users.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI69-103M module.

Module Power Up

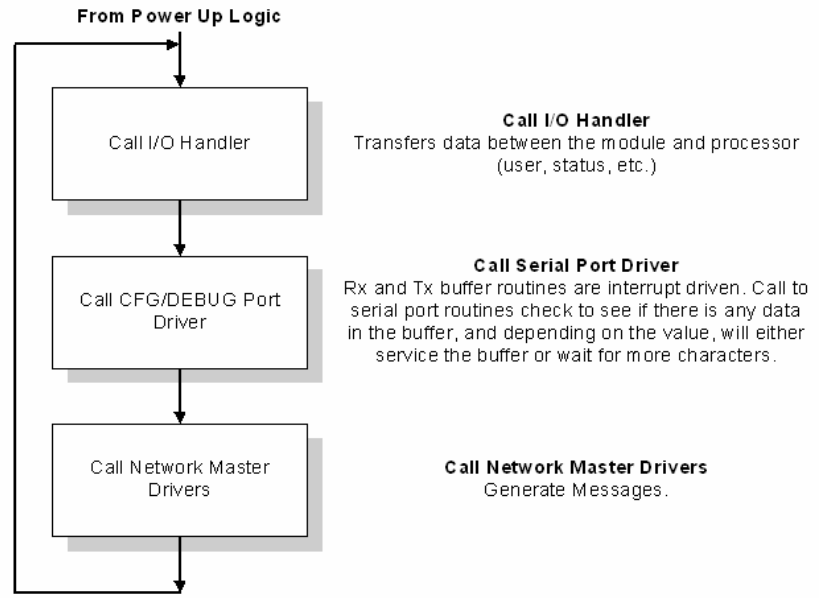
On power up the module begins performing the following logical functions:

1. Initialize hardware components
   - Initialize CompactLogix or MicroLogix backplane driver
   - Test and clear all RAM
   - Initialize the serial communication ports
2. Read configuration for module from IEC103M.CFG file on Compact Flash Disk
3. Initialize the databases and ports
4. Set up the serial communication interface for the debug/configuration port

After the module has received the configuration, the module will begin receiving and transmitting messages with devices on the serial networks.

Main Logic Loop

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



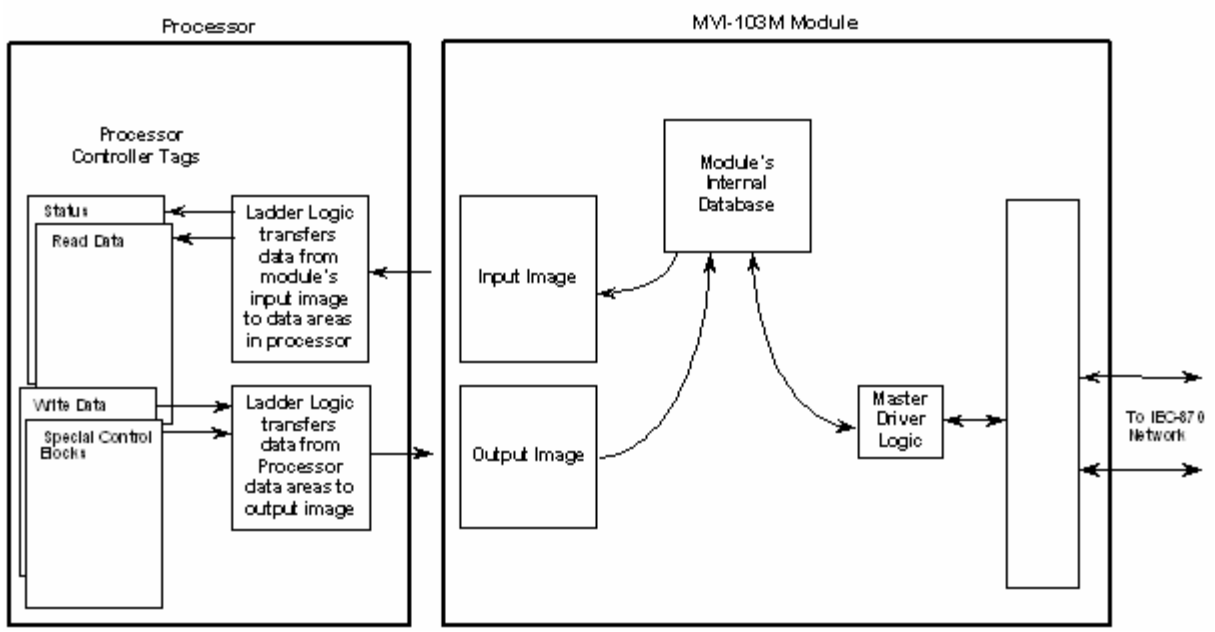
Backplane Data Transfer

The MVI69-103M module communicates directly over the CompactLogix or MicroLogix backplane. Data travels between the module and the CompactLogix or MicroLogix processor across the backplane using the module's input and output images. The update frequency of the data is determined by the scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 1 to 10 milliseconds.

Data received by the master drivers is placed in the module's input image. This data is processed by the ladder logic in the CompactLogix or MicroLogix processor.

The processor inserts data in the module's output image to transfer to the module. The module's program extracts the data and transmits the data out to the master driver to the serial network. Additionally, the CompactLogix or MicroLogix processor can send special control blocks to the module to instruct it to perform a special task.

The following illustration shows the data transfer method used to move data between the CompactLogix or MicroLogix processor, the MVI69-103M module, and the serial network.



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 CompactLogix or MicroLogix processor to interface the input and output image data defined in the controller tags. The user is responsible for handling and interpreting all data received on the application ports and transferred to the input image.

Data Types and Mapping

When interfacing data in the processor to that of the IEC 60870-5-103 protocol, it is important that the user understand the mapping of the data types to their corresponding representation in the modules database. The following table lists the data types supported by the module and their associated storage representation:

Type ID	Description	Data Representation
1	Time-tagged messages with each data point represented by two bits.	Dual-bit status (7.2.6.5 with 00b (0 decimal) = not used, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = not used
2	Time-tagged messages with relative time with each point represented by two bits.	Dual-bit status (7.2.6.5 with 00b (0 decimal) = not used, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = not used

Type ID	Description	Data Representation
3	Measurands with quality descriptor. The lower 3 bits of the values represented in this data type contain status information. The upper 13 bits of the value contained a signed, 12-bit number. This data type will return from 1 to 4 values. The number of words received is dependant on the information object number and the slave device.	Measurand with quality descriptor (7.2.6.8) Bit 0: 0 = No overflow, 1 = Overflow Bit 1: 0 = Valid, 1 = Invalid Bit 2: Reserved Bits 3 to 25: Value from -1..+1-2^{12}
4	Time-tagged measurands with relative time with the value in the packet represented by a single floating point number.*	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.20)
5	Identification data composed of 12 characters of data. Each point in defined of this data type should reserve 12 bytes (6-word addresses) in the database for the data received.	Byte data as defined in 7.2.6.2. First 8 bytes are characters 1 to 8 and last 4 bytes are manufacture bytes either decimal (0 to 255) or as ASCII characters.
9	Measurands with quality descriptor. The lower 3 bits of the values represented in this data type contain status information. The upper 13 bits of the value contained a signed, 12-bit number. This data type will return from 1 to 9 values (some slaves may return up to 16 values). The number of words received is dependant on the information object number and the slave device.	Measurand with quality descriptor (7.2.6.8) Bit 0: 0 = No overflow, 1 = Overflow Bit 1: 0 = Valid, 1 = Invalid Bit 2: Reserved Bits 3 to 25: Value from -1..+1-2^{12}
20	General command to control a dual-point object. Each command issued by the module uses the values of two adjacent bits in the database or an override value specified by the user command.	Dual-bit status (7.2.6.4 with 00b (0 decimal) = not used, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = not used

* The words should be swapped in the ladder logic.

As shown in the previous table, all bit types are addressed as bits in the modules database.

Addressing the Data Types

The following table shows an example of how to address the data types in the MVI69-103M:

Data Type	Address Type	Length	Example
1	Bit	2 bits	Address 160 refers to first and second bits from word 10.
2	Bit	2 bits	Address 160 refers to first and second bits from word 10.
3	Word	4 words	Address 50 refers to word 50, 51, 52, and 53.
4	Double-word	2 words	Address 40 refers to two consecutive words starting at word 80.
5	Bit	12 bits	Address 180 refers to 12 consecutive bits starting at the LSB of word 90.

Data Type	Address Type	Length	Example
9	Word	5 words	Address 100 refers to 9 consecutive words starting at word 100.
20	Bit	2 bits	Address 160 refers to the first and second bits from word 10.

Therefore, address 16000 represents bit zero in word 1000 of the module's database. Short floating-point, 32-bit strings and integrated total values each occupy a double-word space in the database. Therefore, short float database address of 100 represents the two words, 200 and 201, in the modules database. Identification objects are stored as byte values in the modules database. Identification object address 1000 is stored in the module's database in word addresses 500 to 505.

When setting the monitored data and commands, each point is defined by its ASDU type, function code, and information number. Valid function codes are listed in the following table.

Function Code	Symbol	Number
Distance Protection	T(z)	128
Over-current Protection	I>>	160
Transformer Differential Protection	rIt	176
Line Differential Protection	rIl	192
Global Function Type	GLB	255

Refer to Protocol Interoperability Documentation for a full listing of the protocol support offered by the module.

As blocks are transferred between the module and the processor, each block contains block identification codes that define the content or function of the block of data transferred. The block identification codes used by the module are displayed in the following table.

Block Range	Descriptions
-1	Null block
0	Null block
1 to 20	Read or write data
9901	User Constructed Command
9902	Command Control Block (Add command to Command List Queue)
9903	Event Messages from Master port
9950	Command List Error data
9970	Set PLC time using module's time
9971	Set module's time using CompactLogix or MicroLogix time
9998	Warm Boot Request from CompactLogix or MicroLogix (Block contains no data)
9999	Cold Boot Request from CompactLogix or MicroLogix (Block contains no data)

Blocks -1 and 0 transfer status data from the module to the processor and they contain no data when transferred from the processor to the module. Blocks 1 to 20 are utilized to transfer data stored or to be stored in the module's database. These data blocks send data from module to the processor (monitored data received from the devices on the serial network) and to send data from the processor to the module (control data to send to the end devices). Block identification codes 9901 to 9999 are used for special control blocks to control the module.

Normal Data Transfer

Normal data transfer includes the transferring of data received by or to be transmitted to the master drivers and the status data. These data are transferred through read (input image) and write (output image) blocks.

Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required. The following topics discuss the structure and function of each block.

Read Block

These blocks of data transfer information from the module to the CompactLogix or MicroLogix processor. The structure of the input image used to transfer this data is shown in the following table.

Offset	Description	Length
0	Read Block ID	1
1	Write Block ID	1
2 to (n+1)	Read Data	n

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Read Block ID is an index value used to determine the location of where the data will be placed in the CompactLogix or MicroLogix processor controller tag array of module read data. The number of data words per transfer depends on the configured Block Transfer Size parameter in the configuration file (possible values are 60, 120, or 240).

The Write Block ID associated with the block requests data from the CompactLogix or MicroLogix processor. Under normal, program operation, the module sequentially sends read blocks and requests write blocks. For example, if three read and two write blocks are used with the application, the sequence will be as follows:

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

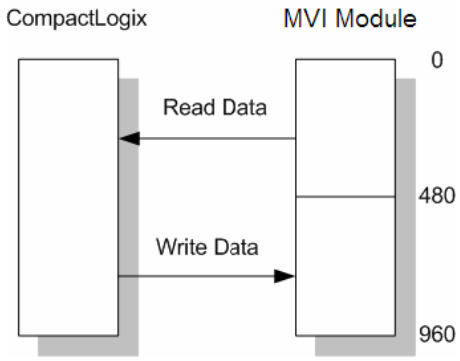
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the 103M network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

Assume that the backplane parameters are configured as follows:

```
Read Register Start:    0
Read Register Count:  480
Write Register Start: 480
Write Register Count: 480
```

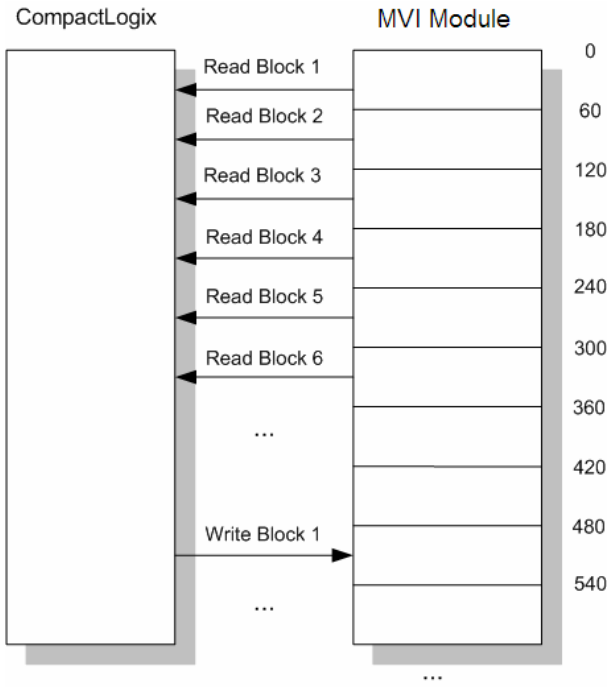
The backplane communication would be configured as follows:



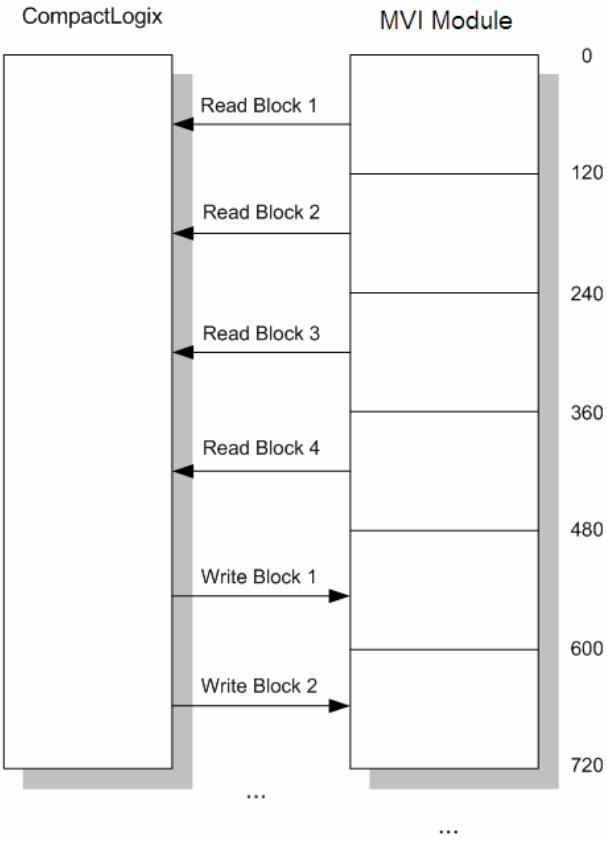
Database address 0 to 479 will be continuously transferred from the module to the processor. Database address 480 to 959 will continuously be transferred from the processor to the module.

The Block Transfer Size parameter basically configures how the Read Data and Write Data areas are broken down into data blocks (60, 120, or 240).

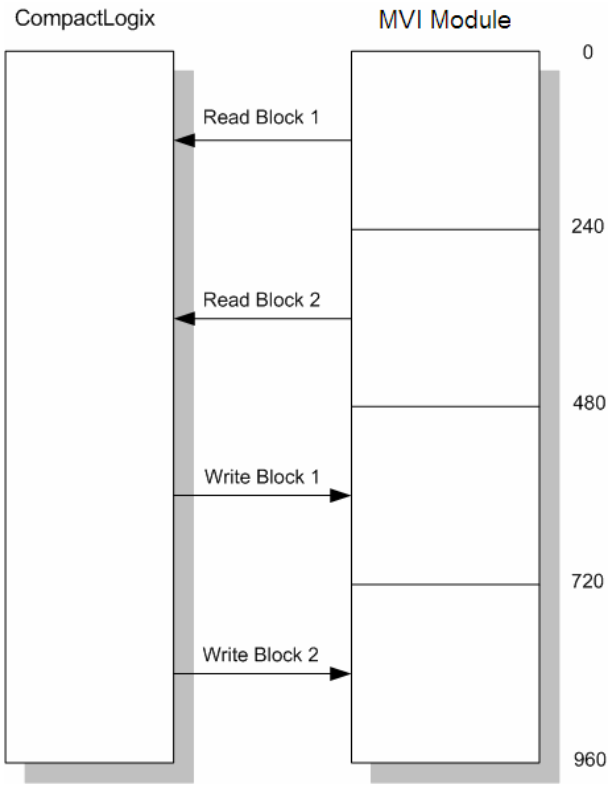
If Block Transfer Size = 60:



If Block Transfer Size = 120:



If Block Transfer Size = 240:



Write Block

These blocks of data transfer information from the CompactLogix or MicroLogix processor to the module. The structure of the output image used to transfer this data is shown in the following table.

Offset	Description	Length
0	Write Block ID	1
1 to n	Write Data	n

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed.

How Data is Transferred

In order to understand how the data is transferred between the processor and the module, you must understand the Read Data and Write Data area concept in the module's database. The module's database can be partially, or totally divided into Read Data Areas and Write Data Areas.

These areas are defined by the user when the configuration file is being edited. The following parameters define the Read and Write data areas:

Read Register Start = 0

Read Register Count = 120

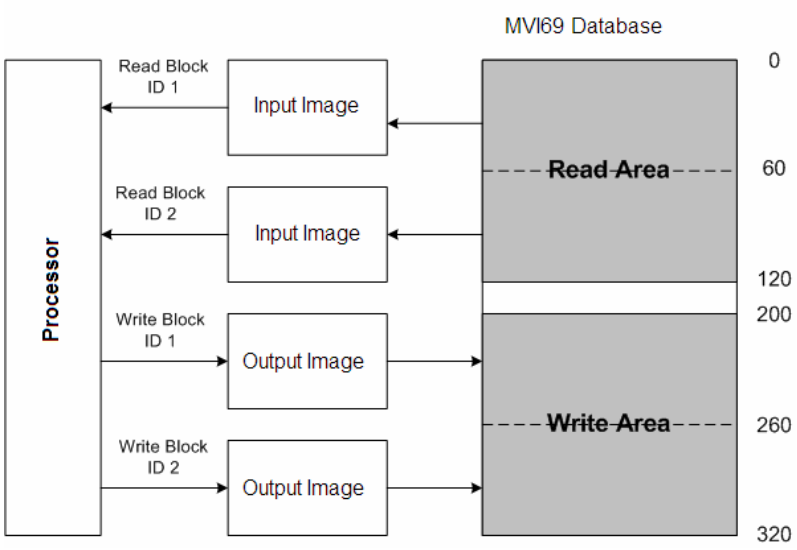
Write Register Start = 200

Write Register Count = 120

Each area is broken down into blocks of 60 words. Therefore, the Read Register Count and Write Register Count parameters should be multiples of 60.

The Read Data Area will be transferred from the module to the CompactLogix or MicroLogix processor. The Write Data Area will be transferred from the CompactLogix or MicroLogix processor to the module.

The following example shows the resulting data flow:



Command Control Blocks

Block identification codes greater than 9900 are utilized to perform special functions in the module. Each control block recognized and used by the module is defined in the following topics.

User Constructed Command Block (9901)

Block identification code 9901 issues one or more user constructed commands. When the module receives a block 9901 identification code, it will place the included commands into the command queue.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Command Count	This field defines the number of user commands contained in the block. The valid range for the field is 1 to 10.

Word Offset in Block	Data Field(s)	Description
2 to 11	Command #1	Data required to build the user defined command in the command queue.
12 to 21	Command #2	Data required to build the user defined command in the command queue.
22 to 31	Command #3	Data required to build the user defined command in the command queue.
32 to 41	Command #4	Data required to build the user defined command in the command queue.
42 to 51	Command #5	Data required to build the user defined command in the command queue.

The following fields are used for each 10-word record in the command list:

Word Offset	Definitions	Description
0	Database Index	Address in module to associate with the command.
1	Session Index	Session index defined in the module to associate with the command.
2	Sector Index	Sector index for session as defined in the module.
3	Data Type	ASDU data type associated with the command.
4	Function Code	Function Code for the command.
5	Point Index * * Information Number	Information object address for the point on which the command operates.
6	Override Flag	Override flag for general command.
7	Override Value	Override value for general command.
8	Reserved	Reserved for future use.
9	Reserved	Reserved for future use.

Refer to the command list section of this documentation for a detailed definition of the fields contained in this block. They are the same as those used in constructed the commands in the command list.

There is no response block built by the module to send back to the processor after the block is processed. The commands are placed in the command queue and issued at a high priority.

This block and block 9902 should be used when controlling double-point data points in remote units using general commands. This provides complete control of the slave devices under ladder logic control. Alternatively, the slaves can be controlled by changing data in the database and having the data be transferred using pre-constructed commands in the user's command list. Some points only accept value of on for control (that is, LED reset or activate characteristic). For these points, block 9901 and 9902 should only be utilized.

Command Control Block (9902)

The block 9902 identification code is used by the processor to send a list of commands to be placed in the command queue from the user configured command list. Commands placed in the queue with this method need not have their enable bit set in the command list.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9902 identifying the enable command to the module.
1	Command count	This field contains the number of commands to enable in the command list. Valid values for this field are 1 to 50.
2 to 61	Command Numbers to enable	These 50 words of data contain the command numbers in the command list to enable. The commands in the list will be placed in the command queue for immediate processing by the module. The first command in the list has an index of 0.
62 to n	Spare	Not Used

There is no response to this block by the module. The module will place the selected commands into the command queue. If the command references a unit that is not defined, the command will not be placed in the command queue. Normal processing of the command list will continue after the commands specified in this block are processed.

For digital output control, the use of block 9901 and 9902 is preferred to the use of the command list. The exact state of the output can be specified in the command list and then the command can be enabled through the use of block 9902. When the user wishes to execute this command (knowing the state of the command), can enable the command with the block 9902 request.

Event Message Block (9903)

Block identification code 9903 sends event messages received on the master port to the processor.

Note: Events are recognized when using a COT = Spontaneous.

Word Offset in Block	Data Field(s)	Description
0	Read Block ID	9903
1	Write Block ID	
2	Event Count	This field contains the number of events present in the block. Values of 1 to 20 are valid.
3 to 12	Event 1	Event message
13 to 22	Event 2	Event message
23 to 32	Event 3	Event message
33 to 42	Event 4	Event message
43 to 52	Event 5	Event message

The format of each 10-word data region in the block is as follows:

Word Offset	Definitions	Description
0	Session Index/Sector Index	This field contains the session and sector indices used to define the controlled unit in the module from which the event was generated. The MSB contains the session index and the LSB contains the sector index.
1	ASDU Type	This field contains the ASDU type code for the data contained in the message.
2	Function Code/Point Index*	This field contains the function code and the point index associated with the event message. The MSB contains the function code and the LSB contains the point index.

Word Offset	Definitions	Description
3	Fault Number	This is the fault number for the event if applicable. Only valid for ASDU types 2 and 4.
4	Sec/milliseconds	This word contains the seconds and millisecond values with a range of 0 to 59999 time at which the message was generated by the slave device.
5	Hr/Min.	This word contains the hour and minutes the message was generated by the slave. The MSB contains the hour and the LSB contains the minute value.
6	Invalid/DST	This word contains two bits that relate to the time value recorded in the slave device for the message. Bit 0 corresponds to the validity of the time (0=valid, 1=invalid) and Bit 1 defines if daylight savings time is used in the time (0=no, 1=yes).
7	Relative Time	This field contains the relative time value if applicable to the object. Only valid for ASDU types 2 and 4.
8 to 9	Value	This double-word value contains the value for the point index/function code in the event message. For ASDU types 1 and 2, this value is only 2 bits wide. For ASDU type 4, this double-word value contains the floating-point number (short circuit location).

In order for this feature to be activated, the event pass-through parameter must be set. When a master driver receives an event message from a controlled station, it will build an event message corresponding to the event in the event buffer of the module. This buffer is then sent to the processor when any messages are present. Therefore, these blocks are sent to the processor on a high priority. After the block is sent, the event message is removed from the module's event buffer.

If too many events are present in the buffer (>200), the module will set the event message overflow flag in the error/status data area of the normal read data block. There is no response block to be received by the module from the processor.

Command List Error Data Block (9950)

Block 9950 identification code requests the Command List Error Table from the module for the 1000 user configurable commands. The format for the block is shown below:

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the module.
1	Number of Commands to report	This field contains the number of commands to report in the response message. The value has a range of 1 to 200.
2	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands -1.
3 to n	Spare	Not Used

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The module will respond to a valid request with a block containing the requested error information. The format for the block is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Read Block ID	Read Block Identification
1	Write Block ID	Write Block Identification (9950)
2	Number of Commands reported	This field contains the number of commands contained in the block that must be processed by the PLC. This field will have a value of 1 to 200.
3	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 0 to MaxCommands-1.
4 to 53	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the block. The number of valid command errors in the block is set in word 2 of the block. Refer to the command error list to interpret the error codes reported.

Set CompactLogix or MicroLogix Time Block (9970)

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

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to n	Not Used	Not Used

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The module will respond to a valid block 9970 request with a block containing the requested date and time. The format for the block is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Read Block ID	Read Block Identification
1	Write Block ID	This is the next block requested by the module (9970).
2	Year	This field contains the four-digit year to be used with the new time value.
3	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.

Word Offset in Block	Data Field(s)	Description
8	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
9 to 53	Not Used	Not Used

Set Module Time Block (9971)

Block identification code 9971 passes the clock time in the CompactLogix or MicroLogix to the module. The date and time provided will be used to set the module's clock.

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

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The module does not send a response block to the processor after receiving this block.

Warm Boot Block (9998)

Block 9998 performs a warm-boot operation on the module. The format of the block constructed by the processor is as follows:

Offset	Description	Length
0	9998	1
1 to n	Spare	n

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

In this version of the module, the warm and cold boot processes perform the same operation as many of the variables that must be initialized are fixed when the module first boots and cannot be changed after the application starts.

Cold Boot Block (9999)

Block 9999 performs a cold-boot operation on the module. The format of the block constructed by the processor is as follows:

Offset	Description	Length
0	9999	1
1 to n	Spare	n

where

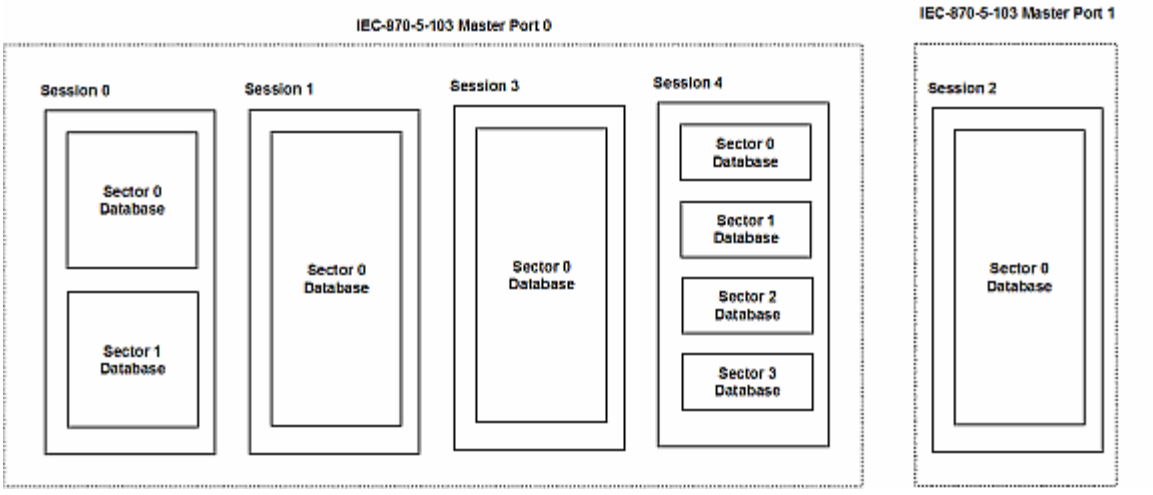
n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

In this version of the module, the warm and cold boot processes perform the same operation as many of the variables that must be initialized are fixed when the module first boots and cannot be changed after the application starts.

5.2.2 Master Driver

The master driver supported on each application port of the module emulates an IEC 60870-5-103 Master device. Configuration of each port is independent and should be connected to different serial networks.

Each port on the module communicates with one or more controlled stations on what are referred to as sessions. A session represents a controlled device with a unique data link layer address. Each session (controlled device) contains one or more data sets (sectors) that are defined by the vendor of the device. The following illustration shows these relationships.



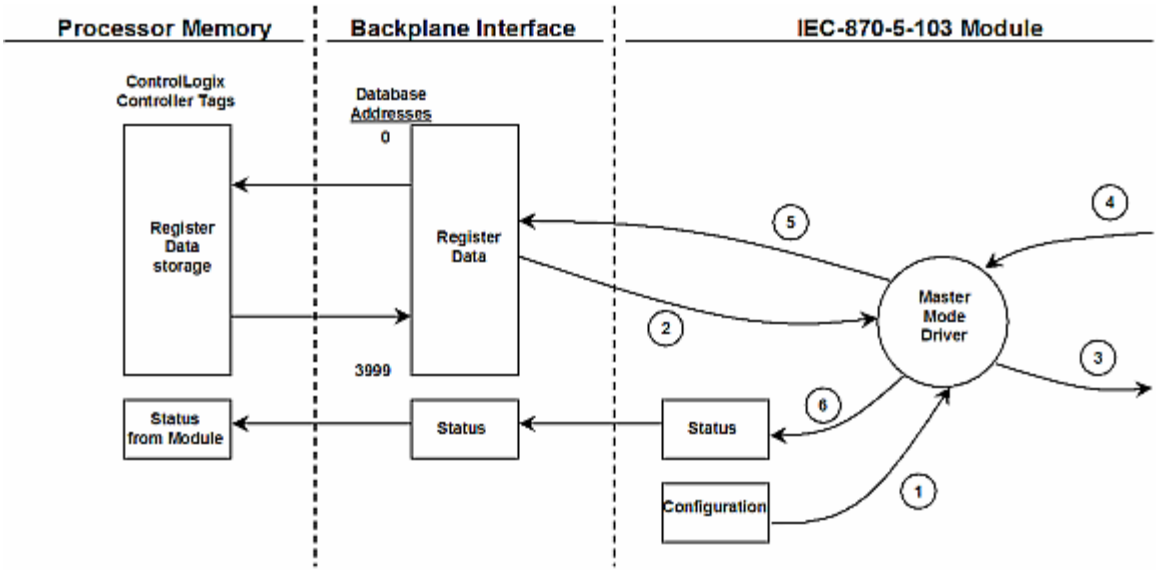
Port 0 on the module communicates with 4 sessions (0, 1, 3 and 4) each of which has their own data set(s). Session 1 only has one sector (all data for device contained in a single database). This sector is addressed by the master using the Common address of ASDU value set for the sector in the configuration file. Session 0 contains two sectors each with their own unique Common address of ASDU value to identify the sector.

Port 1 is connected to one device on the network. This device is defined in the Session 2 section of the configuration file. In this example, all data of the device is stored in a single sector.

Note: The IEC 60870-5-103 specification only supports the unbalanced mode. No support is given in the protocol for the balanced mode and the module does not support this mode.

The module supports two application ports. Thirty-two session can be defined on the module with each session being assigned to an application port. Within each session, up to five sectors can be defined. This system permits a very flexible assignment of resources in the module. The definition of the data associated with each sector in the system is defined by the user in the configuration file.

The following diagram shows the functionality of the master driver:



1 The master driver is configured as specified by the IEC103M.CFG file
2 The master will construct control commands using the data in the database
3 The master will send these commands and class polls out on the serial network
4 Response messages or spontaneous messages generated by controlled devices on the serial network are received by the master driver
5 Monitor data received by the master is passed to the module's database and passed to the processor
6 Additionally, status data for the module is passed to the processor

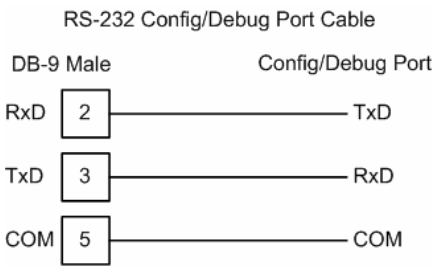
5.3 Cable Connections

The application ports on the MVI69-103M module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

5.3.1 RS-232 Configuration/Debug Port

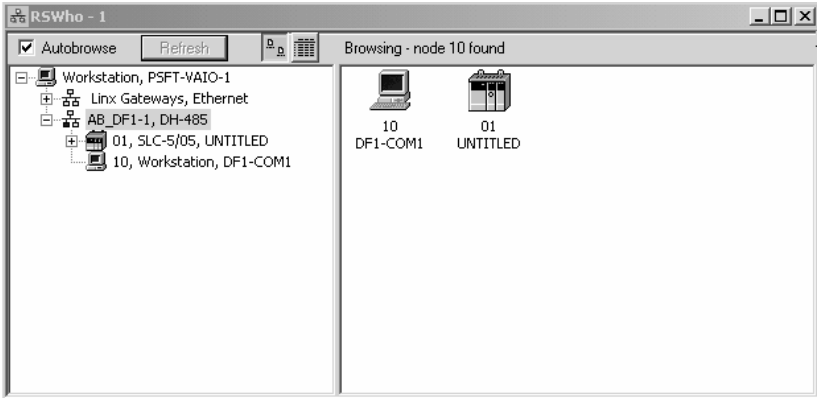
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following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



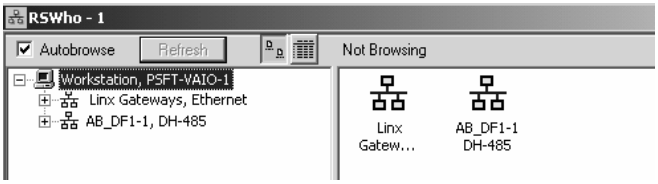
Disabling the RSLinx Driver for the Com Port on the PC

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

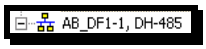
1. Open RSLinx and go to Communications>RSWho
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

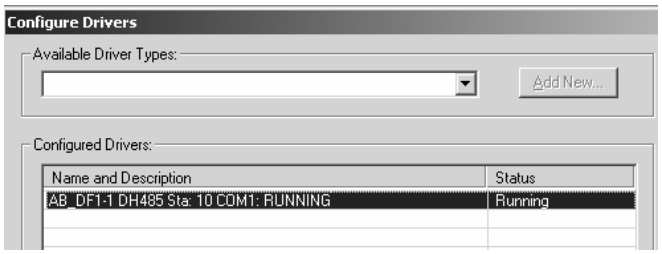


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.

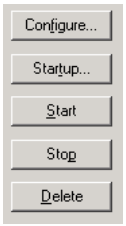


4 When you have verified that the driver is not being browsed, go to

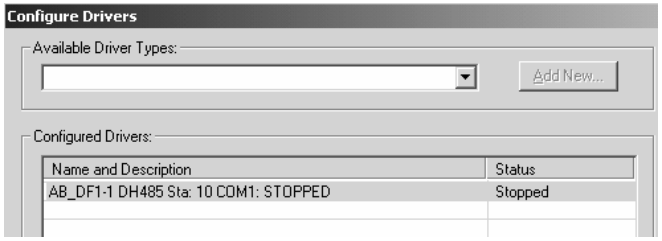
Communications>Configure Drivers
You may see something like this:



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:

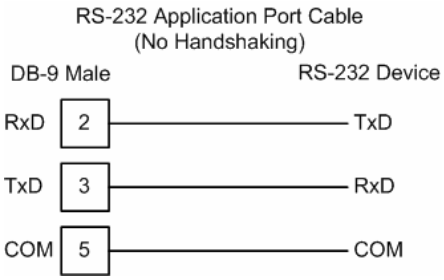


6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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.

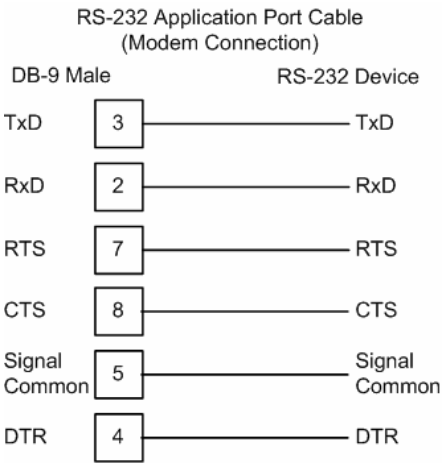
5.3.2 RS-232

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

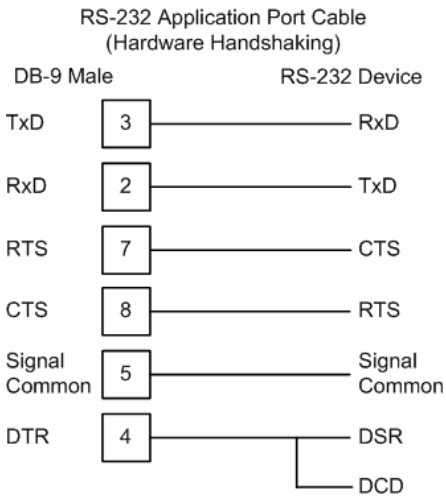
This type of connection is required between the module and a modem or other communication device.



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

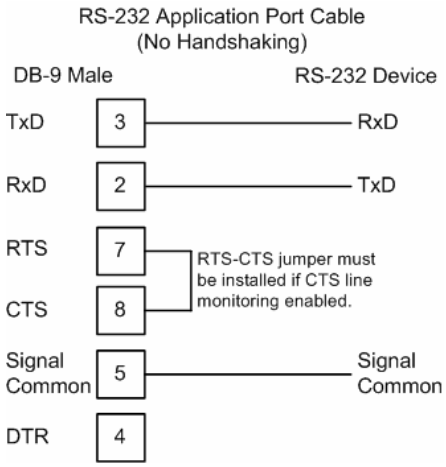
RS-232: Null Modem Connection (Hardware Handshaking)

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



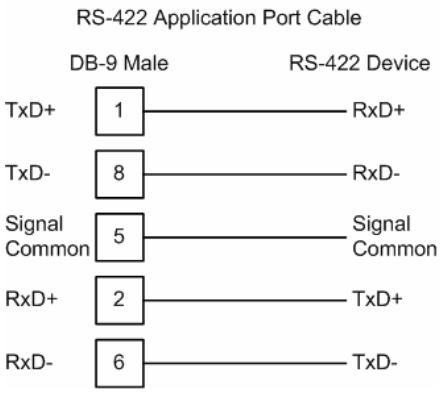
RS-232: Null Modem Connection (No Hardware Handshaking)

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



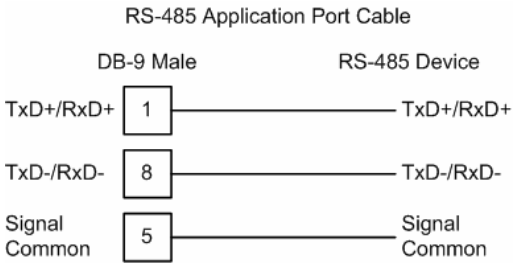
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

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:

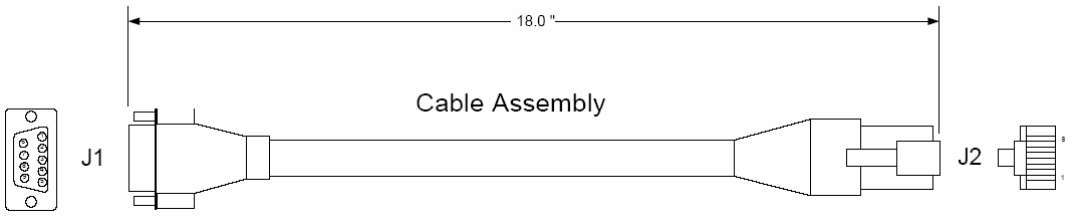


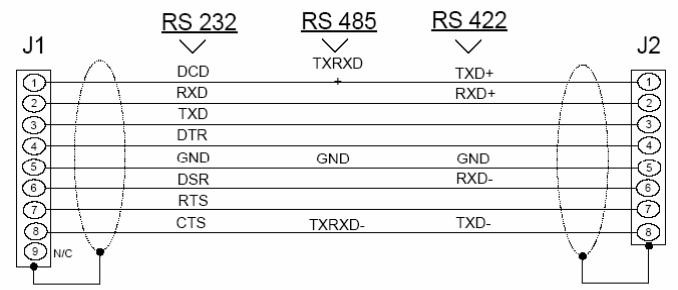
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

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 MVI69-103M Status Data Area

This section contains a listing of the data contained in the MVI69-103M status data object, configuration error word and module error codes.

5.4.1 MVI69-IEC 60870-5-103 Master Communication Module Error/Status Data Format

Offset	Parameter	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name. These words contain the text 'I3M5' for the MVI69 platform.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Event Msg Cnt	This word contains the number of event messages waiting to send to the processor.
14	Event Msg Overflow	This word contains a value of 0 if the event message buffer has not overflowed. If the event buffer overflows, this word will be set to a value of 1.

Offset	Parameter	Description
15	Session Count	This word contains the number of session configured in the module.
16	Current Cmd	This word contains the index of the current command being executed in the command list.
17	Cmd Busy Flag	This word is set to zero if no command is currently being executed and waiting on a response. If the word is set to 1, a command is currently executing.
18	Cmd Count	This word contains the count of the number of commands configured for the module.
19	Cmd Delay	This word contains the command delay counter preset. There is a fixed delay between each command to permit the module to perform class polls on controlled stations.
20	Cmd Queue	This word is set to zero if the command executing is from the command list. If the executing command is from the command queue, the word will be set to 1.
21	Cmd Queue Count	This word contains the number of active commands in the command queue for the module. Up to 100 commands can be buffered in this queue. These commands are transferred from the processor to the module using special command blocks.
22 to 23	Online Status	This double word value contains a bit for each of the 32 potential sessions in the module. If the bit is set for a session in the double word, the station is online. If the bit is clear, the station is offline. Use this value to determine if commands sent from the processor will have a chance of succeeding.
24	CH 0 State	This word contains the state machine value for channel 0.
25	Cmd Req	This word contains the number of commands transferred out channel 0.
26	Cmd Resp	This word contains the number of command response messages received on channel 0.
27	Cmd Err	This word contains the number of command errors recognized on channel 0.
28	Requests	This word contains the total number of messages transmitted on channel 0.
29	Responses	This word contains the total number of messages received on channel 0.
30	Err Sent	This word contains the number of error messages sent on channel 0.
31	Err Received	This word contains the number of error messages received on channel 0.
32	Cfg Err	This bit mapped word recognizes any configuration errors for channel 0. Refer to the configuration error word table for a definition of each bit.
33	Current Error	This word contains the error code for the current command executing on channel 0.
34	Last Error	This word contains the error code for the last error recognized on channel 0.
35	CH 1 State	This word contains the state machine value for channel 1.
36	Cmd Req	This word contains the number of commands transferred out channel 1.
37	Cmd Resp	This word contains the number of command response messages received on channel 1.
38	Cmd Err	This word contains the number of command errors recognized on channel 1.
39	Requests	This word contains the total number of messages transmitted on channel 1.

Offset	Parameter	Description
40	Responses	This word contains the total number of messages received on channel 1.
41	Err Sent	This word contains the number of error messages sent on channel 1.
42	Err Received	This word contains the number of error messages received on channel 1.
43	Cfg Err	This bit mapped word recognizes any configuration errors for channel 1. Refer to the configuration error word table for a definition of each bit.
44	Current Error	This word contains the error code for the current command executing on channel 1.
45	Last Error	This word contains the error code for the last error recognized on channel 1.
46	Block Transfer Size	Value configured by the user for block transfer size parameter.

The following table defines the contents of the configuration error word. Each bit in the word corresponds to an error condition recognized when the module is configured. There is a separate word for each application port. This data is reported in the status data area previously defined.

Bit	Code	Description
0	0x0001	Invalid baud rate selected
1	0x0002	Invalid parity selected
2	0x0004	Received timeout set to 0
3	0x0008	Invalid Port selected for a session
4	0x0010	Invalid sector count for session
5	0x0020	Could not allocate memory for sector of a session.
6	0x0040	
7	0x0080	Invalid failure delay or confirm timeout for session.
8	0x0100	
9	0x0200	
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

The following table lists all potential errors that can be generated by the IEC 60870-5-103 Master driver:

5.4.2 MVI69-IEC 60870-5-103 Master Communication Module Error Codes

Error	Description
51	Physical layer error - Error transmitting message
52	Physical layer error - Intercharacter timeout occurred before message fully received.
53	Physical layer error - Frame not entirely received before timeout condition.
54	Physical layer error - Invalid frame length.
101	Link layer error - Invalid checksum received
102	Link layer error - Address unknown to module
103	Link layer error - Link established

Error	Description
104	Link layer error - Link failed
105	Link layer error - Received primary
106	Link layer error - FCB error discard
107	Link layer error - FCB error repeat
108	Link layer error - Invalid start character received
109	Link layer error - Invalid second character received
110	Link layer error - Invalid ending character received
111	Link layer error - Length mismatch error
112	Link layer error - Illegal function
113	Link layer error - No confirmation received
114	Link layer error - No ACK received
115	Link layer error - Sequence unknown
116	Link layer error - Out of sequence
117	Link layer error - Remote close
118	Link layer error - Unexpected ACK
119	Link layer error - Request cancelled
201	Application layer error - Length mismatch
202	Application layer error - Address unknown
203	Application layer error - Response late
251	RBE error - Clock event buffer overflow
252	RBE error - Event buffer overflow
271	Data error - Address unknown
281	Control error - Illegal operation
282	Control error - Illegal value
283	Control error - Not selected
301	Initialization error - Database
302	Initialization error - Out of memory
401	Channel open error
501	Session error - Database
502	Session error - Configuration
601	No memory to receive message
602	Session not reserved
603	Illegal session
604	Session is reserved
605	Session is not available
701	No memory to transmit message
702	ASDU not supported
703	Duplicate request
704	Illegal sector
705	Control mode is illegal
801	Partial stop request
802	Stop request failed
901	Response timeout
902	Negative COT in response
903	Session is offline
904	Session is disabled
905	Select confirmation received, waiting to execute
906	Execute confirmation has not be received

5.5 Database Form

5.5.1 Form to Define Sector Database

Session Index #:

Sector Index #:

Data Type (1, 2, 3, 4, 5 or 9)	Database Address	Function Code	Point Index

Session Index #:

Sector Index #:

Data Type (1, 2, 3, 4, 5 or 9)	**Database Address**	**Function Code**	**Point Index**

5.6 Command List Form

5.6.1 Form to Define Command List

Enable Code	Database Index	Poll Interval	Session Index	Sector Index	Data Type (0,6,7,20)	Function Code	Point Index	Override Flag	Override Value

Enable Code	Database Index	Poll Interval	Session Index	Sector Index	Data Type (0,6,7,20)	Function Code	Point Index	Override Flag	Override Value

5.7 Protocol Support

This section contains a listing of the IEC 60870-5-103 protocol support provided by the module.

Note: Shaded areas are not supported by the module.

5.7.1 List of Type Identification Codes

In Monitor Direction

Type	Description
1	time-tagged message
2	time-tagged message with relative time
3	measurands I
4	time-tagged measurands with relative time
5	identification
6	time synchronization
8	general interrogation termination
9	measurands II
10	generic data
11	generic identification
23	list of recorded disturbances
26	ready for transmission of disturbance data
27	ready for transmission of channel
28	ready for transmission of tags
29	transmission of tags
30	transmission of disturbance values
31	end of transmission

In Control Direction

Type	Description
6	time synchronization
7	general interrogation
10	generic data
20	general command
21	generic command
24	order for disturbance data transmission
25	ack for disturbance data transmission

5.7.2 List of Cause of Transmission Codes

In Monitor Direction

COT	Description
1	spontaneous
2	cyclic
3	reset frame count bit (FCB)
4	reset communication unit (CU)
5	start/restart
6	power on
7	test mode
8	time synchronization
9	general interrogation
10	termination of general interrogation
11	local operation
12	remote operation
20	positive ack of command
21	negative ack of command
31	transmission of disturbance data
40	positive ack of generic write command
41	negative ack of generic write command
42	valid data response to generic read command
43	invalid data response to generic read command
44	generic write confirmation

In Control Direction

COT	Description
8	time synchronization
9	initiation of general interrogation
20	general command
31	transmission of disturbance data
40	generic write command
42	generic read command

5.7.3 List of Function Types

Fun	Description	SYMBOL
128	Distance protection	t(z)
160	Overcurrent protection	I>>
176	Transformer differential protection	ΔI_T
192	Line differential protection	ΔI_L
254	Generic function type	GEN
255	Global function type	GLB

5.7.4 Information Numbers Used In Monitor Direction

System functions

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
0	end of general interrogation		8						X
0	time synchronization		6						X
2	reset FCB		5						*
3	reset CU		5						*
4	start/restart		5						*
5	power on		5						*
* - According to main function									

Status Indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
16	auto-recloser active	X	1	X	X		X		
17	teleprotection active	X	1	X	X				
18	protection active	X	1	X	X	X	X		
19	LED reset		1	X	X	X	X		
20	monitor direction blocked	X	1	X	X	X	X		
21	test mode	X	1	X	X	X	X		
22	local parameter setting	X	1	X	X	X	X		
23	characteristic 1	X	1	X					
24	characteristic 2	X	1	X					
25	characteristic 3	X	1	X					
26	characteristic 4	X	1	X					
27	auxiliary input 1	X	1	X	X	X	X		
28	auxiliary input 2	X	1	X	X	X	X		

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
29	auxiliary input 3	X	1	X	X	X	X		
30	auxiliary input 4	X	1	X	X	X	X		

Supervision indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
32	measured supervision I	X	1	X	X				
33	measured supervision V	X	1	X	X				
35	phase sequence supervision	X	1	X	X				
36	trip circuit supervision	X	1	X	X	X	X		
37	I>>back-up operation	X	1	X					
38	VT fuse failure	X	1	X	X				
39	teleprotection disturbed	X	1	X	X		X		
46	group warning	X	1	X	X	X	X		
47	group alarm	X	1	X	X	X	X		

Earth fault indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
48	earth fault L_1	X	1	X	X				
49	earth fault L_2	X	1	X	X				
50	earth fault L_3	X	1	X	X				
51	earth fault forward (that is, line)	X	1	X	X				
52	earth fault reverse (that is, busbar)	X	1	X	X				

Fault indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
64	start/pick-up L_1	X	2	X	X		X		
65	start/pick-up L_2	X	2	X	X		X		
66	start/pick-up L_3	X	2	X	X		X		
67	start/pick-up N	X	2	X	X		X		
68	general trip		2	X	X	X	X		
69	trip L_1		2	X	X	X	X		

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
70	trip L_2		2	X	X	X	X		
71	trip L_3		2	X	X	X	X		
72	trip I>> (back-up operation)		4	X					
73	fault location X in ohms		2	X	X				
74	fault forward/line		2	X	X				
75	fault reverse/busbar		2	X	X				
76	teleprotection signal transmitted		2	X	X				
77	teleprotection signal received		2	X	X				
78	zone 1		2	X					
79	zone 2		2	X					
80	zone 3		2	X					
81	zone 4		2	X					
82	zone 5		2	X					
83	zone 6		2	X					
84	general start/pick-up	X	2	X	X	X	X		
85	breaker failure		2	X	X				
86	trip measuring system L_1		2			X			
87	trip measuring system L_2		2			X			
88	trip measuring system L_3		2			X			
89	trip measuring system E		2			X			
90	trip I>		2		X				
91	trip I>>		2		X				
92	trip IN>		2		X				
93	trip IN>>		2		X				

Auto-reclosure indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
128	CB 'on' by AR		1	X	X		X		
129	CB 'on' by long-time AR		1	X	X		X		
130	AR blocked	X	1	X	X		X		

Measurands

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
144	measurand I		3.1	X	X				
145	measurands I, V		3.2	X	X				

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
146	measurands I, V, P, Q		3.3	X					
147	measurands I_N, V_{EN}		3.4	X	X				
148	measurands $I_{L1,2,3}$, $V_{L1,2,3}$, P, Q, f		9	X					

Generic functions

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
240	read headings of all defined groups		10					X	
241	read values or attributes of all entries of one group		10					X	
243	read directory of a single entry		11					X	
244	read value or attribute of a single entry	(x)	10					X	
245	end of general interrogation of generic data		10					X	
249	write entry with confirmation		10					X	
250	write entry with execution		10					X	
251	write entry aborted		10					X	

5.7.5 Information Numbers Used In Control Direction

System functions

Inf	Description	Type	Fun					
			128	160	176	192	254	255
			t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
0	initiation of general interrogation	7						X
0	time synchronization	6						X

General commands

Inf	Description	Com	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
16	auto-recloser on/off	On/Off	20	X	X		X		
17	teleprotection on/off	On/Off	20	X	X				
18	protection on/off	On/Off	20	X	X	X	X		
19	LED reset	On	20	X	X	X	X		
23	activate characteristic 1	On	20	X					

Inf	Description	Com	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
24	activate characteristic 2	On	20	X					
25	activate characteristic 3	On	20	X					
26	activate characteristic 4	On	20	X					

Generic functions

Inf	Description	Type	Fun					
			128	160	176	192	254	255
			t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
240	read headings of all defined groups	21					X	
241	read values or attributes of all entries of one group	21					X	
243	read directory of a single entry	21					X	
244	read value or attribute of a single entry	21					X	
245	general interrogation of generic data	21					X	
248	write entry	10					X	
249	write entry with confirmation	10					X	
250	write entry with execution	10					X	
251	write entry abort	10					X	

5.7.6 Definition and Presentation of ASDUs In Monitor Direction

Type 1: Time-tagged

Bytes	Description	Spec
1	Value of 1	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	DPI (0 to 3)	7.2.6.5
4	4-octet binary time	7.2.6.28
1	SIN	7.2.6.23

Type 2: Time-tagged with relative time

Bytes	Description	Spec
1	Value of 2	
1	0x81	

Bytes	Description	Spec
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	DPI (0 to 3)	7.2.6.5
2	RET (relative time)	7.2.6.15
2	FAN (fault number)	7.2.6.6
4	4-octet binary time	7.2.6.28
1	SIN	7.2.6.23

Type 3: Measurands I

Bytes	Description	Spec
1	Value of 3	
1	i = value of 1, 2 or 4	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
2	Current L_2	7.2.6.8
2	Voltage $L_{1\ to}\ L_2$	7.2.6.8
2	Active power P	7.2.6.8
2	Reactive power Q	7.2.6.8
	ASD U3.1: i=1	
	ASDU 3.2: i=2	
	ASDU 3.3: i=4	
	ASDU 3.4: i=2 val1=I_N and val2=V_{EN}	

Type 4: Time-tagged measurands with relative time

Bytes	Description	Spec
1	Value of 4	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
4	SCL (short-circuit location (real))	7.2.6.20
2	RET (relative time)	7.2.6.15
2	FAN (fault number)	7.2.6.6
4	4-octet binary time	7.2.6.28

Type 5: Identification

Bytes	Description	Spec
1	Value of 5	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	COL (compatibility level 2 or 3)	7.2.6.3
1	Char 1	7.2.6.2
1	Char 2	7.2.6.2
1	Char 3	7.2.6.2
1	Char 4	7.2.6.2
1	Char 5	7.2.6.2
1	Char 6	7.2.6.2
1	Char 7	7.2.6.2
1	Char 8	7.2.6.2
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment

Type 6: Time synchronization

Bytes	Description	Spec
1	Value of 6	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
7	7-Octet binary time	7.2.6.29

Type 8: Termination of general interrogation

Bytes	Description	Spec
1	Value of 8	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	SCN (scan number 0 to 255)	7.2.6.21

Type 9: Measurands II

Bytes	Description	Spec
1	Value of 9	
1	I = 1 to 9 for number of values	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
2	Current L_1	7.2.6.8
2	Current L_2	7.2.6.8
2	Current L_3	7.2.6.8
2	Voltage $L_{1 \text{ to } E}$	7.2.6.8
2	Voltage $L_{2\text{-}E}$	7.2.6.8
2	Voltage $L_{3\text{-}E}$	7.2.6.8
2	Active power P	7.2.6.8
2	Reactive power Q	7.2.6.8
2	Frequency f	7.2.6.8

Value of I determines number of parameters passed in the message with a range of 1 to 9 but always starts with the L_1 value.

5.7.7 Definition and Presentation Of ASDUs In Control Direction

Type 6: Time synchronization

Bytes	Description	Spec
1	Value of 6	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type = GLB (255)	
1	Information number	
7	7-Octet binary time	7.2.6.29

Type 7: Initiation of general interrogation

Bytes	Description	Spec
1	Value of 7	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type = GLB (255)	
1	Information number	
1	SCN (scan number 0 to 255)	7.2.6.21

Type 20: General command

Bytes	Description	Spec
1	Value of 20	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	DCO (1=Off, 2=On)	7.2.6.4
1	RII	7.2.6.19

RII is not to be processed within the protection equipment, but to be used as a SIN of the return message.

5.8 Protocol Interoperability Documentation

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

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

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

☐ Function or ASDU is not used

☒ Function or ASDU is used as standardized (default)

R Function or ASDU is used in reverse mode

B Function or ASDU is used in standard and reverse mode

The possible selection (blank, X, R, B) is specified for each specific clause or parameter.

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

The pages in this section have been extracted from the 60870-5-103 © IEC:1997, pages 159 to 171.

INTEROPERABILITY

5.8.1 Physical Layer

5.8.2 Electrical Interface

☒ EIA RS-485

☒ Number of loads 32. For one protection equipment

NOTE - EIA RS-485 standard defines unit loads so that 32 of them can be operated on one line. For detailed information refer to clause 3 of EIA RS-485 standard.

5.8.3 Optical Interface

☐ Glass fiber

☐ Plastic fiber

☐ F-SMA type connector

☐ BFOC/2,5 type connector

5.8.4 Transmission speed

☒ 9 600 bit/s

☒ 19 200 bit/s

5.8.5 Link Layer

There are no choices for the link layer.

5.8.6 Application Layer

5.8.7 Transmission mode for application data

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

5.8.8 Common Address of ASDU

☒ One Common Address of ASDU (identical with station address)

☒ More than one Common Address of ASDU

5.8.9 Selection of standard information numbers in monitor direction

5.8.10 System functions in monitor direction

	INF	Semantics
☒	<0>	End of general interrogation
☒	<0>	Time synchronization
☒	<2>	Reset FCB
☒	<3>	Reset CU
☒	<4>	Start/restart
☒	<5>	Power on

5.8.11 Status indications in monitor direction

	INF	Semantics
☒	<16>	Auto-recloser active
☒	<17>	Teleprotection active
☒	<18>	Protection active
☒	<19>	LED reset
☒	<20>	Monitor direction blocked
☒	<21>	Test mode
☒	<22>	Local parameter setting
☒	<23>	Characteristic 1
☒	<24>	Characteristic 2
☒	<25>	Characteristic 3
☒	<26>	Characteristic 4
☒	<27>	Auxiliary input 1
☒	<28>	Auxiliary input 2
☒	<29>	Auxiliary input 3
☒	<30>	Auxiliary input 4

5.8.12 Supervision indications in monitor direction

	INF	Semantics
☒	<32>	Measurand supervision I
☒	<33>	Measurand supervision V
☒	<35>	Phase sequence supervision

☒	<36>	Trip circuit supervision
☒	<37>	I>> back-up operation
☒	<38>	VT fuse failure
☒	<39>	Teleprotection disturbed
☒	<46>	Group warning
☒	<47>	Group alarm

5.8.13 Earth fault indications in monitor direction

	INF	Semantics
☒	<48>	Earth fault L_1
☒	<49>	Earth fault L_2
☒	<50>	Earth fault L_3
☒	<51>	Earth fault forward, that is, line
☒	<52>	Earth fault reverse, that is, busbar

5.8.14 Fault indications in monitor direction

	INF	Semantics
☒	<64>	Start /pick-up L_1
☒	<65>	Start /pick-up L_2
☒	<66>	Start /pick-up L_3
☒	<67>	Start /pick-up N
☒	<68>	General trip
☒	<69>	Trip L_1
☒	<70>	Trip L_2
☒	<71>	Trip L_3
☒	<72>	Trip I>> (back-up operation)
☒	<73>	Fault location X in ohms
☒	<74>	Fault forward/line
☒	<75>	Fault reverse/busbar
☒	<76>	Teleprotection signal transmitted
☒	<77>	Teleprotection signal received
☒	<78>	Zone 1
☒	<79>	Zone 2
☒	<80>	Zone 3
☒	<81>	Zone 4

☒ <82> Zone 5
☒ <83> Zone 6
☒ <84> General start/pick-up
☒ <85> Breaker failure
☒ <86> Trip measuring system L_1
☒ <87> Trip measuring system L_2
☒ <88> Trip measuring system L_3
☒ <89> Trip measuring system E
☒ <90> Trip I>
☒ <91> Trip I>>
☒ <92> Trip IN>
☒ <93> Trip IN>>

5.8.15 Auto-reclosure indications in monitor direction

	INF	Semantics
☒	<128>	CB 'on' by AR
☒	<129>	CB 'on' by long-time AR
☒	<130>	AR blocked

5.8.16 Measurands in monitor direction

	INF	Semantics
☒	<144>	Measurand I
☒	<145>	Measurands I, V
☒	<146>	Measurands I, V, P, Q
☒	<147>	Measurands I_N, V_{EN}
☒	<148>	Measurands $I_{L1,2,3}$, $V_{L1,2,3}$, P, Q, f

5.8.17 Generic functions in monitor direction

	INF	Semantics
☐	<240>	Read headings of all defined groups
☐	<241>	Read values or attributes of all entries of one group
☐	<243>	Read directory of a single entry
☐	<244>	Read value or attribute of a single entry
☐	<245>	End of general interrogation of generic data
☐	<249>	Write entry with confirmation

☐ <250> Write entry with execution

☐ <251> Write entry aborted

5.8.18 Selection of standard information numbers in control direction

5.8.19 System functions in control direction

	INF	Semantics
☒	<0>	Initiation of general interrogation
☒	<0>	Time synchronization

5.8.20 General commands in control direction

	INF	Semantics
☒	<16>	Auto-recloser on/off
☒	<17>	Teleprotection on/off
☒	<18>	Protection on/off
☒	<19>	LED reset
☒	<23>	Activate characteristic 1
☒	<24>	Activate characteristic 2
☒	<25>	Activate characteristic 3
☒	<26>	Activate characteristic 4

5.8.21 Generic functions in control direction

	INF	Semantics
☐	<240>	Read headings of all defined groups
☐	<241>	Read values or attributes of all entries in one group
☐	<243>	Read directory of a single entry
☐	<244>	Read value or attribute of a single entry
☐	<245>	General interrogation of generic data
☐	<248>	Write entry
☐	<249>	Write entry with confirmation
☐	<250>	Write entry with execution
☐	<251>	Write entry abort

5.8.22 Basic application functions

☒ Test mode

☒ Blocking of monitor direction

☐ Disturbance data

☐ Generic services

☒ Private data *(if ASDU type is supported by module)*

5.8.23 Miscellaneous

Measurands are transmitted with ASDU 3 as well as with ASDU 9. As defined in 7.2.6.8, the maximum MVAL can either be 1,2 or 2,4 times the rated value. No different rating shall be used in ASDU 3 and ASDU 9, that is, for each measurand there is only one choice.

Measurand	**Max. MVAL = rated value times**		
	1,2	**or**	**2,4**
Current L_1	☒		☒
Current L_2	☒		☒
Current L_3	☒		☒
Voltage L_{1-E}	☒		☒
Voltage L_{2-E}	☒		☒
Voltage L_{3-E}	☒		☒
Active power P		☒	☒
Reactive power Q		☒	☒
Frequency f	☒		☒
Voltage L_1 - L_2	☒		☒

6 Support, Service & Warranty

In This Chapter

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

1 Product Version Number
2 System architecture
3 Network details

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

1 Module configuration and contents of file
   - Module Operation
   - Configuration/Debug status information
   - LED patterns

2 Information about the processor and user data files as viewed through and LED patterns on the processor.
3 Details about the serial devices interfaced, if any.

6.1 How to Contact Us: Technical Support

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

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com (mailto:support.asia@prosoft-technology.com)

Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com (mailto:support.emea@prosoft-technology.com)

Languages spoken include: French, English

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

+1.661.716.5100, support@prosoft-technology.com (mailto:support@prosoft-technology.com)

Languages spoken include: English, Spanish

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

Brasil (location in Sao Paulo)

+55-11-5084-5178 , eduardo@prosoft-technology.com (mailto:eduardo@prosoft-technology.com)

Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns:

a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 121). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, etc.

6.2.2 Procedures for Return of Units Under Warranty:

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

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

6.2.3 Procedures for Return of Units Out of Warranty:

a) Customer sends unit in for evaluation
b) If no defect is found, Customer will be charged the equivalent of $100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.
c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

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

6.2.4 Purchasing Warranty Extension:

a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 124)". The Warranty Period may be extended at the time of equipment purchase 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

6.3 LIMITED WARRANTY

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

6.3.1 What Is Covered By This Warranty

a) *Warranty On New Products*: ProSoft warrants, to the original purchaser, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires three years from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 39 months. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft.
b) *Warranty On Services*: Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranteed in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.3.2 What Is Not Covered By This Warranty

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

b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.
c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

Product manufactured or supplied by ProSoft is not fault tolerant and is not designed, manufactured or intended for use in hazardous environments requiring fail-safe performance including and without limitation: the operation of nuclear facilities, aircraft navigation 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.

6.3.4 Intellectual Property Indemnity

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

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

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

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 124) 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.

*6.3.6 Limitation of Remedies ***

In no event will ProSoft or its Dealer be liable for any special, incidental or consequential damages based on breach of warranty, breach of contract, negligence, strict tort or any other legal theory. Damages that ProSoft or its Dealer will not be responsible for included, but are not limited to: Loss of profits; loss of savings or revenue; loss of use of the product or any associated equipment; loss of data; cost of capital; cost of any substitute equipment, facilities, or services; downtime; the claims of third parties including, customers of the Purchaser; and, injury to property.

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

6.3.7 Time Limit for Bringing Suit

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

6.3.8 No Other Warranties

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

6.3.9 Allocation of Risks

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

6.3.10 Controlling Law and Severability

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

Index

S

T

U

V

W

Y