ProSoft TECHNOLOGY

Where Automation Connects.

MVI56-101M

ControlLogix Platform

IEC 60870-5-101 Master Communication Module

August 5, 2021

USER MANUAL

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 our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology, Inc.
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

MVI56-101M User Manual

August 5, 2021

ProSoft Technology® Product Documentation

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

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.
North America: +1.661.716.5100
Asia Pacific: +603.7724.2080
Europe, Middle East, Africa: +33 (0) 5.3436.87.20
Latin America: +1.281.298.9109

For professional users in the European Union
If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Agency Approvals & Certifications

Please visit our website: www.prosoft-technology.com

Important Installation Instructions

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

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

MVI (Multi Vendor Interface) Modules

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

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

Warnings

North America Warnings

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

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
C Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

Avertissement - Risque d'explosion - Avant de déconnecter l'équipement, couper le courant ou s'assurer que l'emplacement est désigné non dangereux.

D Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Battery Life Advisory

The MVI46, MVI56, MVI56E, MVI69, and MVI71 modules use a rechargeable Lithium Vanadium Pentoxide battery to backup the 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 the battery becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and the real-time clock for approximately 21 days. When the battery is fully discharged, the module will revert to the default BIOS and clock settings.

Note: The battery is not user replaceable.

Markings

Electrical Ratings

- Backplane Current Load: 800 mA @ 5.1 Vdc; 3 mA @ 24 Vdc
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Shock: 30 g, operational; 50 g, non-operational; Vibration: 5 g from 10 Hz to 150 Hz
- Relative Humidity: 5% to 95% with no condensation
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Label Markings

ATEX

II 3 G

EEx nA IIC T6

0°C <= Ta <= 60°C

cULus

E183151

Class I Div 2 Groups A,B,C,D

T6

-30°C <= Ta <= 60°C

Contents

Guide to the MVI56-101M User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 13)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 63)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 85) Product Specifications (page 86) Functional Overview	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 123) Index	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

To get the most benefit from this User Manual, you should have the following skills:

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect 101 and ControlLogix devices to a power source and to the MVI56-101M module's application port(s)

1.1 System Requirements

The MVI56-101M module requires the following minimum hardware and software components:

- Rockwell Automation ControlLogix™ processor, with compatible power supply and one free slot in the rack, for the MVI56-101M module. The module requires 800 mA of available power.
- Rockwell Automation RSLogix 5000 programming software version 2.51 or higher
- 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)
- ProSoft Configuration Builder, HyperTerminal or other terminal emulator program.

Note: You can install the module in a local or remote rack. For remote rack installation, the module requires EtherNet/IP or ControlNet communication with the processor.

1.2 Package Contents

The following components are included with your MVI56-101M 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	MVI56-101M Module	MVI56-101M	IEC 60870-5-101 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

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

1.3 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the module. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

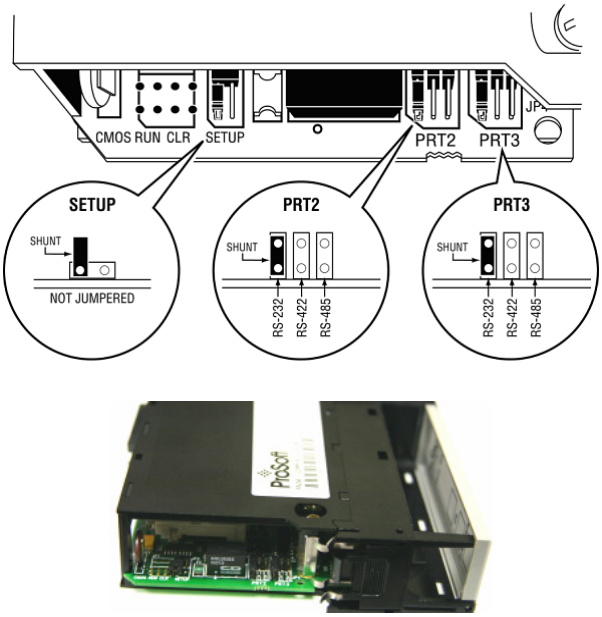
Installing ProSoft Configuration Builder from the ProSoft website

1. Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
2. Click the **DOWNLOAD HERE** link to download the latest version of *ProSoft Configuration Builder*.
3. Choose **SAVE** or **SAVE FILE** when prompted.
4. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

1.4 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to match the interface. There are three jumpers located at the bottom of the module.

The following illustration shows the MVI56-101M jumper configuration:



1. Set the PRT 2 (for application port 1) and PRT 3 (for application port 2) jumpers for RS232, RS422, or RS485 to match the wiring needed for your application. The default jumper setting for both application ports is RS-232.
2. 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.

1.5 Installing the Module in the Rack

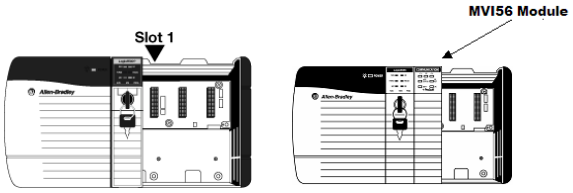
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-101M module. Refer to your Rockwell Automation product documentation for installation instructions.

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

After you have checked the placement of the jumpers, insert MVI56-101M into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

Warning: When you insert or remove the module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Verify that power is removed or the area is non-hazardous before proceeding. Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

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



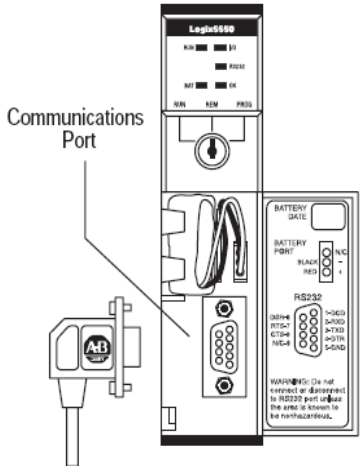
3. With a firm but steady push, snap the module into place.
4. Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5. Make a note of the slot location. You must identify the slot in which the module is installed in order for the sample program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

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

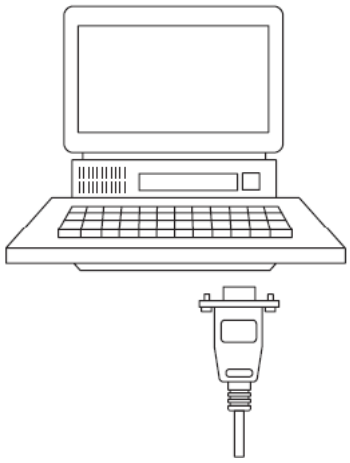
1.6 Connecting Your PC to the ControlLogix Processor

There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface. It is not mandatory that you use the processor's serial interface. You may access the processor through whatever network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

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.



1.7 Opening the Sample Ladder Logic

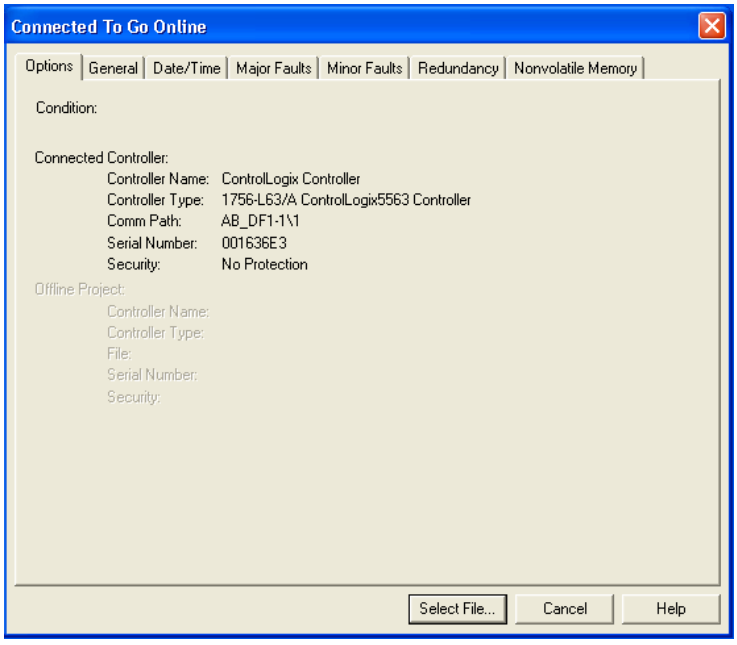
The sample program for your MVI56-101M module includes custom tags, data types and ladder logic for data I/O and status monitoring. For most applications, you can run the sample ladder program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

The version number appended to the file name corresponds with the firmware version number of your ControlLogix processor. The firmware version and sample program version must match.

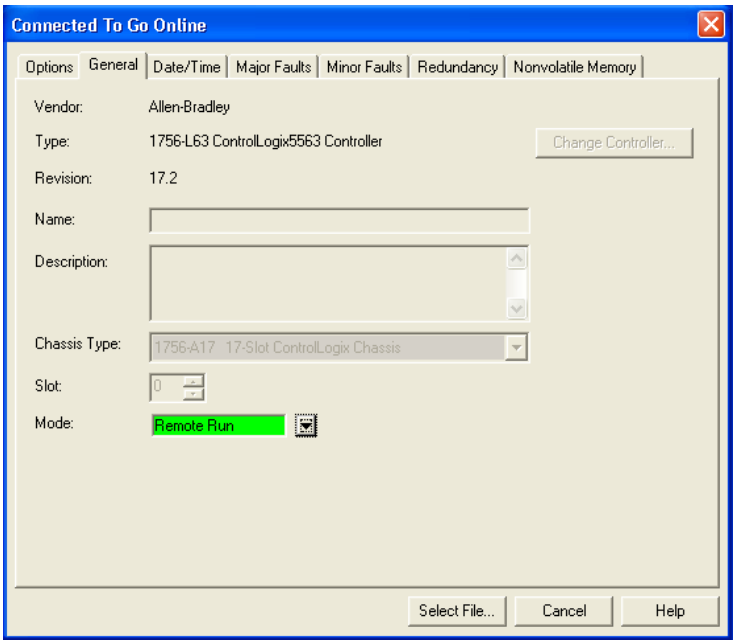
1.7.1 Determining the Firmware Version of Your Processor

Important: The RSLinx service must be installed and running on your computer in order for RSLogix to communicate with the processor. Refer to your RSLinx and RSLogix documentation for help configuring and troubleshooting these applications.

1. Connect an RS-232 serial cable from the COM (serial) port on your PC to the communication port on the front of the processor.
2. Start RSLogix 5000 and close any existing project that may be loaded.
3. Open the **COMMUNICATIONS** menu and choose **GO ONLINE**. RSLogix will establish communication with the processor. This may take a few moments.
4. When RSLogix has established communication with the processor, the *Connected To Go Online* dialog box will open.



5 On the *Connected To Go Online* dialog box, click the **GENERAL** tab. This tab shows information about the processor, including the *Revision* (firmware) version. In the following illustration, the firmware version is 17.2.



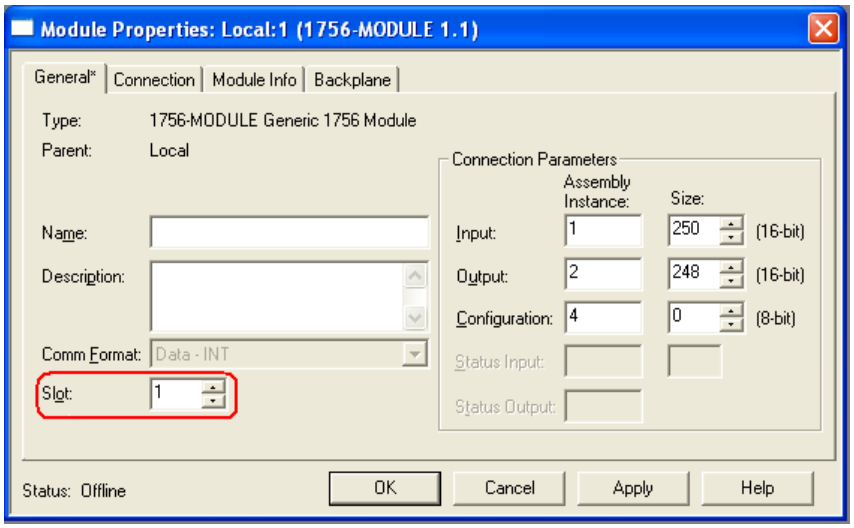
1.7.2 Selecting the Slot Number for the Module

This sample application is for a module installed in Slot 1 in a ControlLogix rack. The ladder logic uses the slot number to identify the module. If you are installing the module in a different slot, you must update the ladder logic so that program tags and variables are correct, and do not conflict with other modules in the rack.

Changing the slot number

1 In the *Controller Organization* list, select the module and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, choose **PROPERTIES**. This action opens the *Module Properties* dialog box.



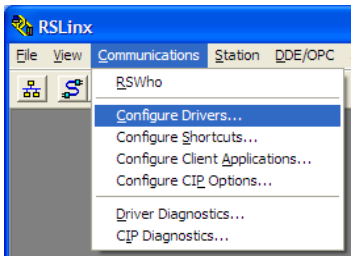
3 In the *Slot* field, use the spinners on the right side of the field to select the slot number where the module will reside in the rack, and then click **OK**.

RSLogix will automatically apply the slot number change to all tags, variables and ladder logic rungs that use the MVI56-101M slot number for computation.

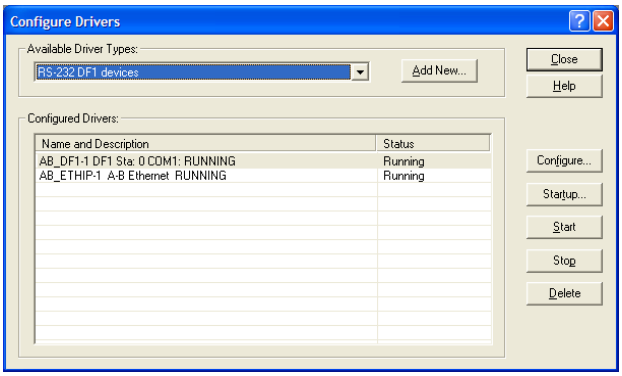
1.7.3 Configuring the RSLinx Driver for the PC COM Port

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**.

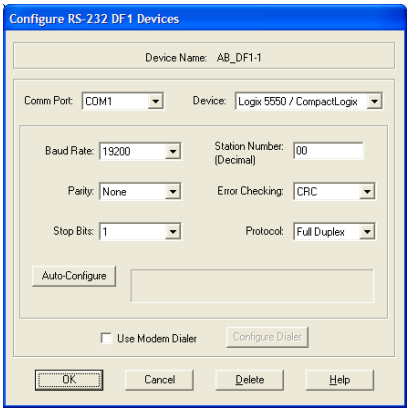


This action opens the *Configure Drivers* dialog box.



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 RS-232 DF1 Devices* dialog box.



4 Click the **AUTO-CONFIGURE** button. RSLinx will attempt to configure your serial port to work with the selected driver.

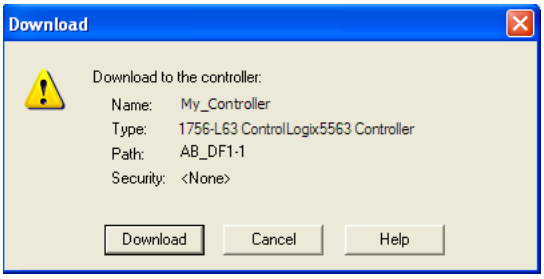
5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

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.

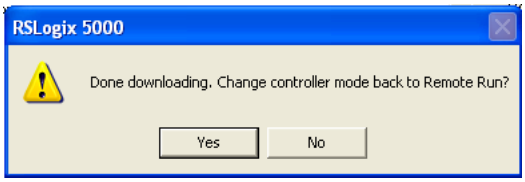
1.8 Downloading the Sample Program to the Processor

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

1. If you are not already online with the processor, open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.
2. When communication is established, RSLogix 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.

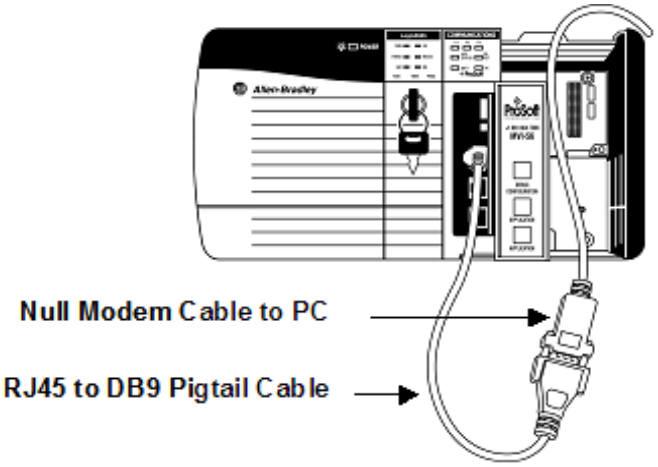


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.9 Connecting 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.



2 Configuring the MVI56-101M Module

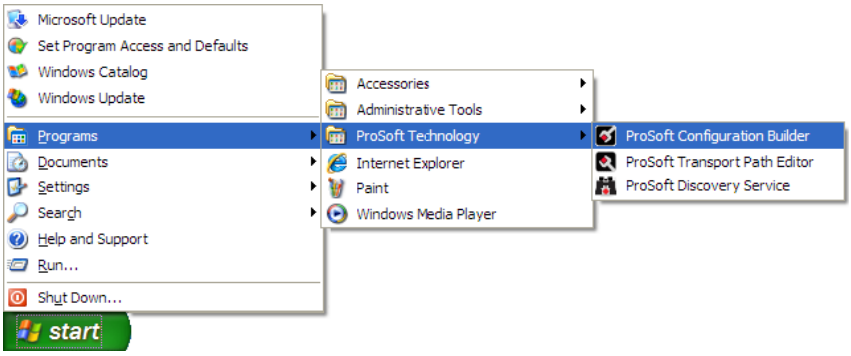
In This Chapter

2.1 Using ProSoft Configuration Builder

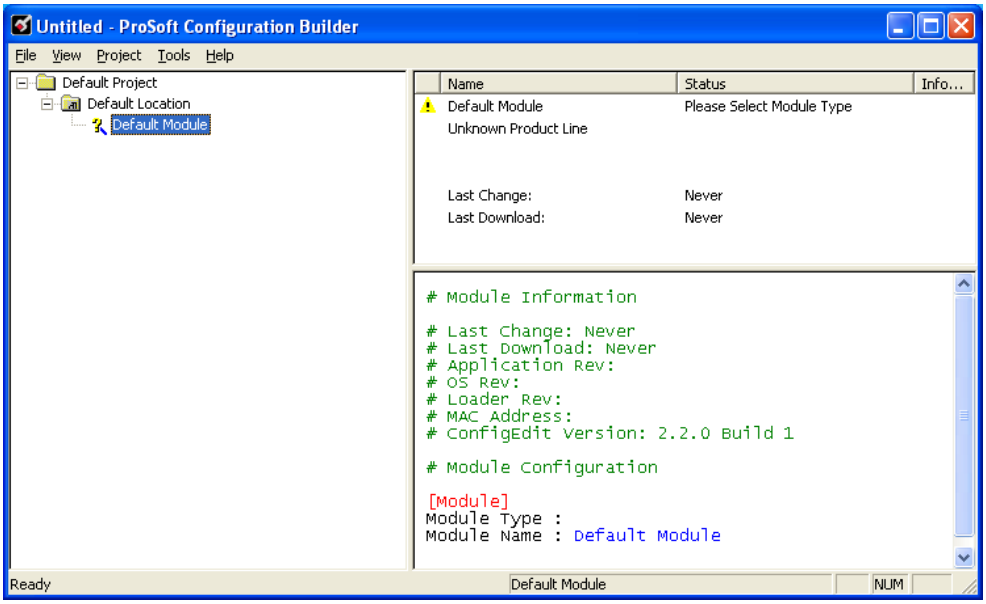
ProSoft Configuration Builder (PCB) provides a quick and easy way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

2.1.1 Setting Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).

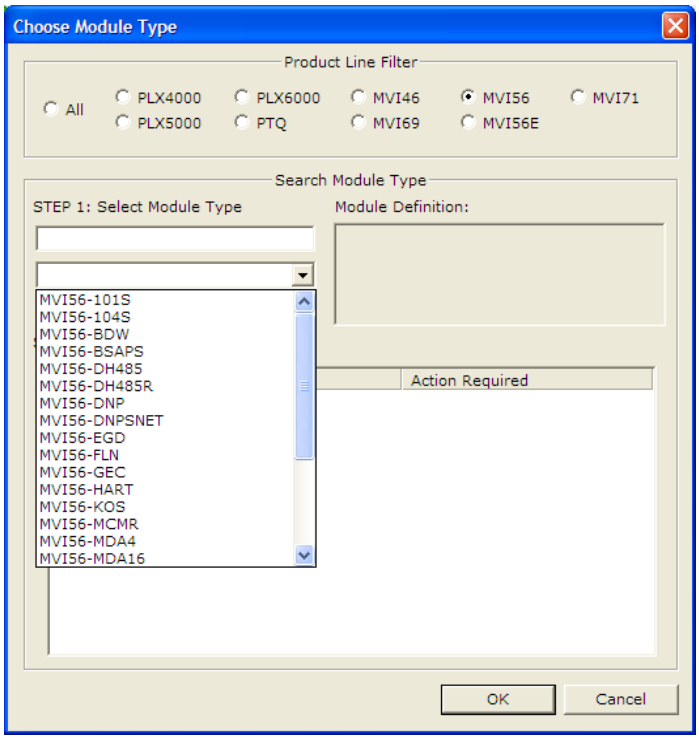


If you have used other Windows configuration tools before, you will find the screen layout familiar. *PCB*'s window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When you first start *PCB*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *PCB* window with a new project.



Adding the MVI56-101M module to the project

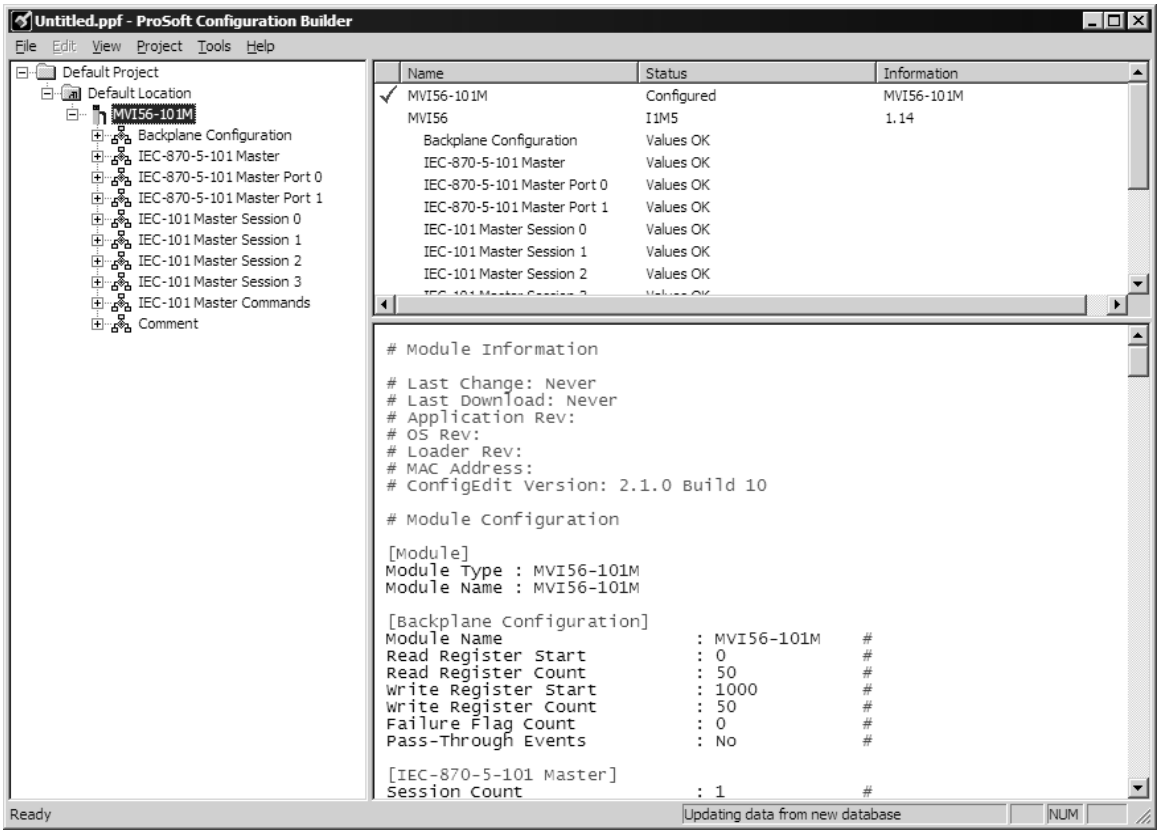
1. Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



3. In the *Product Line Filter* area of the dialog box, select **MVI56**. In the *Select Module Type* dropdown list, select **MVI56-101M**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

2.1.2 Renaming PCB Objects

Notice that the contents of the information pane and the configuration pane changed when you added the module to the project.



At this time, you may wish to rename the *Default Project* and *Default Location* folders in the tree view.

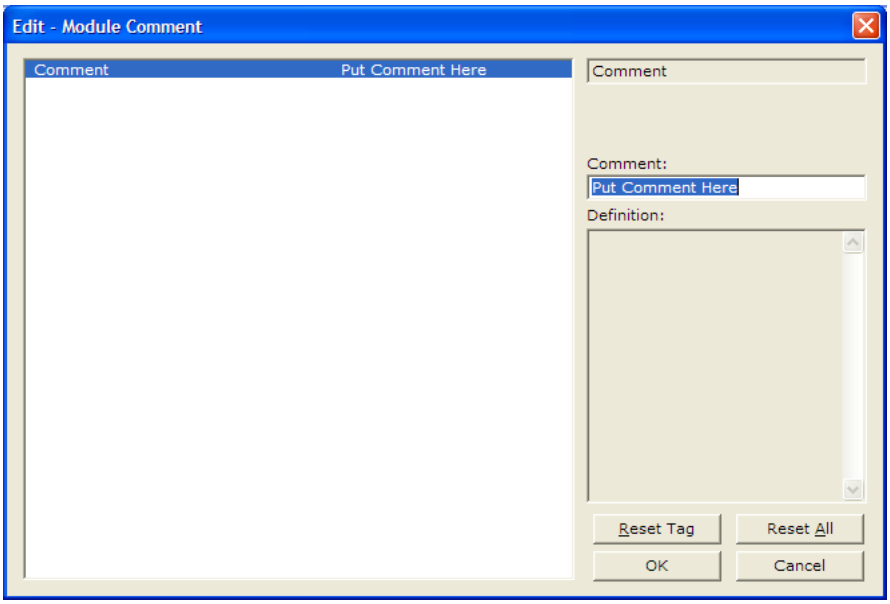
1. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2. Type the name to assign to the object.
3. Click *away* from the object to save the new name.

Configuring Module Parameters

1. Click on the **[+]** sign next to the module icon to expand module information.
2. Click on the **[+]** sign next to any icon to view module information and configuration options.
3. Double-click any icon to open an *Edit* dialog box.
4. To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
5. Click **OK** to save your changes.

Creating Optional Comment Entries

1 Click the **[+]** to the left of the Comment icon to expand the module comments.
2 Double-click the Module Comment icon. The *Edit - Module Comment* dialog box appears.



3 Enter your comment and click **OK** to save your changes.

Printing a Configuration File

1 Select the module icon, and then click the right mouse button to open a shortcut menu.
2 On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.
3 In the *View Configuration* window, open the **FILE** menu, and choose **PRINT.** This action opens the *Print* dialog box.
4 In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK.**

2.2 [Backplane Configuration]

This section contains the backplane configuration information. It is used to determine the number of registers moved between the module and the PLC processor. Additionally, the backplane failure condition is defined as to whether to continue using the protocol.

2.2.1 Read Register Start

0 to **3999**

This parameter specifies the starting register address of a block of data registers to transfer from the module to the processor.

2.2.2 Write Register Count

0 to **4000**

This parameter specifies the number of registers to transfer from the processor to the module.

2.2.3 Write Register Start

0 to **3999**

This parameter specifies the starting register address of a module register block where data transferred from the processor will be stored.

2.2.4 Write Register Count

Range 0 to 4000

This parameter specifies the number of registers to transfer from the processor to the module. Valid entry for this parameter is 0 to 4000.

2.2.5 Failure Flag Count

0 through **65535**

This parameter specifies the number of successive transfer errors that must occur before halting communication on the application port(s). If the parameter is set to **0**, the application port(s) 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.

2.2.6 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.

2.3 [IEC-870-5-101 Master Port x]

2.3.1 Baud Rate

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

Refer to Communication Parameters for more information on configuring serial communications for the port.

2.3.2 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

2.3.3 RTS On

0 to **65535** milliseconds

This parameter sets the number of milliseconds to delay after *Ready To Send* (RTS) is asserted before data will be transmitted.

2.3.4 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.

2.3.5 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.

2.3.6 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 set dependent on the baud rate. A value of 2000 should work with most applications.

2.3.7 Single char ACK F0, 1 or 3

Y - YES or **N - NO**

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

2.3.8 Use Balanced Mode

Y - YES or **N - NO**

This parameter specifies if the port will use balanced mode. If balanced mode is used, only one controlled station will be permitted on the port. If unbalanced mode is used, multiple controlled stations can be used on a port.

Refer to module Initialization for more information on these modes.

2.4 [IEC-101 Master Commands]

This section contains the commands for the module. 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 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).

2.4.1 Enable Code

0 = Disabled

1 = Enabled, will execute using Poll Interval parameter (page 35) (seconds)

2 = Conditional (executed when point in database changes)

This field defines whether the command is to be executed, and under what conditions. To disable the command, set this parameter to 0 (Disabled). You can still execute commands through the processor, using a *Special Function* block.

To enable the command, set this parameter to 1.

- Set the Poll Interval Time to 0 to execute the command during each scan of the command list.
- Set the Poll Interval Time to a value in seconds, to execute the command at the specified interval (page 35).

To execute the command only if the internal data associated with the command changes, set this parameter to 2. This value is valid only for write commands.

2.4.2 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 (page 36) field determines the meaning of the index.

2.4.3 Poll Interval

This parameter specifies the minimum frequency at which the module should execute the command when the Enable Code is set to one **1**. 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, as quickly as possible.

2.4.4 Session Index

0 to 31

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.

2.4.5 Sector Index

Sector Index represents the sector index for the specific session. The range of values for this field is 0 to 4.

2.4.6 Data Type

Data type file represents the ASDU type as follows:

Type	Description	DB Index type
45	Single point command	Bit address
46	Double point command	Bit address
47	Regulating Step point command	Byte address
48	Setpoint, normalized point command	Word address
49	Setpoint, scaled point command	Word address
50	Setpoint, short float point command	Double-word address
51	Bitstring (32-bits) point command	Double-word address
100	Group interrogation command	NA
101	Counter interrogation command	NA
102	Read command	NA
103	Clock Synchronization	NA
104	Test command (101 standard)	NA
105	Reset process command	NA
107	Test command (104 standard)	NA
110	Parameter, normalized measured value	Word address
111	Parameter, scaled measured value	Word address
112	Parameter, short float value	Double-word address
113	Parameter activation command	NA
242	BCD integrated setpoint command	3 word address
255	Send a class 2 poll	NA

Note: The last item in the *Data type* dropdown list is user-defined. If you select **USER DEFINED** from the dropdown list, a text box will appear below the list. You may enter any data type code in this text box that will be accepted by the destination slave.

2.4.7 Point Index

The Point Index specifies the point address to be accessed in the remote slave device.

Index Value	Description
1	Bit address with each point occupying 2 bits
2	Bit address with each point occupying 2 bits
3	Word address with each point occupying 4 words
4	Double-word address for the single float value
5	Byte address with each point occupying 12 bytes
9	Word address with each point occupying 9 words

2.4.8 Qualifier Parameter

The Qualifier Parameter field defined for a command depends on the data type used in the command. In order to compute the qualifier for a command, add all the values for the features to use with a command together to form a single number. This number should be entered in the command record. Each data set is discussed below:

Single Point, Double Point, and Regulating Setup

The format of the field for **Single Point (45)**, **Double Point (46)**, **and Regulating Step (47)** commands is as follows:

Single Point, Double Point and Regulating Step Point Commands

Bit	Single	Double	Step
0	Value	Control	Control
1	0	Value	Value
2 3 4 5 6	Qualifier Code		
7	Select/Execute Code		
8	Deselect Code		
9	Use Override Value		
10 to 15	Not Utilized		

The value field for the different data types can be derived from the module's database or that set in the command. The User Override bit is utilized to select the source of the data value. The values for each data type are defined below:

Single Point Value:

- 0=Off
- 1=On

Double Point Value:

- 0=Not permitted
- 1=Off
- 2=On
- 3=Not Permitted

Regulating Point Value (Set by module using database value -1=next lower, 1=next higher unless override enabled):

- 0=Not permitted
- 1=Next step lower if database point is set to -1
- 2=Next step high if database point set to +1
- 3=Not Permitted

The Qualifier Code area defines the operation to perform as defined below:

Qualifier Code (Select one of the following):

- 0=No additional definition (slave dependent)
- 4=Short pulse duration
- 8=Long pulse duration
- 12=Persistent output

The Select/Execute area defines if the command should perform a direct execute or select before execute command sequence. The values for this field are as follows (Select one of the values for the following list):

- 0=Direct execution without select
- 128=Select executed followed by execute
- 256=Deselect command

The value field for the qualifier can be derived from the module's database or be that defined in the qualifier. If the override flag is used, the module will issue the command using the values contained in the qualifier defined for the command. If the override flag is not set, the module will use the value in the database to send to the controlled device. The values to use for the override flag are as follows:

- 0=Use value in database (value field should be set to zero for qualifier parameter)
- 512=Use override value for state (preferred when using block 9902 with value field set for command to execute)

Normalized, Scaled, and Short Float

The format of the field for *Normalized (48), Scaled (49), and Short Float (50)* setpoint command is as follows:

Normalized, Scaled and Short Float Setpoint Commands

Bit	Description
0	Select/Execute Code
1	Deselect Code
2 to 15	Not Utilized

The value read from database for point specified is used with this qualifier to build a command.

The Qualifier Parameter uses one of the following codes:

- 0=Direct execution without select
- 1=Select executed followed by execute
- 2=Deselect command

Bitstring for 32 Bits

The format of the field for *32-Bitstring (51)* setpoint command is as follows:

Bitstring of 32 Bit Command

Bit	Description
0 to 15	Not Utilized

The value read from database for point specified is used with this qualifier to build a command.

The Qualifier Parameter is not currently used to construct commands.

The format of the field for **Interrogation Command (100)** is as follows:

Interrogation Command

Bit	Description
0 to 7	Interrogation Group
8 to 15	Not Utilized

No database value is associated with the construction of this command.

The Qualifier Parameter used with this command defines the interrogation group to request. Only a single group can be requested in a single command. The codes to use for this field are as follows:

- 0=Not used
- 1 to 19 = Reserved by standard
- 20=Station interrogation (global)
- 21=Interrogation group 1
- 22=Interrogation group 2
- 23=Interrogation group 3
- 24=Interrogation group 4
- 25=Interrogation group 5
- 26=Interrogation group 6
- 27=Interrogation group 7
- 28=Interrogation group 8
- 29=Interrogation group 9
- 30=Interrogation group 10
- 31=Interrogation group 11
- 32=Interrogation group 12
- 33=Interrogation group 13
- 34=Interrogation group 14
- 35=Interrogation group 15
- 36=Interrogation group 16
- 37 to 63 = Reserved by standard
- 64 to 255 = Reserved for special use (private range)

Counter Interrogation

The format of the field for *Counter Interrogation Command (101)* is as follows:

Counter Interrogation Command

Bit	Description
0 to 5	Counter Interrogation Group
6 to 7	Freeze/Reset Qualifier
8 to 15	Not Utilized

No database value is associated with the construction of this command.

The Qualifier Parameter used with this command defines the counter interrogation group to request. Only a single group can be requested in a single command. The qualifier also contains the freeze/reset operation to be utilized with the command The codes to use for this field are as follows:

Counter Interrogation Group:

- 0=No counter requested
- 1=Request counter group 1
- 2=Request counter group 2
- 3=Request counter group 3
- 4=Request counter group 4
- 5=Request general counter group
- 6 to 31 = Reserved by standard
- 32 to 63 = Reserved for special use (private range)

Freeze/Reset Qualifier:

- 0=No freeze or reset
- 64=Counter freeze without reset
- 128=Counter freeze with reset
- 192=No freeze with counter reset

Read Command

The format of this field for the *Read (102)* command is as follows:

Read Command

Bit	Description
0 to 15	Not Utilized

No database value is associated with the construction of this command and no qualifier value is used in this release of the software.

Clock Synchronization

The format of this field for the *Clock Synchronization (103)* command is as follows:

Clock Synchronization Command

Bit	Description
0	Synchronization Mode
1	Qualifier
0 to 15	Not Utilized

No database value is associated with the construction of this command.

The Qualifier Parameter for this command has one of the following values:

- 0=Clock synchronization with out delay utilized
- 1=Synchronize clock with delay set
- 2=Measure delay, load delay then synchronize clock

Test Command

The format of this field for the *test command (104 and 107)* is as follows:

Test Command (both 101 and 104 versions)

Bit	Description
0 to 15	Not Utilized

No database value is associated with the construction of this command and no qualifier value is used in this release of the software.

Reset Process

The format of this field for the *Reset Process (105)* command is as follows:

Reset Process Command

Bit	Description
0 to 7	Reset Qualifier
8 to 15	Not Utilized

No database value is associated with the construction of this command.

The Qualifier Parameter has one of the following values as define in the protocol specification:

- 0=Not used
- 1=General reset of process
- 2=Reset pending information with time tag of the event buffer
- 3 to 127 = Reserved by standard
- 128 to 255 = Reserved for special use (private range)

Parameter Setting

The format of this field for the *Parameter Setting (110=Normalized, 111=Scaled, 112=Short float) i*s as follows:

Parameter Setting (Normalized, Scaled, Short Float) Command

Bit	Description
0 to 5	Kind of Parameter
6	Local change
7	Operation
8 to 15	Not Utilized

The value from module's database utilized to build the command.

The Qualifier Parameter used with this command is determined by summing the options from lists that follow:

Kind of parameter:

- 0=Not used
- 1=Threshold value
- 2=Smoothing factor (filter time constant)
- 3=Low limit for transmission of measured values
- 3=High limit for transmission of measured values
- 5 to 31 = Reserved by standard
- 32 to 63 = Reserved for special use (private range)

Local parameter change:

- 0=No change
- 64=Change

Parameter in operation:

- 0=Operation
- 128=Not in operation

Parameter Activation

The format of this field for the *Parameter Activation (113)* is as follows:

Parameter Activation Command

Bit	Description
0 to 7	Parameter Qualifier
8	Activation Qualifier
9 to 15	Not Utilized

No database value used with the construction of this command.

The Qualifier Parameter used with the command is determined by summing the options from the lists that follow:

Parameter Qualifier:

- 0=Not used
- 1=Act/Deact of previously loaded parameters (point index = 0)
- 2=Act/Deact of the parameter of the point index specified
- 3=Act/Deact of persistent cyclic or periodic transmission of the addressed object
- 4 to 127 = Reserved by standard
- 128 to 255 = Reserved for special use (private range)

Activation Qualifier:

- 0=Deactivate
- 256=Activate

BCD Integrated Setpoint

The format of this field for *BCD Integrated Setpoint (242)* command is as follows:

BCD Integrated total Setpoint Command

Bit	Description
0	Select/Execute Code
1	Deselect Code
2 to 15	Not Utilized

The value in database is utilized for this command . The data resides in a 6-byte data area in the module.

The Qualifier Parameter used with this command is selected from the following list:

- 0=Direct execution without select
- 1=Select executed followed by execute
- 2=Deselect command

The format of this field for the Class 2 poll (255) command is as follows:

Class 2 Poll

Bit	Description
0 to 15	Not Utilized

No database or qualifier is used with this command.

2.5 [IEC-870-5-101 Master]

This is the configuration for the IEC-870-5-101 master port emulated on the module.

2.5.1 Session Count

1 to 32

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 all ports.

2.6 [IEC-101 Master Session x]

This section defines Session *y,* which runs on Port *x*.The *Session Count* parameter in the *[IEC-870-5-101 Master]* section of the configuration (page 47) determines the number of sessions (controlled devices) for this port.

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]*).

The parameters in *[IEC-101 Master Session y]* define the characteristics of the specific controlled device to interface.

2.6.1 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.

2.6.2 Sector Count

1 to 5

This parameter sets the number of sectors contained in this controlled device. The range of values is from 1 to 5. A sector section is required for each sector in a session to define its database and settings.

2.6.3 Data Link Address

0 to 254 or 0 to 65534

This parameter uniquely defines the data link address for this unit on the communication channel The ranges of values depends on the value set in the DL Address Length parameter.

2.6.4 Common address of ASDU Len

1 or 2

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

2.6.5 Inform. Object address Len

1, 2 or 3

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

2.6.6 COT octet count

1 or 2

This parameter sets the number of octets used for the COT field in each message. If a value of 2 is selected, the value entered for the Originator Address For COT will accompany each message from the controlling unit.

2.6.7 Originator address for COT

0 to 255

This parameter sets the address to be passed with each message when the COT Octet Count parameter is set to 2.

2.6.8 Failure Delay

0 to 2000 seconds

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.

2.6.9 Confirm Timeout

0 to 4,294,967,295 (two raised to the power of 32, minus one) milliseconds

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

2.6.10 Retry Count

0 to 255

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

2.6.11 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.

2.6.12 Class 1 Polls

0 to 65535

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.

2.6.13 Class 1 Pend Delay

0 to 4,294,967,295 (two raised to the power of 32, minus one) milliseconds

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

2.6.14 Class 2 Pend Delay

0 to 4,294,967,295 (two raised to the power of 32, minus one) milliseconds

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

2.6.15 Class 1 Poll Delay

0 to 4,294,967,295 (two raised to the power of 32, minus one) milliseconds

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

2.6.16 Class 2 Poll Delay

0 to 4,294,967,295 (two raised to the power of 32, minus one) milliseconds

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

2.6.17 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.

2.6.18 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.

2.6.19 Response Timeout

0 to 4,294,967,295 milliseconds

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

2.6.20 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.

2.7 [IEC-101 Master Session x Sector y]

This section defines Sector *z*, which belongs to Session *y*. The *Sector Count* parameter within each session definition specifies the number of sectors for the session.

Each sector has a corresponding *[IEC-101 Master Session y Sector z]* section, where *y* represents the session index and *z* represents the sector index.

The sectors are referenced by a zero-based index value. For example, if Session 0 is configured for four sectors, the configuration file should contain sections for Sectors 0 to 3 (that is, *[IEC-101 Master Session 0 Sector 0]* to *[IEC-101 Master Session 0 Sector 3]*).

The parameter set and database for Sector *z* is defined in this section.

2.7.1 Common ASDU Address

0 to 255 (1 oct) or 0 to 65535 (2 oct)

This parameter sets the common ASDU address to association with this sector of the specified session. The range of address for this parameter are dependent on the length value set in the session section.

2.7.2 Use Time tag commands

Yes or No

This parameter specifies if a time tag field is to be included with commands. This is as specified in the IEC-870-5-104 specification and should only be utilized if the controlled device supports these new data types. If the parameter is set to Yes, a time tag will be added to all commands. If the parameter is set to No, the normal IEC 60870-5-101 data type messages will be utilized.

2.7.3 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.

2.7.4 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.

2.7.5 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.

2.7.6 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.

2.7.7 Database Definition

Database definition for this session/sector.

Data Types are as follows:

Monitored Data

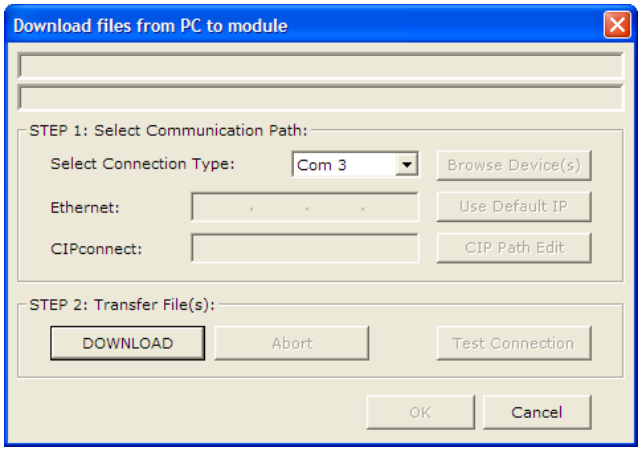
- 1 = Single point
- 3 = Double point
- 5 = Step point
- 7 = Bitstring of 32-bits
- 9 = Measured normalized points
- 11 = Measured scaled points
- 13 = Measured short float points
- 15 = Integrated totals
- 110 = Measured normalized parameter (word-addressing/1 point = 1 data word)
- 111 = Measured scaled parameter (word-addressing/1 point = 1 data word)
- 112 = Measured short float parameters (double-word-addressing/1 point = 2 data words)
- 240 = Integrated totals BCD format (3 word-addressing/1 point = 3 data words)

Note: The last item in the *Data type* dropdown list is user-defined. If you select **USER DEFINED** from the dropdown list, a text box will appear below the list. You may enter any data type code in this text box that will be accepted by the destination slave.

2.8 Downloading the Project to the Module Using a Serial COM port

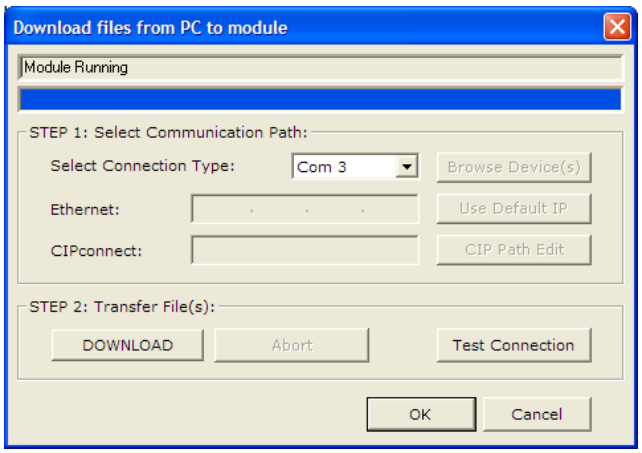
For the module to use the settings you configured, you must download (copy) the updated *Project* file from your PC to the module.

1. In the tree view in *ProSoft Configuration Builder*, click once to select the module.
2. Open the *Project* menu, and then choose **MODULE/DOWNLOAD**. The program will scan your PC for a valid com port (this may take a few seconds). When *PCB* has found a valid COM port, the *Download* dialog box will open.



3. Choose the COM port to use from the dropdown list, and then click the **DOWNLOAD** button.

   The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the *Download* dialog box will display the message *Module Running*.



3 Ladder Logic

In This Chapter

Ladder logic is required for application of the MVI56-101M 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 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

All data related to the MVI56-101M module 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.

3.1.1 Module Status Data and Variables (101MmoduleDef)

All status and variable data related to the MVI56-101M 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 following table describes the structure of the object.

Name	Data Type	Description
GenStat	I101MStat	General status information
BP	I101MBackplane	Data to handle backplane logic

This object contains objects that define variables for 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 (101MStat)

This object stores the status data of the module. The 101MStat 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".

Name	**Data Type**	**Description**
Scan_Cnt	INT	Program Scan Counter
Product_Name	SINT[4]	Product Code
Rev_Level	SINT[4]	Revision
Op_Sys	SINT[4]	Operating system revision
Run_Number	SINT[4]	Run number
Blk_Rd_Count	INT	Number of block read transfers
Blk_Wr_Count	INT	Number of block write transfers
Blk_Parse_Cnt	INT	Number of blocks parsed by module
Blk_Err	INT	Number of block errors
Evnt_Cnt	INT	Number of event messages in buffer
Evnt_Overflow	INT	Flag to indicate event message buffer overflow (1=overflow)
SesCnt	INT	Number of sesseions configured
CurCmd	INT	Index of command executing
CmdBusy	INT	Command busy flag
CmdMax	INT	Maximum number of commands configured
CmdDelay	INT	Command delay counter
CmdQueue	INT	Command Queue Flag
CmdQCnt	INT	Number of commands in command queue
Online	DINT	Online status bits for each session
ChStat	I101MCHStat[2]	Channel Status Data

Within the 101MStat objects are objects containing the status information for each application port (101MCHStat). Refer to the Reference chapter for a complete listing of the data stored in this object.

Channel Status Object (101MCHStat)

The 101MCHStat object holds the status data related to a single IEC 60870-5-101 master port. The following table describes the structure of the object.

Name	Data Type	Description
State	INT	State machine value
CmdReq	INT	Number of command requests
CmdResp	INT	Number of command responses
CmdErr	INT	Number of command errors
Req	INT	Number of request messages
Resp	INT	Number of responses
ErrSent	INT	Number of errors sent
ErrRec	INT	Number of errors received
CfgErr	INT	Configuration Error Word for channel
CurErr	INT	Current error code for channel
LastErr	INT	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 101MCHStat 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	Invalid length data for session: Data link length Command address of ASDU length Information object address length COT octet count
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	

Backplane Object (101MBackplane)

The 101MBackplane 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 following table describes the structure of the object.

Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table

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

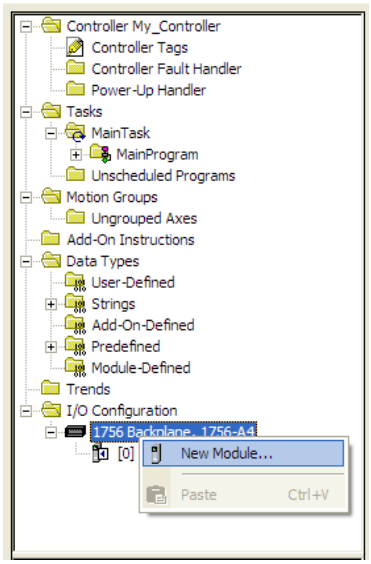
3.1.2 Data Object (101MData)

The 101MData object is defined in the example ladder logic to demonstrate how the data from a module can be stored in the processor. This object can be used to temporarily store a data set received. A structure of this type can be constructed for each session or sector or can be set as in the example as one large database. The user should decide the best data storage method for their application. Another factor in determining the database layout is the simplicity of the ladder logic. In the example ladder logic, only one data copy operation is required for each data type as the data is packed in the module. If separate data areas are defined for each session, more copy instructions would be required. This could simplify data management. The following table describes the structure of the object.

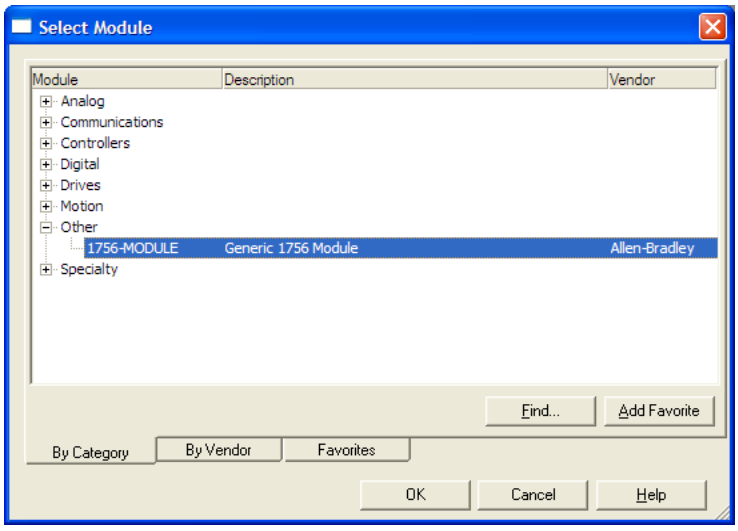
Name	Data Type	Description
I101M_SP	INT[2]	Data area to hold single point data
I101M_DP	INT[2]	Data area to hold double-point data
I101M_ST	SINT[4]	Data area to hold step data
I101M_BO	DINT[4]	Data area to hold bitstring data (32-bits/point)
I101M_MN	INT[10]	Data area to hold measured, normalized data
I101M_MS	INT[5]	Data area to hold measured, scaled data
I101M_MF	REAL[5]	Data area to hold measured, short float data
I101M_IT	DINT[10]	Data area to hold integrated total data
I101M_CSP	INT[2]	Data area to hold command single point data
I101M_CDP	INT[2]	Data area to hold command double point data
I101M_CRS	SINT[6]	Data area to hold command regulating step data
I101M_CSN	INT[10]	Data area to hold command normalized setpoints
I101M_CSS	INT[11]	Data area to hold command scaled setpoints
I101M_CSF	REAL[6]	Data area to hold command float setpoints
I101M_CSB	DINT[2]	Data area to hold bitstring setpoints

3.2 Adding the Module to an Existing Project

1 Select the *I/O Configuration* folder in the *Controller Organization* window of RSLogix 5000, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE**.



This action opens the *Select Module* dialog box:

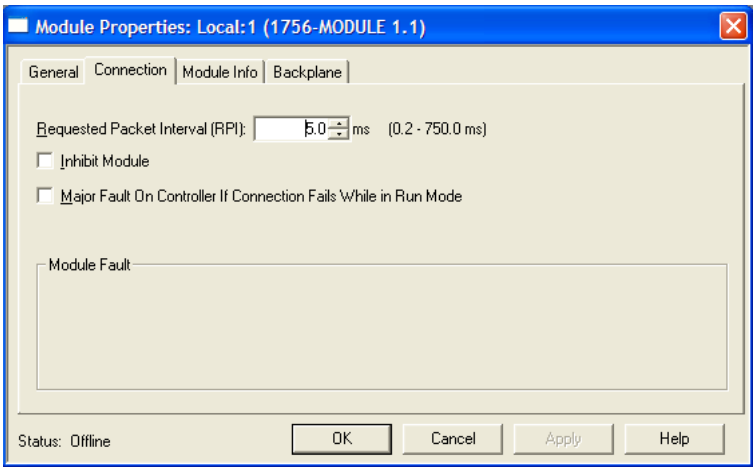


2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK**. This action opens the *New Module* dialog box.

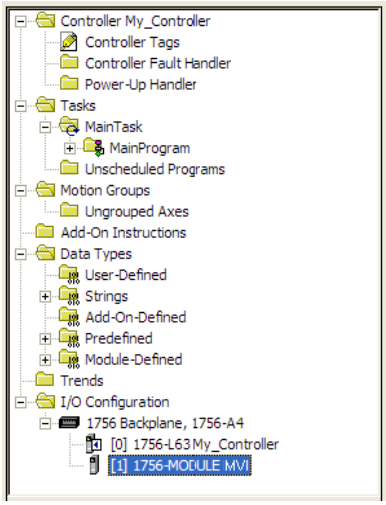
3 Enter the *Name, Description* and *Slot* options for your application. You must select the *Comm Format* as **DATA - INT** in the dialog box, otherwise the module will not communicate. Click **OK** to continue.

Parameter	Value
Name	Enter a module identification string. Example: **_2**
Description	Enter a description for the module. Example: **IEC 60870-5-101 MASTER COMMUNICATION MODULE**
Comm Format	Select **DATA-INT**.
Slot	Enter the slot number in the rack where the MVI56-101M module is located.
Input Assembly Instance	**1**
Input Size	**250**
Output Assembly Instance	**2**
Output Size	**248**
Configuration Assembly Instance	**4**
Configuration Size	**0**

4 Select the *Requested Packet Interval* value for scanning the I/O on the module. This value represents the minimum frequency that the module will handle scheduled events. This value should not be set to less than **1** millisecond. The default value is **5** milliseconds. Values between **1** and **10** milliseconds should work with most applications.



5 Save the module. Click **OK** to dismiss the dialog box. The *Controller Organization* window now displays the module's presence.



6 Copy the *User-Defined Data Types* from the sample program into your existing RSLogix 5000 project.
7 Copy the *Controller Tags* from the sample program into your project.
8 Copy the *Ladder Rungs* from the sample program into your project.

4 Diagnostics and Troubleshooting

In This Chapter

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

- LED status indicators on the front of the module provide general information on the module's status.
- Status data contained in the module can be viewed through the Configuration/Debug port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- Status data values can be transferred from the module to processor memory and can be monitored there manually or by customer-created logic.

4.1 LED Status Indicators

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

LED	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 MVI56-101M is working normally.
		On	The MVI56-101M 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.

4.1.1 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 correct configuration data is being transferred to the module from the ControlLogix controller.

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 Technical Support.

4.1.2 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 in the I/O Configuration of RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56-101M. Verify that all modules in the rack are correctly configured in the ladder logic.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56E modules with scrolling LED display: *<Backplane Status>* condition reads ERR	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 or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
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 it, and then restore power to the rack.

4.2 Using ProSoft Configuration Builder (PCB) for Diagnostics

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 submenus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

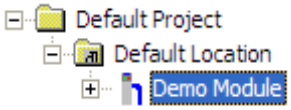
Because this is a text-based menu system, you enter commands by typing the [command letter] from your computer keyboard in the *Diagnostic* window in *ProSoft Configuration Builder (PCB)*. The module does not respond to mouse movements or clicks. The command executes as soon as you press the **[COMMAND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

4.2.1 Using the Diagnostic Window in ProSoft Configuration Builder

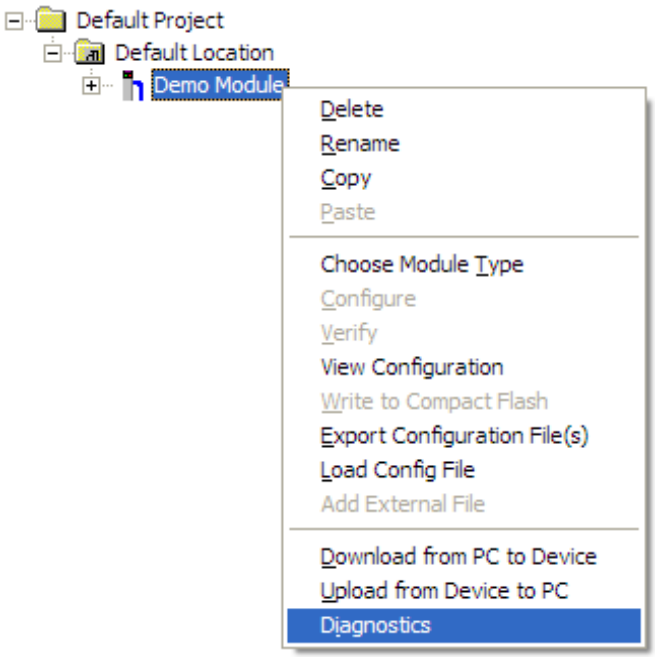
Tip: You can have a ProSoft Configuration Builder Diagnostics window open for more than one module at a time.

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

1. Start *PCB*, and then select the module to test. Click the right mouse button to open a shortcut menu.

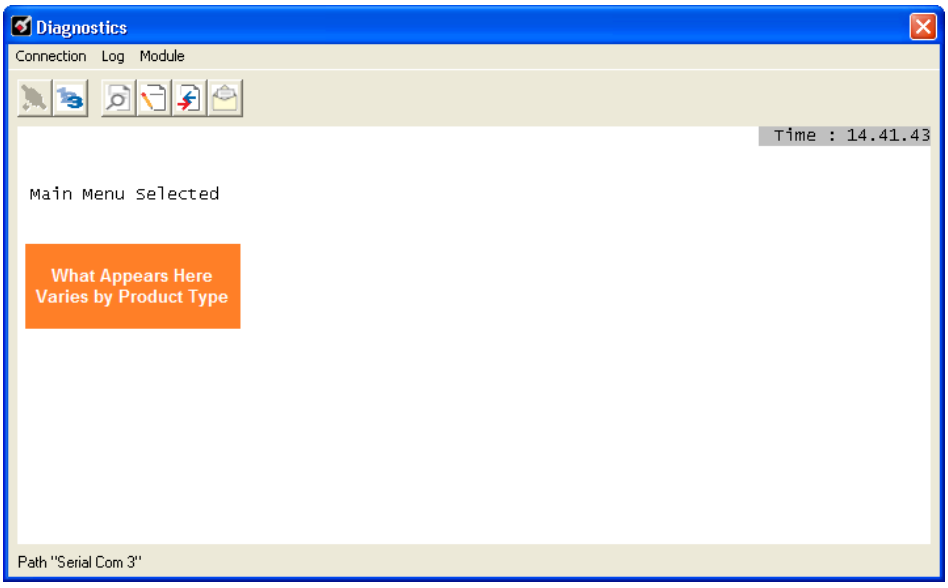


2 On the shortcut menu, choose **DIAGNOSTICS.**



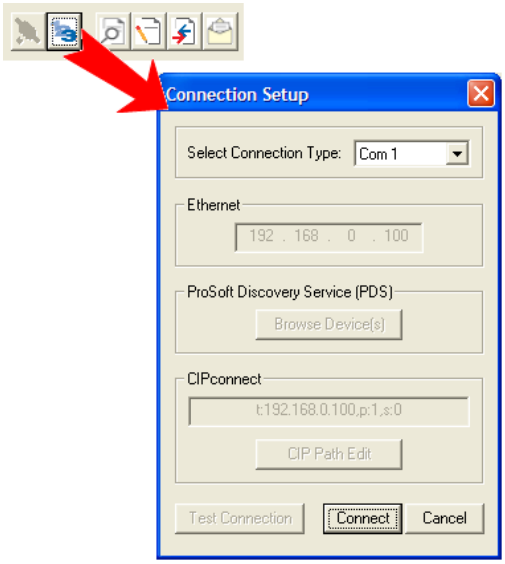
This action opens the *Diagnostics* dialog box.

3 Press **[?]** to open the *Main* menu.



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

1 Click to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.



2 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.

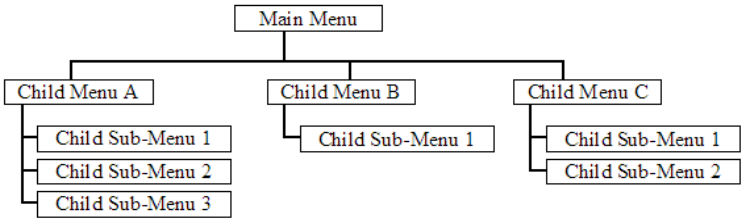
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

Keystrokes

The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase 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** and */*.

Also, take care to distinguish the different uses for uppercase letter "eye" (**I**), lowercase letter "el" (**L**), and the number one (**1**). Likewise, uppercase letter "oh" (**O**) and the number zero (**0**) are not interchangeable. Although these characters look alike on the screen, they perform different actions on the module and may not be used interchangeably.

4.2.2 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.

```
IEC-870-5-101 MASTER COMMUNICATION MODULE
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-101 Master Menu
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  Esc=Exit Program
```

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

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.

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 View Menu

Press **[D]** to open the *Database View* menu.

Use this menu command to view the current contents of the module's database. For more information about this submenu, see Database View Menu (page 71).

Opening the IEC-101 Master Menu

Press **[I]** from the Main Menu to open the IEC-870-5-101 Master Driver Menu. Use this menu command to view detailed configuration information for the module.

Transferring the Configuration File from the PC to the Module

On the Diagnostics Menu this is referred to as *Receive Module Configuration*.

Press **[R]** to receive (download) the configuration file from your PC to the module and store the file on the module's Compact Flash Card (Personality Module) or Flash RAM.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, the module will restart the program and load the new configuration information. Review the new configuration using menu commands **[6]** and **[0]** to verify that the module is configured correctly.

Transferring the Configuration File from The Module to the PC

On the Diagnostics Menu this is referred to as *Send Module Configuration*.

Press **[S]** to send (upload) the configuration file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully uploaded, you can open and edit the file to change the module's configuration.

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

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

4.2.3 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 database. Press **[?]** to view a list of commands available on this menu.

```
DB Menu Selected

DATABASE VIEW MENU
 ?=Display Menu
 0-9=Display 0-9000
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 D=Decimal Display
 H=Hexadecimal Display
 F=Float Display
 A=ASCII Display
 M=Main Menu
```

Viewing Register Pages

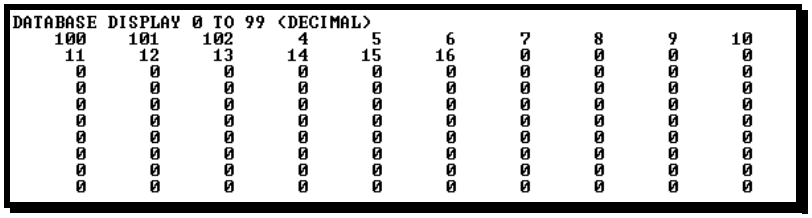
To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again

Press **[S]** from the *Database View* menu to show the current page of registers again.



This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the *Database View* menu to skip five pages back in the database to see the 100 registers of data starting 500 registers before the currently displayed page.

Moving Forward (Skipping) Through 5 Pages of Registers

Press **[+]** from the *Database View* menu to skip five pages ahead in the database to see the 100 registers of data starting 500 registers after the currently displayed page.

Viewing the Previous Page of Registers

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

Viewing the Next Page of Registers

Press **[N]** from the *Database View* menu to display the next page of data.

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating-Point Format

Press **[F]** from the *Database View* menu 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

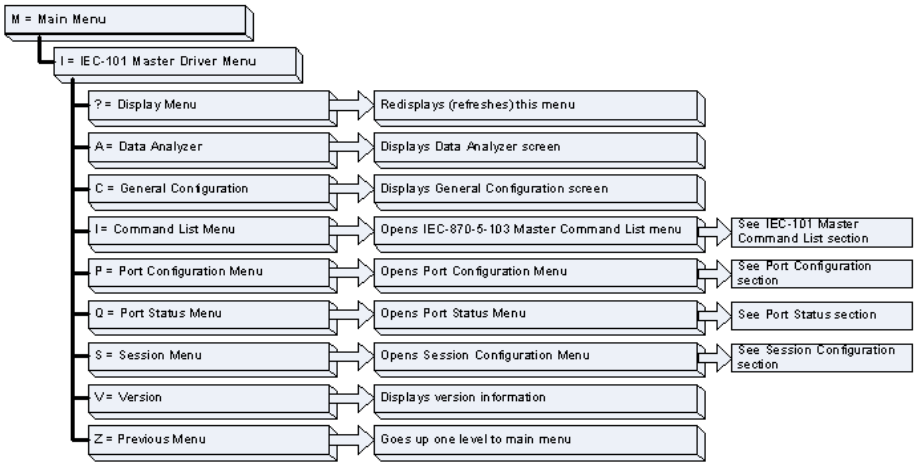
Press **[A]** from the *Database View* menu to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.2.4 IEC-101M Master Menu

Press **[I]** from the Main Menu to open the ICE-870-5-101 Master Driver Menu. Use this menu command to view detailed configuration information 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.

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 (page 75) 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 Protocol Configuration

Press **[C]** to view configuration information for the protocol.

Opening the Client Command List Menu

Press **[I]** to open the Client Command List menu. Use this command to view the configured command list for the module.

Viewing Port Configuration

Press **[P]** to view configuration information for the application port.

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

Viewing Port Communication Status

Press **[Q]** to view the port communication status for the application port.

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

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* for more information about the commands on this menu.

4.2.5 Data Analyzer

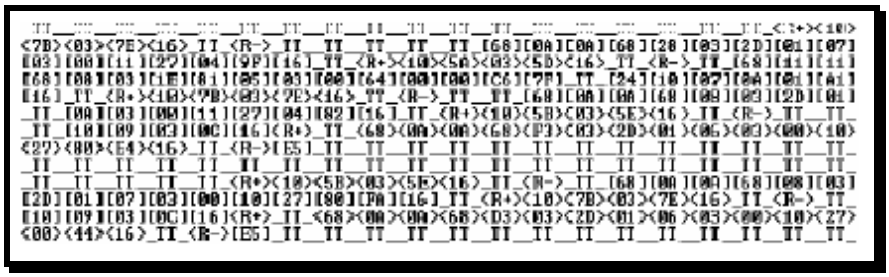
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

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
[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

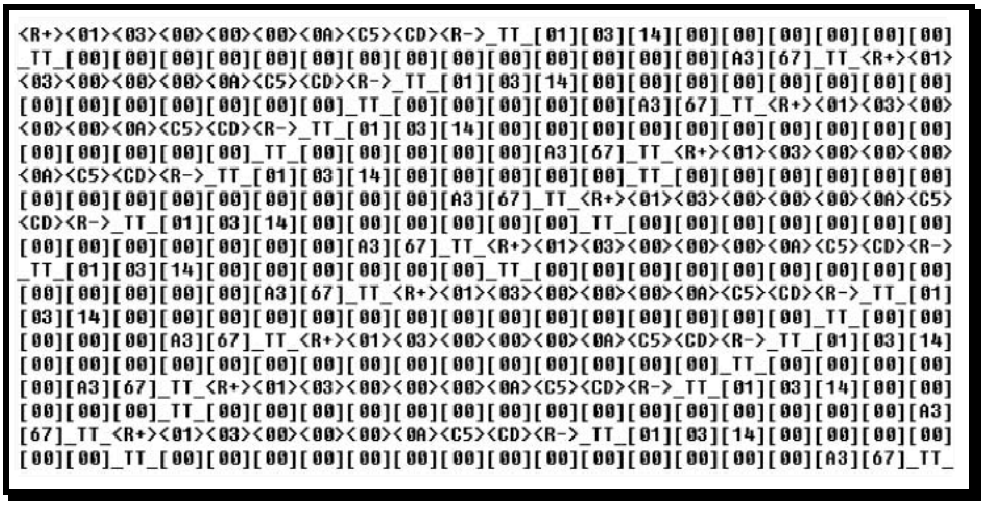
Press **[H]** from the *Database View* menu to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

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



The Data Analyzer displays the following special characters:

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

Stopping the Data Analyzer

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

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

Data Analyzer Tips

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

```
Data Analyzer Mode Selected
```

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

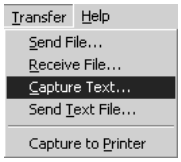
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

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

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

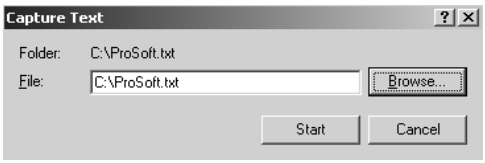
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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

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



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



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

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

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

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

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

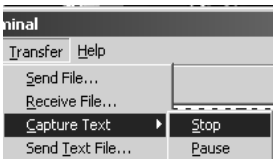
All characters shown in <> brackets are characters being sent out by the module.

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

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

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

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



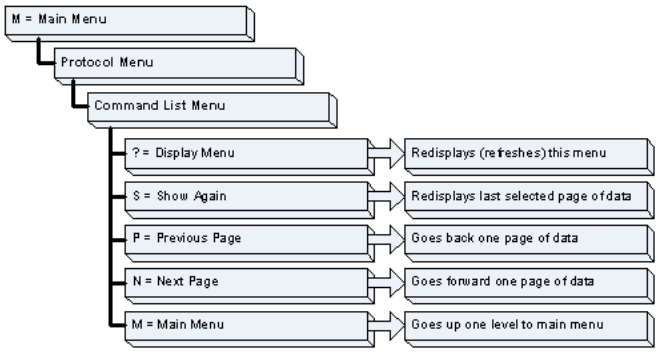
You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

Returning to the Main Menu

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

4.2.6 Master Command List Menu

Use this menu to view the command list for the module. Press **[?]** to view a list of commands available on this menu.



Redisplaying the Current Page

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

Viewing the Previous 50 Commands

Press **[-]** to view the previous 50 commands.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next 50 Commands

Press **[+]** to view the next 50 commands from the master command list.

Viewing the Next Page of Commands

Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.2.7 Session Configuration Menu

Press **[S]** from the IEC-101 Master Driver Menu to open the Session Configuration menu. Use this command to view the session configuration for each controlled device.

```
IEC-870-5-101 MASTER SESSION 0 CONFIGURATION
  Online State           = 1
  Communication Port     = 0
  Sector Count           = 1
  Data Link Address      = 3
  Common ASDU Length     = 2
  IOA Length             = 2
  COT Octet Count        = 1
  COT Originator Address = 1
  Failure Delay          = 120
  Confirm Timeout        = 1000
  Retry Count            = 2
  C1/C2 Poll Count Pend  = 100
  Class 1 Polls          = 20
  Class 1 Pend Delay     = 10
  Class 2 Pend Delay     = 10
  Class 1 Poll Delay     = 10
  Class 2 Poll Delay     = 10
  Auto Clk Sync Mode     = 0
  Propagation Delay      = 0
  Response Timeout       = 2000
  ACTTERM with setpoint  = 1
```

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.

4.2.8 Sector Configuration Menu

Press **[1]** from the IEC-101 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.

```
SECTOR CONFIGURATION MENU
 ?=Display Menu
 S=Show again
 0=Single-point data
 1=Double-point data
 2=Step point data
 3=Bitstring point data
 4=Normalized measured point data
 5=Scaled measured point data
 6=Short float measured point data
 7=Integrated total point data
 8=Parameter, normalized data
 9=Parameter, scaled data
 A=Parameter, short float data
 B=BCD integrated total data
 M=Return to Sector
```

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

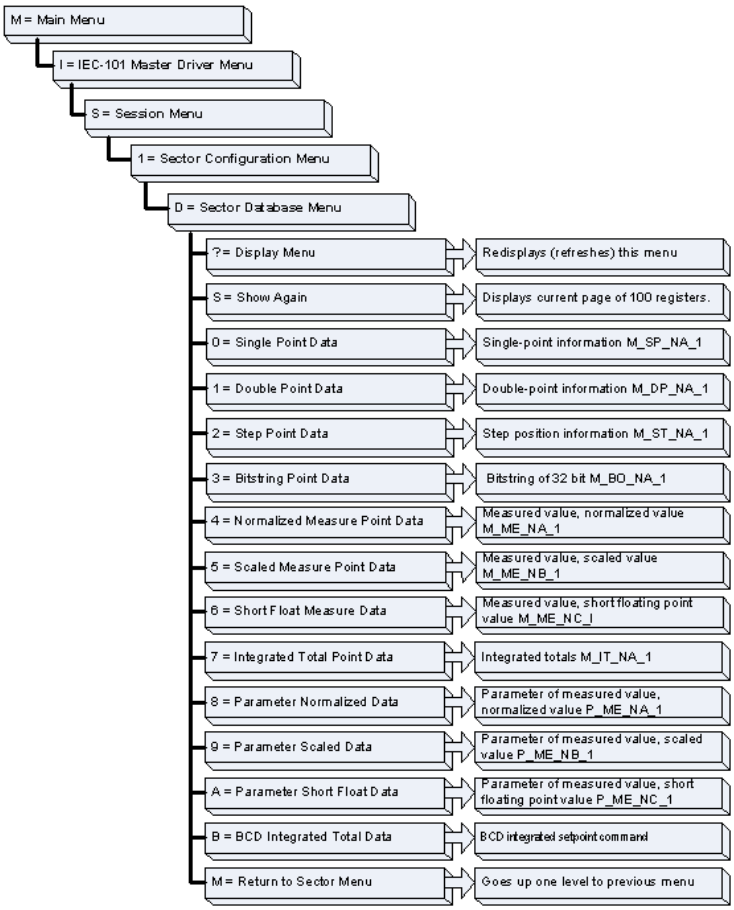
Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Returning to the Main Menu

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

4.2.9 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.

Returning to the Main Menu

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

4.3 Reading Status Data from the Module

The MVI56-101M module returns a status data set to the ControlLogix processor in each read block. This data is transferred to the ControlLogix 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
- Facility to upload and download the module's configuration file

5 Reference

In This Chapter

5.1 Product Specifications

The MVI56 IEC 60870-5-101 Master Communication Module allows ControlLogix to interface easily with IEC 60870-5-101 Master (controlling unit) devices.

The MVI56-101M module interfaces up to 32 serial communication devices with the <Processor Model>. Two communication ports on the module act as controlling devices (masters) to interface with controlled devices on their own networks. Each port is individually configurable and can be set for balanced or unbalanced mode. Data is exchanged between the serial network and the <Processor Model> through the database contained in the module and direct control by the processor's ladder logic.

5.1.1 General Specifications

- Single Slot - 1756 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
- Local or remote rack

5.1.2 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 V DC 3mA @ 24V DC
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Shock	30g Operational 50g non-operational Vibration: 5 g from 10 to 150 Hz
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Module Status Backplane Transfer Status Application Status Serial Activity
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Application ports (PRT1 & PRT2) **Full hardware handshaking control, providing radio, modem and multi-drop support**	
Software configurable communication parameters	Baud rate: 110 to 38,400 baud RS-232 and 422 Parity: none, odd or even Data bits: 5, 6, 7, or 8 Stop bits: 1 or 2 RTS on/off delay: 0 to 65535 milliseconds
App Ports (P1,P2) (Serial modules)	RJ45 (DB-9M with supplied cable) RS-232 handshaking configurable 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

- Built in accordance to the approved international specification
- Two independent master ports completely user configurable
- Support for balanced and unbalanced mode
- Up to 32 sessions
- Up to five sectors (separate databases) for each session
- Individual database definition for each sector
- 1000 commands to control stations
- Processor can issue control commands directly to the module or a controlled device (10 at each scan)
- Pass-through of event messages from controlled device to processor for logging of time-tagged events
- Operation via simple ladder logic
- Supports clock synchronization from/to the processor
- Receives events from the slave and sends them to the processor
- Supports monitored data
  - Single-point
  - Double-point
  - Step-point
  - Measured-point
  - Bitstring 32-bit
  - Integrated total point
- Class 1 and Class 2 delay parameter in the configuration file
- Complete set up and monitoring of module through RSLogix 5000 software and user constructed configuration file (IEC101M.CFG)
- All data related to the module is contained in user data files to simplify monitoring and interfacing with the module

5.2 Functional Overview

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

PUBLICATION	TITLE
IEC 60870-5-101	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-101 Amendment 1	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions
IEC 60870-5-104	Network access for IEC 60870-5-101 using standard transport profiles

Detailed questions about the protocol can be answered by reading these documents. 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.

The following discussion explains several concepts that are important for understanding module operation.

5.2.1 General Concepts

he following discussion explains several concepts that are important for understanding module operation.

Module Power Up

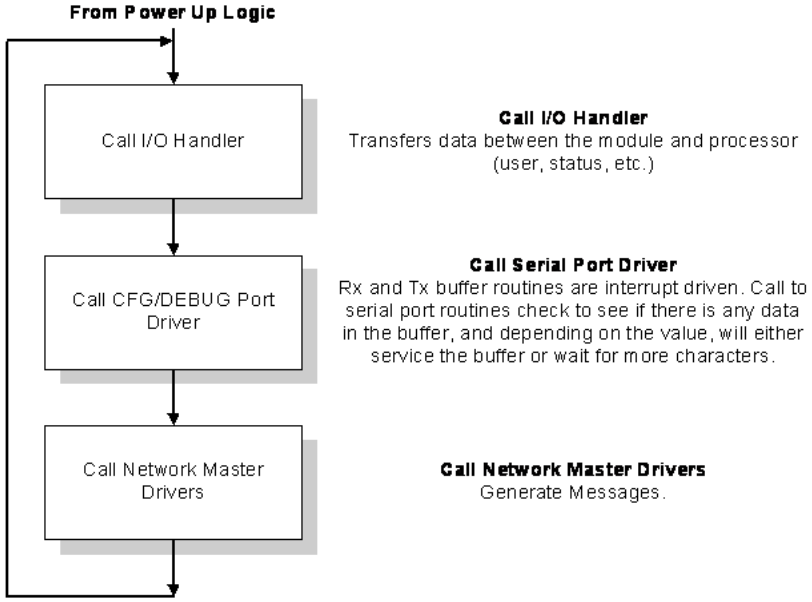
On power up the module begins performing the following logical functions:

1. Initialize hardware components
   - Initialize ControlLogix backplane driver
   - Test and clear all RAM
   - Initialize the serial communication ports
2. Read configuration for module from IEC101M.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:



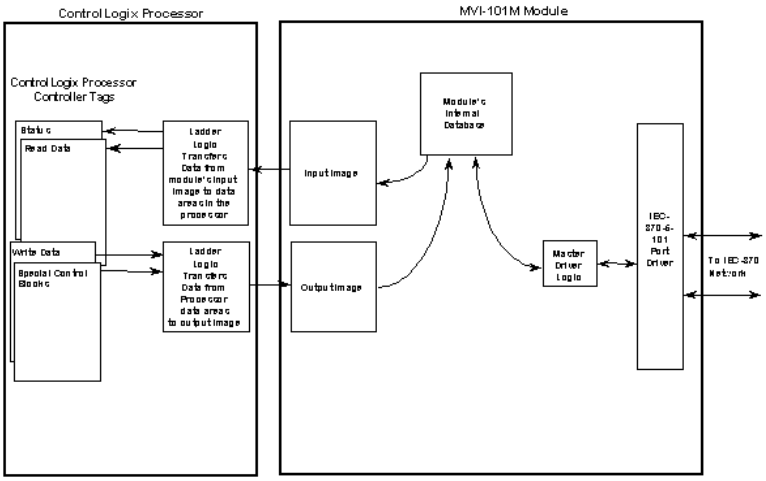
Backplane Data Transfer

The MVI56-101M module communicates directly over the ControlLogix backplane. Data travels between the module and the ControlLogix 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 ControlLogix processor. The input image for the module is set to 500 bytes. This large data area permits fast throughput of data between the module and the 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 ControlLogix processor can send special control blocks to the module to instruct it to perform a special task. The output image for the module is set to 496 bytes. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-101M 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 ControlLogix 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.

5.2.2 Data Type Mapping and Addressing

When interfacing data in the processor to that of the IEC 60870-5-101 protocol, it is important that the user understand the mapping of the data types to their corresponding representation in the modules database. The table that follows lists the data types supported by the module and their associated storage representation.

IEC-870-5-101 Data Types

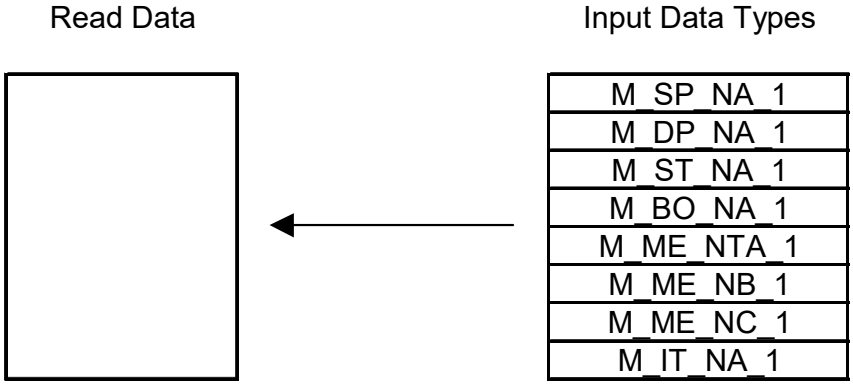
Type ID	Type	Description	Data representation
1	M_SP_NA_1 (7.3.1.1)	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value (7.2.6.1) with 0=Off and 1=On.
3	M_DP_NA_1 (7.3.1.3)	Monitored Dual-point Information: This data type stores a dual-point binary input value (that is, valve status). Associated time-tagged event information for this type are M_DP_TA_1 (4) and M_DP_TB_1 (31).	Dual-bit status (7.2.6.2) with 00b (0 decimal) = indeterminate or intermediate, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = indeterminate.
5	M_ST_NA_1 (7.3.1.5)	Monitored Step-point Information: This data type is used for step position of transformers or other step position information. The value for the position ranges from -64 to 63. Associated time-tagged event information for this type are M_ST_TA_1 (6) and M_ST_TB_1 (32).	Step data (7.2.6.5) is stored in a single character value with bits 0 to 6 (-64 to +63) representing the step position and bit 7 representing the following states: 0 = Equipment is not in transient state 1 = Equipment in transient state
7	M_BO_NA_1 (7.3.1.7)	Monitored Bitstring of 32-bit data --This data type stores 32-bit data in binary form. Each bit in the string has a value of 0 or 1. Associated time-tagged event information for this type are M_BO_TA_1 (8) and M_BO_TB_1 (33).	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).
9	M_ME_NA_1 (7.3.1.9)	Monitored Normalized Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TA_1 (10) and M_ME_TD_1 (34).	Normalized values (7.2.6.6) are stored in a word (16-bit) data area with a range of -1..+1-2^{-15}
11	M_ME_NB_1 (7.3.1.11)	Monitored Scaled Measured Value --This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TB_1 (12) and M_ME_TE_1 (35).	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
13	M_ME_NC_1 (7.3.1.13)	Monitored Measured Value, Short Floating-Point Number: This data type is used for analog input data stored in floating point format according to the IEEE STD 754, QDS format. Associated time-tagged event information for this type are M_ME_TC_1 (14) and M_ME_TE_1 (36).	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
15	M_IT_NA_1 (7.3.1.15)	Monitored Integrated Total-point Information -- This data type stores meter or other count data. Associated time-tagged event information for this type are M_IT_TA_1 (15)and M_IT_TB_1 (37).	Binary counter data (7.2.6.9) is stored in a double-word (32-bit) value with a range of -2^{31}..+2^{31}-1.
45	C_SC_NA_1 (7.3.2.1)	Single-point Command: This command controls a single binary point such as a relay.	Single bit value (7.2.6.15) with 0 = Off and 1 = On
46	C_DC_NA_1 (7.3.2.2)	Double-point Command: This command controls a dual-point binary control device such as a trip/close relay.	Double Command (7.2.6.16) with 0 = Not permitted 1 = Off 2 = On 3 = Not permitted
47	C_RC_NA_1 (7.3.2.3)	Regulating Step Command: This command controls a stepping device such as a transformer.	Regulating Step Command (7.2.6.17) with 0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted

Type ID	Type	Description	Data representation
48	C_SE_NA_1 (7.3.2.4)	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values (7.2.6.6) are stored in a word (16-bit)data area with a range of -1..+1-2^{-15}
49	C_SE_NB_1 (7.3.2.5)	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
50	C_SE_NC_1 (7.3.2.6)	Setpoint Command, Short Floating-Point Format: This command controls an analog device accepting an IEEE STD 754 floating-point format value.	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
51	C_BO_NA_1 (7.3.2.7)	Setpoint Command, 32-bit Bitstring: This command controls a bitstring in a device.	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).

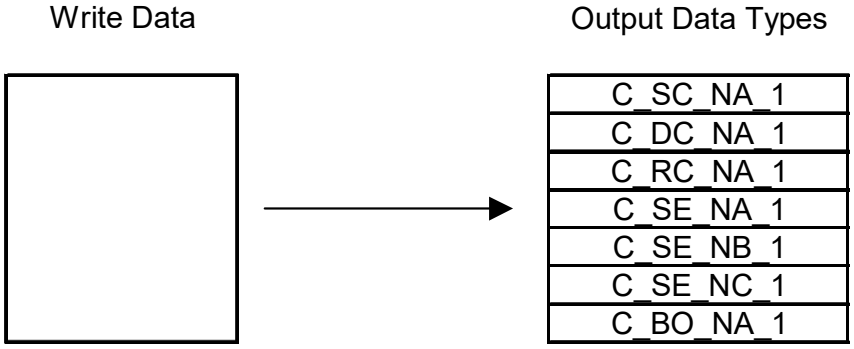
The data addressing is resumed in the following table.

Data	Size	Example
Single Point	1 bit	Address 1600 refers to word 100, bit 1 in database
Dual Point	2 bits	Address 1600 refers to word 100, bits 1 and 2 in database
Step Point	1 byte	Address 200 refers to word 100, lower byte in database
Bitstring 32 bit	2 words	Address 50 refers to word 100 and 101 in database
Normalized Measured Value	1 word	Address 100 refers to word 100 in database
Scaled Measured Value	1 word	Address 100 refers to word 100 in database
Short Float Point Measured Value	2 words	Address 50 refers to words 100 and 101 in database
Integrated Total Point	2 words	Address 50 refers to words 100 and 101 in database

Another important concept to understand is the direction of data transfer for the different data types with reference to the controller. The following illustration shows the data types (monitored data) that are transferred from the module to the processor.



The next diagram shows the movement of control data from the processor to the module. This data is then sent to the controlled devices on the serial networks.



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 PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (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 Blocks

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 ControlLogix processor. The following table describes the structure of the input image.

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2 to 201	Read Data	200
202 to 247	Error/Status Data	46
248	Spare	1
249	Read Block ID	1

The Block Identification Code (word 249) is used to signal to the ControlLogix processor that a new block is ready for processing and informs the processor of the contents of the block. If the value of the code is set to1, the block contains the first 200 words of data contained in the database of the module. Additionally, the status data contained in the block should be copied to the status data area in the module. This information can be used to determine the "health" and activity of the module.

Refer to the Reference chapter for a detailed listing of the area and its contents.

The block also contains the block identification code the module expects to receive from the processor (word 1 in the block). Under normal data transfer conditions, the ladder logic should use the code to build the appropriate block for the module in the output image.

Write Block

These blocks of data transfer information from the <Processor Model> to the module. The following table describes the structure of the output image.

Offset	Description	Length
0	Write Block ID	1
1 to 200	Write Data	200
201 to 247	Spare	47

The Write Block ID code defines the content of the data area contained in the block. If the code is set to -1 or 0, the data area contains no valid data. If the word contains a value from 1 to 20, the data contained in the block will come from the appropriate position of the module’s database. Under normal conditions, the value used for the Write Block ID should be that received in the Read Block from the module.

Special Function Blocks

Block identification codes greater than 9900 are utilized to perform special functions in the module. Each Special Function 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.
2 to 7	Command #1	Data required to build the user defined command in the command queue.
8 to 13	Command #2	Data required to build the user defined command in the command queue.
14 to 19	Command #3	Data required to build the user defined command in the command queue.
20 to 25	Command #4	Data required to build the user defined command in the command queue.
26 to 31	Command #5	Data required to build the user defined command in the command queue.
32 to 37	Command #6	Data required to build the user defined command in the command queue.
38 to 43	Command #7	Data required to build the user defined command in the command queue.
44 to 49	Command #8	Data required to build the user defined command in the command queue.
50 to 55	Command #9	Data required to build the user defined command in the command queue.
56 to 61	Command #10	Data required to build the user defined command in the command queue.
62 to 247	Spare	Not Used

The following fields are used for each 6-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	Point Index	Information object address for the point on which command operates.
5	Qualifier	Qualifier as defined for the command parameter is data type dependent. list. This

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 single-point, double-point and regulating step data points in remote units. If the persistent output qualifier is used in a command list, the command list can be utilized. For points that are controlled using the short or long pulse or no definition given qualifier, block 9901 should be used to control the output. For example, if the command list controls a short pulse output for a single point using the enable code of 2, any time the point's value in the database changes the command will be executed. When the point changes to a value of 0, the short pulse off command will be executed. When the point changes to a value of 1, the short pulse on command will be executed. This may not be what is desired for the application. If block 9901 is utilized instead, the ladder logic can execute a short pulse on command using a database point it that is set to 1.

For step control, if the database value is set to -1, the regulating step command of next lower will be executed. If the value in the database is 1, the regulating step command of next higher will be executed. Using block 9901 control of end device may be more reliable.

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 60.
2 to 61	Command Numbers to enable	These 60 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 247	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. Therefore, the command list can contain an short pulse on command. 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.

Word Offset in Block	Data Field(s)	Description
0	Reserved	
1	Block ID	This is the next block requested by the module.
2	Event Count	This field contains the number of events present in the block. Values of 1 to 17 are valid.
3 to 16	Event 1	Event message
17 to 30	Event 2	Event message
31 to 44	Event 3	Event message
45 to 58	Event 4	Event message
59 to 72	Event 5	Event message
73 to 86	Event 6	Event message
87 to 100	Event 7	Event message
101 to 114	Event 8	Event message
115 to 128	Event 9	Event message
129 to 142	Event 10	Event message
143 to 156	Event 11	Event message
157 to 170	Event 12	Event message
171 to 184	Event 13	Event message
185 to 198	Event 14	Event message
199 to 212	Event 15	Event message
213 to 226	Event 16	Event message
227 to 240	Event 17	Event message
241 to 248	Spare	Not Used
249	Block ID	This field contains the block identification code of 9903 for the block.

The format of each 14 word data region in the block is as follows:

Word Offset	Definitions	Description
0	Session Index	This field contains the session index used to define the controlled unit in the module from which the event was generated.
1	Sector Index	This field contains the sector index used to define the database within the controlled unit from which the event was generated.
2	COT	This field contains the COT for the event message received from the IED. If the size of the COT is a single byte, the originator address will always be zero. The COT is in the LSB and the originator address is in the MSB.
3	Reserved	This field is reserved for future use and is added here to keep the structure double-word aligned for all platforms.
4 to 5	Point Index	This field contains the point index in the remote device that generated the event.
6	ASDU Type	This field contains the ASDU type code for the data contained in the message.
7	Milliseconds and Seconds	This word contains the seconds and milliseconds when the event occurred.
8	Minutes and Hours	This field contains the minutes and hours the event occurred.
9	Month and Day	This field contains the month and day of the month the event occurred.
10	Year	This field contains the year the event occurred.
11	Qualifier	This field contains the point qualifier, quality or sequence value as described in the protocol specification.
12 to 13	Value	This field contains the a double word value for the point associated with the event message.

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.

Read Command Error List Block (9950)

Block 9950 identification code requests the Command List Error Table from the module for the 1000 user configurable commands. The following table describes the format of this block.

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 247	Spare	Not Used

The module will respond to a valid request with a block containing the requested error information. The following table describes the format of this block.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	This is the next block requested by the module.
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 203	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.
204 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9950 identifying the block type to the PLC.

Set Processor Time Block (9970)

Block 9970 requests the module's date and time. This data can be used to set the PLC clock.

Block Format from Processor (4x Register Data)

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the value of 9970 identifying the block type to the module.
2 to 63	Not Used	Not Used

Block Format from Module (3x Register Data)

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9970 for the block.
2	Year	This field contains the four-digit year for 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.
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 63	Not Used	Not Used

Set Module Time Block (9971)

Block identification code 9971 passes the clock time 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	Block ID	This field contains the block identification code of 9971 for the block.
1	Year	This field contains the four-digit year for 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 247	Not Used	Not Used

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 following table describes the format of the block constructed by the processor.

Offset	Description	Length
0	9998	1
1 to 247	Spare	247

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 following table describes the format of the block constructed by the processor.

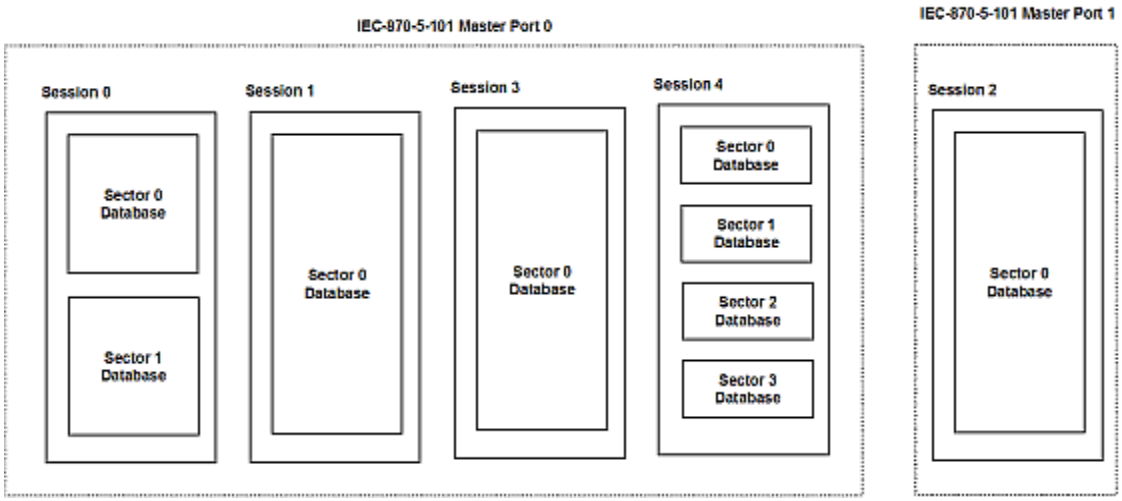
Offset	Description	Length
0	9999	1
1 to 247	Spare	247

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.

5.2.3 Master Driver

The master driver supported on each application port of the module emulates an IEC 60870-5-101 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.

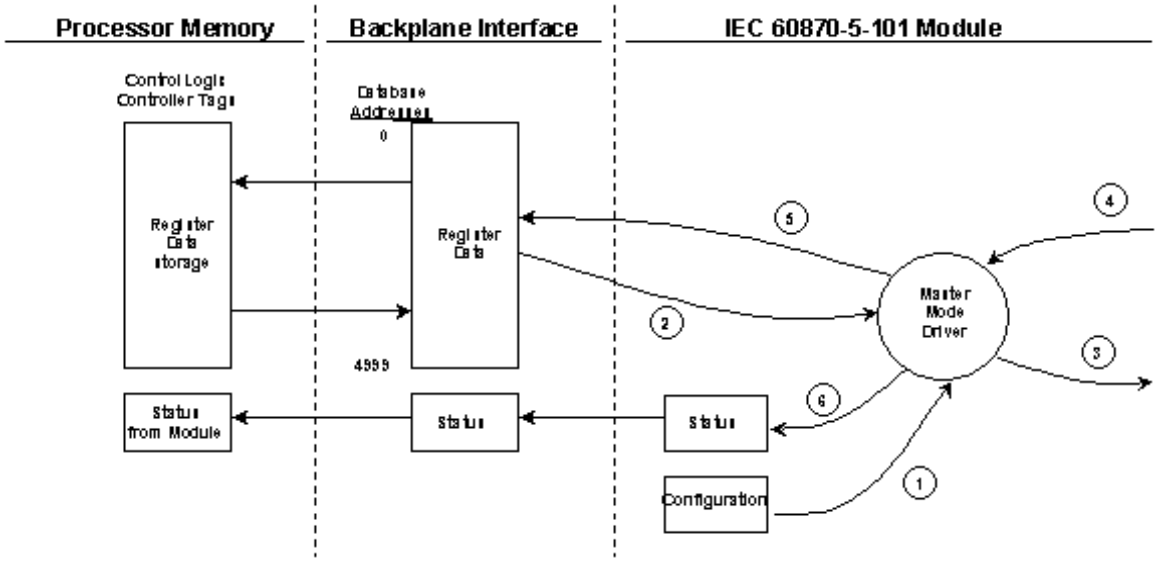


Port 0 on the module communicates with 4 sessions (0, 1, 3 and 4) each of which has their own data set(s). Session 1 only has one sector (all data for device contained in a single database). This sector is addressed by the master using the Common address of ASDU value set for the sector in the configuration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

Port 1 may operate in balanced mode as it only contains 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.

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 IEC101M.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 (static and event) 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

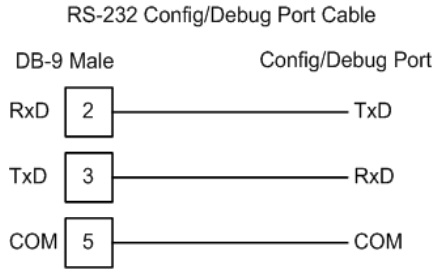
5.3 Cable Connections

The application ports on the MVI56-101M 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.

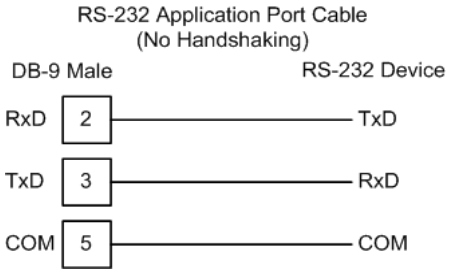
5.3.1 RS-232 Configuration/Debug Port

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:



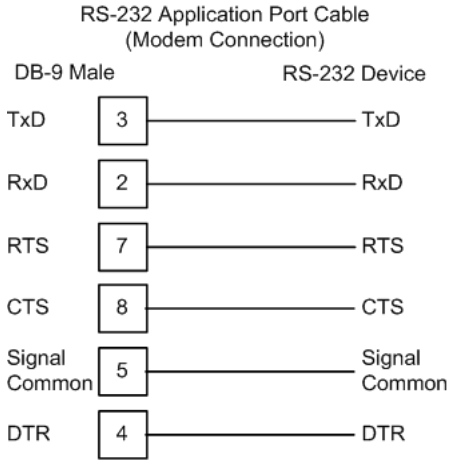
5.3.2 RS-232 Application Port(s)

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, here are the cable pinouts to connect to the port.



RS-232: Modem Connection (Hardware Handshaking Required)

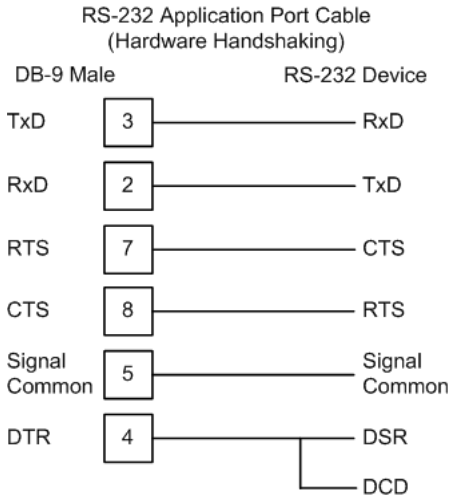
This type of connection is required between the module and a modem or other communication device.



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

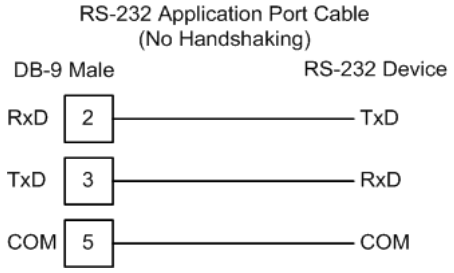
RS-232: Null Modem Connection (Hardware Handshaking)

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

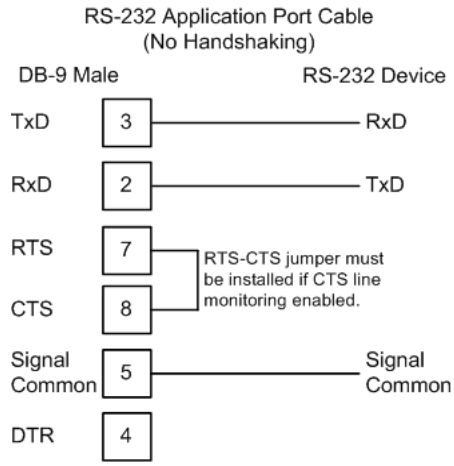


RS-232: Null Modem Connection (No Hardware Handshaking)

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

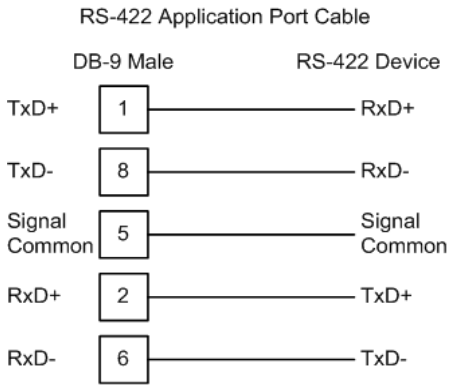


Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to **N** and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to **Y**, then a jumper is required between the RTS and the CTS lines on the port connection.



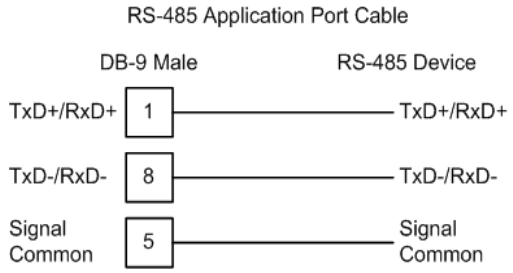
5.3.3 RS-422

The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



5.3.4 RS-485 Application Port(s)

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

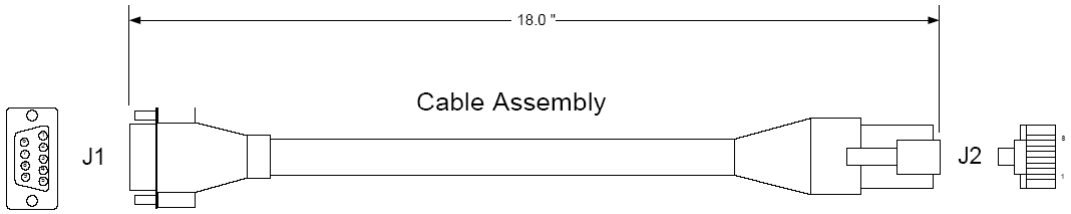


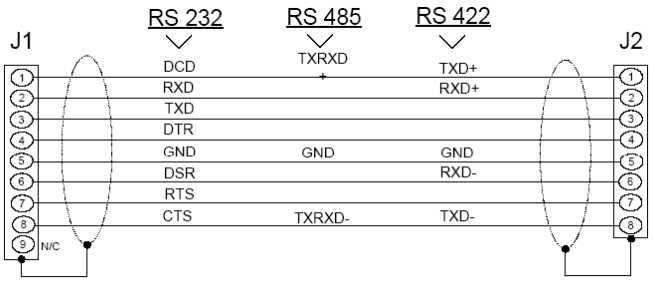
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In these cases, installing a 120-ohm terminating resistor between pins 1 and 8 on the module connector end of the RS-485 line may improve communication quality.

RS-485 and RS-422 Tip

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

5.3.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 MVI56-101M Status Data Area

This section contains a listing of the data contained in the MVI56-101M status data object, configuration error word and module error codes.

5.4.1 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 "87S5" for the PTQ 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.
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.

Offset	Parameter	Description
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.
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.

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	Invalid length data for session: ▪ Data link length ▪ Command address of ASDU length ▪ Information object address length ▪ COT octet count
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	

5.4.2 Error Codes

The following table lists all potential errors that can be generated by the IEC 60870-5-101 master driver:

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
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

Error	Description
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

Error	Description
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	**Point Index**	**Database Address**

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	Point Index	Qualifier Parameter

5.7 Interoperability

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.

The selected parameters should be crossed in the white boxes (replace "☐" with "☒").

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.

Network configuration

(network-specific parameter)

☒ Point-to-point ☒ Multipoint-party line

☒ Multiple point-to-point ☒ Multipoint-star

Physical Layer

(network-specific parameter)

Transmission speed (control direction)

Unbalanced interchange circuit V.24/V.28 Standard	Unbalanced interchange circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange circuit X.24/X.27
☐ 100 bit/s	☒ 2400 bit/s	☒ 2400 bit/s ☐ 56000 bit/s
☐ 200 bit/s	☒ 4800 bit/s	☒ 4800 bit/s ☐ 64000 bit/s
☒ 300 bit/s	☒ 9600 bit/s	☒ 9600 bit/s
☒ 600 bit/s	☒ 19200 bit/s	
☒ 1200 bit/s	☒ 38400 bit/s	

Transmission speed (monitor direction)

Unbalanced interchange circuit V.24/V.28 Standard	Unbalanced interchange circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange circuit X.24/X.27
☐ 100 bit/s	☒ 2400 bit/s	☒ 2400 bit/s ☐ 56000 bit/s

☐	200 bit/s	☒	4800 bit/s	☒	4800 bit/s	☐	64000 bit/s
☒	300 bit/s	☒	9600 bit/s	☒	9600 bit/s		
☒	600 bit/s			☒	19200 bit/s		
☒	1200 bit/s			☒	38400 bit/s		

Link Layer

(network-specific parameter)

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

Link transmission procedure **Address field of link**

☒ Balanced transmission ☒ Not present (balanced transmission only)

☒ Unbalanced transmission ☒ One octet

☒ Two octets

☒ Structured

Frame length ☒ Unstructured

255 Maximum length L (number of octets)

Application Layer

Transmission mode for application data

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

Common address of ASDU

(system-specific parameter)

☒ One octet ☒ Two octets

Information object address

(system-specific parameter)

☒ One octet ☒ structured

☒ Two octets ☒ unstructured

☒ Three octets

Cause of transmission

(system-specific parameter)

☒ One octet ☒ Two octets (with originator address)

Selection of standard ASDUs

Process information in monitor direction

(station-specific parameter)

☒ <1> := Single-point information M_SP_NA_1

☒ <2> := Single-point information with time tag M_SP_TA_1

☒ <3> := Double-point information M_DP_NA_1

☒ <4> := Double-point information with time tag M_DP_TA_1

☒ <5> := Step position information M_ST_NA_1

☒ <6> := Step position information with time tag M_ST_TA_1
☒ <7> := Bitstring of 32 bit M_BO_NA_1
☒ <8> := Bitstring of 32 bit with time tag M_BO_TA_1
☒ <9> := Measured value, normalized value M_ME_NA_1
☒ <10> := Measured value, normalized value with time tag M_ME_TA_1
☒ <11> := Measured value, scaled value M_ME_NB_1
☒ <12> := Measured value, scaled value with time tag M_ME_TB_1
☒ <13> := Measured value, short floating point value M_ME_NC_1
☒ <14> := Measured value, short floating point value with time tag M_ME_TC_1
☒ <15> := Integrated totals M_IT_NA_1
☒ <16> := Integrated totals with time tag M_IT_TA_1
☐ <17> := Event of protection equipment with time tag M_EP_TA_1
☐ <18> := Packed start events of protection equipment with time tag M_EP_TB_1
☐ <19> := Packed output circuit information of protection equipment with time tag M_EP_TC_1
☐ <20> := Packed single-point information with status change detection M_PS_NA_1
☐ <21> := Measured value, normalized value without quality descriptor M_ME_ND_1
☒ <30> := Single-point information with time tag CP56Time2a M_SP_TB_1
☒ <31> := Double-point information with time tag CP56Time2A M_DP_TB_1
☒ <32> := Step position information with time tag CP56Time2A M_ST_TB_1
☒ <33> := Bitstring of 32 bit with time tag CP56Time2A M_BO_TB_1
☒ <34> := Measured value, normalized value with time tag CP56Time2A M_ME_TD_1
☒ <35> := Measured value, scaled value with time tag CP56Time2A M_ME_TE_1
☒ <36> := Measured value, short floating point value with time tag CP56Time2A M_ME_TF_1
☒ <37> := Integrated totals with time tag CP56Time2A M_IT_TB_1
☐ <38> := Event of protection equipment with time tag CP56Time2A M_EP_TD_1
☐ <39> := Packed start events of protection equipment with time tag CP56time2A M_EP_TE_1
☐ <40> := Packed output circuit information of protection equipment with time tag CPT56Time2a M_EP_TF_1

Process information in control direction

(station-specific parameter)

☒ <45> := Single command C_SC_NA_1

☒ <46> := Double command C_DC_NA_1

☒ <47> := Regulating step command C_RC_NA_1

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

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

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

☒ <51> := Bitstring of 32 bit C_BO_NA_1

System information in monitor direction

(station-specific parameter)

☒ <70> := End of initialization M_EI_NA_1

System information in control direction

(station-specific parameter)

☒ <100> := Interrogation command C_IC_NA_1

☒ <101> := Counter interrogation command C_CI_NA_1

☒ <102> := Read command C_RD_NA_1

☒ <103> := Clock synchronization command C_CS_NA_1

☒ <104> := Test command C_TS_NB_1

☒ <105> := Reset process command C_RP_NC_1

☒ <106> := Delay acquisition command C_CD_NA_1

Parameter in control direction

(station-specific parameter)

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

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

☒⑨<112> := Parameter of measured value, short floating point value P_ME_NC_1

☒ <113> := Parameter activation P_AC_NA_1

File transfer

(station-specific parameter)

☐ <120> := File ready F_FR_NA_1

☐ <121> := Section ready F_SR_NA_1

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

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

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

☐ <125> := Segment F_SG_NA_1

☐ <126> := Directory F_DR_TA_1

Basic Application Functions

Station initialization

(station-specific parameter)

☒ Remote initialization

General Interrogation

(system- or station-specific parameter)

☒ global

☒ group 1	☒ group 7	☒ group 13
☒ group 2	☒ group 8	☒ group 14
☒ group 3	☒ group 9	☒ group 15
☒ group 4	☒ group 10	☒ group 16
☒ group 5	☒ group 11	
☒ group 6	☒ group 12	

Addresses per group have to be defined

Clock synchronization

(station-specific parameter)

☒ Clock synchronization

Command transmission

(object-specific parameter)

☒ Direct command transmission ☒ Select and execute command

☒ Direct set point command transmission ☒ Select/execute set point cmd

☒ C_SE_ACTTERM used

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation)

☒ Long pulse duration (duration determined by a system parameter in the outstation)

☒ Persistent output

Transmission of Integrated totals

(station- or object-specific parameter)

☒ Counter request	☒ General request counter
☒ Counter freeze without reset	☒ Request counter group 1
☒ Counter freeze with reset	☒ Request counter group 2
☒ Counter reset	☒ Request counter group 3
	☒ Request counter group 4

Addresses per group have to be defined

Parameter loading

(object-specific parameter)

☒ Threshold value

☒ Smoothing factor

☒ Low limit for transmission of measured value

☒ High limit for transmission of measured value

Parameter activation

(object-specific parameter)

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

File transfer

(station-specific parameter)

☐ File transfer in monitor direction

☐ File transfer in control direction

7 Support, Service & Warranty

7.1 Contacting Technical Support

ProSoft Technology, Inc. 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 associated ladder files, if any
2. Module operation and any unusual behavior
3. Configuration/Debug status information
4. LED patterns
5. Details about the serial, Ethernet or Fieldbus devices interfaced to the module, if any.

Note: *For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues.*

North America (Corporate Location)	**Europe / Middle East / Africa Regional Office**
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:
https://www.prosoft-technology.com/About-Us/Contact-Us.

7.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS please see the documents at:
www.prosoft-technology/legal