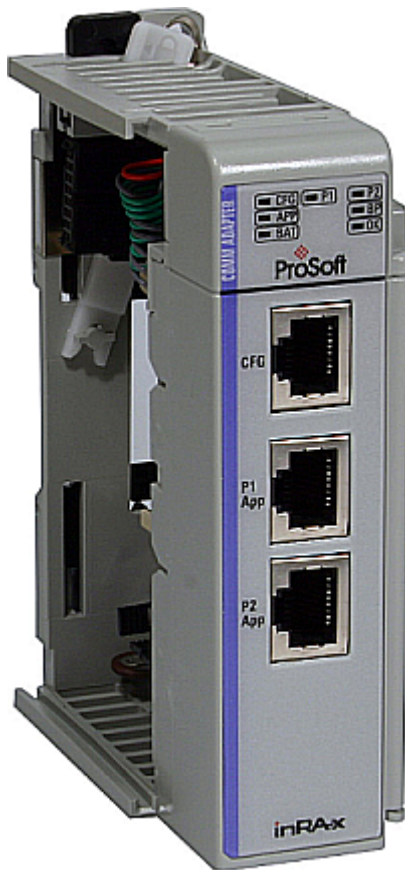




Where Automation Connects.



inRax[®]

MVI69-101S

CompactLogix or MicroLogix Platform

IEC 60870-5-101 Slave Communication
Module

3/16/2009

USER MANUAL

MVI (Multi Vendor Interface) Modules

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

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

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

Temp Code T5

II 3 G

Ex nA nL IIC T4 X

0° C ≤ Ta ≤ 60° C

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

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

G – Equipment protected against explosive gasses.

Important Installation Instructions

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

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

Warnings

North America Warnings

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

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.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: -40 to 85°C (-40 to 185°F)
- Shock: 30g Operational; 50g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings:



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

Battery Life Advisory

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

Note: The battery is not user replaceable.

Your Feedback Please

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

ProSoft Technology

1675 Chester Avenue, Fourth Floor
Bakersfield, CA 93301
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
<http://www.prosoft-technology.com>

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

MVI69-101S User Manual
3/16/2009

ProSoft Technology®, ProLinx®, inRAx®, ProTalk® and RadioLinx® are Registered Trademarks of ProSoft Technology, Inc.

ProSoft® Product Documentation

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

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

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33.5.34.36.87.20

Latin America: +1.281.298.9109

North America: +1.661.716.5100

Contents

MVI (Multi Vendor Interface) Modules	2
Important Installation Instructions	2
Warnings	2
Battery Life Advisory	3
Your Feedback Please.....	3
ProSoft® Product Documentation.....	3

Guide to the MVI69-101S User Manual **7**

1 Start Here **9**

1.1	System Requirements	9
1.2	Package Contents	10
1.3	Install ProSoft Configuration Builder Software	10
1.4	Setting Jumpers	12
1.5	Install the Module in the Rack	12
1.6	Connect your PC to the Processor.....	16
1.7	Download the Sample Program to the Processor.....	16
1.8	Connect your PC to the Module	19

2 Configuring the MVI69-101S Module **21**

2.1	ProSoft Configuration Builder.....	21
2.2	[Backplane Configuration]	26
2.3	[IEC-870-5-101 Port 0]	28
2.4	[IEC-870-5-101 Database]	35
2.5	[M_SP_NA_1].....	37
2.6	[M_DP_NA_1]	38
2.7	[M_ST_NA_1]	38
2.8	[M_ME_NA_1]	38
2.9	[M_ME_NB_1]	39
2.10	[M_ME_NC_1].....	39
2.11	[M_IT_NA_1]	40
2.12	[C_SC_NA_1]	40
2.13	[C_DC_NA_1].....	41
2.14	[C_RC_NA_1].....	41
2.15	[C_SE_NA_1]	42
2.16	[C_SE_NB_1]	42
2.17	[C_SE_NC_1].....	43
2.18	Download the Project to the Module	43

3 Ladder Logic **45**

3.1	Module Data Object: MVI69101S_ModuleDef	45
3.2	Adding the Module to an Existing CompactLogix Project	49
3.3	Adding the Module to an Existing MicroLogix Project.....	53

4	Diagnostics and Troubleshooting	57
4.1	Reading Status Data from the Module	57
4.2	LED Status Indicators	71
5	Reference	73
5.1	Product Specifications	73
5.2	Functional Overview	75
5.3	IEC-60870-5-101 Protocol Implementation	91
5.4	Cable Connections	119
5.5	MVI69-101S Status Data Definition	125
5.6	MVI69-101S Database Design Forms	126
5.7	MVI69-101S Error Status Table.....	130
5.8	IEC 60870-5-101 Slave Interoperability Document	131
6	Support, Service & Warranty	139
6.1	How to Contact Us: Technical Support.....	139
6.2	Return Material Authorization (RMA) Policies and Conditions	140
6.3	LIMITED WARRANTY	141
	Index	147

Guide to the MVI69-101S User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 71) Diagnostics and Troubleshooting (page 57)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 73) Functional Overview (page 75) Product Specifications (page 73)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 139)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- ❖ System Requirements 9
- ❖ Package Contents 10
- ❖ Install ProSoft Configuration Builder Software..... 10
- ❖ Setting Jumpers 12
- ❖ Install the Module in the Rack 12
- ❖ Connect your PC to the Processor 16
- ❖ Download the Sample Program to the Processor..... 16
- ❖ Connect your PC to the Module 19

Installing the MVI69-101S module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-101S Module and the application in which they will be used.



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

1.1 System Requirements

The MVI69-101S module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix processor, with compatible power supply and one free slot in the rack, for the MVI69-101S module. The module requires 800mA of available power.

Important: The MVI69-101S module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).

Important: For 1769-L23x processors, please make note of the following limitations.

- 1769-L23-QBFC1B = 800mA at 5Vdc (1 MVI69-101S will use all 800mA of available power. No other modules can be used with an MVI69 module connected to this processor).
- 1769-L23E-QB1B = 1000mA at 5Vdc (1 MVI69-101S will use 800mA of available power. One other module can be used on this rack provided it consumes less than 200mA at 5Vdc.
- 1769-L23E-QBFC1B = 450mA at 5Vdc (no MVI69 module can be used with this processor)
- Rockwell Automation RSLogix 5000 (CompactLogix) or RSLogix 500 (MicroLogix) programming software
- Rockwell Automation RSLinx communication software

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

1.2 Package Contents

The following components are included with your MVI69-101S module, and are all required for installation and configuration.

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

Qty.	Part Name	Part Number	Part Description
1	MVI69-101S Module	MVI69-101S	IEC 60870-5-101 Slave Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI69-101S module.

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

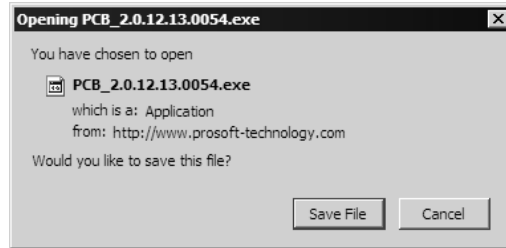
1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the MVI69-101S module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

To install ProSoft Configuration Builder from the ProSoft Web Site

- 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. The following illustrations show the file download prompt for two of the most common web browsers.



- 4 Make a note of the location where you saved the file, for example "Desktop", or "My Documents", so you can start the installation program.
- 5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install ProSoft Configuration Builder from the ProSoft Solutions CD-ROM, included in the package with your MVI69-101S module.

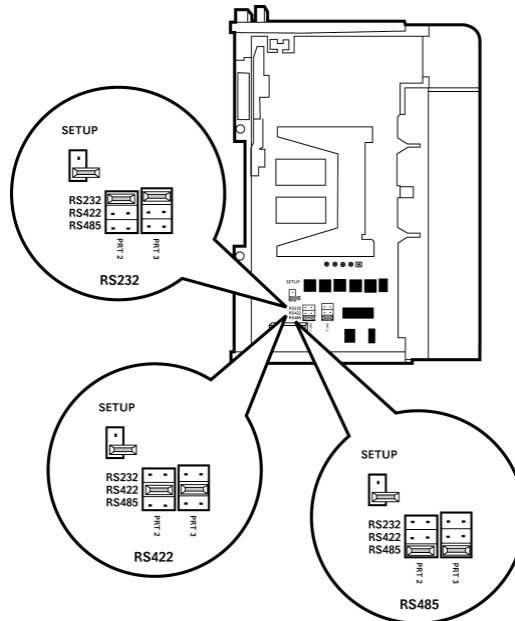
To install ProSoft Configuration Builder from the Product CD

- 1 Insert the ProSoft Solutions Product CD into the CD drive of your PC. Wait for the startup screen to appear.
- 2 On the startup screen, click *Product Documentation*. This action opens an explorer window.
- 3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.
- 4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

Note: Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the Utilities folder on the CD-ROM to a convenient location on your hard drive.

1.4 Setting Jumpers

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



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

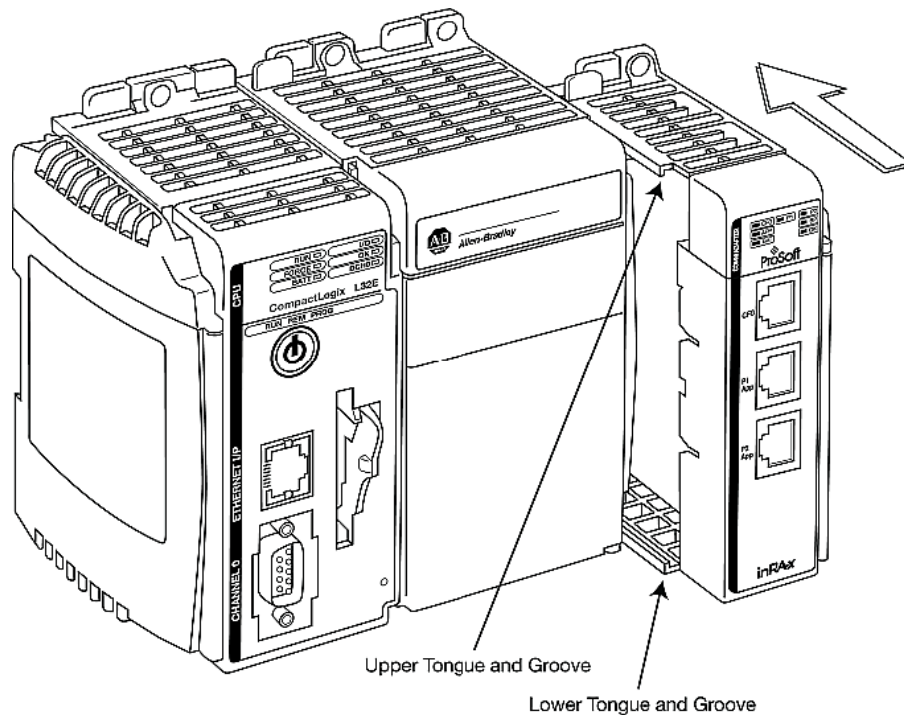
1.5 Install the Module in the Rack

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

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

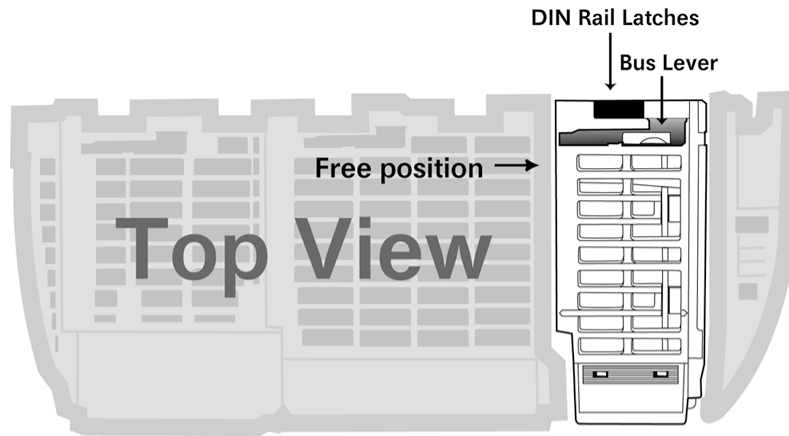
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

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

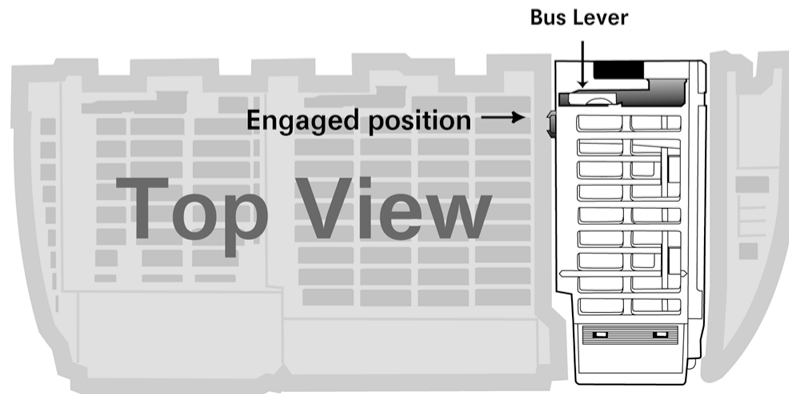


- 2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

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

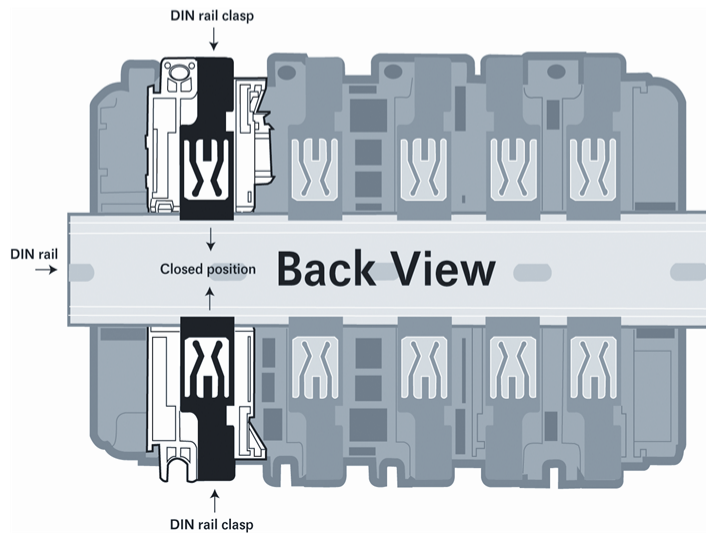
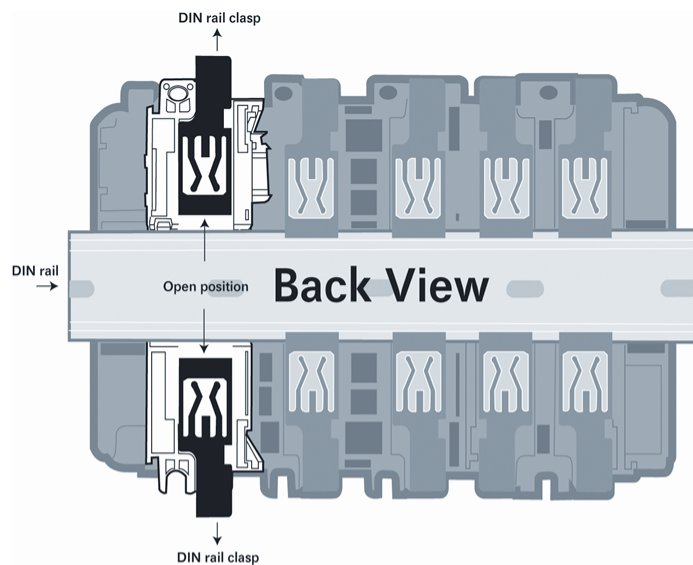


Move the Bus Lever to the left
until it clicks



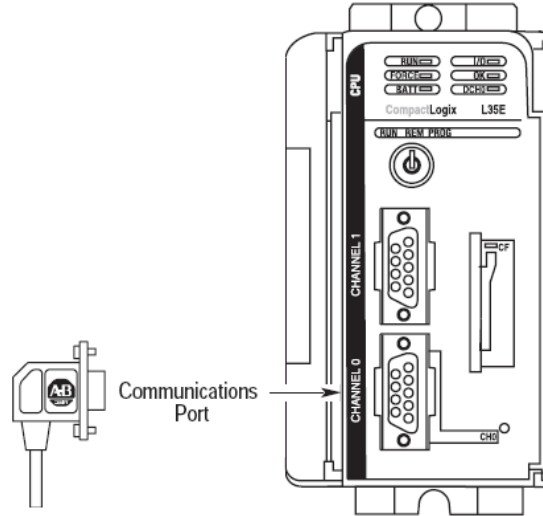
- 4 Close all DIN rail latches.

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

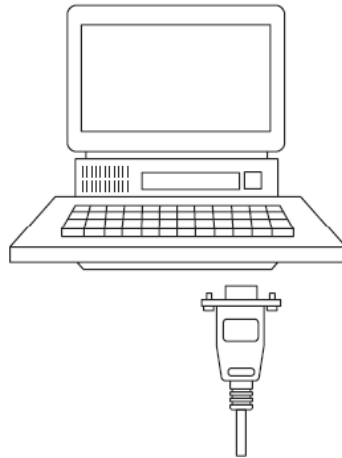


1.6 Connect your PC to the Processor

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



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



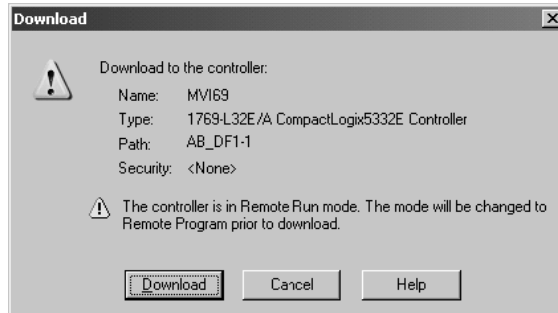
1.7 Download the Sample Program to the Processor

Important: For most applications, the sample program will work without modification.

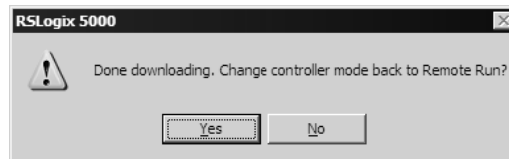
Note: The key switch on the front of the CompactLogix processor must be in the REM position.

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

- 2 When communication is established, RSLogix will open a confirmation dialog box. Click the Download button to transfer the sample program to the processor.



- 3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.
- 4 When the download is complete, RSLogix will open another confirmation dialog box. Click OK to switch the processor from Program mode to Run mode.

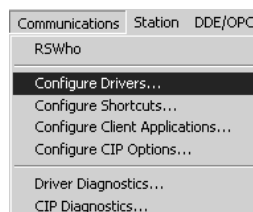


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

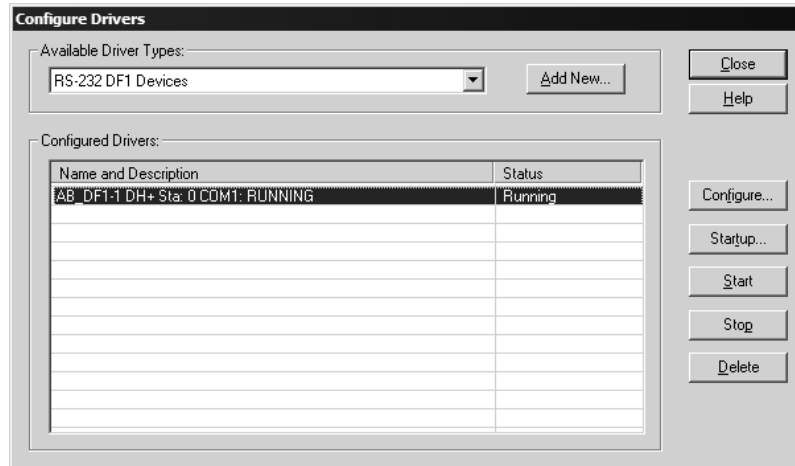
1.7.1 Configuring RSLinx

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

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



This action opens the Configure Drivers dialog box.



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

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



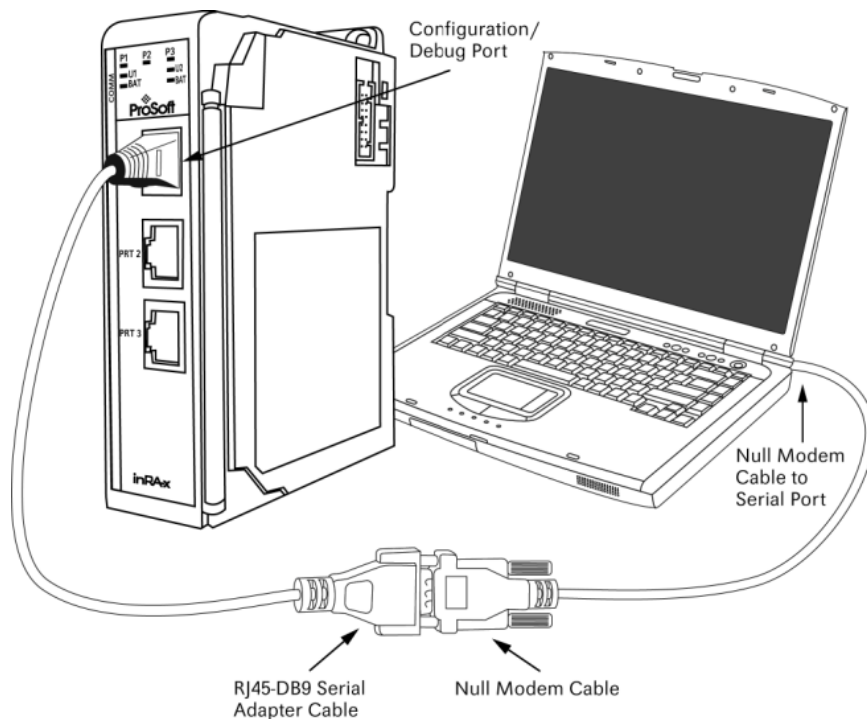
- 4 Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.
- 5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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

1.8 Connect your PC to the Module

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

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



2 Configuring the MVI69-101S Module

In This Chapter

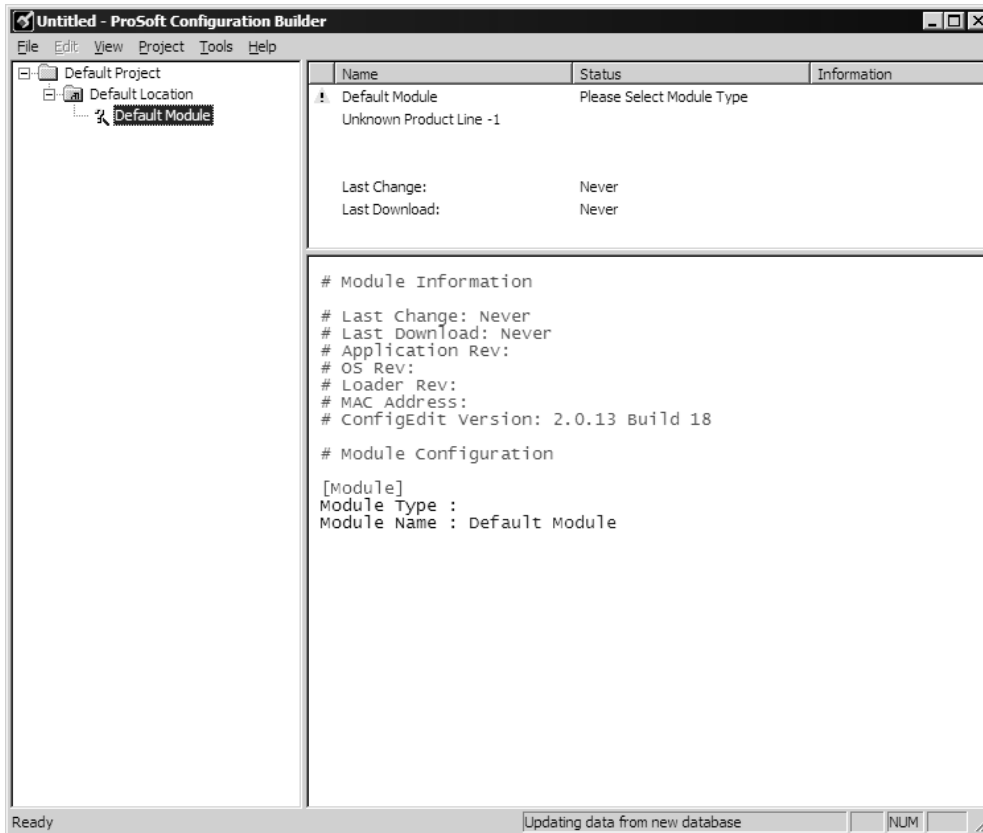
❖ ProSoft Configuration Builder	21
❖ [Backplane Configuration]	26
❖ [IEC-870-5-101 Port 0]	28
❖ [IEC-870-5-101 Database].....	35
❖ [M_SP_NA_1].....	37
❖ [M_DP_NA_1]	38
❖ [M_ST_NA_1].....	38
❖ [M_ME_NA_1].....	38
❖ [M_ME_NB_1].....	39
❖ [M_ME_NC_1].....	39
❖ [M_IT_NA_1].....	40
❖ [C_SC_NA_1].....	40
❖ [C_DC_NA_1].....	41
❖ [C_RC_NA_1].....	41
❖ [C_SE_NA_1].....	42
❖ [C_SE_NB_1].....	42
❖ [C_SE_NC_1].....	43
❖ Download the Project to the Module.....	43

2.1 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 Set Up the Project

To begin, start ProSoft Configuration Builder. If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window. When you first start ProSoft Configuration Builder, 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 ProSoft Configuration Builder window with a new project.



Your first task is to add the MVI69-101S 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 MVI69. In the Select Module Type dropdown list, select MVI69-101S, and then click OK to save your settings and return to the ProSoft Configuration Builder window.

Adding a Project

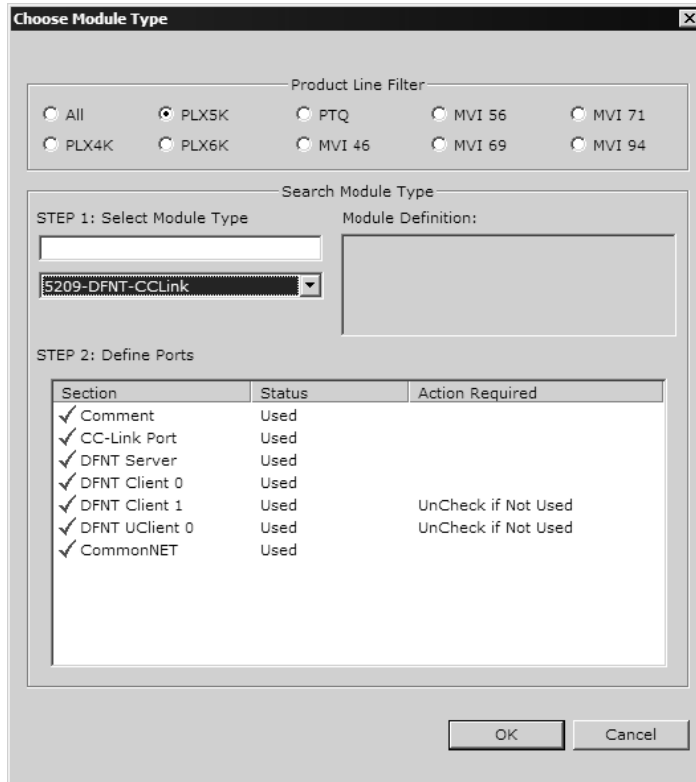
To add a project to an existing project file:

- 1 Select the Default Project icon.
- 2 Choose Project from the Project menu, then choose Add Project. A new project folder appears.

Adding a Module

To add a module to your project:

- 1 Double-click the Default Module icon to open the Choose Module Type dialog box.



- 2 On the Choose Module Type dialog box, select the module type.

Or

- 1 Open the Project menu and choose Location.
- 2 On the Location menu, choose Add Module.

To add a module to a different location:

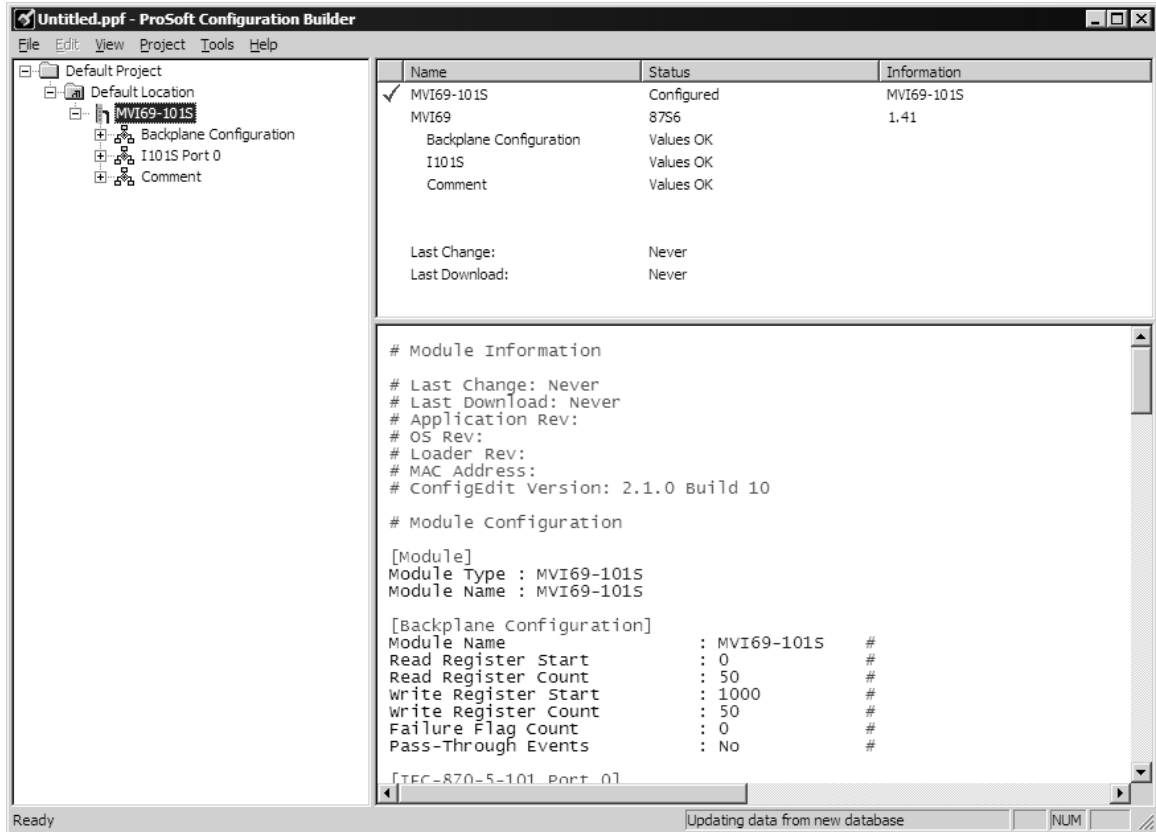
- 1 Right-click the Location folder and choose Add Module. A new module icon appears.

Or

- 1 Select the Location icon.
- 2 From the Project menu, select Location, then select Add Module.

2.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the MVI69-101S module to the project.





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

To rename an object:

- 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.



Module Entries

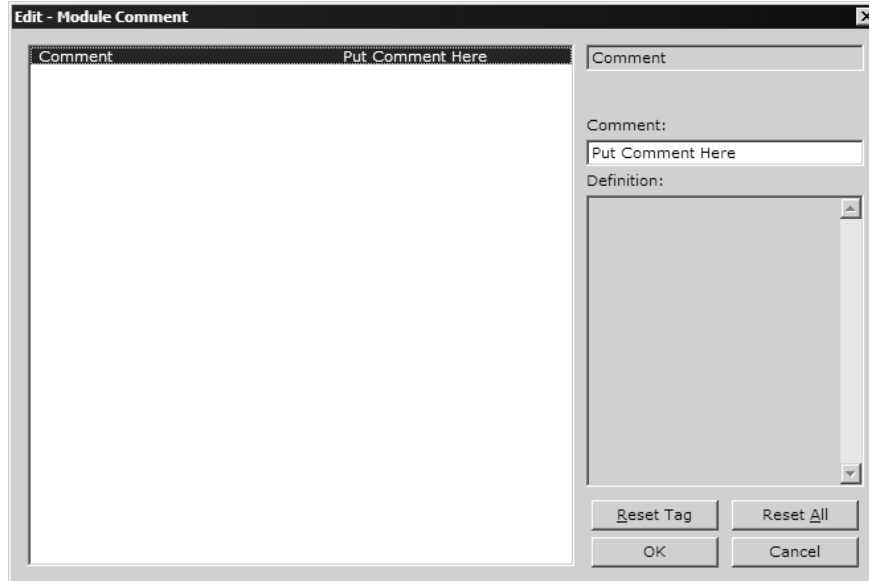
To configure module parameters

- 1 Click on the plus sign next to the icon  Comment to expand module information.
- 2 Double-click the  Module Comment icon to open the Edit dialog box.
- 3 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
- 4 Click OK to save your changes.

Comment Entries

To add comments to your configuration file:

- 1 Click the plus sign to the left of the  Comment icon to expand the Module Comments.
- 2 Double-click the  Module Comment icon. The Edit - Module Comment dialog appears.



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

Printing a Configuration File

To print 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 On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.
- 4 On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

2.2 [Backplane Configuration]

This section provides the module with:

- a unique name,
- designates database addresses for input and output on the module and on the processor,
- identifies the method of failure for the communications for the module if the PLC is not in run mode
- describes how to initialize the module upon startup.

2.2.1 Module Name

0 to 80 characters

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

2.2.2 Read Register Start

Range 0 to 3999

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

2.2.3 Read Register Count

0 to 4000

The Read Register Count parameter defines the size of the module's input database, up to a maximum value of 4000 words.

2.2.4 Write Register Start

0 to 3999

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

2.2.5 Read Register Count

0 to 4000

The Read Register Count parameter defines the size of the module's input database, up to a maximum value of 4000 words.

2.2.6 Failure Flag Count

0 through 65535

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

2.2.7 Error Offset

0 to 8899, -1 to disable

This parameter specifies the register location in the module's database where module status data will be stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 8966, the data will be placed in the modules database.

2.2.8 Block Transfer Size

60, 120 or 240

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

2.3 [IEC-870-5-101 Port 0]

This section provides information required to configure a slave application with the module. Most entries contained within this section are self-explanatory. Modify each parameter according to the needs of your application:

2.3.1 Enabled

0 = Disable port, 1 = Enable the port.

This parameter enables or disables the protocol port.

2.3.2 Time DB Offset

Time DB Offset : 2000 #DB location of IEC time

This parameter defines the location in the database where the time maintained for the IEC protocol is copied. This time is updated when ever a time synchronization command is received from the host and continually as the program runs.

2.3.3 Data Link address

Data link address : 1 #Range is 0 to 65535

This parameter defines the data link address for the device emulated on the module. This address identifies the module on the network along with the common address of ASDU. Valid values are 0 to 65535.

2.3.4 Data Link address Length

0, 1 or 2

This parameter sets the number of octets used to define the data link address for the session. A value of 0 is only permitted when balanced mode is used.

2.3.5 Common address of ASDU

Common address of ASDU : 1 #Range 0 to 65535

This parameter specifies the common address of the ASDU (section address) for access to data in the module. There is only one value entered for access to all data in the module.

2.3.6 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.

2.3.7 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.

2.3.8 Select/Operate Timeout

Select/Operate Timeout : 2000 #Milliseconds before select timeout

This parameter sets the number of milliseconds after a select command is received in which to wait for a valid execute command. If the parameter is set to 0, the feature will be disabled.

2.3.9 Cyclic Data Transmission

Cyclic data transmission : 60000 #Numb of milliseconds between cyclic
#updates

This parameter defines the number of milliseconds between cyclic updates. If the parameter is set to 0, cyclic data reporting will be disabled.

2.3.10 Use ACTTERM with Setpoint

Use ACTTERM with setpoint : 1 #1=Yes, 0=No aCTCON is last response
#to cmd

This parameter determines if an ACTTERM will be sent. If the parameter is set to Y, then setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to N, aCTCON is the last response to a setpoint command.

2.3.11 Use ACTTERM with Step

Use ACCTERM with step : 1 #1=Yes, 0=No ACTCON is last response
#to cmd

This parameter determines if an ACCTERM will be sent. If the parameter is set to Y, then step commands will issue an ACCTERM when the command is complete. If the parameter is set to N, ACTCON is the last response to a step command.

2.3.12 Single Char ACK F0, 1 or 3

Yes or No

If set to Y, 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 N, the fixed length ACK will be sent.

2.3.13 Single Char ACK C1 or C2

Single char ACK C1 or C2 : 1 #1=Single E5 resp when no C1 or C2 data

If set to Y, a single character ACK (0xE5) will be sent instead of a fixed length NACK (secondary function code 9) when no response user data is available. If set to N, the fixed length NACK will be sent.

2.3.14 Maximum ASDU Resp Len

Maximum ASDU Resp Len : 249 #Max ASDU response message length may
#have value from 25 to 252 (usually 252).

This parameter limits the maximum size of the ASDU portion of a response message. Most applications will use a value of 252.

2.3.15 Cause of Trans Octets

Cause of Trans Octets : 2 #Number of COT octets (1 or 2)

This parameter sets the COT length to 1 or 2. The second octet stores the originator address passed in the register. Spontaneous and cyclic data will always respond with the originator address set to 0.

2.3.16 Freeze Start Type

D=Day, H=Hour, M=Minute, N=Not used

The Freeze Start Type parameter defines when the module starts sending the M_IT messages.

2.3.17 Interval for Freeze

0 to 65535

Freeze Start Type and Interval for Freeze are used if Mode A operation is to be used for the counter freeze operation. If they are not used, the module will operate in Mode D.

2.3.18 Set Priority Queues

Yes or No

This section defines priority queues for the module. The data types that can return events can be assigned priorities so that events of data types will be returned before other data types. This may cause events to be lost as the event buffers for low priority queues may overflow. If this feature is utilized, each data type must be assigned a unique index from 0 to 6. The lower the index, the higher the priority (0=highest priority).

M_SP_Na Priority	:	1	#Unique index for this data type in queue (0-5)
M_DP_Na Priority	:	0	#Unique index for this data type in queue (0-5)
M_ST_Na Priority	:	5	#Unique index for this data type in queue (0-5)
M_ME_Na Priority	:	4	#Unique index for this data type in queue (0-5)
M_ME_NB Priority	:	3	#Unique index for this data type in queue (0-5)
M_ME_NC Priority	:	2	#Unique index for this data type in queue (0-5)
M_IT_Na Priority	:	6	#Unique index for this data type in queue (0-5)

Each of the ASDUs affected by this feature must be assigned a unique priority index from 0 to 6. Events of the ASDU with a priority of 0 will always be reported before any others when they are present.

For more information, refer to Event Priority (page 117).

2.3.19 Cyclic Set IV Time

Minimum 3 times larger than IV Check Delay Time parameter

0 to disable

The Cyclic Set IV Time parameter determines how frequently the IV Checks will be performed. If the IV bit is ON for a number of times given by the IV Fail Count (page 32) parameter, the module will consider the point as invalid.

If the IV Fail Count parameter is zero, the feature will be disabled.

For more information on using this feature, refer to Using Monitor Points (page 97).

2.3.20 IV Check Delay Time

Value in Seconds

This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. The Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

2.3.21 IV Fail Count

0 to disable

This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. If you enable this feature, the processor can determine the individual IV quality bit status of each point you configured.

To disable this feature, set the IV Fail Count parameter to 0. If used, the Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

2.3.22 Event Scan Delay

1 to 65535

0 to disable

If set to 0, the feature will be disabled and the module will not generate any events. If set from 1 to 65535, the parameter represents the number of milliseconds between event scanning. This parameter defines how often the program will scan for new events in the databases.

2.3.23 Scan Events

Scan for Events

No Scanning

Determines if events of this point type will be generated by the module. If "No Scanning", then events will not be generated. If "Scan for events", events will be scanned and generated on change.

2.3.24 Time Type

None, CP24 or CP56

This parameter defines the time format used with data events. 0=None, 1=CP24 and 2=CP56 time formats.

2.3.25 Use Balanced Mode

Yes or 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. Select Yes to use balanced mode and No to use unbalanced mode.

2.3.26 Retry Count

0 to 255

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

2.3.27 Response Timeout

0 to 65535

This parameter specifies the minimum number of milliseconds to wait for a response to a primary message. Do not set this parameter too small or timeout conditions may prevent successful data transmission. If the timeout is recognized, the message will be retransmitted up to the number of times specified in the Retry Count parameter. This parameter is only used in balance mode.

2.3.28 Baud Rate

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

2.3.29 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.

2.3.30 RTS On

0 to 65535 milliseconds

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

2.3.31 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.

2.3.32 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.

2.3.33 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.

2.3.34 Backup Port Enabled

Yes or No

This parameter enables or disables the backup port (Port 2).

2.3.35 Backup Port Baud Rate

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

2.3.36 Backup Port 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.

2.3.37 Backup Port RTS On

0 to 65535 milliseconds

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

2.3.38 Backup Port 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.

2.3.39 Backup Port Min 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.

2.3.40 Backup Port Handshaking

None, RTS/CTS, DTR/DSR, Modem

Selects the hardware handshaking type. Typically RTS/CTS, DTR/DSR and Modem options are used for modems. RTS/CTS and DTR/DSR are normally used for leased modems. Modem can be used also for dial-up modems. When the messages must go out if the modem is not connected, the module will assert the DTR line which will cause the modem to dial the number in the modem's configuration. The module does not have the functionality to dial the number to be called for the modem. The module reconnects after 30 seconds of inactivity.

2.4 [IEC-870-5-101 Database]

This section describes the [IEC-870-5-101 Database] Section.

2.4.1 Short Pulse Time

Short Pulse Time : 2000 #MSec for short pulse command

This parameter defines the number of milliseconds to be associated with a short pulse command.

2.4.2 Long Pulse Time

Long Pulse Time : 10000 #MSec for long pulse command

This parameter defines the number of milliseconds to be associated with a long pulse command.

2.4.3 Point Count

Point Count configuration ranges in the following configuration items are based on the assumption that you will be using only one of the available data types for your application. The number of point counts you configure will have an effect on module performance, in particular the accuracy of the module's internal clock.

M_SP_Na point count : 10 #Number of monitored single-points

This parameter specifies the number of point values assigned in monitored single-point database. Range is 0 to 1000.

M_DP_NA point count : 10 #Number of monitored dual-points

This parameter specifies the number of point values assigned in monitored dual-point database. Range is 0 to 1000.

M_ST_NA point count : 10 #Number of monitored step-points

This parameter specifies the number of point values assigned in monitored step-point database. Range is 0 to 1000.

M_ME_NA point count : 10 #Number of monitored normalized-points

This parameter specifies the number of point values assigned in monitored normalized-point database. Range is 0 to 1000.

M_ME_NB point count : 10 #Number of monitored scaled-points

This parameter specifies the number of point values assigned in monitored scaled-point database. Range is 0 to 1000.

M_ME_NC point count : 10 #Number of monitored scaled-points

This parameter specifies the number of point values assigned in monitored scaled short-float point database. Range is 0 to 50.

M_IT_NA point count : 10 #Number of monitored counter-points

This parameter specifies the number of point values assigned in monitored counter-point database. Range is 0 to 99.

C_SC_NA point count : 10 #Number of command single-points

This parameter specifies the number of point values assigned in command single-point database. Range is 0 to 1000.

C_DC_NA point count : 10 #Number of command dual-points

This parameter specifies the number of point values assigned in command dual-point database. Range is 0 to 1000.

C_RC_NA point count : 10 #Number of command step-points

This parameter specifies the number of point values assigned in command step-point database. Range is 0 to 1000.

C_SE_NA point count : 10 #Number of command normalized-points

This parameter specifies the number of point values assigned in command normalized-point database. Range is 0 to 1000.

C_SE_NB point count : 10 #Number of command scaled-points

This parameter specifies the number of point values assigned in command scaled-point database. Range is 0 to 1000.

C_SE_NC point count : 5 #Number of command short-float points

This parameter specifies the number of point values assigned in command short-float point database. Range is 0 to 50.

2.4.4 Sequence Flag

M_SP_Na Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

M_DP_Na Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

M_ME_Na Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

M_ME_NB Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

M_ME_NC Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

M_IT_Na Sequence : Yes #Y=ASDU in sequence with SQ=1, N=report separate (SQ=0)

In order to save bandwidth, you can configure the module to use the Sequence Flag feature. If this feature is not selected, the module will send the object address and its value at every monitored response to the master.

If this parameter is selected, the module will turn the Sequence Flag on every monitored response sending the address for the first point along with all point values. The master assumes that all other points use information object addresses in a contiguous order (using the first point as the reference).

2.4.5 Parameter Offset

```
M_ME_Na Parameter Offset : 2000 #M_ME_Na IOa offset for parameter data
M_ME_NB Parameter Offset : 2000 #M_ME_NB IOa offset for parameter data
M_ME_NC Parameter Offset : 2000 #M_ME_NC IOa offset for parameter data
```

This parameter specifies the IOa offset to the parameter data for the normalized parameter data. The value entered is added to the Information Object address for the associated point to compute the parameter IOa address.

The Master may send a "Parameter of Measured Normalized" or "Parameter of Measured Scaled" command using the parameter IOa in order to change the deadband values for specific points.

Note: The Low Limit and High Limit values are always calculated based on the deadband value as described in the following table.

Point	Value
Threshold	Determined by the deadband set in the configuration file or altered by the write command.
Low	Last reported event value - threshold.
High	Last reported event value + threshold.

For example, if the MVI69-101S configuration sets two M_ME_Na points with IOa (Point #) of 600 and 601 and a M_ME_Na Parameter Offset value of 3000, it would result in the parameter points shown in the table.

2.5 [M_SP_NA_1]

This section defines the monitored single-point database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies 1 bit. (1 = On, 0 = Off state).

This section takes the following parameters:

- Point #
- DB address
- Groups
- IV DB Bit

Each point is one bit and the DB address value corresponds to the bit offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.6 [M_DP_NA_1]

This section defines the monitored dual-point database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies two bits. (00 = intermediate, 01 = off, 10 = on and 11 = intermediate).

This section takes the following parameters:

- Point #
- DB address
- Groups
- IV DB Bit

Each point is two bits and the DB address value corresponds to the bit offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.7 [M_ST_NA_1]

This section defines the monitored step database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point in the database occupies one byte.

This section takes the following parameters:

- Point #
- DB address
- Groups
- IV DB Bit

Each point is one byte and the DB address value corresponds to the byte offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.8 [M_ME_NA_1]

This section defines the monitored measured value, normalized database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies a word position in the database. The IOa for the parameters are for each object and are determined by adding the Point # in the following section to the value of the M_ME_Na parameter offset parameter set in the previous section.

This section takes the following parameters:

- Point #
- DB address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB address value corresponds to the word offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.9 [M_ME_NB_1]

This section defines the monitored measured value, scaled database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies a word position in the database. The IOa for the parameters for each object are determined by adding the Point # in the following section to the value of the M_ME_NB parameter offset parameter set in the previous section.

This section takes the following parameters:

- Point #
- DB address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB address value corresponds to the word offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.10 [M_ME_NC_1]

This section defines the monitored short-float point database for the slave device emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies 4-byte positions in the database. The IOa for the parameters for each object are determined by adding the Point # in the following section to the value of the M_ME_NC Parameter Offset parameter set in the previous section.

This section takes the following parameters:

- Point #
- DB address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB address value corresponds to the word offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.11 [M_IT_NA_1]

This section defines the monitored integrated totals (counter) database for the slave emulated. This information is sourced from the database and is transferred to the remote master unit. Each point occupies two words in the database (4 bytes).

This section takes the following parameters:

- Point #
- DB address
- Groups
- IV DB Bit

Each point is two words and the DB address value corresponds to the double-word offset in the database.

The Database address value should be located in a database area that is constantly being moved from the CompactLogix or MicroLogix processor to the MVI69-101S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes section for a listing of Group Codes.

2.12 [C_SC_NA_1]

This section defines the single point command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a single bit position in the database. You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB address values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB address
- Monitor Point #
- Monitor DB addr
- Require Select

Each point is one bit and the DB address value corresponds to the bit offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

2.13 [C_DC_NA_1]

This section defines the double point command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies two bits in the database. You can associate a command with a monitored double point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB addr values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB address
- Monitor Point #
- Monitor DB addr
- Require Select

Each point is two bits and the DB address value corresponds to the bit offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

2.14 [C_RC_NA_1]

This section defines the step command database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a byte in the database. The control value can be associated with a monitored point as described in the previous example.

This section takes the following parameters:

- Point #
- DB address

- Monitor Point #
- Monitor DB addr

Each point is one byte and the DB address value corresponds to the byte offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

2.15 [C_SE_NA_1]

This section defines the normalized setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a word position in the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB address
- Monitor Point #
- Monitor DB addr
- Require Select

Each point is one word and the DB address value corresponds to the word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

2.16 [C_SE_NB_1]

This section defines the scaled setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB address
- Monitor Point #
- Monitor DB addr
- Require Select

Each point is one word and the DB address value corresponds to the word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

2.17 [C_SE_NC_1]

This section defines the short-float setpoint database for the slave emulated. This information is sourced from the remote master and is transferred to the database. Each point occupies a double-word position in the database. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB address
- Monitor Point #
- Monitor DB addr
- Require Select

Each point is two words and the DB address value corresponds to the double-word offset in the database.

The database address should be located in a database area that is being constantly moved from the MVI69-101S to the CompactLogix or MicroLogix processor. You should configure the DB address parameter in an area that is on a Backplane Command Function 2.

Note:

Monitor Point # and Monitor DB Addr

You can also associate a monitor point with each command point. Therefore, every time the module responds to a command, it will include the monitor point with the information object address given by the Monitor Point # parameter. Its value will be copied from the database address location given by the Monitor DB Address parameter.

Require Select

This parameter specifies if the point requires a "select" before an "Operation" command.

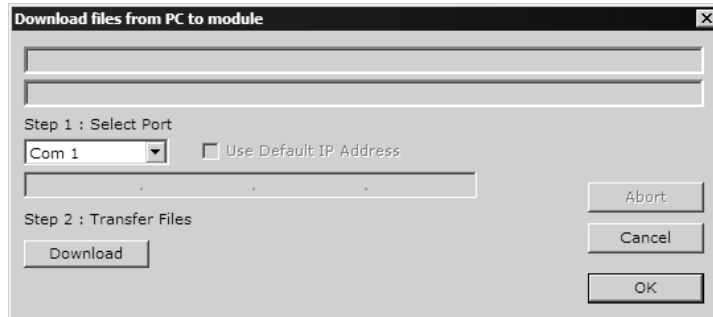
2.18 Download the Project to the Module

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

To Download the Project File

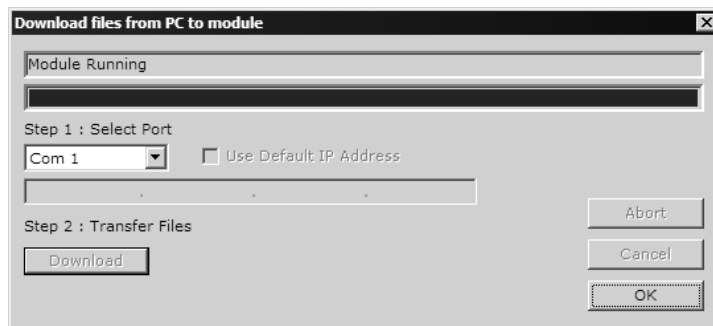
- 1 In the tree view in ProSoft Configuration Builder, click once to select the MVI69-101S module.

- 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 following 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 ProSoft Configuration Builder will be updated with the message *"Module Running"*.



3 Ladder Logic

In This Chapter

- ❖ Module Data Object: MVI69101S_ModuleDef..... 45
- ❖ Adding the Module to an Existing CompactLogix Project 49
- ❖ Adding the Module to an Existing MicroLogix Project..... 53

Ladder logic is required for application of the MVI69-101S module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt.

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

3.1 Module Data Object: MVI69101S_ModuleDef

All data related to the MVI69-101S 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.

The structure of the object is displayed in the following table.

Name	Data Type	Description
DATA	MV69101S_DATA	
CONTROL	MVI69101S_CONTROL	
STATUS	MVI69101S_STATUS	
UTIL	MVI69101S_UTIL	

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

3.1.1 User Data Objects - MV69101S_DATA

These objects hold data to be transferred between the processor and the MVI69-101S module. The user data is the read and write data transferred between the processor and the module as "pages" of data. The size of each page will depend on the Block Transfer Size parameter value (60, 120 or 240 words). The sample ladder logic reserves 10 points for each data type to illustrate how to copy data using the ReadData and WriteData arrays. The array size must be re-dimensioned for each application.

Name	Data Type	Description
ReadData	INT[480]	Read buffer where data is initially copied from the 101S (command) data types and then copied from the buffer to the module
WriteData	INT[480]	Write buffer where (monitored) data is initially copied from the module and then parsed to one of the 101S data types
M_SP_NA	BOOL[32]	Monitored Single Points
M_DP_NA	INT[10]	Monitored Double Points
M_ST_NA	SINT[10]	Monitored Step-Points
M_ME_NA	INT[10]	Monitored Normalized Measured Points
M_ME_NB	INT[10]	Monitored Scaled Measured Points
M_ME_NC	REAL[10]	Monitored Floating Point Measured Points
M_IT_NA	DINT[10]	Monitored Integrated Total
C_SC_NA	BOOL[32]	Controlled (Command) Single Point
C_DC_NA	INT[10]	Controlled (Command) Double Point
C_RC_NA	SINT[10]	Controlled (Command) Step point
C_SE_NA	INT[10]	Controlled (Command) Set-point Normalized value
C_SE_NB	INT[10]	Controlled (Command) Set-Point Scaled Value
C_SE_NC	REAL[10]	Controlled (Command) Set-Point Short Floating Point Number

The read data (**ReadData**) is an array set to match the value entered in the **Read Register Count** parameter of the 101S.CFG file. For ease of use, this array should be dimensioned as an even increment of the value configured through the block transfer size parameter (60, 120 or 240 words). This data is paged up to 60, 120 or 240 words at a time from the module to the processor. The ReadData task places the data received into the proper position in the read data array. The ReadData array contains the control points received from the controlling station.

The write data (**WriteData**) is an array set to match the value entered in the **Write Register Count** parameter of the 101S.CFG file. For ease of use, this array should be dimensioned as an even increment of the value configured through the block transfer size parameter (60, 120 or 240 words). This data is paged up to 60, 120 or 240 words at a time from the processor to the module. The WriteData task places the write data into the output image for transfer to the module. The WriteData array contains the monitor points to be sent to the controlling station.

3.1.2 MVI69101S_CONTROL

The CONTROL tag contains special tags for special tasks to be requested to the module.

Name	Data Type	Description
Event	MVI69101S_EventBlock	Block 9958: Event message
SetTime	MVI69101S_Clock	Block 9971: Writes the clock data to the module
GetTime	MVI69101S_Clock	Block 9970: Reads the clock data from the module

MVI69101S_EventBlock

Name	Data Type	Description
Activate	BOOL	Set this bit to 1 to activate event block 9958 to send events to the module
EventCount	INT	Number of events to be sent at this block (between 1 and 3)
Event	MVI69101S_Event[3]	

MVI69101S_Event

The Event tag describes the format of each event to be transferred from the processor to the module.

Name	Data Type	Description
DBIndex	INT	Index of the point in configuration file
ASDU	INT	1=SP, 3=DP, 5=STEP,9=NORM, 11=SCALED, 15=INTEGRATED
Qualifier	INT	Qualifier Code
Year	INT	Year of timestamp
Month	INT	Month of timestamp
Day	INT	Day of timestamp
Hour	INT	Hour of timestamp
Minute	INT	Minute of timestamp
Seconds_Milliseconds	INT	For example, a value of 15742 means 15secs and 742 milliseconds
Data	INT[6]	Data associated to event: data size will depend on point type - check documentation

MVI69101S_Clock

The Clock tag contains the format to read and write the module's date and time information.

Name	Data Type	Description
Activate	BOOL	
Year	INT	
Month	INT	

Name	Data Type	Description
Day	INT	
Hour	INT	
Minute	INT	
Second	INT	
Milliseconds	INT	

3.1.3 Module Status Object: MVI69101S_STATUS

This object views the status of the module. The MVI69101S_STATUS object shown below is updated each time a read block is received by the processor. Use this data to monitor the state of the module at a "real-time rate".

Name	Data Type	Description
ProgramScanCounter	INT	Program cycle counter for module
ProductCode	INT[2]	Product code for module
ProductVersion	INT[2]	Revision level of module's code
OperatingSystem	INT[2]	Operating system version of module
RunNumber	INT[2]	Run number for module
Backplane_Read_Count	INT	Incremented at every backplane read
Backplane_Write_Count	INT	Incremented at every backplane write
Backplane_Parse_Count	INT	Incremented at every backplane parse
Backplane_Error_Count	INT	Incremented at every backplane error
PortSelect	INT	Port selected (0 or 1)
BadCRC	INT	CRC errors
SyncError	INT	Synchronization errors
LengthError	INT	Message length errors
Timeouts	INT	Message timeouts
RX_Frames	INT	Number of messages received
TX_Frames	INT	Number of messages transmitted
M_SP_NA_EventBufferFree	INT	Number of available events in event buffer
M_DP_NA_EventBufferFree	INT	Number of available events in event buffer
M_ST_NA_EventBufferFree	INT	Number of available events in event buffer
M_ME_NA_EventBufferFree	INT	Number of available events in event buffer
M_ME_NB_EventBufferFree	INT	Number of available events in event buffer
M_ME_NC_EventBufferFree	INT	Number of available events in event buffer
M_ME_IT_EventBufferFree	INT	Number of available events in event buffer

3.1.4 Block Control Data - MVI69101S_UTIL

This data object stores the variables required for the data transfer between the processor and the MVI69-101S module. The structure of the object is shown in the following table.

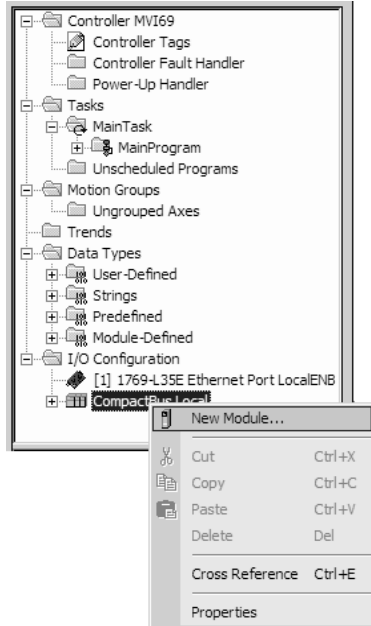
Name	Data Type	Description
Warmboot	BOOL	Requests warmboot of the module
ColdBoot	BOOL	Requests coldboot of the module
BPLastRead	INT	Index of last read
BPLastWrite	INT	Index of last write
BlockIndex	INT	Computed block offset for data table
BootTimer	TIMER	Used to clear output block ID after power up
LoopIndex	INT	Used as index during Loops
BlockTransferSize	INT	Must match the configuration parameter Block Transfer (60, 120, 240). This value must be set in ladder by the user

3.2 Adding the Module to an Existing CompactLogix Project

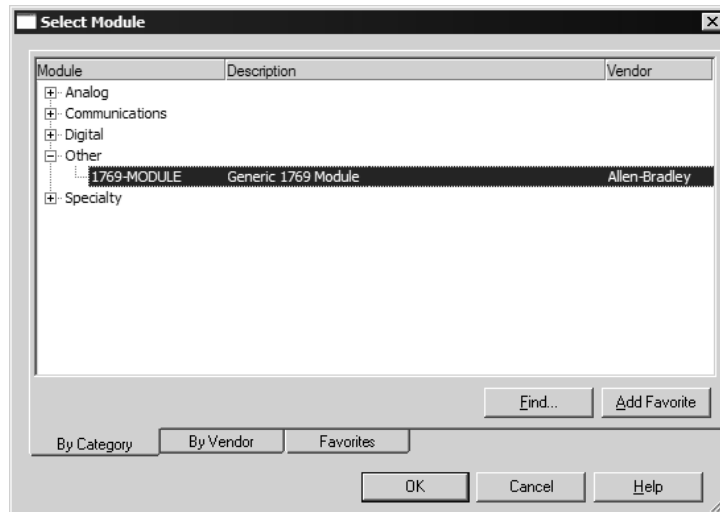
Important: The MVI69-101S module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

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

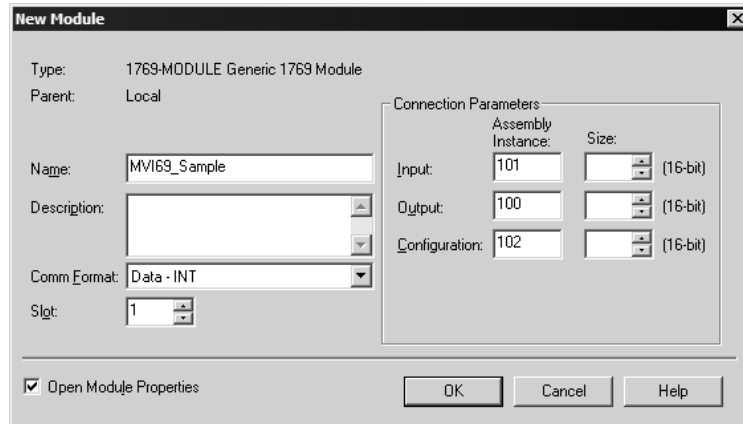
- 1 **Add the MVI69-101S module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



This action opens the following dialog box:



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



- Enter the Name, Description and Slot options for your application, using the values in the illustration above. You must select the **Comm Format** as **Data - INT** in the dialog box, otherwise the module will not communicate over the backplane of the CompactLogix rack.
- Configure the Connection Parameters to match to the Block Transfer Size parameter in the configuration file. Use the values in the table corresponding with the block transfer size you configured.

Block Transfer Size = 60

Field	Recommended Value
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	62
Output Assembly Instance	100
Output Size	61
Configuration Assembly Instance	102
Configuration Size	0

Block Transfer Size = 120

Field	Recommended Value
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed

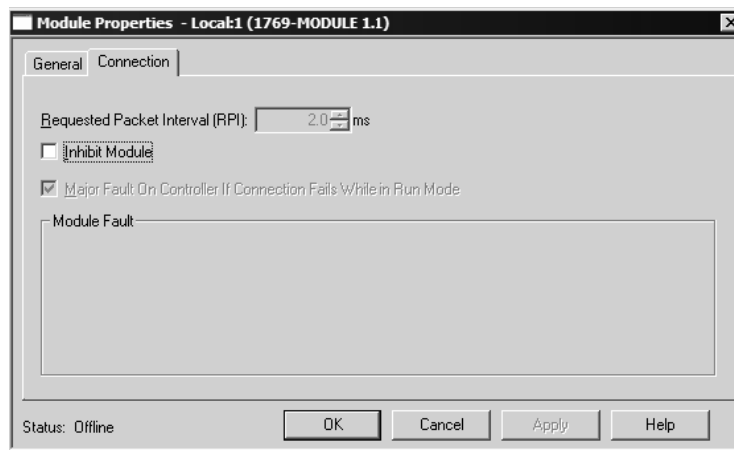
Block Transfer Size = 120

Field	Recommended Value
Input Assembly Instance	101
Input Size	122
Output Assembly Instance	100
Output Size	121
Configuration Assembly Instance	102
Configuration Size	0

Block Transfer Size = 240

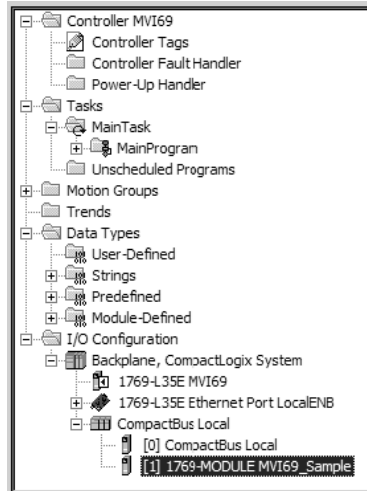
Field	Recommended Value
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	242
Output Assembly Instance	100
Output Size	241
Configuration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

5 Click **Next** to continue.



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

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



- 8 Copy the Controller Tags from the sample program.
- 9 Copy the User Defined Data Types from the sample program.
- 10 Copy the Ladder Rungs from the sample program.
- 11 Save and Download the new application to the controller and place the processor in run mode.

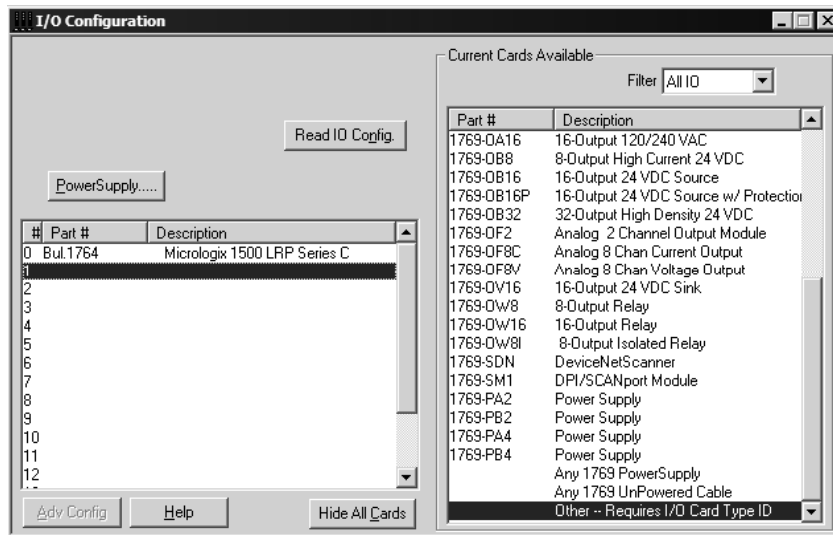
3.3 Adding the Module to an Existing MicroLogix Project

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

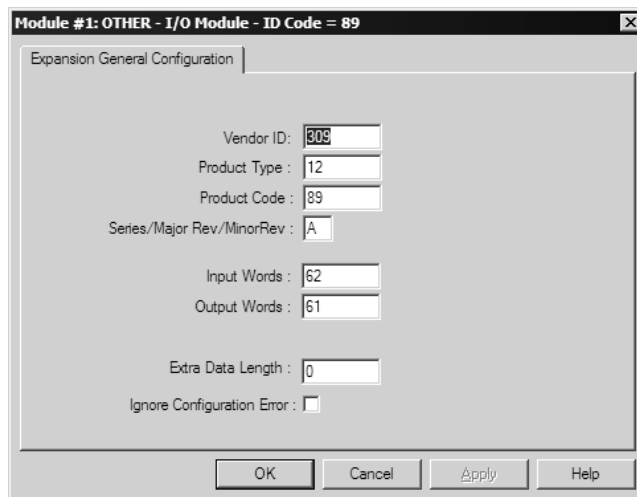
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

- 1 In RSLogix, open your existing application, or start a new application, depending on your requirements.

- 2 Double-click the I/O Configuration icon located in the Controller folder in the project tree. This action opens the I/O Configuration dialog box.



- 3 On the I/O Configuration dialog box, select "Other - Requires I/O Card Type ID" at the bottom of the list in the right pane, and then double-click to open the Module dialog box.
- 4 Enter the values shown in the following illustration to define the module correctly for the MicroLogix processor, and then click OK to save your configuration.



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

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

- 5 Click **Next** to continue.
- 6 After completing the module setup, the I/O configuration dialog box will display the module's presence.

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

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 57) for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

4 Diagnostics and Troubleshooting

In This Chapter

- ❖ Reading Status Data from the Module 57
- ❖ LED Status Indicators..... 71

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

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

4.1 Reading Status Data from the Module

The MVI69-101S module returns a 20-word Status Data block that can be used to determine the module's operating status. This data can be located in the module's database at registers at the location specified in the configuration. This data is transferred to the CompactLogix or MicroLogix processor continuously with each read block.

4.1.1 Required Hardware

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

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

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

4.1.2 The Configuration/Debug Menu

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

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the 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.

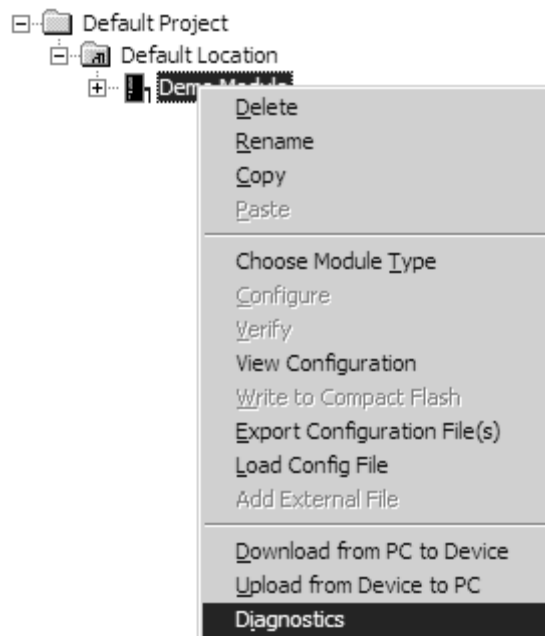
Using the Diagnostic Window in ProSoft Configuration Builder

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

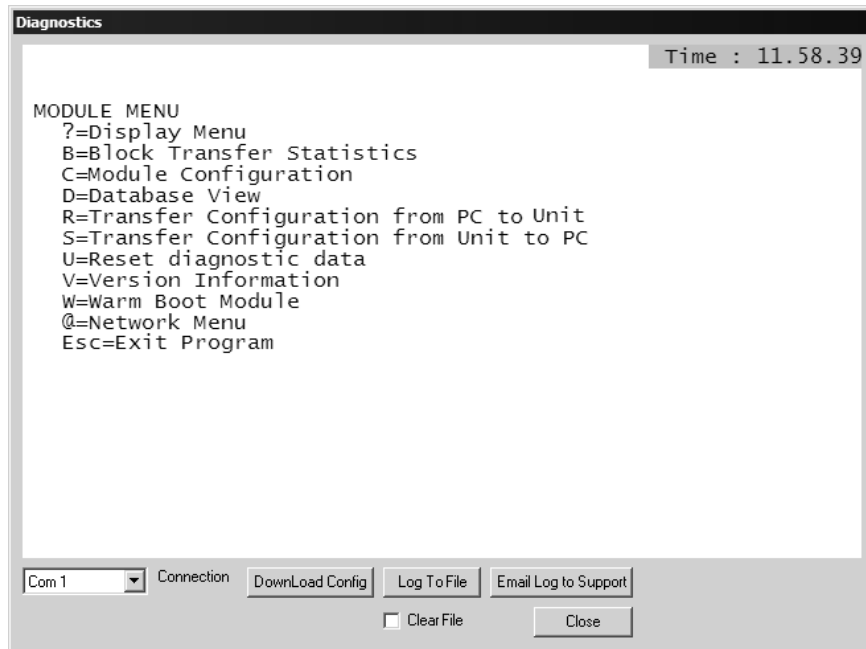
- 1 Start PCB program with the application file to be tested. Right click over the module icon.



- 2 On the shortcut menu, choose Diagnostics.



- 3 This action opens the Diagnostics dialog box. Press "?" to display the Main Menu.



Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

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

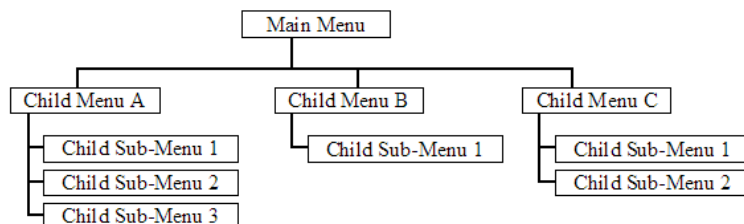
- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

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



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

Keystrokes

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

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

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

4.1.3 Main Menu

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

```
IEC-870-5-101 SLAVE COMMUNICATION MODULE MENU
?=Display Menu
A=Data Analyzer
B=Block Transfer Statistics
C=Module Configuration
D=Database View
E=Display Program Status
F=Priority Queues
G=Invalid State Monitoring
P=Port Cfg
R=Receive Configuration File
S=Send Configuration File
V=Version Information
W=Warm Boot Module
1 = M_SP_NA Setup          2 = M_DP_NA Setup
3 = M_ST_NA Setup          4 = M_ME_NA Setup
5 = M_ME_NB Setup          6 = M_IT_NA Setup
^ = M_ME_NC Setup
7 = C_SC_NA Setup          8 = C_DC_NA Setup
9 = C_RC_NA Setup          0 = C_SE_NA Setup
! = C_SE_NB Setup          $ = C_SE_NC Setup
@ = IEC-870 Database Cfg   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.

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.

Opening the Data Analyzer Menu

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

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

Viewing Module Configuration

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

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

Opening the Database Menu

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

Viewing Error and Status Data

Press **[E]** to display the error/status data for the module.

Viewing Port Configuration

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

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

Receiving the Configuration File

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

Sending the Configuration File

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

Viewing Version Information

Press **[V]** to view Version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The Program Scan Counter value is incremented each time a module's program cycle is complete.

Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Warm Booting the Module

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

Press **[W]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to re-boot.

Viewing Data Type Setup

To view setup information for each data type, press the matching key from the Menu.

Key	Data Type	Screen Example
1	M_SP_NA	<pre> M_SP_NA Setup Menu Selected M_SP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value 0 11 0 80000001 0 1 12 1 80000001 0 </pre>
2	M_DP_NA	<pre> M_DP_NA Setup Menu Selected M_DP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Bits 0 21 16 00000002 0 0 1 22 17 00000002 0 0 </pre>

Key	Data Type	Screen Example																									
3	M_ST_NA	M_ST_NA Setup Menu Selected																									
		M_ST_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>Group(s)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>31</td> <td>6</td> <td></td> <td>00000004</td> <td>0</td> </tr> <tr> <td>1</td> <td>32</td> <td>7</td> <td></td> <td>00000004</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	Group(s)	Value	0	31	6		00000004	0	1	32	7		00000004	0							
Index	Point#	DB	Addr	Group(s)	Value																						
0	31	6		00000004	0																						
1	32	7		00000004	0																						
4	M_ME_NA	M_ME_NA Setup Menu Selected																									
		M_ME_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>Group(s)</th> <th>Deadband</th> <th>Value</th> <th>Norm. Va</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>41</td> <td>4</td> <td></td> <td>00000008</td> <td>0</td> <td>0</td> <td>0.00000</td> </tr> <tr> <td>1</td> <td>42</td> <td>5</td> <td></td> <td>00000008</td> <td>0</td> <td>0</td> <td>0.00000</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	Group(s)	Deadband	Value	Norm. Va	0	41	4		00000008	0	0	0.00000	1	42	5		00000008	0	0	0.00000	
Index	Point#	DB	Addr	Group(s)	Deadband	Value	Norm. Va																				
0	41	4		00000008	0	0	0.00000																				
1	42	5		00000008	0	0	0.00000																				
5	M_ME_NB	M_ME_NB Setup Menu Selected																									
		M_ME_NB Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>Group(s)</th> <th>Deadband</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>51</td> <td>6</td> <td></td> <td>00000010</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>52</td> <td>7</td> <td></td> <td>00000010</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	Group(s)	Deadband	Value	0	51	6		00000010	1	0	1	52	7		00000010	1	0				
Index	Point#	DB	Addr	Group(s)	Deadband	Value																					
0	51	6		00000010	1	0																					
1	52	7		00000010	1	0																					
6	M_IT_NA	M_IT_NA Setup Menu Selected																									
		M_IT_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>Group(s)</th> <th>Value</th> <th>FROZEN</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>61</td> <td>4</td> <td></td> <td>00020000</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>62</td> <td>5</td> <td></td> <td>00040000</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	Group(s)	Value	FROZEN	0	61	4		00020000	0	0	1	62	5		00040000	0	0				
Index	Point#	DB	Addr	Group(s)	Value	FROZEN																					
0	61	4		00020000	0	0																					
1	62	5		00040000	0	0																					
7	C_SC_NA	C_SC_NA Setup Menu Selected																									
		C_SC_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>700</td> <td></td> <td>1600</td> <td>2000</td> <td>3200</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>701</td> <td></td> <td>1601</td> <td>2010</td> <td>3201</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	0	700		1600	2000	3200	0	0	1	701		1601	2010	3201	0	0	
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value																				
0	700		1600	2000	3200	0	0																				
1	701		1601	2010	3201	0	0																				
8	C_DC_NA	C_DC_NA Setup Menu Selected																									
		C_DC_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Bits</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>800</td> <td></td> <td>1616</td> <td>3000</td> <td>3216</td> <td>0</td> <td>0 0</td> </tr> <tr> <td>1</td> <td>801</td> <td></td> <td>1617</td> <td>3090</td> <td>3217</td> <td>0</td> <td>0 0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Bits	0	800		1616	3000	3216	0	0 0	1	801		1617	3090	3217	0	0 0	
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Bits																				
0	800		1616	3000	3216	0	0 0																				
1	801		1617	3090	3217	0	0 0																				
9	C_RC_NA	C_RC_NA Setup Menu Selected																									
		C_RC_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>900</td> <td></td> <td>51</td> <td>8000</td> <td>30</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>901</td> <td></td> <td>52</td> <td>8888</td> <td>31</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	0	900		51	8000	30	0	0	1	901		52	8888	31	0	0	
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value																				
0	900		51	8000	30	0	0																				
1	901		52	8888	31	0	0																				
0	C_SE_NA	C_SE_NA Setup Menu Selected																									
		C_SE_NA Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Value</th> <th>Norm</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1000</td> <td></td> <td>105</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0.0</td> </tr> <tr> <td>1</td> <td>1001</td> <td></td> <td>106</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	Norm	0	1000		105	0	0	0	0	0.0	1	1001		106	0	0	0
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	Norm																			
0	1000		105	0	0	0	0	0.0																			
1	1001		106	0	0	0	0	0.0																			
Shift 1	C_SE_NB	C_SE_NB Setup Menu Selected																									
		C_SE_NB Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1100</td> <td></td> <td>107</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1101</td> <td></td> <td>108</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	0	1100		107	0	0	0	0	1	1101		108	0	0	0	0	
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value																				
0	1100		107	0	0	0	0																				
1	1101		108	0	0	0	0																				
Shift 4	C_SE_NC	C_SE_NC Setup Menu Selected																									
		C_SE_NC Setup (0 to 2)																									
		<table border="1"> <thead> <tr> <th>Index</th> <th>Point#</th> <th>DB</th> <th>Addr</th> <th>MPnt#</th> <th>MPntDb</th> <th>ReqSel</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1200</td> <td></td> <td>55</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1201</td> <td></td> <td>56</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value	0	1200		55	0	0	0	0	1	1201		56	0	0	0	0	
Index	Point#	DB	Addr	MPnt#	MPntDb	ReqSel	Value																				
0	1200		55	0	0	0	0																				
1	1201		56	0	0	0	0																				

Viewing IEC 60870 Database Configuration

Press [**@**] to view the database size configuration information.

```

IEC-870-5-101 DATABASE CONFIGURATION:
PMENA Offs: 2000    PMENB Off: 2000
M_SP_NA point count = 10    Event Scanning Enabled : Yes
M_DP_NA point count = 10    Event Scanning Enabled : Yes
M_ST_NA point count = 10    Event Scanning Enabled : Yes
M_ME_NA point count = 10    Event Scanning Enabled : Yes
M_ME_NB point count = 10    Event Scanning Enabled : Yes
M_IT_NA point count = 4
C_SC_NA point count = 10
C_DC_NA point count = 10
C_RC_NA point count = 10
C_SE_NA point count = 10
C_SE_NB point count = 10
    
```

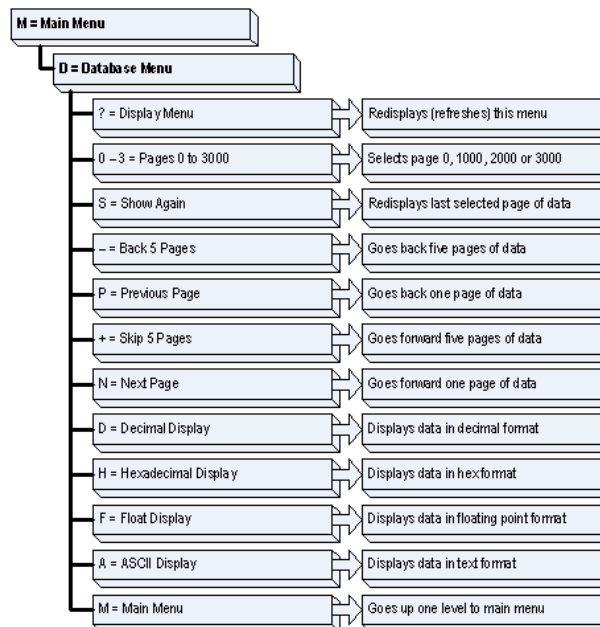
Exiting the Program

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

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

4.1.4 Database View Menu

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



Viewing 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

DATABASE	DISPLAY	0 TO 99	<DECIMAL>							
100	101	102	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	

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 back to the previous 500 registers of data.

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

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

Returning to the Main Menu

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

4.1.5 Data Analyzer

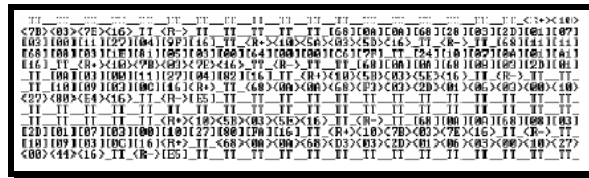
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

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

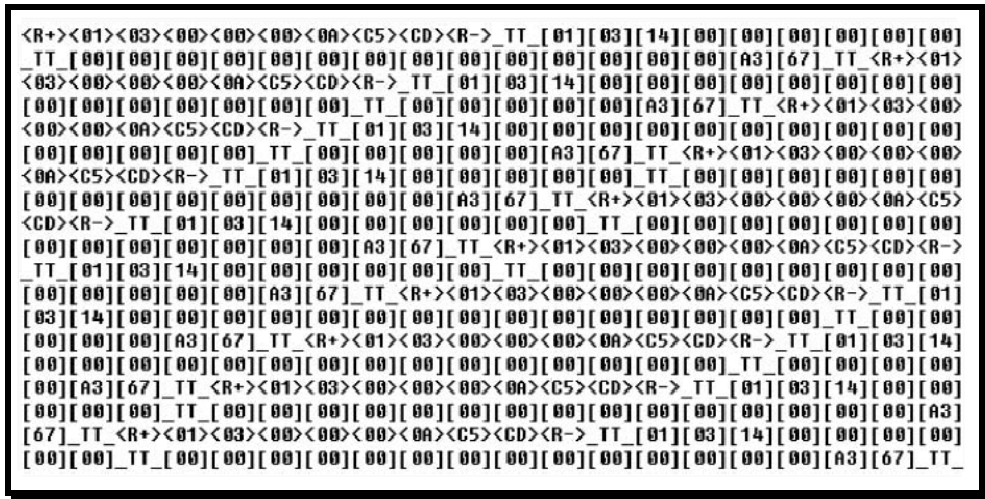
Press [H] to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

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



The Data Analyzer displays the following special characters:

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

Stopping the Data Analyzer

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

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

Returning to the Main Menu

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

4.1.6 Data Analyzer Tips

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

```
Data Analyzer Mode Selected
```

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

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

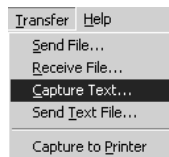
Port = 1, Format=HEX, Tick=10
```

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

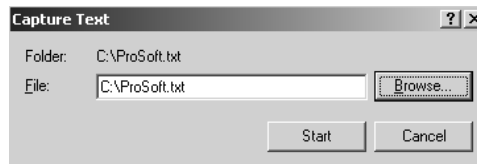
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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

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



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



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

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

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

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

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

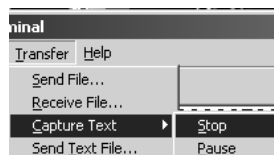
All characters shown in <> brackets are characters being sent out by the module.

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

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

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

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



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

4.2 LED Status Indicators

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

Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the network on Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI69-101S is working normally.
		On	The MVI69-101S module program has recognized a communication error on one of its ports.
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.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

4.2.1 Clearing a Fault Condition

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

To clear the condition, follow these steps:

- 1 Turn off power to the rack
- 2 Remove the card from the rack
- 3 Verify that all jumpers are set correctly
- 4 If the module requires a Compact Flash card, verify that the card is installed correctly

- 5 Re-insert the card in the rack and turn the power back on
- 6 Verify the configuration data being transferred to the module from the CompactLogix or MicroLogix processor.

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

4.2.2 Troubleshooting

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

Processor Errors

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

Module Errors

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

5 Reference

In This Chapter

❖ Product Specifications	73
❖ Functional Overview	75
❖ IEC-60870-5-101 Protocol Implementation	91
❖ Cable Connections	119
❖ MVI69-101S Status Data Definition	125
❖ MVI69-101S Database Design Forms	126
❖ MVI69-101S Error Status Table	130
❖ IEC 60870-5-101 Slave Interoperability Document	131

5.1 Product Specifications

The MVI69 IEC 60870-5-101 Slave Communication Module allows CompactLogix and MicroLogix I/O compatible processors to interface easily with IEC 60870-5-101 protocol compatible hosts. The module's two powerful and highly configurable ports allow the many SCADA host systems supporting the IEC protocol to be integrated into the CompactLogix platform.

The MVI69-101S module is the fastest and easiest way to add IEC 60870-5-101 protocol interface support to the CompactLogix and MicroLogix platforms. It is a single slot, backplane compatible solution for the CompactLogix and MicroLogix I/O compatible processors. This module has two powerful and highly configurable IEC 60870-5-101 Slave ports, allowing the many SCADA and field devices supporting the IEC protocol to be integrated into the CompactLogix platform.

The MVI69-101S module acts as an input/output module between the IEC 60870-5-101 telecontrol network and the CompactLogix processor. The data transfer from the processor is asynchronous from the actions on the network. A 5000-word register space in the module exchanges data between the processor and the telecontrol network.

The standards used in developing the product 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

Refer to these standards for any questions on the protocol supported.

5.1.1 General Specifications

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

5.1.2 Hardware Specifications

Specification	Description
Dimensions	Standard 1769 Single-slot module
Current Load	800 mA max@ 5 VDC Power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)
Operating Temp.	0 to 60°C (32 to 140°F)
Storage Temp.	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Battery and Module Status Application Status Serial Port Activity CFG Port Activity
CFG Port (CFG)	RJ45 (DB-9F with supplied cable) RS-232 only No hardware handshaking
App Ports (P1,P2) (Serial modules)	RS-232, RS-485 or RS-422 (jumper selectable) RJ45 (DB-9F 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

The MVI69-101S module accepts commands from an attached master unit. A port configured as a virtual slave permits a remote master to interact with all data contained in the module. This data can be derived from the CompactLogix processor. The remote master device uses the fully-configured databases in the module to control outputs and monitor inputs. The module can operate in balanced or unbalanced mode.

- Supports time stamp events
- Supports time and data synchronization from a master or the processor

- Supports monitored data
- Event queue supports 100 points for each data type
- Supports Master Class 1 and Class 2 polls with configurable parameters
- Acknowledgement transmission is handled internally by the module
- Configurable data link address (session) and common ASDU address (sector)
- Configurable pulse duration

5.2 Functional Overview

This section provides an overview of how the MVI69-101S module transfers data using the 101S protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

Important: The MVI69-101S module complies with section 6.2.2.5 of the Norwegian User Convention. Users should not assume that the entire Norwegian User Convention specifications are supported.

5.2.1 General Concepts

The following topics describe several concepts that are important for understanding the operation of the MVI69-101S module.

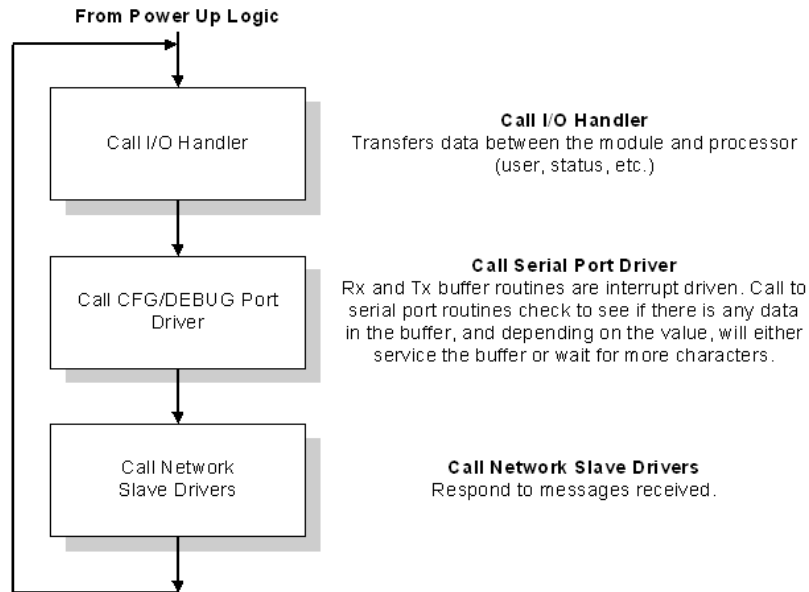
On power up the module begins performing the following logical functions:

- 1 Initialize hardware components
 - Initialize CompactLogix or MicroLogix backplane driver
 - Test and Clear all RAM
 - Initialize the serial communication ports
- 2 Reads configuration from Compact Flash Disk
- 3 Initialize Module Register space
- 4 Enable Slave Driver on selected ports

After the module has received the Module Configuration, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.



CompactLogix or MicroLogix Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (that is, Fault or PGM), backplane communications will halt between processor and module but not on the 101S network. When the processor is returned to a running state, the module will resume communication with the Processor.

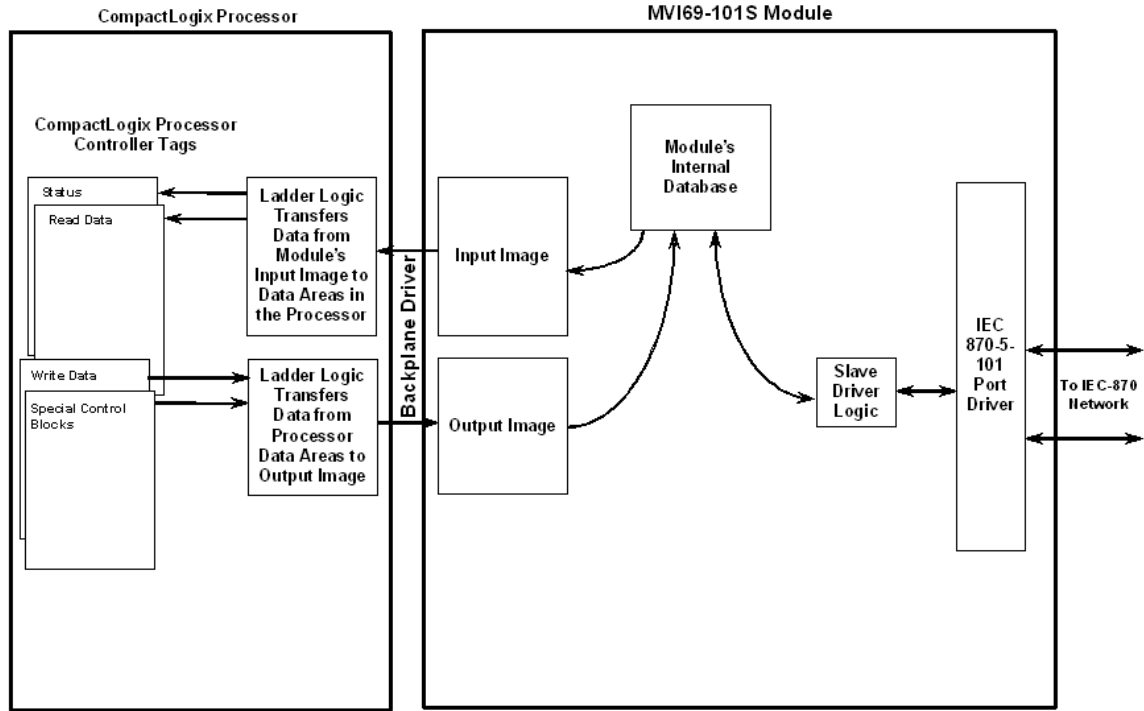
Backplane Data Transfer

The MVI69-101S module communicates directly over the CompactLogix or MicroLogix backplane. Data is paged between the module and the CompactLogix or MicroLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled 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.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is user-configurable at 60, 120 or 240 words. This large data area permits fast throughput of data between the module and the processor.

The processor inserts data into the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module is user-configurable at 60, 120 or 240 words.. This large data area permits fast throughput of data from the processor to the module

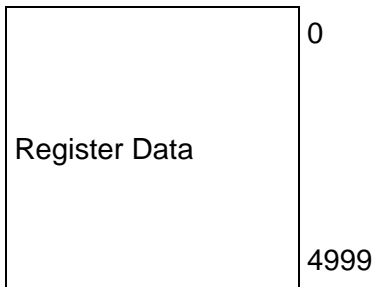
The following illustration shows the data transfer method used to move data between the CompactLogix or MicroLogix processor, the MVI69-101S module and the IEC 60870-5-101 network.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the CompactLogix or MicroLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Module's Internal Database Structure

5000 registers for user data



Data contained in this database is paged through the input and output images by coordination of the CompactLogix or MicroLogix ladder logic and the MVI69-101S module's program. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The module uses the following block numbers:

Block Range	Descriptions
0, -1	Status Block
1 to 67	Read or Write Data
1000 to 1067	Request Output Data from Processor
9958	Event Messages from CLX
9970	Set PLC time using module's time
9971	Set module's time using CLX time
9998	Warm-boot control block
9999	Cold-boot control block

Each image has a defined structure depending on the data content and the function of the data transfer.

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999 and the status data. These data are transferred through read (input image) and write (output image) blocks. Refer to the **Module Configuration** and **Ladder Logic** sections for a description of the data objects used with the blocks and the ladder logic required. The structure and function of each block is discussed in the following topics.

Read Block

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

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

where

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

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

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

R1W1 → R2W2 → R3W1 → R1W2 → R2W1 → R3W2 → R1W1 →

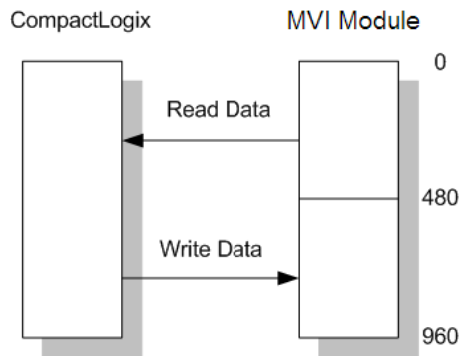
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the 101S network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

Assume that the backplane parameters are configured as follows:

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

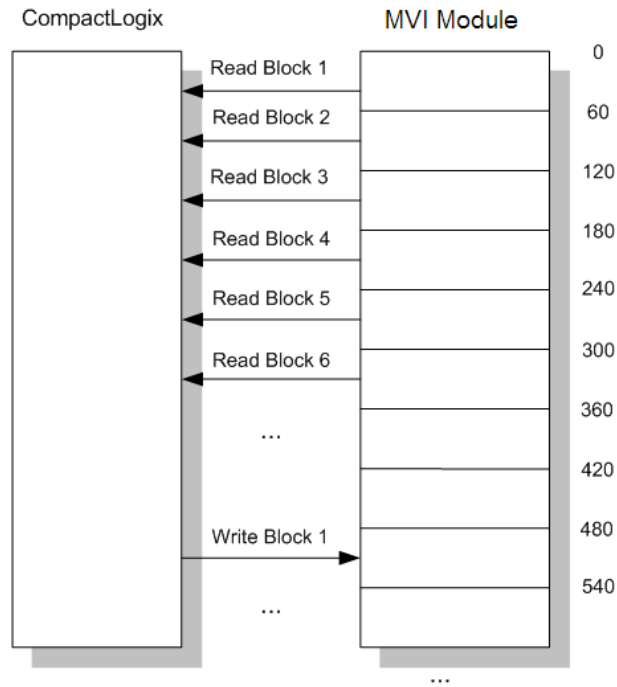
The backplane communication would be configured as follows:



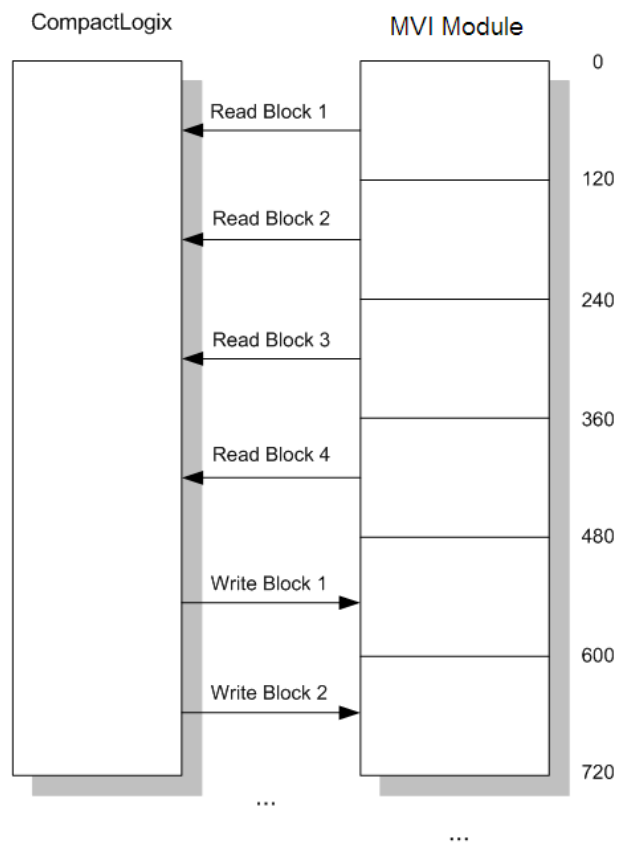
Database address 0 to 479 will be continuously transferred from the module to the processor. Database address 480 to 959 will continuously be transferred from the processor to the module.

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

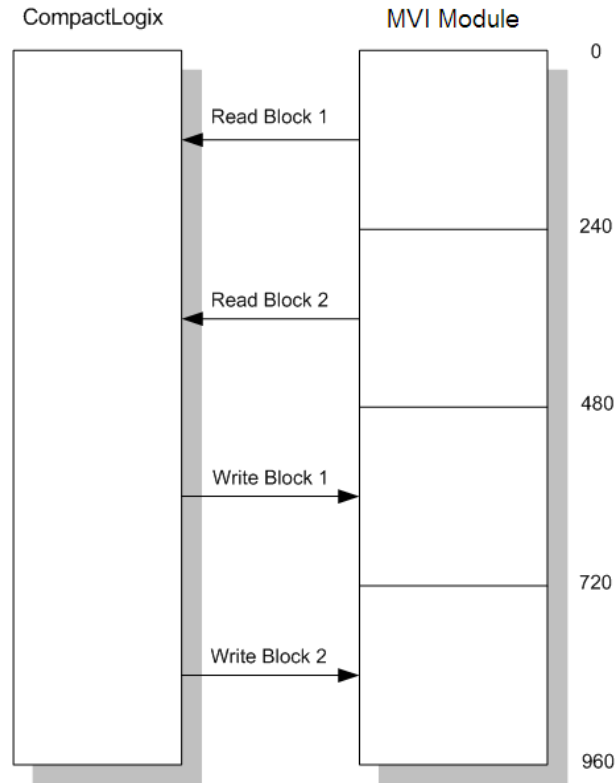
If Block Transfer Size = 60:



If Block Transfer Size = 120:



If Block Transfer Size = 240:



Write Block

These blocks of data transfer information from the CompactLogix or MicroLogix processor to the module and source the input (monitored) data to be used by the remote client. The structure of the output image used to transfer this data is shown in the following table.

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

where $n = 60, 120, \text{ or } 240$ depending on the Block Transfer Size parameter (refer to the configuration file).

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed. Each transfer can move up to 200 words (block offsets 1 to 200) of data.

How Data is Transferred

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

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

Read Register Start = 0

Read Register Count = 120

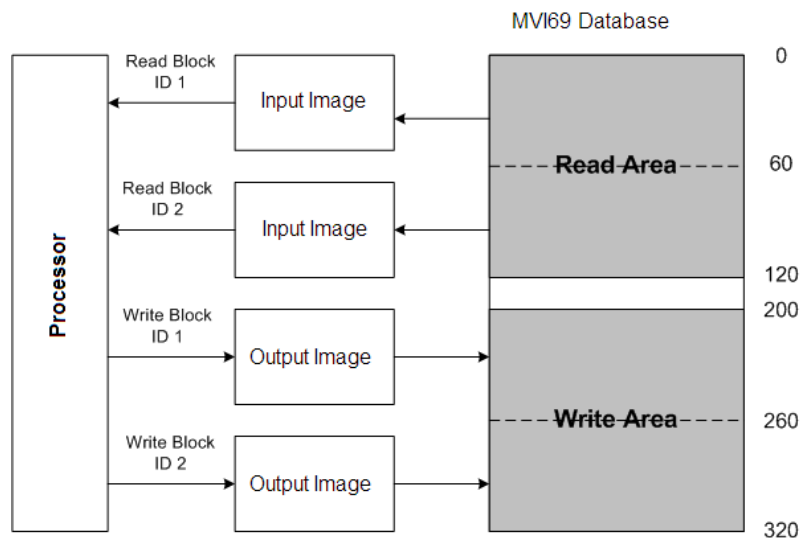
Write Register Start = 200

Write Register Count = 120

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

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

The following example shows the resulting data flow:



5.2.3 Status Block

The module periodically sends blocks 0 and -1 containing status information, as described in the following table.

Offset	Description	Length
0	Block Read ID	1
1	Block Write ID	1
2	Program Scan Counter	2
4	Product Code	2
6	Product Version	2
8	Operating System	2
10	Run Number	2
11	Backplane Read Count	1

Offset	Description	Length
12	Backplane Write Count	1
13	Backplane Parse Count	1
14	Backplane Error Count	1
15	Port Select	1
16	Bad CRC	1
17	Sync Error Count	1
18	Length Error Count	1
19	Timeouts Count	1
20	RX Frame Count	1
21	TX Frame Count	1
22	M_SP_NA Event Buffer Free	1
23	M_DP_NA Event Buffer Free	1
24	M_ST_NA Event Buffer Free	1
25	M_ME_NA Event Buffer Free	1
26	M_ME_NC Event Buffer Free	1
27	M_ME_IT Event Buffer Free	1

5.2.4 Sending Events

The module operates by sending data when the master sends Class 2 polls. When the slave has an event, it sends the event to the module queue. If the communication mode is unbalanced, the slave notifies the master that it has an event to be sent and the master's next poll should be a Class 1 poll. The slave then replies to this poll sending the first event to the queue.

There are two ways of sending timestamp events to the module queue. The first is when a configured point changes its value in the module's database. This is actually the way the module would normally send events.

The second method is by using block 9958 (refer to the next section), in which case ladder logic should be used to send events in this manner. This method is limited due to the number of events that can be sent to the master. The important issue about events is that the slave should have its clock synchronized with the master. The master should send a Sync. Command to the MVI module in order to synchronize both clocks. You can also synchronize the module and the processor clocks using blocks 9970 and 9971 (Refer to the next section).

5.2.5 Command Control Blocks

Command control blocks are special blocks used to control the module. The current version of the software supports the following command control blocks:

- Request Output Data from Processor (page 84)
- Event Message Block (Block 9958) (page 84)
- Get Module Time (Block 9970) (page 85)
- Set Module Time (Block 9971) (page 86)
- Warm Boot (page 87, page 62)
- Cold Boot (page 87)

Introduction

The following table lists all control blocks supported by the module.

Block Range	Descriptions
1000 to 1067	Request Output Data from Processor
9958	Event message block from processor
9970	Get module time from module
9971	Set module time from processor
9998	Warm-boot control block
9999	Cold-boot control block

Request Output Data from Processor

In order to use this functionality the Initialize Output Data parameter in the configuration file must be enabled. The idea is to allow the module after a power up to update its Read Data area with the last data that sent to the module. This allows the module to start communications with its database updated with the last values that were read from the remote device. Upon power up the module will request these special blocks and the ladder logic must copy the last read data blocks received back to the module. Please refer to sample ladder logic for a program example that shows how to use this functionality.

Write Request Output Block

Offset	Description	Length
0	Write Block ID 1000 to 1067	1
1	Read Data	60/120 or 240

Event Message Block (Block 9958)

This block is used to send timestamp event messages from the processor to the module. Please refer to the sample ladder logic for an example on how to use this functionality.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9958 identifying the block type to the module.
1	Event Count	Number of events present in the block. This field can have a value from 1 to 3.
2 to 16	Event #1	Event data to add to event message queue.
17 to 31	Event #2	Event data to add to event message queue.
32 to 46	Event #3	Event data to add to event message queue.
47 to x	Not Used	Not Used

The structure of each event record in the block is shown below:

Word Offset in Event Record	Data Field(s)	Description
0	DB Index	This is the index for the point in the module's database. This corresponds to the order of point definition for the module data types. This is not the point address for the event.
1	ASDU	This is the ASDU data type for the event message. Valid entries for this field are as follows: 1=single point, 3=double-point, 5=step, 9=normalized, 11=scaled and 15=integrated total.
2	Qualifier	This is the qualifier code to be used with the event message. Refer to the IEC protocol specification for a full listing of valid qualifier codes for each ASDU type.
3	Year	This field contains the four-digit year to be used with the event.
4	Month	This field contains the month value for the event. Valid entry for this field is in the range of 1 to 12.
5	Day	This field contains the day value for the event. Valid entry for this field is in the range of 1 to 31.
6	Hour	This field contains the hour value for the event. Valid entry for this field is in the range of 0 to 23.
7	Minute	This field contains the minute value for the event. Valid entry for this field is in the range of 0 to 59.
8	Seconds & Milliseconds	This field contains the seconds and milliseconds value for the event. Valid entry for this field is in the range of 0 to 59,999.
9 to 14	Data	These words contain the data to be used with the event. For single- and double-point, step and measured value events, the first word is used. For integrated total events, the first two words are used.

Get Module Time (Block 9970)

This block can be used to retrieve the date and time information from the module.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to 61	Not Used	Not Used

Response to a block 9970 request: The module will respond to a valid block 9970 request with a block containing the requested date and time. The format for the block is shown below:

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block Read ID	This field contains the block identification code of 9970 for the block.
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year to be used with the new time value.

Word Offset in Block	Data Field(s)	Description
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 61	Not Used	Not Used

Set Module Time (Block 9971)

This block can be used to set the date and time information to the module.

Block Format for Write

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 to be used with the new time value.
2	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
4	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
5	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
8 to 61	Not Used	Not Used

Warm Boot

This block is sent from the CompactLogix or MicroLogix processor to the module when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made in the controller tags data area. This forces the module to read the new configuration information and to restart. The structure of the control block is shown in the following table.

Block Request

Offset	Description	Length
0	9998	1
1 to x	Not used	blk_size

Cold Boot

This block is sent from the CompactLogix or MicroLogix processor to the module when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The structure of the control block is shown in the following table.

Block Request

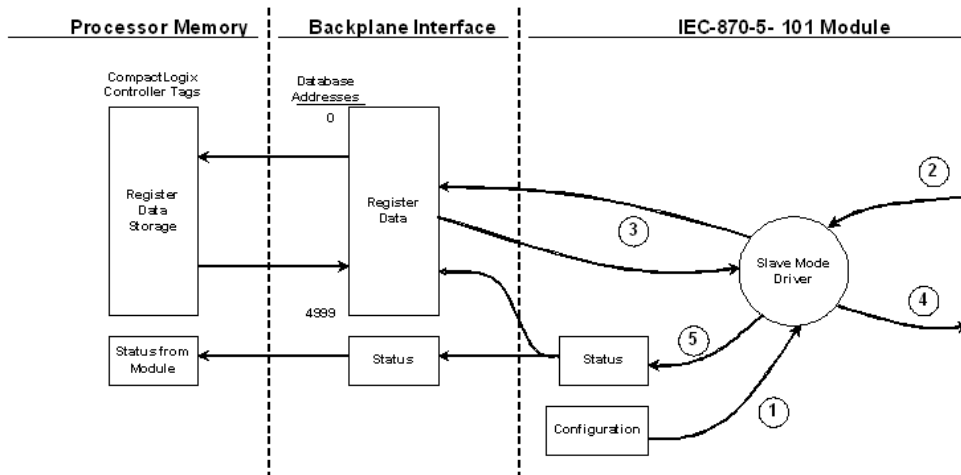
Offset	Description	Length
0	9999	1
1 to x	Not used	blk_size

5.2.6 Data Flow Between MVI69-101S Module and CompactLogix or MicroLogix Processor

The following topics describe the flow of data between the two pieces of hardware (CompactLogix or MicroLogix processor and MVI69-101S module) and the IEC 60870-5-101 master unit. Each port on the module is configured to emulate a common slave device. The database used in the module is used for both ports and only one port can be utilized at one time.

Slave Driver

The Slave Driver allows the MVI69-101S module to respond to data read and write commands issued by a master unit on the telecontrol network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The slave port driver receives the configuration information from the Compact Flash Disk in the module. This information configures the serial port and define the slave node characteristics.
2	A Host device issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the message, the data is immediately transferred to or from the internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

Review the **Installing and Configuring the Module** section for a complete list of the parameters that must be defined for a slave port. The IEC 60870-5-101 Interoperability Document for the MVI69-101S Slave Module contains a listing of the protocol support supplied in the module.

5.2.7 Databases

The read and write areas can be placed anywhere in the module's database area. Because each point is defined individually to the module, the data for a specific type need not be contiguous in the module's database. This means that the module error/status data area can be passed to the controlling station using the M_ME_NB_1 database. In the database definition for the type, establish a point for each status value to be monitored by the controlling station and set the module's database address for the point in the definition.

This section discusses the databases utilized by the module to support the IEC 60870-5-101 protocol. The user is responsible for defining the databases for their specific application. In the module's configuration file, the size of each database and definition of each point is established. Refer to the Module Configuration section for a complete discussion of the configuration file. The following table summarizes the data types used in each of the 11 individual databases.

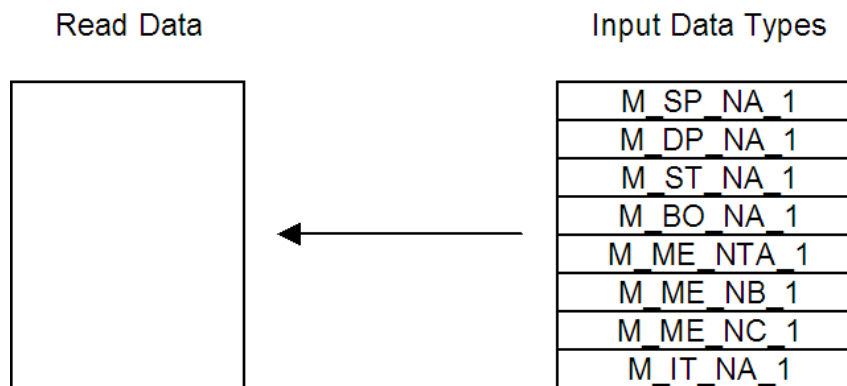
Type ID	Type	Description	Data Representation
1	M_SP_NA_1 (7.3.1.1)	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value with (7.2.6.1) 0=Off and 1=On.
3	M_DP_NA_1 (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
9	M_ME_TA_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 (16-bit) word 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 (16-bit) word 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 with (7.2.6.17) 0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted

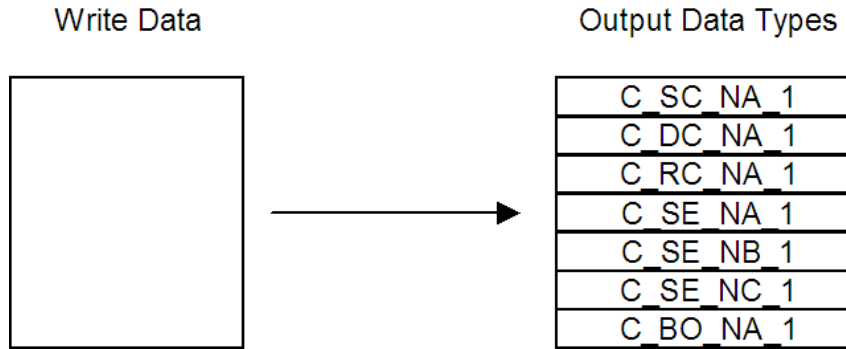
Type ID	Type	Description	Data Representation
48	C_SE_NA_1 (7.3.2.4)	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values (7.2.6.6) are stored in a (16-bit) word data area with a range of -1..+1-2 ⁻¹⁵
49	C_SE_NB_1 (7.3.2.5)	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values (7.2.6.7) are stored in a (16-bit) word data area with a range of -2 ¹⁵ ..+2 ¹⁵ -1

A key concept in interfacing the protocol with the CompactLogix or MicroLogix processor is the relationship between the databases and the data transfer operation between the module and the processor. The module transfers data to the processor in read blocks using the input image. These blocks should contain the information received from the controlling unit (output data) and includes the following data types: C_SC_NA_1, C_DC_NA_1, C_RC_NA_1, C_SE_NA_1 and C_SE_NB_1.

This data is all sourced from the master unit and passed to the processor for control. Databases associated with these data types should place the points in the read data area of the module's database. The Read Register Start and Read Register Count parameters in the configuration file establish the portion of the database to transfer to the processor. Ladder logic extracts the data from the read data area and places it in the proper location for use by the processor.

Similarly, data to be monitored (input data) by the master unit (all databases associated with the "M_" data types) must all be placed in the write data area of the module. The Write Register Start and Write Register Count parameters establish the portion of the database to receive data from the processor. This data is sourced from the processor and passed through the module to the remote controlling unit. Ladder logic is required to place the data in the correct position in the write data area. The relationship between the data types and the read and write data areas is displayed in the following diagram:





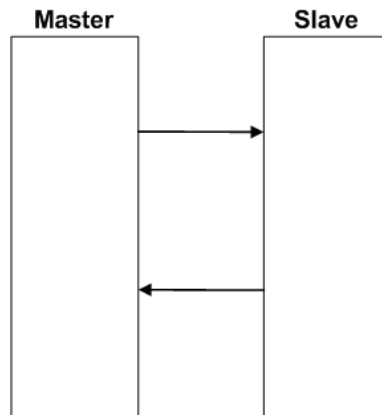
The read and write areas can be placed anywhere in the module's 5000-word database area. Because each point is defined individually to the module, the data for a specific type need not be contiguous in the module's database. This means that the module error/status data area can be passed to the controlling station using the M_ME_NB_1 database. In the database definition for the type, establish a point for each status value to be monitored by the controlling station and set the module's database address for the point in the definition.

5.3 IEC-60870-5-101 Protocol Implementation

This section presents an overview of how the MVI69-101S works, while skipping the complex details of the protocol specification. If you require more information about the implementation of the protocol, refer to the protocol specification (IEC 60870-5-101 2003). For more information on the configuring and modifying the backplane implementation of the protocol with the MVI69-101S, refer to Customizing the Sample Configuration File.

The IEC-60870-5-101 protocol applies to telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes.

Any application using the IEC-60870-5-101 protocol will have a master (controlling station) and one or more slaves (controlled stations). The master will constantly monitor and control the data from each slave in the network.



The MVI69-101S module works as a IEC-60870-5-101 slave. It can send monitor data, receive commands or generate events to the master unit, as explained in the following topics.

5.3.1 General Parameter Configuration

Communication Parameters

The following parameters should be configured for serial communication:

```
Baud Rate      : 9600 #Baud rate for port 110-38400
Parity         : 2 #0=None, 1=Odd, 2=Even
RTS On         : 0 #0-65536 mSec before message
RTS Off        : 0 #0-65536 mSec after message
Minimum Delay  : 20 #Minimum # of mSec before response sent
Receive Timeout : 2000 #Maximum mSec from first char to last to wait
Hardware Handshaking: 0 #Hardware handshaking 0=None, 1=RTS/CTS, 2=DTR/DSR,
                    #3=modem
```

Adjust these parameters for your application.

Data Link Configuration

The protocol specification document IEC 60870-5-2 specifies an unambiguous address (number) for each link. Each address may be unique within a specific system, or may be unique within a group of links sharing a common channel. The protocol specification defines that the Data Link Address may have 0, 1 or 2 octets. The first option should be used only during Balanced Mode.

Configure the following parameter to set the number of octets to be used for the Data Link Address value. It is essential that the master unit also uses the same number of octets configured for the MVI69-101S.

```
Data link address length : 2 #0, 1, or 2 octets used for DL address
```

You must also configure the actual Data Link Address value using the following parameter:

```
Data link address : 1 #Range depends on the configured DL
                  #Length
```

This value identifies the module's address in the network.

ASDU Configuration

The protocol specification document IEC 60870-5-3 describes the Basic Application Data Units that are used in the protocol. It also defines the Application Service Data Unit (ASDU) used by the protocol for data communication.

You can configure the number of bytes to be used for the following ASDU components:

Component	Abbreviation	Possible Number of Octets
Cause of Transmission	COT	1 or 2
Common Address of ASDU	CASDU	1 or 2
Information Object Address	IOA	1, 2 or 3

These parameters must be configured to match the master's configuration, as shown in the following example:

```
Common Address of ASDU len:      2 #Range is 1 or 2 octets
Inform. Object Address len:     3 #Range is 1, 2, or 3 octets
Cause of Trans Octets          :   2 #Number of COT octets (1 or 2)
```

The Common Address of ASDU is the station address. The module only allows the addressing of the whole station (some devices allows different Common Addresses to identify particular station sectors). You should configure the Common ASDU Address with the following parameter:

```
Common Address of ASDU          :   1 # Range depends on number of octets for
                                   # CASDU
```

You should also configure the maximum number of bytes that the module will support for each ASDU response to the master unit. The range lies between 25 and 252 bytes. You should verify the maximum number of bytes supported by the master because some IEC 60870-5-101 master devices only support messages with less than 252 bytes.

Configure the following parameter to set the maximum number of bytes to be transferred at every ASDU response.

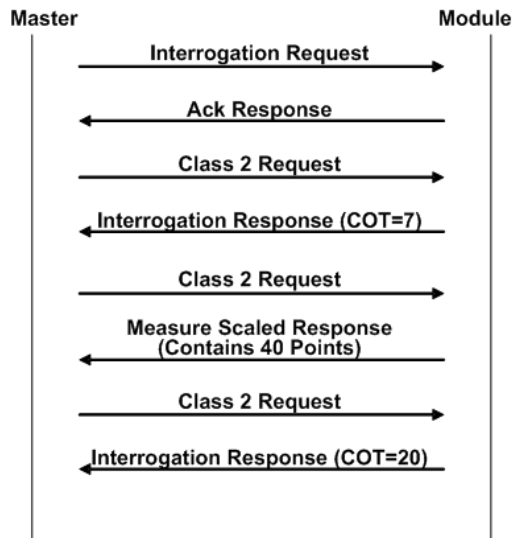
```
Maximum ASDU Resp Len          :   252 #Max ASDU response message
```

A value less than 252 can cause the module to break down the response to send all points using more response messages, with each message containing fewer bytes.

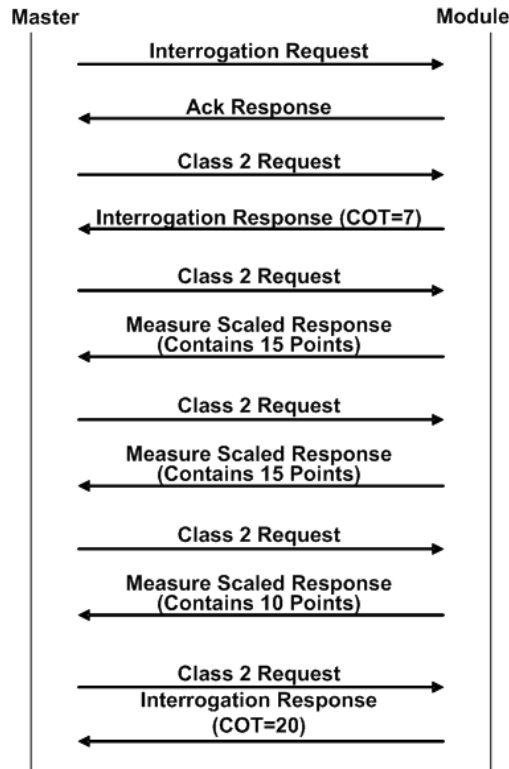
Example - Changing the ASDU Length:

Considering that the master sends a General Interrogation request to poll forty M_ME_NB points (measured scaled points in Unbalanced Mode):

If Maximum ASDU Resp Len = 252:



This example shows that the module sends all 40 measured scaled points in one single message, if the message is not greater than 252 bytes. However, some master devices may not support messages containing this number of bytes. If the master for a given application supports only 100 bytes, the following communication procedure would occur:



As shown in the previous diagram, the module sends out 15 messages (15 points, 15 points and 10 points) instead of sending the whole 40 points in one single message.

Note: This example shows the case where IOA Length = 3 bytes, COT Length = 2 bytes and CASDU Length = 2 bytes.

Important: If the database parser gets a point index that is not valid, the whole database is set as invalid and no points are reported. Because the index 0 is not valid (the protocol does not support this index value), the driver considers it invalid. For example, if you set the size of the ASDU to 1 and you set a value of 1000 for a point index, this is also invalid as the indexes can only go from 1 to 255.

Balanced and Unbalanced Modes

The module supports balanced and unbalanced modes. In balanced mode, each station may initiate message transfers. If the links from the master unit to several slaves share a common physical channel, then these links must be operated in an unbalanced mode to avoid the possibility of more than one device attempting to transmit on the channel at the same time.

Select the communication mode with the following parameter:

Use Balanced Mode : 0 #0=No, 1=Use balanced mode

To use balanced mode, configure the following parameters:

Retry Count : 2 #0 to 255 retries for balanced mode if no
#response

Response Timeout : 1000 #0 to 65535 milliseconds to wait for response

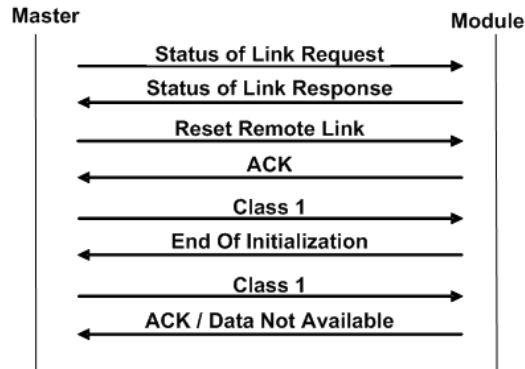
5.3.2 Module Initialization

After the module powers up, a specific initialization procedure occurs, depending on the communication mode you selected (Balanced or Unbalanced).

Unbalanced mode

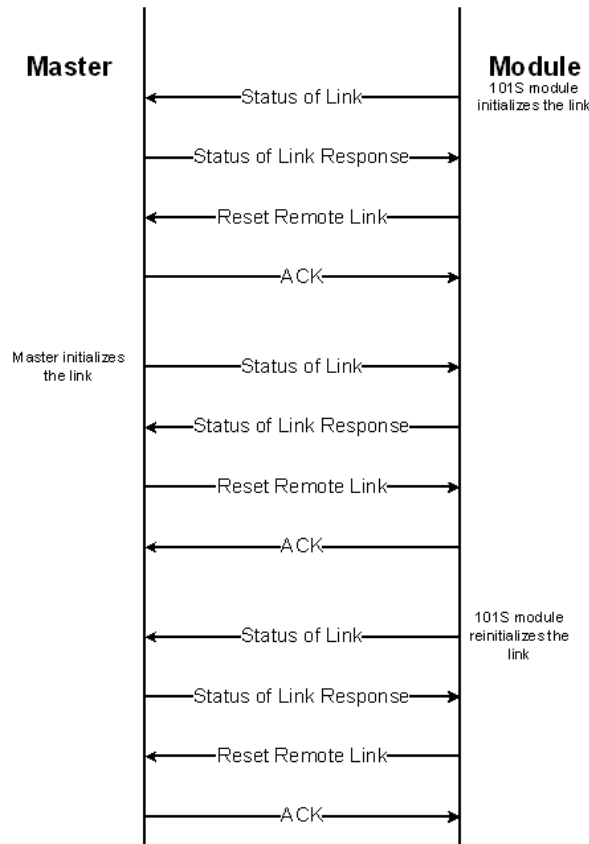
In order to start communications between the master and the slave units, the master tries to establish the link connection by transmitting repeated "Request Status of Link" at specific time out intervals. When the module's link is available it will respond with a "Status of Link" response. Then, the master transmits a "Reset of Remote Link" message and the slave responds with an ACK response. Then the master sends two consecutive Class 1 requests. The slave responds the first Class 1 request with a "End of Initialization" response and the second Class 1 request with an Ack message.

The following illustration shows a typical initialization procedure for the unbalanced mode:



Balanced Mode

During balanced mode, the link must be initialized in both directions. The MVI69-101S module also always reinitializes the link after it receives an initialization request from the master. Therefore, the following initialization occurs during balanced mode, after MVI69-101S boots.



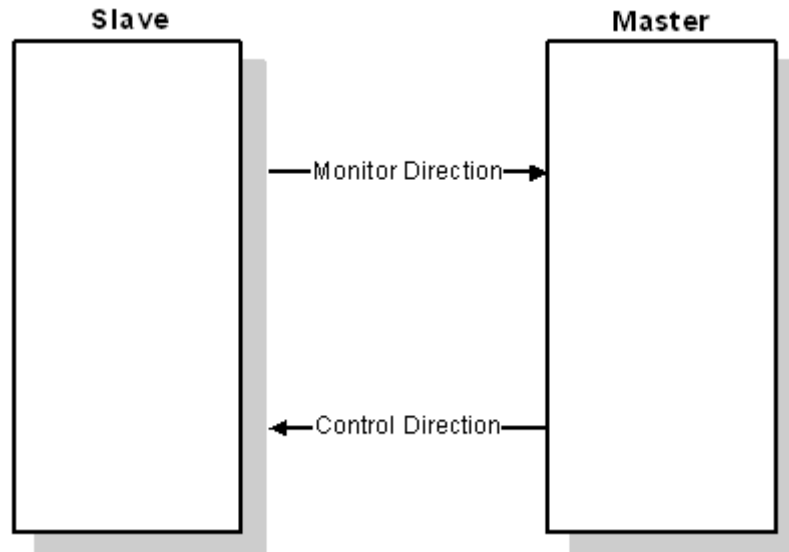
After the initialization procedure is completed, the master and the MVI69-101S start communicating. If during communication the master fails to respond to a message from the module, the MVI69-101S will retry for a configured number of times. If the master still fails to respond, the module will initialize the line again.

5.3.3 Monitor Direction and Control Direction

The protocol specification defines two directions of data: monitor direction and control direction. These directions are defined by the protocol specification as follows:

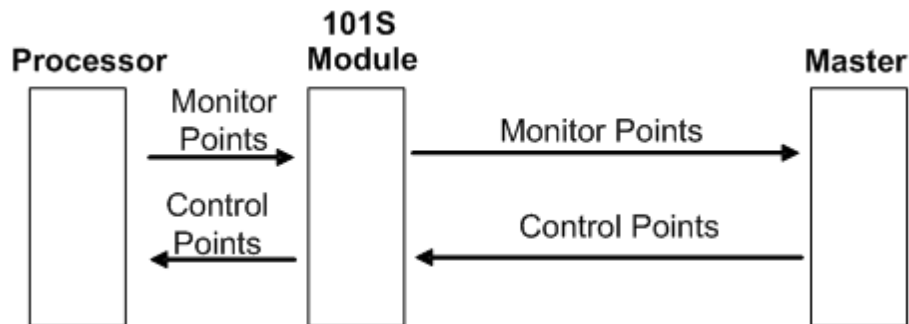
Monitor Direction: The direction of transmission from a slave to the master

Control Direction: The direction of transmission from the master to a slave



The points that are typically transferred from the slave to the master are also known as Monitor Points (or Monitor Information Objects). The points that are typically transferred from the master to the slave are also known as Control Points (or Command Information Objects).

For the MVI69-101S, the control and monitor points would be transferred as follows:



You must make sure that all points are configured in the correct location in the MVI69-101S module database in order to be properly updated from/to the processor.

5.3.4 Using Monitor Points

The following monitor points are supported by the MVI69-101S module:

Symbol	Description	Data Size in Database	Addressing Type
M-SP-NA	Monitored Single-Points	1 bit	Bit
M-DP-NA	Monitored Dual-Points	2 bits	Bit
M-ST-NA	Monitored Step-Points	1 byte	Byte

Symbol	Description	Data Size in Database	Addressing Type
M-ME-NA	Monitored Measured Normalized-Points	1 word	Word
M-ME-NB	Monitored Measured Scaled-Points	1 word	Word
M-ME-NC	Monitored Measured Short Floating Points	2 words	Double-Word
M-IT-NA	Monitored Counter-Points	2 words	Double-Word

Each monitor point is identified by its Information Object Address (it should be unique for each Common ASDU Address in the network). For each monitor point, configure the following parameters:

Point # - The information object address of the point. It identifies the point in the network.

DB Address - The database location in the MVI69-101S module associated with the point. You must associate each point to a database address in the MVI69-101S module. The interpretation of this parameter depends on the point type configured. For example, for an M_SP_NA point, this value represents the bit address. For a M_ME_NA point, this value represents the Word address.

Group(s) - This is the group definition for the point. It sets how the point will be polled by the master (cyclic or group interrogation). It can also be used to enable or disable the event generation for one specific point. The group parameter is discussed in the Data Communication section.

Deadband - Sets the deadband for each Measured point. If the value changes from more than the configured deadband, the module will generate an event for this point.

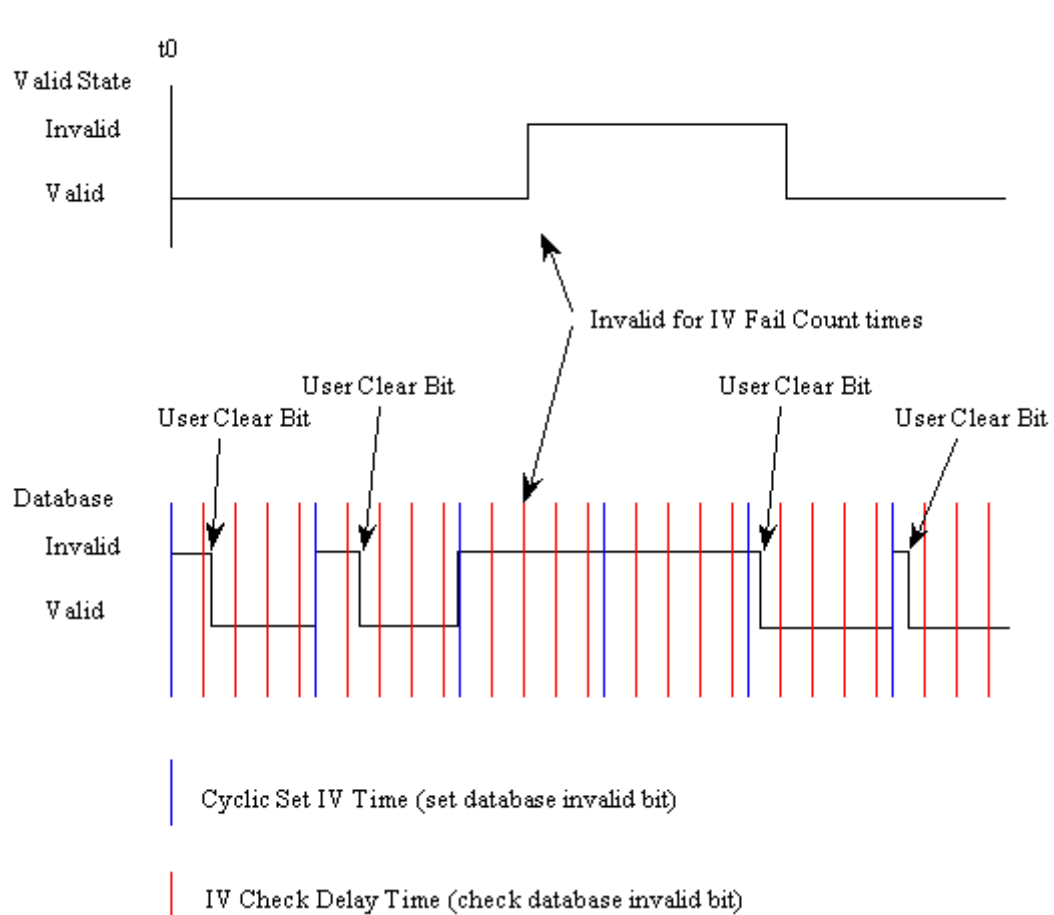
IV DB Bit - This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. If you enable this feature, the processor can determine the individual IV quality bit status of each point you configured.

The following parameters must be configured in order to use this feature:

```
Cyclic Set IV Time      :    10 #Number of sec intervals
IV Check Delay Time    :     2 #Number of sec intervals between
                        #investigation
IV Fail Count          :     2 #Number of IV failures recognized
                        #before reporting
```

To disable this feature, set the IV Fail Count parameter to 0. If used, the Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

The Cyclic Set IV Time parameter must be set to determine how frequently the IV Checks will be performed. If the IV bit is ON for a number of times given by the IV Fail Count parameter the module will consider the point as invalid. The following illustration shows how these parameters are implemented:



If the IV bit field is absent or set to 0, the invalid quality state for the point will always be reported as valid.

If a database bit address (1 to 64000) is present, the application may consider the point with an invalid flag if the previous logic checks the iv bit as 1 during consecutive IV Check Delay scans. The iv bits would have to be reset to 0 to set the point to valid state.

The IV DB bit defined for each point can be unique or many points may share the same bit. The last case could be used when the points on an I/O module are to be considered as one set. In this case only a single bit is required. For a point that is the result of a computation, the valid quality state could be set for each point individually.

Monitor Data Transfer

In unbalanced mode the polling procedure is initiated from the master unit through Class 1 and Class 2 requests. In general, ASDUs containing the causes of transmission periodic/cyclic are assigned to be transmitted with class 2 requests and all time tagged or spontaneously transmitted ASDUs are assigned to be transmitted through class 1 requests. Other ASDUs with other causes of transmission of low priority such as background scan may also be assigned to class 2 requests.

Typically, you should properly configure the group code for each monitor point to define how the master will poll for the point. The group codes are defined as follows:

The Group parameter is defined as follows:

Group Code	Description
0x00000001	Interrogated by general interrogation (station or global)
0x00000002	Interrogated by group 1 interrogation
0x00000004	Interrogated by group 2 interrogation
0x00000008	Interrogated by group 3 interrogation
0x00000010	Interrogated by group 4 interrogation
0x00000020	Interrogated by group 5 interrogation
0x00000040	Interrogated by group 6 interrogation
0x00000080	Interrogated by group 7 interrogation
0x00000100	Interrogated by group 8 interrogation
0x00000200	Interrogated by group 9 interrogation
0x00000400	Interrogated by group 10 interrogation
0x00000800	Interrogated by group 11 interrogation
0x00001000	Interrogated by group 12 interrogation
0x00002000	Interrogated by group 13 interrogation
0x00004000	Interrogated by group 14 interrogation
0x00008000	Interrogated by group 15 interrogation
0x00010000	Interrogated by group 16 interrogation
0x00020000	Interrogated by general counter request
0x00040000	Interrogated by group 1 counter request
0x00080000	Interrogated by group 2 counter request
0x00100000	Interrogated by group 3 counter request
0x00200000	Interrogated by group 4 counter request
0x40000000	Disable event scanning of this point
0x80000000	Periodic/cyclic data returned from unit

The module will periodically send all points configured for periodic/cyclic poll (0x80000000) at every x milliseconds, where x is configured with the following parameter:

```
Cyclic data transmission: 20000 #Numb of milliseconds between cyclic
                           #updates
```

Example - Periodic Monitor Polling:

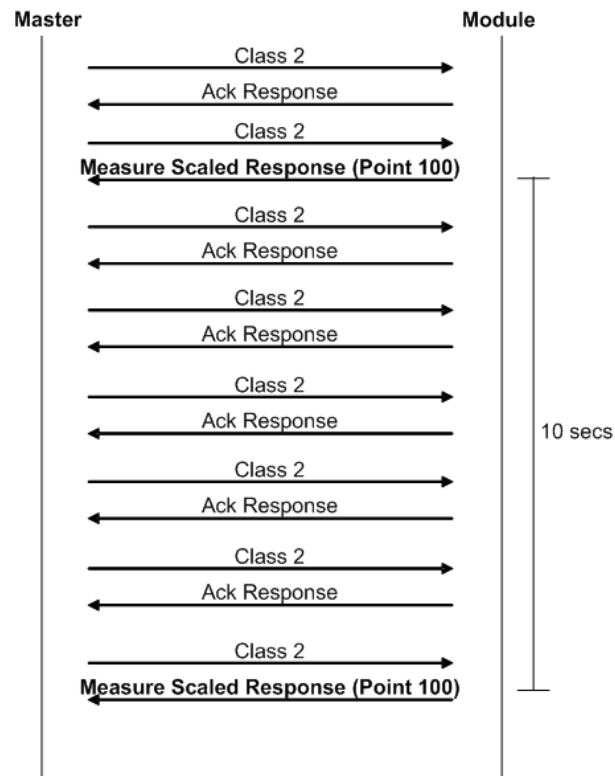
If the following point is configured for monitor polling:

```
[M_ME_NB_1]
#
# Each point is one word and the DB Address value corresponds to the
# word offset into the database.
#
#
#           Default
# Point #   DB Address   Group(s)   Deadband   IV DB Bit
# -----  -
START
    100           0   80000000         0           0
END
```

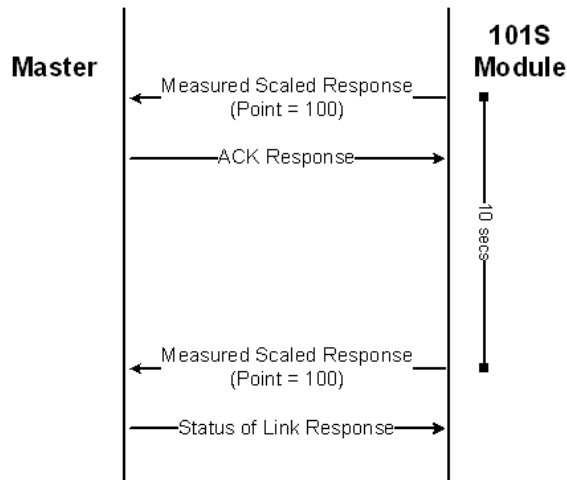
If you configure the periodic polling for 10 seconds (10000 milliseconds) as follows:

```
Cyclic data transmission : 10000 #Numb of milliseconds between
                             #cyclic updates
```

The following illustration shows the communication procedure for unbalanced mode:



The following illustration shows the communication procedure for balanced mode:



Therefore, the point configured for a cyclic poll is periodically reported to the master.

You may also create groups of points allowing the master to poll certain points more frequently than other points. The master may send requests for different groups as follows:

- General Interrogation (station)
- General Interrogation for Group 1
- General Interrogation for Group 2
- ...
- General Interrogation for Group 16

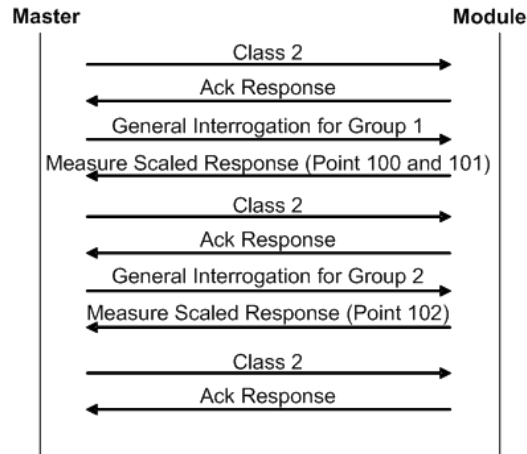
Example - General Interrogation

If the following points are configured for General Interrogation:

If you configure the following data points:

```

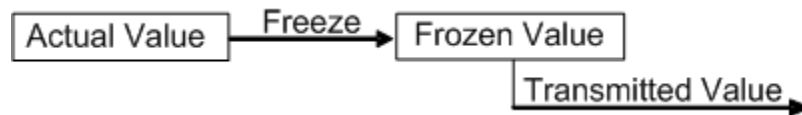
[M_SP_NA_1]
# Point #    DB Address    Group(s)    IV DB Bit
# -----    -
START
    100        1600           00000002    0 # Group 1 Interrogation
    101        1601           00000002    0 # Group 1 Interrogation
    102        1602           00000004    0 # Group 2 Interrogation
END
    
```



This feature allows you to separate the points into different groups according to the priority level that these should be reported to the master. In the example above, points 100 and 101 would be returned with a General Interrogation for Group 1 and point 102 would be returned with a General Interrogation for Group 2:

Counter Points

There are four modes of acquisition of integrated totals (M_IT_NA points) defined by the protocol specification. The actual values may be memorized (copied) periodically to frozen values by a freeze command received from the master or initiated locally within the module.



The module supports the following modes:

Mode A - Local freeze with spontaneous transmission

Mode D - Counter interrogation commands from the master initiate the freeze operation and the frozen values are reported spontaneously.

Example - Mode A

To use Mode A, configure the following parameters:

```

Freeze Start Type      :   D #D=Day, H=Hour, M=Minute, N=Not used
Interval For Freeze   :   15 #Number of seconds after start type
                        :   #(0 to 65535)
  
```

Freeze Start Type

The Freeze Start Type parameter defines when the module starts sending the M_IT messages.

Example I - Freeze Start Type

If the module powers up with the following date and time clock:

03/25/2004 18:07:42

If you configure the Interval For Freeze parameter as follows:

```
Interval For Freeze      :      15 #Number of seconds after start
                          #type (0 to 65535)
```

The module would send the counter messages every 15 seconds. The module would start sending the messages depending on the Freeze Start Type parameter as follows:

Freeze Start Type	Time to Start Sending Messages
D	03/26/2004 00:00:00
H	03/25/2004 19:00:00
M	03/25/2004 18:08:00

Example II - Freeze Start Type

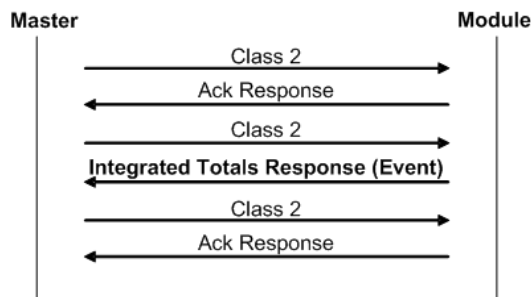
If the module should send the counter points on the hourly turn around time and also 45 minutes later, the Mode A parameters should be configured as follows:

```
Freeze Start Type      :      H #D=Day, H=Hour, M=Minute, N=Not
                          #used
Interval For Freeze    :      2700 #Number of seconds after start
                          #type (0 to 65535)
```

So the module would send events as follows (Hours:Minutes:Seconds):

```
17:00:00
17:45:00
18:00:00
18:45:00
19:00:00
19:45:00
...
```

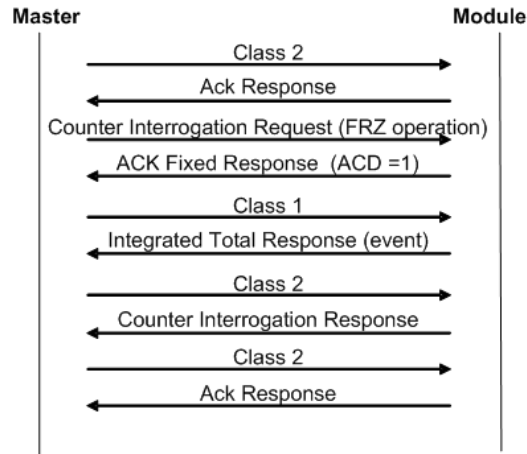
The following illustration shows a typical communication example when the Mode A is selected during unbalanced mode operation:



Mode D

To select the Mode D, configure the Freeze Start Type parameter as "N". For this mode the master would periodically send Counter Interrogation Commands to perform the freeze operation. After the values are frozen the module will return the counter points as events. The counter points must be properly configured for counter interrogation groups for Mode D operation.

The following illustration shows a typical communication example when the Mode D is selected during unbalanced mode operation:



Monitor Points Addressing

As discussed before, the monitor points must be configured in a database area in the MVI69-101S module.

The monitor data types are described in the following table.

Data Type	Data Size	Addressing Type
M_SP_NA	1 bit	Bit
M_DP_NA	2 bits	Bit
M_ST_NA	1 byte	Byte
M_ME_NA	1 word	Word
M_ME_NB	1 word	Word
M_ME_NC	2 word	Double-Word
M_IT_NA	2 word	Double-Word

M_SP_NA and M_DP_NA

The monitored single-point (1 bit) and monitored double-point (2 bits) types both occupy bit-addressing. For example, if you configured the following points:

```

# Point #   DB Address   Group(s)   IV DB Bit
# -----   -
START
    100      1600      80000000   0
    101      1601      00000200   0
    102      1602      00000400   0
END
  
```

These points would be used as follows:

Inf. Object Address	Module Database Address
100	Bit 0 of word 100
101	Bit 1 of word 100
102	Bit 2 of word 100

The monitored double-point uses two bits with bit-addressing. It typically represents the ON/OFF states where:

01 = OFF

10 = ON

M_ST_NA

The monitored step-point uses one byte with byte-addressing.

For example, if you configured the following points:

```
# Point #   DB Address   Group(s)   IV DB Bit
# -----   -
START
    300         40   80000000         0
    301         60   00000200         0
    302         81   00000400         0
END
```

These points would be used as follows:

Inf. Object Address	Module Database Address
300	Low Byte of word 20
301	Low Byte of word 30
302	High Byte of word 40

M_ME_NA and M_ME_NB

The monitored measured normalized and measured scaled points occupy one word with word-addressing.

For example, if you configured the following points:

```
#
# Point #   DB Address   Group(s)   Default Deadband   IV DB Bit
# -----   -
START
    400         10   80000000         0         0
    401         12   00000200         0         0
    402         18   00000400         0         0
END
```

These points would be used as follows:

Inf. Object Address	MVI69-101S Module Database Address
400	Word 10
401	Word 12
402	Word 18

The monitored measured normalized points use a data representation defined by the protocol specification, where each bit represents a value as follows:

Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Value	S	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸	2 ⁻⁹	2 ⁻¹⁰	2 ⁻¹¹	2 ⁻¹²	2 ⁻¹³	2 ⁻¹⁴	2 ⁻¹⁵

Example: a value of 4000hex is interpreted as 0.5

M_ME_NC and M_IT_NA

The monitored measured short floating point and monitored integrated total points occupy two words with double-word addressing.

For example, if you configured the following points:

# Point #	DB Address	Group(s)	Deadband	IV DB Bit
START				
500	20	80000000	0	0
501	32	00000200	0	0
502	52	00000400	0	0
END				

These points would be used as follows:

Inf. Object Address	Module Database Address
500	Words 40 and 41
501	Words 64 and 65
502	Word 104 and 105

5.3.5 Using Control (Command) Points

The following command points are supported by the module:

Symbol	Description	Data Size in Database	Addressing Type
C-SC-NA	Single-Point Command	1 bit	Bit
C-DC-NA	Dual-Point Command	2 bits	Bit
C-RC-NA	Step-Point Command	1 byte	Byte
C-SE-NA	Measured Normalized Point Command	1 word	Word
C-SE-NB	Measured Scaled-Point Command	1 word	Word
C-SE-NC	Measured Short-Floating Point Command	2 words	Double-Word

Each command point is identified by its Information Object Address (it should be unique for each Common ASDU Address in the network). For each monitor point, configure the following parameters:

Point # - This is the information object address of the point. It identifies the point in the network.

DB Address - This is the database location in the MVI69-101S module associated with the point. You must associate each point to a database address in the MVI69-101S. The interpretation of this parameter depends on the point type configured. For example, for an C_SC_NA point, this value represents the bit address. For a C_SE_NA point, this value represents the byte address.

Monitor Point # & Monitor DB Address- You can (optionally) configure a monitor point to be sent by the module when it receives the command for that specific point.

Example - Monitor Point Configuration for Command Points:

```
#
# Point #      DB Address      Monitor Point #  Monitor DB Addr  Require
# -----      -
START
      700          3200          100          1600          0
END
```

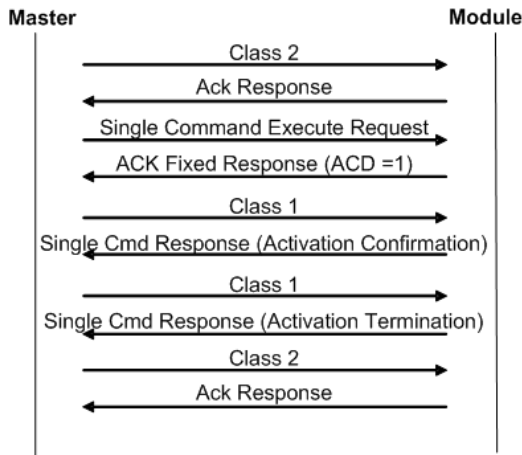
So every time the module receives a command for single-command point 700 it would send a response containing a monitored single-point (information object address 100 with the value at bit-address 1600).

Require Select - This parameter configures the point to require a Select request before the Operate command.

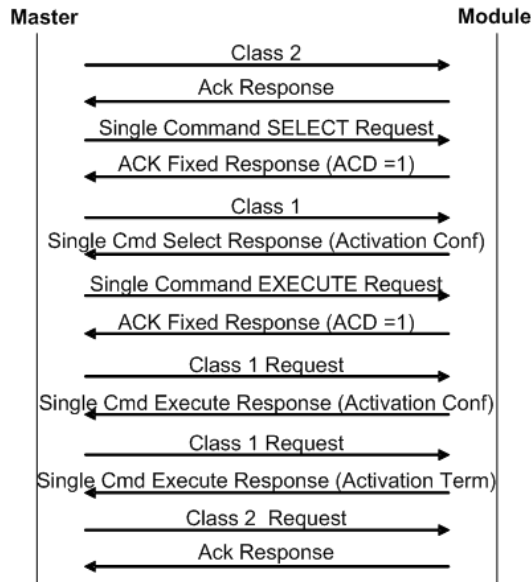
Control Data Transfer

The control communication typically occurs when the master sends a command request to update the module's command points.

The following illustration shows a typical command communication between the master and module during unbalanced operation.



Some of the command points may be configured to be selected before executed. The following illustration shows a typical command communication between the master and module during unbalanced operation with a SELECT operation.



Refer to the following parameter to configure the select/operate timeout period. After the module receives the SELECT operation it will wait for this period of time for the EXECUTE operation. If the module does not receive an EXECUTE operation within this period of time it will require another SELECT operation before the EXECUTE operation.

```
Select/Operate Timeout      : 20000 #Milliseconds before select
                               #timeout
```

Command Points Addressing

As discussed before, the command points must be configured in a database area that is updated at the module. You must associate each point to a database address in the MVI69-101S module. The interpretation of this parameter depends on the point type configured.

C_SC_NA and C_DC_NA

The single-point command and dual-point command points use one bit with bit-addressing. For example, if you configure the following points:

#	Point #	DB Address	Monitor Point #	Monitor DB Addr	Require Select
START	100	1600	0	0	0
	101	1601	0	0	0
	102	1602	0	0	0
END					

These points would be used as follows:

Inf. Object Address	Module Database Address
100	Bit 0 of word 100
101	Bit 1 of word 100
102	Bit 2 of word 100

The protocol specification defines a qualifier value that is set by the master to determine the duration of the pulse (short, long or persistent). Configure the parameters below to set the duration of the short and long pulses:

```
Short Pulse Time      : 2000 #MSec for short pulse command
Long Pulse Time       : 10000 #MSec for long pulse command
```

C_RC_NA

The step-point command uses one byte with byte-addressing.

For example, if you configured the following points:

```
#
# Point #      DB Address      Monitor Point #  Monitor DB Addr
# -----      -
START
      300          40           0             0
      301          60           0             0
      302          81           0             0
END
```

These points would be used as follows:

Inf. Object Address	Module Database Address
300	Low Byte of word 20
301	Low Byte of word 30
302	High Byte of word 40

C_SE_NA and C_SE_NB

The measured normalized point command uses one word with word-addressing. For example, if you configured the following points:

```
#
# Point #      DB Address      Monitor Point #  Monitor DB Addr  Require
# -----      -              -              -              -
START
      400          10           0             0             0
      401          12           0             0             0
      402          18           0             0             0
END
```

These points would be used as follows:

Inf. Object Address	MVI69-101S Module Database Address
400	Word 10
401	Word 12
402	Word 18

The measured normalized points use a data representation defined by the protocol specification, where each bit represents a value as follows:

Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Value	S	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸	2 ⁻⁹	2 ⁻¹⁰	2 ⁻¹¹	2 ⁻¹²	2 ⁻¹³	2 ⁻¹⁴	2 ⁻¹⁵

Example: a value of 4000hex is interpreted as 0.5

C_SE_NC

The measured short floating point command uses two words with double-word addressing. For example, if you configured the following points:

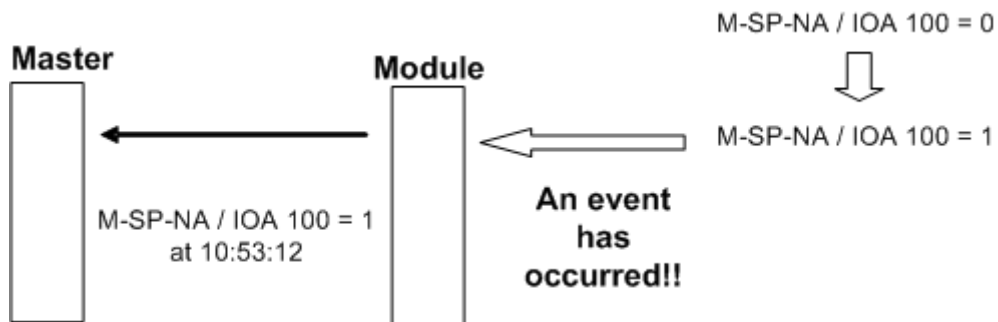
#	DB Address	Monitor	Monitor	Require
# Point #	(word*2)	Point #	DB Addr	Select
#	-----	-----	-----	-----
START				
400	10	0	0	0
401	12	0	0	0
402	18	0	0	0
END				

These points would be used as follows:

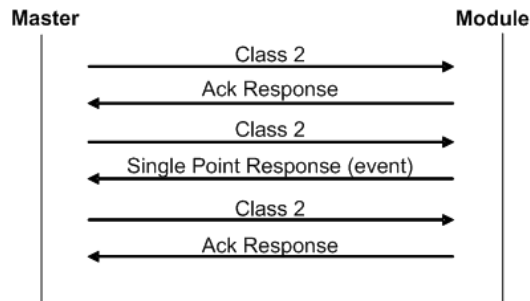
Inf. Object Address	MVI69-101S Module Database Address
400	Words 5 and 6
401	Words 6 and 7
402	Words 9 and 10

5.3.6 Events

In order to improve the communication efficiency, most applications will require the master to periodically poll for data changes with a higher priority than polling for monitor data. Every time data change occurs, the slave should send this information, typically with the date and time information on when the data change occurred.



The following illustration shows the event communication between the master and the module during unbalanced mode:



The module can queue up to 99 events per data type. When the queue is full, the oldest event will be deleted each time a new event is added to the queue. You must configure the master to poll the event queue frequently enough to avoid losing events.

The events can be returned for Class 1 or Class 2 requests according to the data type as follows:

Data Type	Class Request
M_SP_NA	Class 1
M_DP_NA	Class 1
M_ST_NA	Class 1
M_ME_NA	Class 2
M_ME_NB	Class 2
M_ME_NC	Class 2
M_IT_NA	Class 1

Note: In response to a Class 2 poll the module may respond with Class 1 data when there is no Class 2 data available. So eventually the events for M_SP_NA, M_DP_NA and M_ST_NA points may also be sent during a Class 2 response.

Deadbands

A deadband is a range of values within which the module will avoid generating events. The monitored measured points (M_ME_NA and M_ME_NB) will only generate events if the measurement varies by more than the configured deadband value.

For example, if the following point is configured:

```

#
# Point #   DB Address   Group(s)   Default   IV DB Bit
# -----   -
START
    500      105      80000000   100      0
END
    
```


So, if the current value for this point is 130, it would only generate events if:
NEW VALUE is less or equal than 30

OR

NEW VALUE is greater or equal than 230.

You can set the deadband for each monitored measured point through the configuration file.

The master may also dynamically change the deadband for each monitored point. The master may send one of the following commands:

Type	Command
110	Parameter of Measured Normalized Data (M_ME_NA)
111	Parameter of Measured Scaled Data (M_ME_NB)
112	Parameter of Measured Short Floating Point (M_ME_NC)

The protocol specification explains that the qualifier value for these commands should be configured as:

Bits	Value	Description
1 to 6	0	Not Used
	1	Threshold Value (Deadband)
	2	Smoothing Factor (filter time constant) - Not Supported
	3	Low Limit Transmission of Measured Value
	4	High Limit Transmission of Measured Value
	5..31	Reserved
7	0	No Change
	1	Change
8	0	Operation
	1	Not in Operation

The module calculates the Low Limit and High Limit values using the formula below (so these values cannot be changed dynamically)

Low Limit = (LAST REPORTED VALUE) - Deadband

High Limit = (LAST REPORTED VALUE) + Deadband

These commands must be sent to a specific Information Object Address. The module associates each monitor measured point with a parameter point through the following configuration parameters:

```
M_ME_NA Parameter Offset : 2000 #M_ME_NA IOA offset for parameter data
M_ME_NB Parameter Offset : 2000 #M_ME_NB IOA offset for parameter data
M_ME_NC Parameter Offset : 2000 #M_ME_NB IOA offset for parameter data
```

Example - Parameter Data for Monitored Points:

If the following monitored measured points are configured:

```
[M_ME_NA_1]
#
#                               Default
# Point # DB Address  Group(s)  Deadband  IV DB Bit
```

```

# -----
START
    400      100  80000000      100      0# P1 suction pressure
    401      101  00000200      100      0# P1 discharge pressure
    402      102  00000400      100      0# P2 suction pressure
    403      103  00000800      100      0# P2 discharge pressure
    404      104  00001000      100      0# Station discharge
                                           # pressure

END
[M_ME_NB_1]
#
#                               Default
# Point #   DB Address   Group(s)   Deadband   IV DB Bit
# -----
START
    500      105  80000000      100        0
    501      106  00000200      100        0
    502      102  00000400        1         0
    503      103  00000800        1         0
    504      104  00001000        1         0

END
    
```

And if the parameter points are configured as follows:

M_ME_NA Parameter Offset : 2000 #M_ME_NA IOA offset for parameter data
 M_ME_NB Parameter Offset : 2000 #M_ME_NB IOA offset for parameter data

It would imply that the parameter points would be configured as follows:

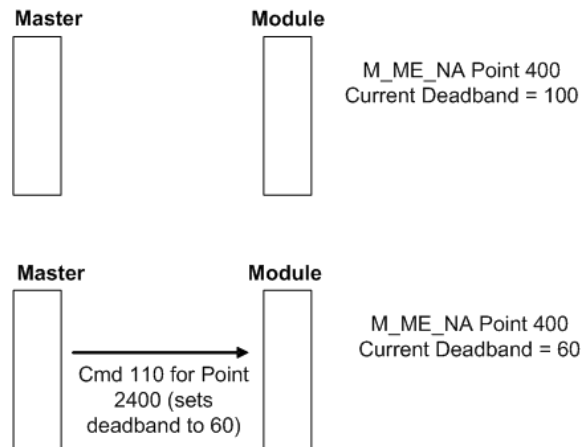
M_ME_NA

Monitored Measured Normalized Point	Associated Parameter Point
400	2400
401	2401
402	2402
403	2403
404	2404

M_ME_NB

Monitored Measured Normalized Point	Associated Parameter Point
500	2500
501	2501
502	2502
503	2503
504	2504

So, in order to send change the deadband for the M_ME_NA point 400, the master would send a command type 110 to point 2400:



Controlling the Generation of Events

Some applications may require that only some points should generate events. Other applications may require that all configured points should generate events.

The module offers considerable flexibility for event control. You can control how events will be generated at 3 different levels:

- 1 General (All Points)
- 2 Data Type Level
- 3 Point Level

General (All Points)

You can control how frequently the module will scan the database for events using the following configuration parameter:

```
Event Scan delay      :      1 #MSec between event scanning (0-65535)
                        #0=Disable
```

If this parameter is set to 0, the module will not generate events for any points. A non-zero value will configure how frequently the module will scan for events in the database.

Data Type Level

You can configure whether a data type should generate events. Each data type has a configuration parameter to control the generation of events:

```
M_SP_NA Scan Events :      1 #0=No scanning, 1=scan for events
M_DP_NA Scan Events :      0 #0=No scanning, 1=scan for events
M_ST_NA Scan Events :      0 #0=No scanning, 1=scan for events
M_ME_NA Scan Events :      0 #0=No scanning, 1=scan for events
M_ME_NB Scan Events :      0 #0=No scanning, 1=scan for events
M_ME_NC Scan Events :      0 #0=No scanning, 1=scan for events
```

In the example above, only the M_SP_NA points would generate events.

Point Level

You can configure whether each point should generate events using the Group field for each point configuration. To disable event generation for a specific port, set the value to 40000000.

```
[M_SP_NA_1 ]
#
#Point # DB Address Group(s) IV DB Bit
#-----
START
    100      1600  40000000          0 # P1-PSHH -- Discharge pressure SD
END
```

Time Information

Each event may also carry the date and time information when it occurred. The module supports the CP56 and CP24 time formats (as defined in the protocol specification).

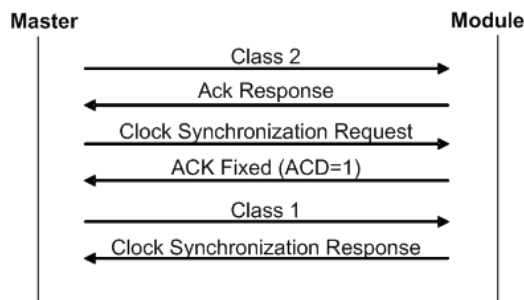
The CP56 time format contains the milliseconds, seconds, minute, hour, day, month and year when the event has occurred.

The CP24 time format contains the milliseconds and minutes when the event has occurred.

The module may also be configured to not send any time information with each event for certain data types. The following parameters may be used to control the time information for each data type:

- M_SP_NA Time Type : 2 #0=None, 1=CP24, 2=CP56 time
- M_DP_NA Time Type : 2 #0=None, 1=CP24, 2=CP56 time
- M_ST_NA Time Type : 2 #0=None, 1=CP24, 2=CP56 time
- M_ME_NA Time Type : 2 #0=None, 1=CP24, 2=CP56 time
- M_ME_NB Time Type : 2 #0=None, 1=CP24, 2=CP56 time
- M_ME_NC Time Type : 1 #0=None, 1=CP24, 2=CP56 time
- M_IT_NA Time Type : 2 #0=None, 1=CP24, 2=CP56 time

The master should periodically send a Time Synchronization command to the module in order to synchronize its date and time information. This is a very important step in order to make sure that the master and the slave are both using the same time information.



You can check the current date and time using the debug menu. Press **[E]** at the main menu to view the Module Status Data screen:

```

MVI46 - HyperTerminal
File Edit View Call Transfer Help
[Icons]

PRODUCT NAME CODE      : 87S4
SOFTWARE REVISION LEVEL : 1.29
OPERATING SYSTEM REVISION : 0504
RUN NUMBER              : 0301

Press ? for menu help.

IEC-870-5-101 STATUS DATA: 05/06/2004 16:01:38.646
Active Port = Primary Port
Errors: CKS= 0 Sync= 0 Len= 0 Tmout= 0
Status: Rx= 49 Tx= 49 Scn= 55524

EVENT BUFFER SPACE FREE:
MSPNA = 100
MDPNA = 100
MSTNA = 100
MMENA = 100
MMENB = 100
MMENC = 100
MITNA = 100

Connected 1:25:48 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

```

You can also configure the module to copy the date and time information to the module database with the following parameter:

```
Time DB Offset          : 2000 #DB location of IEC time (-1=ignore)
```

As the module sends events to the master (serial modules) or client (ethernet modules), it also sends a Clock Synchronization response (spontaneous COT) every hour for the transmission of the clock time to the master (serial modules) or client (ethernet modules).

Event Priority

Event Priority permits ASDUs that generate events to be placed in priority queues that are set by the user. The configuration file contains the following parameters to support this feature:

```

[IEC-870-5-101 Port 0]
Set Priority Queues    : 1 #Set user defined priority queues 1=Yes,
                        #0=No

M_SP_NA Priority      : 1 #Unique index for this data type in queue
                        #(0-5)
M_DP_NA Priority      : 0 #Unique index for this data type in queue
                        #(0-5)
M_ST_NA Priority      : 5 #Unique index for this data type in queue
                        #(0-5)
M_ME_NA Priority      : 4 #Unique index for this data type in queue
                        #(0-5)
M_ME_NB Priority      : 3 #Unique index for this data type in queue
                        #(0-5)
M_ME_NC Priority      : 2 #Unique index for this data type in queue
                        #(0-5)
M_IT_NA Priority      : 6 #Unique index for this data type in queue
                        #(0-5)

```

The Set Priority Queues parameter must be enabled for this feature to be used. Each of the ASDU's affected by this feature must be assigned a unique priority index from 0 to 6. Events of the ASDU with a priority of 0 will always be reported before any others when they are present.

Example - Event Priority

If the module is configured with the example values above, and the event queue contains the events generated in the following order:

Event Order	ASDU
1	M_SP_NA
2	M_SP_NA
3	M_DP_NA
4	M_ST_NA
5	M_DP_NA
6	M_SP_NA

The module will respond to a class one data request from the controlling station by returning the data in the event queue in the order shown in the following table.

Packet Order	Content
1	M_DP_NA events 3 and 5
2	M_SP_NA events 1, 2 and 6
3	M_ST_NA event 4

Note that the events are packed into messages in order to maximize the bandwidth on the network. The following warning must be considered when deciding to use this feature: Because events from the highest priority queues are always reported when present before lower priority queues, events in the lower queues may be lost due to buffer overflow.

If this feature is not utilized, each ASDU's events are stored in their own queue. The module will report each queue containing events in a round-robin fashion with all the data for each ASDU being packed. This methodology limits the possibility of a buffer overflowing and still maximizes the use of bandwidth on the communication channel.

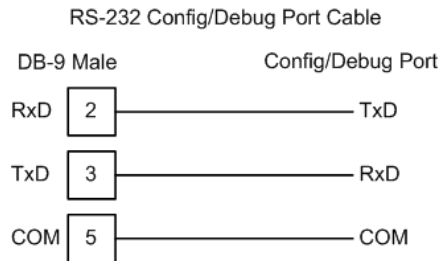
5.4 Cable Connections

The application ports on the MVI69-101S 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.

5.4.1 RS-232 Configuration/Debug Port

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

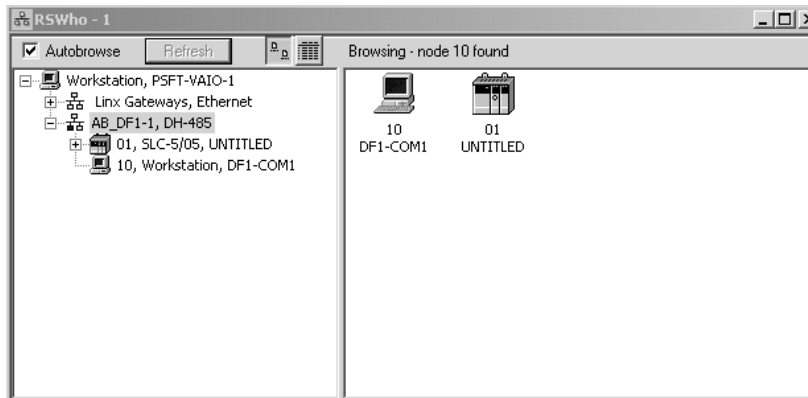


Disabling the RSLinx Driver for the Com Port on the PC

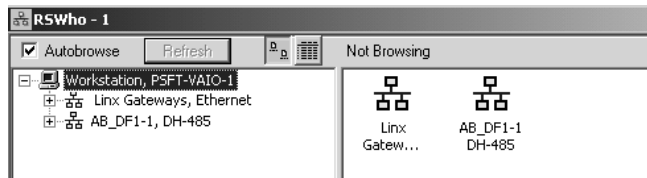
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable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



- 1 Open RSLinx and go to Communications>RSWho

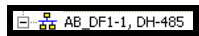
- 2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



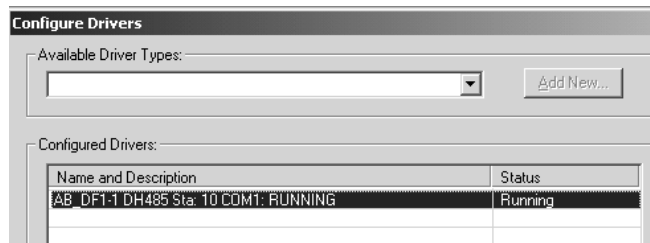
- 3 Notice how the DF1 driver is opened, and the driver is looking for a processor on node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your RSWHo screen should look like this:



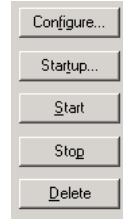
Branches are displayed or hidden by clicking on the  or the  icons.



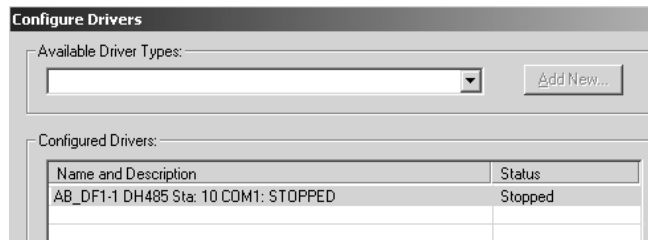
- 4 When you have verified that the driver is not being browsed, go to **Communications>Configure Drivers**
You may see something like this:



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



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

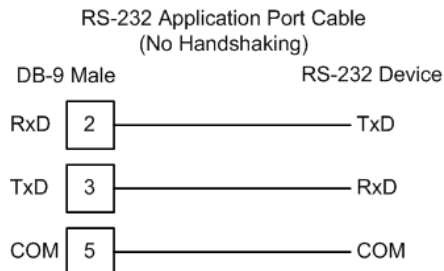


- 6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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

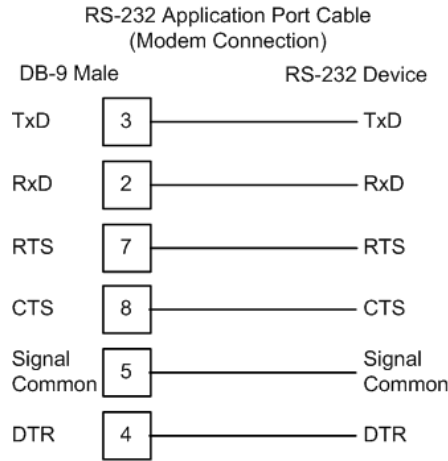
5.4.2 RS-232

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



RS-232: Modem Connection

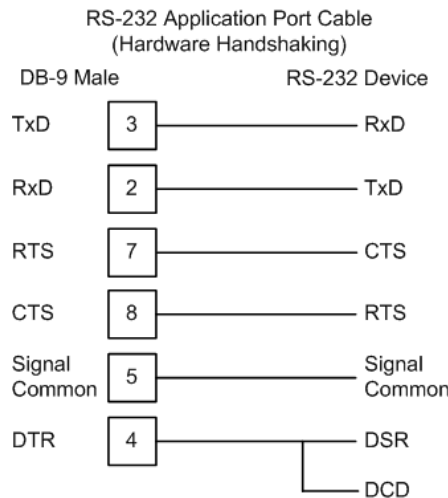
This type of connection is required between the module and a modem or other communication device.



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

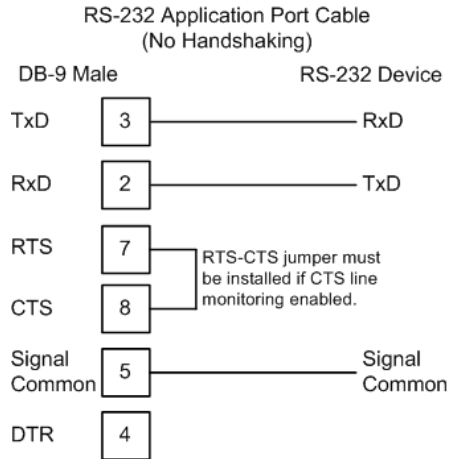
RS-232: Null Modem Connection (Hardware Handshaking)

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

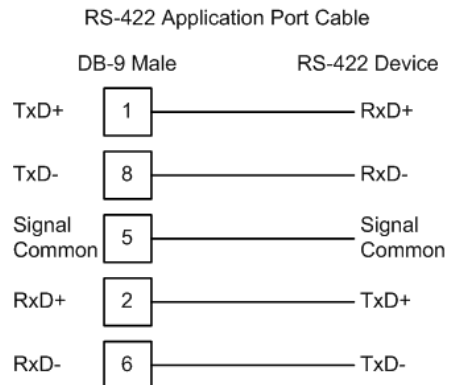


RS-232: Null Modem Connection (No Hardware Handshaking)

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

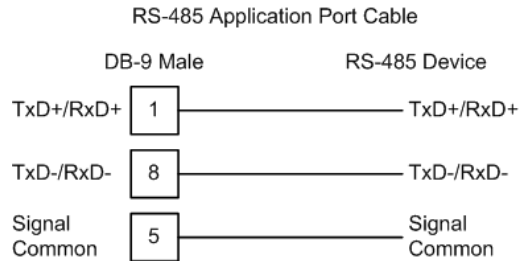


Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.4.3 RS-422

5.4.4 RS-485

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

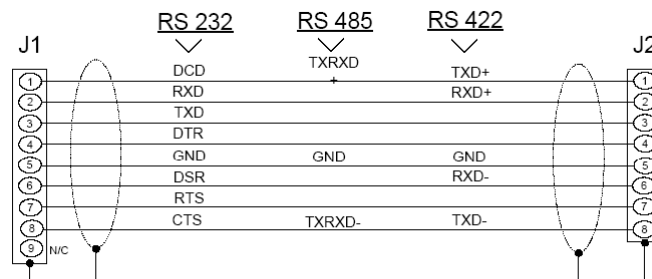
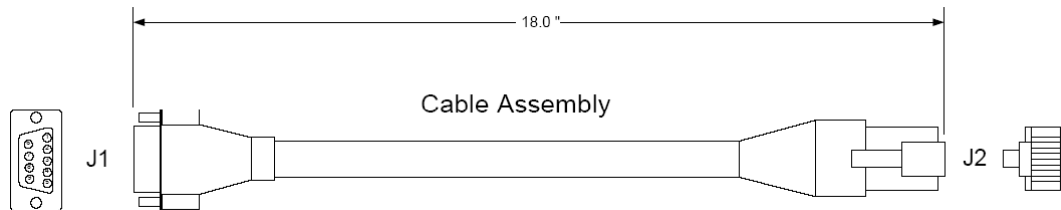


Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

RS-485 and RS-422 Tip

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

5.4.5 DB9 to RJ45 Adaptor (Cable 14)



Wiring Diagram

5.5 MVI69-101S Status Data Definition

This section contains a description of the members present in the **InStat870** object. This data is transferred from the module to the processor as part of each read block.

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 MVI69 platform.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
14	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
15	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
16	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
17	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
18	RX Frames	This word contains the total number of frames received by the module.
19	TX Frames	This word contains the total number of frames transmitted by the module.

5.6 MVI69-101S Database Design Forms

This section contains a set of forms that can be used to design the databases required by the module.

Form for use with the data types **M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 and M_IT_NA_1**

Point Number	Database Address	Group Assignment

Form to be used with data types M_ME_NA_1, M_ME_NB_1 and M_ME_NC_1

Point Number	Database Address	Group Assignment	Default Deadband

Forms to be used with all command data types except C_RC_NA_1

Point Number	Database Address	Monitor Point #	Monitor Database Address	Require Select

Form to be used with C_RC_NA_1 data type

Point Number	Database Address	Monitor Point #	Monitor Database Address

5.7 MVI69-101S Error Status Table

This section contains a listing of the MVI69-101S module's status data area. This file is located at the MVI69-101S database starting at address 4000. You may also configure an additional area using the "Error Offset" parameter.

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.
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	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
14	Bad CRC	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
15	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
16	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
17	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
18	RX Frames	This word contains the total number of frames received by the module.
19	TX Frames	This word contains the total number of frames transmitted by the module.
20	MSP Event Buffer	This parameter shows the number of events available in the event buffer for M_SP_NA points.
21	MDP Event Buffer	This parameter shows the number of events available in the event buffer for M_DP_NA points
22	MST Event Buffer	This parameter shows the number of events available in the event buffer for M_ST_NA points
23	MMENA Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NA points

Offset	Parameter	Description
24	MMENB Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NB points
25	MMENC Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NC points.
26	MIT Event Buffer	This parameter shows the number of events available in the event buffer for M_IT_NA points

5.8 IEC 60870-5-101 Slave Interoperability Document

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 originating from different manufacturers it is necessary that all partners agree on the selected parameters.

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

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

- Function or ASDU is not used
- Function or ASDU is used as standardized (default)

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

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	Unbalanced interchange circuit V.24/V.28	Balanced interchange circuit X.24/X.27
Standard	Recommended if >1 200 bit/s	

- | | | | |
|---|--|---|--------------------------|
| <input type="checkbox"/> 100 bit/s
56000 bit/s | <input checked="" type="checkbox"/> 2400 bit/s | <input checked="" type="checkbox"/> 2400 bit/s | <input type="checkbox"/> |
| <input type="checkbox"/> 200 bit/s
64000 bit/s | <input checked="" type="checkbox"/> 4800 bit/s | <input checked="" type="checkbox"/> 4800 bit/s | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> 300 bit/s | <input checked="" type="checkbox"/> 9600 bit/s | <input checked="" type="checkbox"/> 9600 bit/s | |
| <input checked="" type="checkbox"/> 600 bit/s | | <input checked="" type="checkbox"/> 19200 bit/s | |
| <input checked="" type="checkbox"/> 1200 bit/s | | <input checked="" type="checkbox"/> 38400 bit/s | |

Transmission speed (monitor direction)

Unbalanced interchange circuit V.24/V.28 Unbalanced interchange circuit V.24/V.28 Balanced interchange circuit X.24/X.27

Standard Recommended if >1 200 bit/s

- | | | | |
|---|--|---|--------------------------|
| <input type="checkbox"/> 100 bit/s
56000 bit/s | <input checked="" type="checkbox"/> 2400 bit/s | <input checked="" type="checkbox"/> 2400 bit/s | <input type="checkbox"/> |
| <input type="checkbox"/> 200 bit/s
64000 bit/s | <input checked="" type="checkbox"/> 4800 bit/s | <input checked="" type="checkbox"/> 4800 bit/s | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> 300 bit/s | <input checked="" type="checkbox"/> 9600 bit/s | <input checked="" type="checkbox"/> 9600 bit/s | |
| <input checked="" type="checkbox"/> 600 bit/s | | <input checked="" type="checkbox"/> 19200 bit/s | |
| <input checked="" type="checkbox"/> 1200 bit/s | | <input checked="" type="checkbox"/> 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

- | | |
|---|--|
| <input checked="" type="checkbox"/> Balanced transmission | <input checked="" type="checkbox"/> Not present (balanced transmission only) |
| <input checked="" type="checkbox"/> Unbalanced transmission | <input checked="" type="checkbox"/> One octet |
| | <input checked="" type="checkbox"/> Two octets |
| | <input checked="" type="checkbox"/> Structured |

Frame length

- Unstructured

255 Maximum length L (number of octets) See Note 3.

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)

<input checked="" type="checkbox"/>	<1>	:= Single-point information	M_SP_NA_1
<input checked="" type="checkbox"/>	<2>	:= Single-point information with time tag	M_SP_TA_1
<input checked="" type="checkbox"/>	<3>	:= Double-point information	M_DP_NA_1
<input checked="" type="checkbox"/>	<4>	:= Double-point information with time tag	M_DP_TA_1
<input checked="" type="checkbox"/>	<5>	:= Step position information	M_ST_NA_1
<input checked="" type="checkbox"/>	<6>	:= Step position information with time tag	M_ST_TA_1
<input type="checkbox"/>	<7>	:= Bitstring of 32 bit	M_BO_NA_1
<input type="checkbox"/>	<8>	:= Bitstring of 32 bit with time tag	M_BO_TA_1
<input checked="" type="checkbox"/>	<9>	:= Measured value, normalized value M_ME_NA_1	
<input checked="" type="checkbox"/>	<10>	:= Measured value, normalized value with time tag	M_ME_TA_1
<input checked="" type="checkbox"/>	<11>	:= Measured value, scaled value	M_ME_NB_1
<input checked="" type="checkbox"/>	<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 description
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 CP56Time2a M_EP_TF_1

Process information in control direction
 (Station-specific parameter)

- <45> := Single command C_SC_NA_1

<input checked="" type="checkbox"/>	<46>	:= Double command	C_DC_NA_1
<input checked="" type="checkbox"/>	<47>	:= Regulating step command	C_RC_NA_1
<input checked="" type="checkbox"/>	<48>	:= Set point command, normalized value	C_SE_NA_1
<input checked="" type="checkbox"/>	<49>	:= Set point command, scaled value	C_SE_NB_1
<input checked="" type="checkbox"/>	<50>	:= Set point command, short floating point value	C_SE_NC_1
<input type="checkbox"/>	<51>	:= Bitstring of 32 bit	C_BO_NA_1

System information in monitor direction

(Station-specific parameter)

<input checked="" type="checkbox"/>	<70>	:= End of initialization	M_EI_NA_1
-------------------------------------	------	--------------------------	-----------

System information in control direction

(Station-specific parameter)

<input checked="" type="checkbox"/>	<100>	:= Interrogation command	C_IC_NA_1
<input checked="" type="checkbox"/>	<101>	:= Counter interrogation command	C_CI_NA_1
<input checked="" type="checkbox"/>	<102>	:= Read command	C_RD_NA_1
<input checked="" type="checkbox"/>	<103>	:= Clock synchronization command	C_CS_NA_1
<input checked="" type="checkbox"/>	<104>	:= Test command	C_TS_NB_1
<input checked="" type="checkbox"/>	<105>	:= Reset process command	C_RP_NC_1
<input checked="" type="checkbox"/>	<106>	:= Delay acquisition command ^{Note 1}	C_CD_NA_1

Parameter in control direction

(Station-specific parameter)

<input checked="" type="checkbox"/>	<110>	:= Parameter of measured value, normalized value	P_ME_NA_1
<input checked="" type="checkbox"/>	<111>	:= Parameter of measured value, scaled value	P_ME_NB_1
<input type="checkbox"/>	<112>	:= Parameter of measured value, short floating point value	P_ME_NC_1
<input type="checkbox"/>	<113>	:= Parameter activation	P_AC_NA_1

File transfer

(Station-specific parameter)

<input type="checkbox"/>	<120>	:= File ready	F_FR_NA_1
<input type="checkbox"/>	<121>	:= Section ready	F_SR_NA_1
<input type="checkbox"/>	<122>	:= Call directory, select file, call file, call section	F_SC_NA_1
<input type="checkbox"/>	<123>	:= Last section, last segment	F_LS_NA_1
<input type="checkbox"/>	<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 and execute set point command
- C_SE_ACTTERM used ^{Note 2}
- No additional definition
- Short pulse duration (duration determined by a system parameter in the outstation) ^{Note 4}
- Long pulse duration (duration determined by a system parameter in the outstation) ^{Note 4}
- 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

Note 1: Delay acquisition command supports Load Delay only.

Note 2: C_SE_ACTTERM may be enabled or disabled at time of installation.

Note 3: The ASDU length can be configured by the user (between 25 and 252).

Note 4: Only applies to C_SC_NA_1 and C_DC_NA_1 commands.

Note 5: The low limits and high limit values are calculated based on the Deadband values as follows:

Low Limit: Last reported event value - threshold

High Limit: Last reported event value + threshold

6 Support, Service & Warranty

In This Chapter

- ❖ How to Contact Us: Technical Support..... 139
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 140
- ❖ LIMITED WARRANTY..... 141

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

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

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

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

6.1 How to Contact Us: Technical Support

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

Asia Pacific

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

Europe (location in Toulouse, France)

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

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

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

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

Brasil (location in Sao Paulo)

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

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns:

- a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
- b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 139). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
- c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, etc.

6.2.2 Procedures for Return of Units Under Warranty:

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

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

6.2.3 Procedures for Return of Units Out of Warranty:

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

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

The following is a list of non-repairable units:

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

6.2.4 Purchasing Warranty Extension:

- a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 141)". The Warranty Period may be extended at the time of equipment purchase for an additional charge, as follows:
- Additional 1 year = 10% of list price
 - Additional 2 years = 20% of list price
 - Additional 3 years = 30% of list price

6.3 LIMITED WARRANTY

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

6.3.1 What Is Covered By This Warranty

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

6.3.2 What Is Not Covered By This Warranty

- a) ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.
- b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.

- c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

Product manufactured or supplied by ProSoft is not fault tolerant and is not designed, manufactured or intended for use in hazardous environments requiring fail-safe performance including and without limitation: the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly or indirectly to death, personal injury or severe physical or environmental damage (collectively, "high risk activities"). ProSoft specifically disclaims any express or implied warranty of fitness for high risk activities.

6.3.4 Intellectual Property Indemnity

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

- a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated or reproduced in any form without prior written consent from ProSoft.
- b) ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
- c) Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
- d) Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.

- e) Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.
- f) **Additional Restrictions Relating To Software And Other Intellectual Property**

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

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 142) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

6.3.6 Limitation of Remedies **

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

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

6.3.7 Time Limit for Bringing Suit

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

6.3.8 No Other Warranties

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

6.3.9 Allocation of Risks

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

6.3.10 Controlling Law and Severability

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

Index

I

[Backplane Configuration] • 26
 [C_DC_NA_1] • 41
 [C_RC_NA_1] • 41
 [C_SC_NA_1] • 40
 [C_SE_NA_1] • 42
 [C_SE_NB_1] • 42
 [C_SE_NC_1] • 43
 [IEC-870-5-101 Database] • 35
 [IEC-870-5-101 Port 0] • 28
 [M_DP_NA_1] • 38
 [M_IT_NA_1] • 40
 [M_ME_NA_1] • 38
 [M_ME_NB_1] • 39
 [M_ME_NC_1] • 39
 [M_SP_NA_1] • 37
 [M_ST_NA_1] • 38

A

Adding a Module • 24
 Adding a Project • 23
 Adding the Module to an Existing CompactLogix Project • 49
 Adding the Module to an Existing MicroLogix Project • 53
 All Product Returns • 140
 Allocation of Risks • 145
 Analyzing Data for the first application port • 66
 Analyzing Data for the second application port • 66
 ASDU Configuration • 92

B

Backplane Data Transfer • 76
 Backup Port Baud Rate • 34
 Backup Port Enabled • 34
 Backup Port Handshaking • 35
 Backup Port Min Delay • 34
 Backup Port Parity • 34
 Backup Port RTS Off • 34
 Backup Port RTS On • 34
 Balanced Mode • 96
 Battery Life Advisory • 3
 Baud Rate • 33
 Block Control Data - MVI69101S_UTIL • 49
 Block Format for Read • 85
 Block Format for Write • 85, 86
 Block Transfer Size • 28

C

C_RC_NA • 110
 C_SC_NA and C_DC_NA • 109
 C_SE_NA and C_SE_NB • 110

C_SE_NC • 111
 Cable Connections • 119
 Cause of Trans Octets • 30
 Clearing a Fault Condition • 71
 Cold Boot • 83, 87
 Command Control Blocks • 83
 Command Points Addressing • 109
 Comment Entries • 26
 Common address of ASDU • 29
 Common address of ASDU Len • 29
 Communication Parameters • 92
 CompactLogix or MicroLogix Processor Not in Run • 76
 Configuring RSLinx • 17
 Configuring the MVI69-101S Module • 21
 Connect your PC to the Module • 19
 Connect your PC to the Processor • 16
 Control Data Transfer • 108
 Controlling Law and Severability • 145
 Controlling the Generation of Events • 115
 Cyclic Data Transmission • 29
 Cyclic Set IV Time • 31

D

Data Analyzer • 66
 Data Analyzer Tips • 68
 Data Flow Between MVI69-101S Module and CompactLogix or MicroLogix Processor • 87
 Data Link address • 28
 Data Link address Length • 28
 Data Link Configuration • 92
 Data Type Level • 115
 Database View Menu • 64
 Databases • 88
 DB9 to RJ45 Adaptor (Cable 14) • 124
 Deadbands • 112
 Diagnostics and Troubleshooting • 7, 55, 57
 Disabling the RSLinx Driver for the Com Port on the PC • 119
 Disclaimer of all Other Warranties • 144
 Disclaimer Regarding High Risk Activities • 143
 Displaying the Current Page of Registers Again • 65
 Displaying Timing Marks in the Data Analyzer • 67
 Download the Project to the Module • 43
 Download the Sample Program to the Processor • 16

E

Enabled • 28
 Error Offset • 27
 Event Message Block (Block 9958) • 83, 84
 Event Priority • 31, 117
 Event Scan Delay • 32
 Events • 111
 Exiting the Program • 64

F

Failure Flag Count • 27
 Freeze Start Type • 30
 Functional Overview • 7, 75
 Functional Specifications • 74

G

General (All Points) • 115
General Concepts • 75
General Parameter Configuration • 92
General Specifications • 74
Get Module Time (Block 9970) • 83, 85
Guide to the MVI69-101S User Manual • 7

H

Hardware Specifications • 74
How Data is Transferred • 81
How to Contact Us
 Technical Support • 139, 140

I

IEC 60870-5-101 Slave Interoperability Document • 131
IEC-60870-5-101 Protocol Implementation • 7, 91
Important Installation Instructions • 2
Inform. Object address Len • 29
Install ProSoft Configuration Builder Software • 10
Install the Module in the Rack • 12
Intellectual Property Indemnity • 143
Interval for Freeze • 31
Introduction • 84
IV Check Delay Time • 31
IV Fail Count • 31, 32

K

Keystrokes • 60

L

Ladder Logic • 45
LED Status Indicators • 7, 71
Limitation of Remedies ** • 144
LIMITED WARRANTY • 141
Long Pulse Time • 35

M

M_ME_NA • 114
M_ME_NA and M_ME_NB • 106
M_ME_NB • 114
M_ME_NC and M_IT_NA • 107
M_SP_NA and M_DP_NA • 105
M_ST_NA • 106
Main Logic Loop • 76
Main Menu • 60
Maximum ASDU Resp Len • 30
Minimum Delay • 33
Module Data Object
 MVI69101S_ModuleDef • 45
Module Entries • 25
Module Initialization • 95
Module Name • 27
Module Status Object
 MVI69101S_STATUS • 48
Monitor Data Transfer • 100
Monitor Direction and Control Direction • 96

Monitor Points Addressing • 105
Moving Back Through 5 Pages of Registers • 65
MVI (Multi Vendor Interface) Modules • 2
MVI69-101S Database Design Forms • 126
MVI69-101S Error Status Table • 130
MVI69-101S Status Data Definition • 125
MVI69101S_Clock • 47
MVI69101S_CONTROL • 47
MVI69101S_Event • 47
MVI69101S_EventBlock • 47

N

Navigation • 59
No Other Warranties • 145
Normal Data Transfer • 78

O

Opening the Data Analyzer Menu • 61
Opening the Database Menu • 61

P

Package Contents • 10
Parameter Offset • 37
Parity • 33
Pinouts • 2, 119, 124
Point Count • 35
Point Level • 116
Printing a Configuration File • 26
Procedures for Return of Units Out of Warranty: • 140
Procedures for Return of Units Under Warranty: • 140
Product Specifications • 7, 73
ProSoft Configuration Builder • 21
ProSoft® Product Documentation • 3
Purchasing Warranty Extension: • 141

R

Read Block • 78
Read Register Count • 27
Read Register Start • 27
Reading Status Data from the Module • 57
Receive Timeout • 34
Receiving the Configuration File • 61
Reference • 7, 73
Removing Timing Marks in the Data Analyzer • 67
Request Output Data from Processor • 83, 84
Required Hardware • 57
Response Timeout • 33
Retry Count • 33
Return Material Authorization (RMA) Policies and Conditions • 140
Returning to the Main Menu • 66, 68
RS-232 • 121
 Modem Connection • 122
 Null Modem Connection (Hardware Handshaking) • 122
 Null Modem Connection (No Hardware Handshaking) • 123
RS-232 Configuration/Debug Port • 119
RS-422 • 123
RS-485 • 124

RS-485 and RS-422 Tip • 124
RTS Off • 33
RTS On • 33

S

Scan Events • 32
Select/Operate Timeout • 29
Sending Events • 83
Sending the Configuration File • 62
Sequence Flag • 36
Set Module Parameters • 25
Set Module Time (Block 9971) • 83, 86
Set Priority Queues • 31
Set Up the Project • 22
Setting Jumpers • 12
Short Pulse Time • 35
Single Char ACK C1 or C2 • 30
Single Char ACK F0, 1 or 3 • 30
Skipping 500 Registers of Data • 65
Slave Driver • 88
Start Here • 7, 9
Starting the Data Analyzer • 67
Status Block • 82
Stopping the Data Analyzer • 68
Support, Service & Warranty • 7, 139
System Requirements • 9

T

The Configuration/Debug Menu • 58
Time DB Offset • 28
Time Information • 116
Time Limit for Bringing Suit • 144
Time Type • 32
Troubleshooting • 72

U

Unbalanced mode • 95
Use ACTTERM with Setpoint • 29
Use ACTTERM with Step • 30
Use Balanced Mode • 32
User Data Objects - MV69101S_DATA • 45
Using Control (Command) Points • 107
Using Monitor Points • 31, 97
Using the Diagnostic Window in ProSoft Configuration
Builder • 58

V

Viewing Block Transfer Statistics • 61
Viewing Data in ASCII (Text) Format • 66, 67
Viewing Data in Decimal Format • 65
Viewing Data in Floating Point Format • 66
Viewing Data in Hexadecimal Format • 65, 67
Viewing Data Type Setup • 62
Viewing Error and Status Data • 61
Viewing IEC 60870 Database Configuration • 64
Viewing Module Configuration • 61
Viewing Port Configuration • 61
Viewing Register Pages • 65
Viewing the Next 100 Registers of Data • 65
Viewing the Previous 100 Registers of Data • 65

Viewing Version Information • 62

W

Warm Boot • 83, 87
Warm Booting the Module • 62, 83
Warnings • 2
What Is Covered By This Warranty • 142, 144
What Is Not Covered By This Warranty • 142
Write Block • 81
Write Register Start • 27

Y

Your Feedback Please • 3