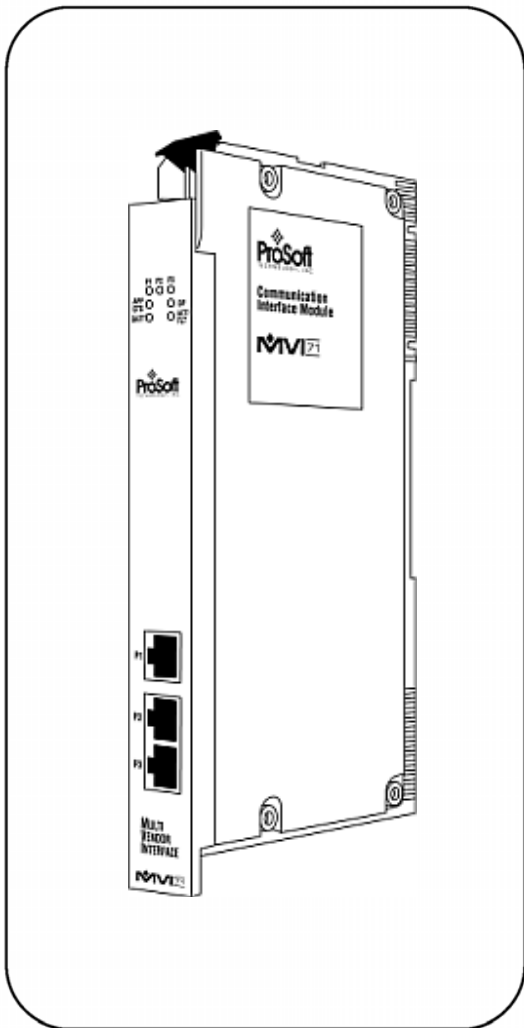


# inRAx



## MVI71-MCM

PLC Platform

Modbus Communication Module

## User Manual

December 28, 2007

  
**ProSoft**  
TECHNOLOGY

## Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation PLC hardware, the MVI71-MCM Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

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

**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.

Power, Input, and Output (I/O) wiring must be in accordance with Class 1, 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.

- A** Warning - Explosion Hazard - Substitution of components may impair suitability for Class 1, Division 2.
- B** Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
- C** Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

## **Your Feedback Please**

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

### **ProSoft Technology**

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

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

MVI71-MCM User Manual  
December 28, 2007  
PSFT.MCM.MVI71.UM.07.12.28

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



# Contents

<b>PLEASE READ THIS NOTICE</b> .....	<b>2</b>
Your Feedback Please .....	3
<b>GUIDE TO THE MVI71-MCM USER MANUAL</b> .....	<b>7</b>
<b>1 START HERE</b> .....	<b>9</b>
1.1 System Requirements.....	9
1.2 Package Contents .....	10
1.3 Setting Jumpers.....	11
1.4 Install the Module in the Rack.....	12
1.5 Connect your PC to the Processor .....	13
1.6 Download the Sample Program to the Processor .....	14
1.6.1 Configuring RSLinx .....	15
1.7 Connect your PC to the Module.....	17
<b>2 INSTALLING AND CONFIGURING THE MODULE</b> .....	<b>19</b>
2.1 Quick Start .....	19
2.2 Configuring the Module.....	22
2.2.1 Power Up .....	22
2.2.2 Configuration Data Transfer .....	23
2.2.3 Changing Parameters During Operation .....	24
2.2.4 Module Data Files .....	24
2.2.5 User Data .....	29
2.2.6 Slave Polling Control and Status .....	29
2.2.7 Using Side-Connect (Requires Side-Connect Adapter).....	29
<b>3 LADDER LOGIC</b> .....	<b>33</b>
<b>4 DIAGNOSTICS AND TROUBLESHOOTING</b> .....	<b>35</b>
4.1 Reading Status Data From the Module.....	35
4.1.1 The Configuration/Debug Menu .....	35
4.1.2 Required Hardware .....	36
4.1.3 Required Software.....	36
4.1.4 Using the Configuration/Debug Port.....	37
4.1.5 Main Menu .....	38
4.1.6 Data Analyzer.....	41
4.1.7 Data Analyzer Tips .....	43
4.1.8 Database View Menu .....	46
4.1.9 Master Command Error List Menu .....	48
4.1.10 Master Command List Menu .....	49
4.2 LED Status Indicators .....	50
4.2.1 Clearing a Fault Condition.....	51
4.2.2 Troubleshooting.....	51
<b>5 REFERENCE</b> .....	<b>53</b>
5.1 Product Specifications .....	53
5.1.1 Features and Benefits .....	53
5.1.2 General Specifications .....	54
5.1.3 Hardware Specifications.....	54
5.1.4 Functional Specifications.....	55
5.2 Functional Overview .....	55
5.2.1 General Concepts .....	55
5.2.2 Main Logic Loop .....	56
5.2.3 PLC Processor Not in Run .....	56
5.2.4 Backplane Data Transfer.....	57
5.2.5 Modbus Addressing.....	59
5.2.6 Using the Read and Write Data Areas .....	59
5.2.7 Normal Data Transfer.....	63

5.2.8	Slave Status Blocks .....	67
5.2.9	Command Control Blocks .....	69
5.2.10	Pass-Through Control Blocks.....	74
5.2.11	Remote Command Control.....	75
5.2.12	Data Flow Between MVI71-MCM Module and PLC Processor .....	76
5.3	Cable Connections .....	79
5.3.1	RS-232 Configuration/Debug Port.....	79
5.3.2	RS-232.....	81
5.3.3	RS-422.....	83
5.3.4	RS-485.....	84
5.3.5	DB9 to RJ45 Adaptor (Cable 14).....	84
5.4	MVI71-MCM Database Definition .....	85
5.5	MVI71-MCM Status Data Definition.....	85
5.6	MVI71-MCM Configuration Data Definition.....	87
5.6.1	Backplane Setup .....	87
5.6.2	Port 1 Setup .....	88
5.6.3	Port 2 Setup .....	90
5.6.4	Port 1 Commands .....	93
5.6.5	Port 2 Commands .....	93
5.6.6	Command Control.....	94
5.7	MVI71-MCM Command Error Codes.....	94
5.7.1	Standard Modbus Protocol Errors.....	94
5.7.2	Module Communication Error Codes .....	94
5.7.3	Command List Entry Errors .....	94
5.8	Configuration Error Codes .....	95
5.8.1	BTR.....	95
5.9	Frequently Asked Questions.....	96
5.9.1	What is the MVI71-MCM? .....	96
5.9.2	Why does the module have to break down the data in blocks of 64 words? .....	96
5.9.3	How do I read and write data between the internal MVI71-MCM database and the PLC?.....	96
5.9.4	What is the procedure to change the module's configuration?.....	97
5.9.5	How do I set a Port Modbus Master command? .....	97
5.9.6	The Modbus Master Command is not working, what should I do? .....	97
5.9.7	What is the Pass-Through feature?.....	97
5.9.8	How can I check a Slave's status connected to an MVI71-MCM Master port? .....	97
5.9.9	Should I use the sample ladder logic? .....	98
5.9.10	Is it possible to control the Master Modbus Commands from ladder logic? .....	98
5.9.11	How is the MVI71-MCM database divided into Discrete Output, Discrete Input, Input Registers, and Holding Registers? .....	98
<b>6</b>	<b>SUPPORT, SERVICE &amp; WARRANTY.....</b>	<b>99</b>
6.1	How to Contact Us: Sales and Support .....	100
6.2	Return Material Authorization (RMA) Policies and Conditions.....	101
6.2.1	All Product Returns .....	101
6.3	Procedures for Return of Units Under Warranty .....	101
6.4	Procedures for Return of Units Out of Warranty .....	102
6.4.1	Un-repairable Units .....	102
6.4.2	Purchasing Warranty Extension .....	103
6.5	LIMITED WARRANTY .....	103
6.5.1	What Is Covered By This Warranty .....	103
6.5.2	What Is Not Covered By This Warranty .....	104
6.5.3	DISCLAIMER REGARDING HIGH RISK ACTIVITIES.....	105
6.5.4	DISCLAIMER OF ALL OTHER WARRANTIES.....	105
6.5.5	LIMITATION OF REMEDIES** .....	105
6.5.6	Time Limit for Bringing Suit .....	105
6.5.7	No Other Warranties .....	106
6.5.8	Intellectual Property .....	106
6.5.9	Additional Restrictions Relating To Software And Other Intellectual Property .....	106
6.5.10	Allocation of risks .....	107
6.5.11	Controlling Law and Severability.....	107
<b>INDEX.....</b>		<b>109</b>

## Guide to the MVI71-MCM User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	<a href="#">Start Here</a> (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	→	<a href="#">Verifying Communication</a> (page 35) <a href="#">Diagnostics and Troubleshooting</a> (page 35)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	<a href="#">Reference</a> (page 53) <a href="#">Functional Overview</a> (page 55) <a href="#">Product Specifications</a> (page 53)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	<a href="#">Support, Service and Warranty</a> (page 99)	This section contains Support, Service and Warranty information. Index of chapters.





# 1 Start Here

## *In This Chapter*

- System Requirements..... 9
- Package Contents..... 10
- Setting Jumpers ..... 11
- Install the Module in the Rack ..... 12
- Connect your PC to the Processor..... 13
- Download the Sample Program to the Processor ..... 14
- Connect your PC to the Module ..... 17

Installing the MVI71-MCM module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI71-MCM 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 MVI71-MCM module requires the following minimum hardware and software components:

- Rockwell Automation PLC processor, with compatible power supply and one free slot in the rack, for the MVI71-MCM module. The module requires 800mA of available power.
- The PLC Processor must provide for at least 64 words of BTR/BTW area, otherwise the module may not function correctly.
- Rockwell Automation RSLogix 5 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® 100 MHz minimum. Pentium III 700 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP
  - Microsoft Windows 2000
  - Microsoft Windows NT v4.0 with Service Pack 3 or greater
  - Microsoft Windows ME
  - Microsoft Windows 98

- 64 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
- 3.5 inch floppy disk drive
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

## 1.2 Package Contents

The following components are included with your MVI71-MCM module, and are all required for installation and configuration.

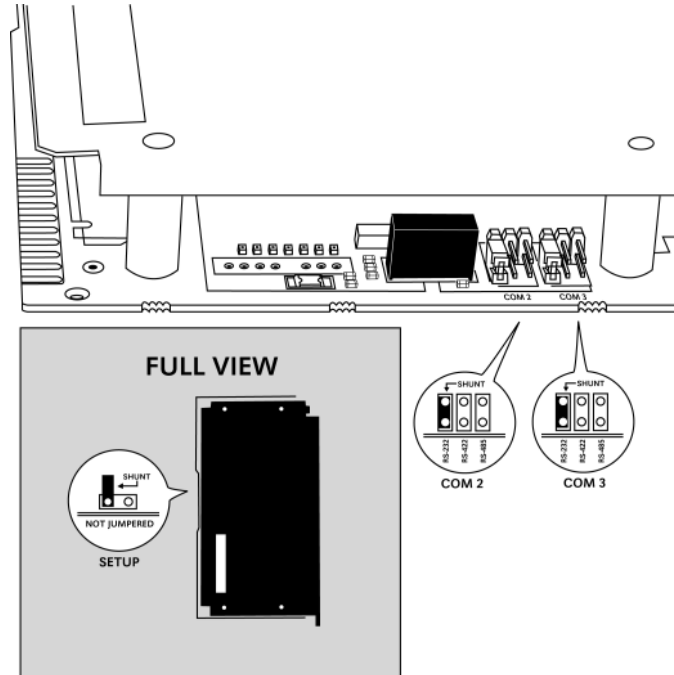
**Important:** Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI71-MCM Module	MVI71-MCM	Modbus Communication Module
1	Cable	RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI71-MCM module.

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

### 1.3 Setting Jumpers

The following illustration shows the jumper configurations for the various RS interfaces. If you are using an interface other than RS-232 (default), you must change the jumpers as shown:



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

## 1.4 Install the Module in the Rack

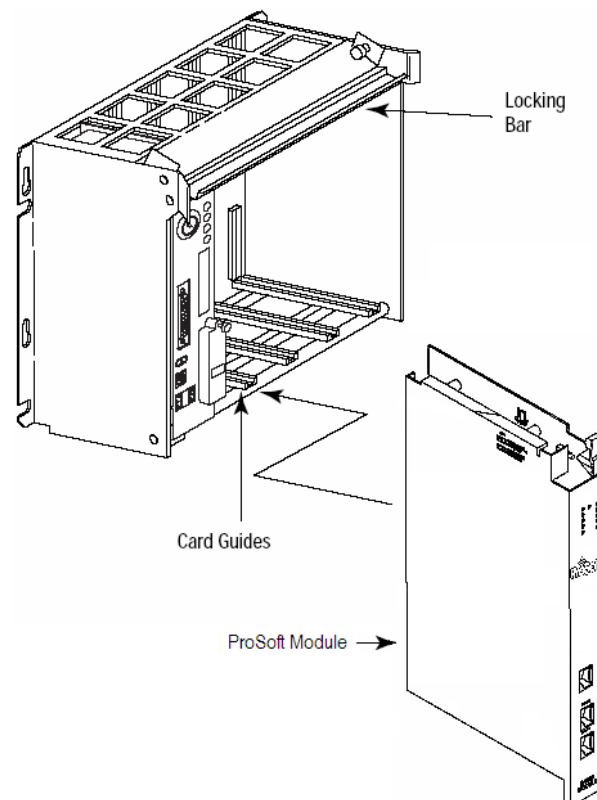
If you have not already installed and configured your PLC processor and power supply, please do so before installing the MVI71-MCM module. Refer to your Rockwell Automation product documentation for installation instructions.

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

After you have checked the placement of the jumpers, insert MVI71-MCM into the PLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install PLC modules.

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

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



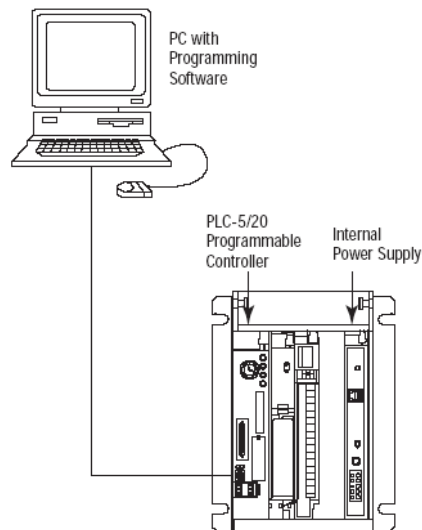
- 3 With a firm but steady push, snap the module into place.

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

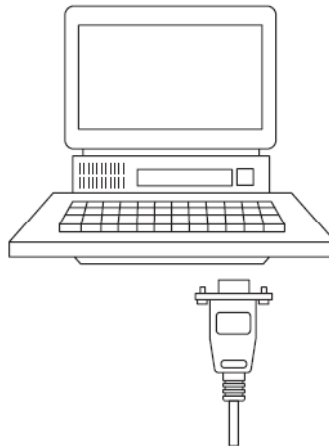
**Note:** If you insert the module improperly, the system may stop working, or may behave unpredictably.

### 1.5 Connect your PC to the Processor

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



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

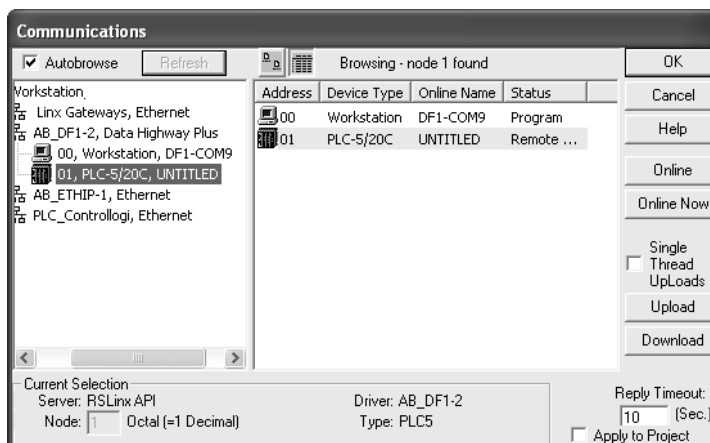


## 1.6 Download the Sample Program to the Processor

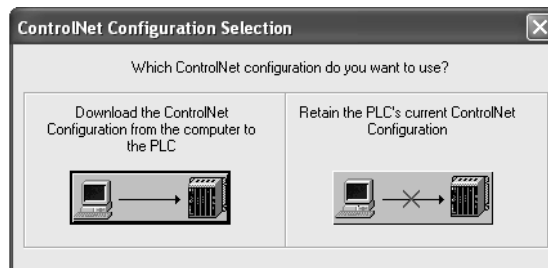
- **To download the sample program from RSLogix 5 to the PLC processor:**

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

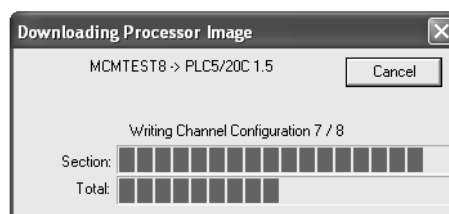
- 1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



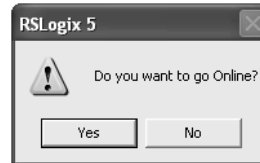
- 2 Click the Download button to transfer the sample program to the processor.
- 3 When prompted, choose Computer to PLC



- 4 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



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

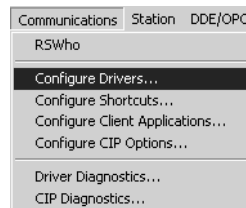


**Note:** If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

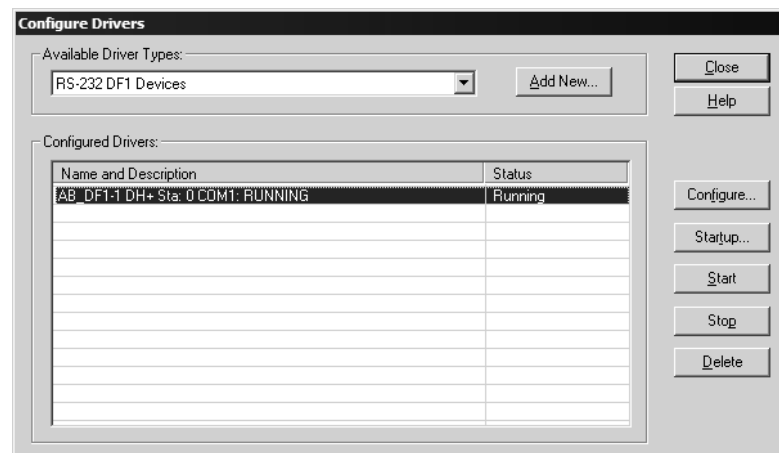
### 1.6.1 Configuring RSLinx

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

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



This action opens the Configure Drivers dialog box.



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

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



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

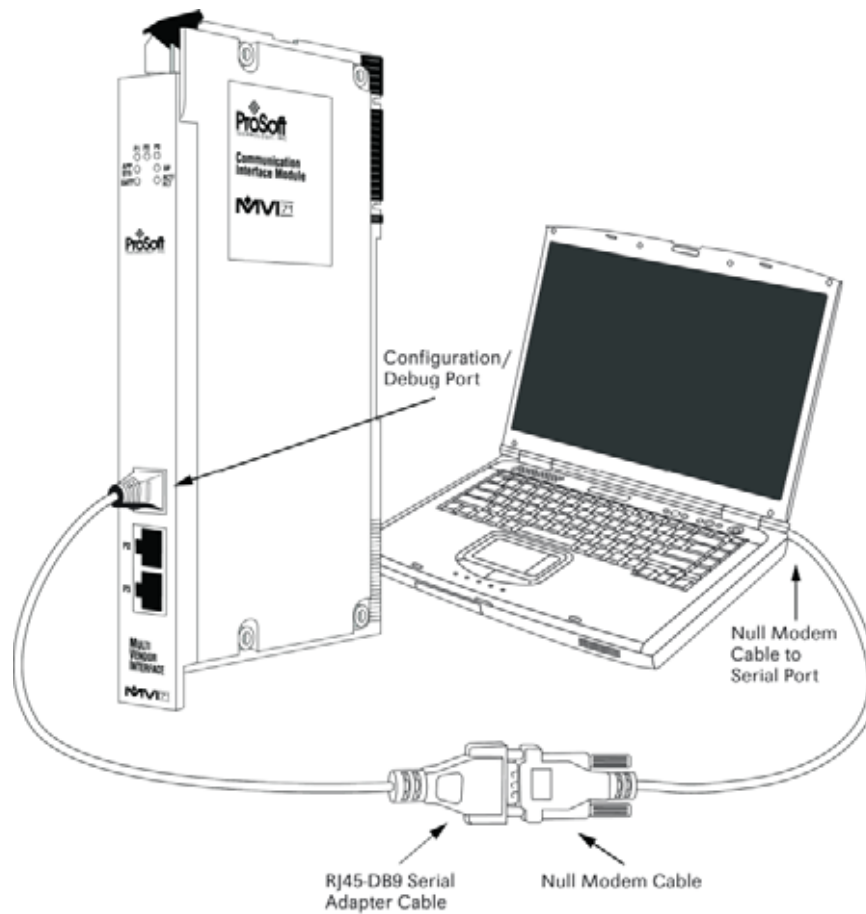
**Note:** If the auto-configuration procedure fails, verify that the cables are connected correctly between the processor and the serial port on your computer, and then try again. If you are still unable to auto-configure the port, refer to your RSLinx documentation for further troubleshooting steps.



## 1.7 Connect your PC to the Module

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

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





## 2 Installing and Configuring the Module

### *In This Chapter*

- Quick Start ..... 19
- Configuring the Module ..... 22

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

- 1 Use RSLogix 5 to identify the module to the processor and add the module to a project.

**Note:** The RSLogix 5 software must be in "offline" mode to add the module to a project.

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

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

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

The first step in installing and configuring the module is to define whether the block transfer or side-connect interface will be used. If the block transfer interface is used, remove the Compact Flash Disk from the module if present and insert the module into the rack with the power turned off.

### 2.1 Quick Start

This section lists the steps required to start module operation. The goal of this section is to provide the user with the necessary actions steps to install and configure the module, however, it is essential that the user read all of this manual in order to fully understand how the module operates.

- 1 Choose the slot where the module will be located in the PLC rack.
- 2 Open the sample ladder file using RSLogix 5.
- 3 Make sure the BTR and BTW instructions match the MVI71-MCM location.
- 4 The sample ladder Block Transfer instructions are configured as:
- 5 Rack=0 , Group=0,Module=0
- 6 Verify that the data files used in the sample ladder will not interfere with the data files you are using in your application. If necessary, edit the sample ladder in order to use a different data file number for the same task.

- 7** The sample ladder uses the following data files:
- B3: Auxiliary Bits
  - N9: Status Data
  - N10: Configuration Data
  - N11: Port 1 Modbus Commands (if Port 1 is configured as a Modbus Master)
  - N12: Port 2 Modbus Commands (if Port 2 is configured as a Modbus Master)
  - N13: Pass-Thru
  - B14: Pass-Thru Bit (used to bit write commands when using pass-thru)
  - N15: Block Transfer Read Data
  - N16: Block Transfer Write Data
  - N50: Event Commands
- 8** Edit the Configuration Data to match your application. The configuration data contains the port configuration and block transfer parameters. In the sample ladder the Configuration Data is located at data file N10 and it is structured according to the table listed below. Use this table to help you configure the module:

Address	Description	Value
N10:0	Write Data Start	
N10:1	Write Data Count	
N10:2	Read Data Start	
N10:3	Read Data Count	
N10:4	Backplane Fail Count	
N10:5	Error Start Address	
N10:6	Port 1 Enable	
N10:7	Port 1 Type	
N10:8	Port 1 Float Flag	
N10:9	Port 1 Float Start	
N10:10	Port 1 Float Offset	
N10:11	Port 1 Protocol	
N10:12	Port 1 Baud Rate	
N10:13	Port 1 Parity	
N10:14	Port 1 Data Bits	
N10:15	Port 1 Stop Bits	
N10:16	Port 1 RTS ON	
N10:17	Port 1 RTS OFF	
N10:18	Port 1 Minimum Response	
N10:19	Port 1 Use CTS	
N10:20	Port 1 Slave ID	
N10:21	Port 1 Bit Input Offset	
N10:22	Port 1 Word Input Offset	
N10:23	Port 1 Bit Output Offset	
N10:24	Port 1 Holding Register Offset	
N10:25	Port 1 Command Count	
N10:26	Port 1 Minimum Command Delay	

Address	Description	Value
N10:27	Port 1 Command Error List Address	
N10:28	Port 1 Command Response Timeout	
N10:29	Port 1 Retry Count	
N10:30	Port 1 Error Delay Count	
N10:31	Port 2 Enable	
N10:32	Port 2 Type	
N10:33	Port 2 Float Flag	
N10:34	Port 2 Float Start	
N10:35	Port 2 Float Offset	
N10:36	Port 2 Protocol	
N10:37	Port 2 Baud Rate	
N10:38	Port 2 Parity	
N10:39	Port 2 Data Bits	
N10:40	Port 2 Stop Bits	
N10:41	Port 2 RTS ON	
N10:42	Port 2 RTS OFF	
N10:43	Port 2 Minimum Response	
N10:44	Port 2 Use CTS	
N10:45	Port 2 Slave ID	
N10:46	Port 2 Bit Input Offset	
N10:47	Port 2 Word Input Offset	
N10:48	Port 2 Bit Output Offset	
N10:49	Port 2 Holding Register Offset	
N10:50	Port 2 Command Count	
N10:51	Port 2 Minimum Command Delay	
N10:52	Port 2 Command Error List Address	
N10:53	Port 2 Command Response Timeout	
N10:54	Port 2 Retry Count	
N10:55	Port 2 Error Delay Count	

**9** If a port is configured as a Modbus Master, you will create Modbus Commands for that port. For example, if Port 1 is configured as Master N11:0 file would be used as:

Start	End	Description
N11:0	N11:9	Command 1
N11:10	N11:19	Command 2
N11:20	N11:29	Command 3
N11:30	N11:39	Command 4
N11:40	N11:49	Command 5
...	...	...

Each Modbus Command has the following structure:

Offset	Description
0	Enable
1	Internal Address
2	Poll Interval
3	Command Count

Offset	Description
4	Swap Code
5	Slave Address
6	Modbus Function
7	Destination Address
8	Spare
9	Spare

- 10 Copy the new ladder file to your existing ladder application.
- 11 Modify the communication set up jumper in the back of the MVI71-MCM, selecting the communication type for each port (485, 422 or 232).
- 12 Insert the module in the rack and connect the port(s) to the Modbus network. Make sure the connection correctly follows the wiring diagrams shown in this manual.
- 13 Turn on the rack power and download the new ladder file to the PLC.

## 2.2 Configuring the Module

In order for the MVI71-MCM module to function, a minimum amount of configuration data must be transferred to the module. The following table provides an overview of the different types of configuration data that the module requires, depending on the operating modes to be supported.

Module Register Address	Functional Modes Affected	Name	Description
5000 to 5009	Data Transfer	General Module Configuration	This section of the configuration data contains the module configuration data that defines the data transfer between the module and the PLC processor.
5010 to 5039 and 5040 to 5069	Master and Slave	Port Configuration	These sections define the characteristics of each of the Modbus serial communication ports on the module. These parameters must be set correctly for proper module operation.
5200 to 6199 and 6400 to 7399	Master	Master Command List	If the module's Master Mode functionality is to be supported on a port, the Master Command List must be set up.

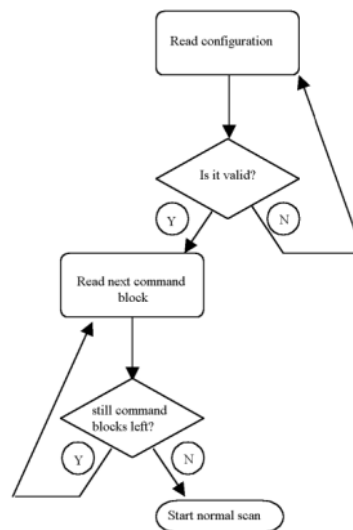
Refer to *Installing and Configuring the Module* for a description of the configuration of the module. The MVI71-MCM module must be configured at least once when the card is first powered, and any time thereafter when the parameters must be changed.

### 2.2.1 Power Up

On power up, the module enters into a logical loop waiting to receive configuration data from the processor. Upon receipt, the module will begin execution of the command list if it is present.

### 2.2.2 Configuration Data Transfer

When the module performs a restart operation, it will request configuration information from the PLC processor. This data is transferred to the module in specially formatted write blocks (output image). The module will poll for each block by setting the required write block number in a read block (input image). Refer to the **Configuring the Module** section for a description of the data objects used with the blocks and the ladder logic required. The module will request all command blocks, according to the number of commands configured by the user for each Master port. The following illustration shows this procedure:



The format of the blocks for configuration is given in the following topics.

#### Module Configuration

This block sends general configuration information from the processor to the module. The data is transferred in a block with an identification code of 9000. The structure of the block is displayed in the following table:

#### Write Block

Offset	Description	Length
0	9000	1
1 to 6	Backplane Setup	6
7 to 31	Port 1 Configuration	25
32 to 56	Port 2 Configuration	25
57 to 63	Spare	7

The read block used to request the configuration has the following structure (Refer to MVI71-MCM Configuration Data Definition (page 87) for a listing of configuration data):

Read Block

Offset	Description	Length
0	-2	1
1	9000	1
2	Module Configuration Errors	1
3	Port 1 Configuration Errors	1
4	Port 2 Configuration Errors	1
5 to 63	Spare	59

If there are any errors in the configuration, the bit associated with the error will be set in one of the three configuration error words. The error must be corrected before the module starts operating.

Refer to [Configuration Error Codes](#) (page 95) for a listing of error words associated with invalid configuration blocks.

### 2.2.3 Changing Parameters During Operation

A copy of the module's configuration data is mapped in the module's database as displayed in the table above. These values are initialized when the module first receives its configuration from the PLC processor. Any node on the network can change this data. A master port on the module may poll a slave for the data or a slave port could receive the data from a remote master unit. The module will not use this data until it is commanded. Ladder logic can be written to issue a Write Configuration command block (9997) to the module. A remote device can set a value of 9997 at address 6800 in the module to download the configuration to the processor. Alternatively, the configuration/debug port on the module can be used to issue the command directly to the module. All three of these methods will force the module to download the configuration to the PLC processor. Ladder logic must exist in the processor to accept the blocks sent by the module. If everything is configured correctly, the module can receive its configuration from a remote device.

### 2.2.4 Module Data Files

All data related to the MVI71-MCM is stored in user-defined data files. The user is responsible for setting up the data areas to match the specific application for the module. Each data area is discussed in the following topics.

#### Configuration Data

Configuration of the module is performed by filling in a user-defined data table. In the example ladder logic (block transfer), file N10 stores the general module configuration information. N11 stores the command list for Port 1. N12 stores the command list for Port 2. Each register in the files has an associated symbol and description to aid in filling in the data. The Reference chapter contains a list of items that must be configured for the module and their associated location in the file. When the side-connect interface is used, the file used for configuration must match that set in the SC\_DATA.TXT file using the setdnpssc.exe program.



#### Data Transfer Parameter

The first five parameters of the configuration relate to the data transfer between the module and the processor as follows:

- **Write start register** - Offset in the module's database where write data is placed
- **Write register count** - Number of registers to transfer from processor to module
- **Read start register** - Offset in the module's database where read data is sourced
- **Read register count** - Number of registers to transfer from the module to the processor
- **Backplane fail** - Number of successive transfer failures that cause communication shutdown
- **Error Status Pointer** - Offset in the module's database where the error status is stored.

These parameters apply to both the side-connect and block transfer interfaces. For the block transfer interface, the number of blocks to transfer between the module and the processor is determined by the count values set in these parameters. Each block can transfer a maximum of 60 words. For example, if the Write Register Count parameter is set to 240, four write blocks will be transferred (1 to 4) between the processor and the module. When the side-connect interface is used, each block can transfer up to 1000 words of data. The user data files must be set to match the values entered in this parameter set. For example, if the Read Register Count parameter is set to 2100, three user data files must be defined. The first two must contain 1000 elements and the last one must contain at least 100 elements.

The Backplane Fail parameter determines if the module should continue communicating on the Modbus network when the backplane transfer operation fails. A value of zero indicates that the module should continue communicating when the backplane is not operational. If the value is greater than 0, the backplane will be retried the entered number of times before a failure will be reported and communication will cease on the ports. When backplane communication is restored, the module will resume communication on the network. For example, if you enter a value of 10 for the parameter, the module will stop all Modbus communications if 10 successive backplane errors are recognized. When a successful transfer is recognized, the module resumes communications.

The Error Status Pointer parameter defines the location in the module's database where the error/status data will be stored. If the value is set to -1, the data will not be stored in the user data area. A value between 0 and 4939 will cause the module's program to store the data at the specified location.

### Modbus Port Parameters

These parameters define the operation of each of the Modbus ports on the module. Refer to [MVI71-MCM Configuration Data Definition](#) (page 87) for a definition of each parameter. These parameters are contained in the configuration at offsets 6 to 55. Care must be taken when filling in this data area for successful operation of the module in a user application.

### Modbus Master Commands

These records define the commands in the master command list.

Each command has the following structure:

Offset	Parameter
0	Enable
1	Internal Address
2	Poll Interval
3	Count
4	Swap
5	Slave Address
6	Function
7	Destination Address
8	Spare
9	Spare

**Enable** - This parameter defines if the command will be executed or disregarded. The following values are valid:

0=Disables the command

1=The command is considered for execution at each scan of the command list and will be controlled by the PollInt parameter

2=The command will only execute if the data associated with the command has changed since the command was last issued. This option is only available for write commands.

**Internal Address** - This parameter specifies the starting internal register address to be associated with the command. Valid entries for this parameter are 0 to 9999.

For example, if using a read holding register function (function 3), the Internal Address parameter will have the address in the MVI71-MCM database in which the value read from a Modbus slave is located. If using a write function (function 16), the Internal Address will have the source address in the MVI71-MCM database which contains the value to be written to a Modbus slave device.

**Important:** The Internal Address must be used according to the Modbus function. When using holding registers (Functions 3,6,16), or input registers (Function 4), the Internal Address indicates the word (16 bits) address in the MVI71-MCM internal database. For example, if using Internal Address = 500, this indicates that word 500 is the Start Internal Address. When using bits (functions 1,2,5,15), the Internal Address parameter indicates the bit address in the MVI71-MCM internal database. For example, if using Internal Address = 32 and function 15, this indicates that the source address starts at bit 0 at word 2

**PollInt** - This parameter defines the minimum number of seconds to wait between the execution of continuous commands (Enable=1). This poll interval command can be used to lighten the communications load on a busy network. Valid entries for this parameter are 0 through 65535.

**Count** - This parameter defines the number of registers to be considered by the command. Valid entries for this parameter are 1 to 100 for functions 3, 4, and 16 and 800 for functions 1, 2, and 15. The Count parameter must be interpreted according to the Modbus function used. Functions 1, 2, and 15: Count = Number of Coils (1 bit). Functions 3, 4, and 16: Count = Number of Words (16 bits).

**Swap** - This parameter specifies if the data used in the command must be altered when reading data from a node on the network (Function 3). Values that can be assigned are as follows:

- 0=No data swapping
- 1=Swap word values
- 2=Swap word and byte values
- 3=Swap byte values

This option is used when interfacing the module with ASCII and floating-point data on other devices.

**Slave Address** - This parameter assigns the Modbus slave node address for the module to reach with the command on the network. This parameter can use values from 0 to 255. Most Modbus networks limit the upper value to 247.

**Func** - This parameter specifies the function to be performed by the command. Valid entries are 1, 2, 3, 4, 5, 6, 15, and 16.

**Destination Address:** This parameter stores the destination address in the slave associated with the command.

For example, if using a Read holding register function (function 3), the Destination Address parameter will contain the address to be read in the Modbus slave connected to a MVI71-MCM master port. If using a Write function (function 16), the Destination Address will contain the slave address to which to command will be written.

**Important:** The Destination Address must be used according to the Modbus function in the command. When using holding registers (functions 3,6,16), or input registers (function 4), the Destination Address is expressed as the word (16 bits) address in the slave. For example, Destination Address = 500 indicates that word 500 is the start Destination Address. When using bits (functions 1,2,5,15,) the Destination Address parameter is expressed as bit address in the MVI71-MCM Destination database. For example, Destination Address = 32, with function 15, indicates that the source address starts at the bit 0 at word 2.

When the side-connect interface is used, the command must be contiguous in the file, each one occupying a 10-word area. The file numbers used are also fixed. Port 1 commands 0 to 99 will reside in the file after the configuration file. Port 2 commands 0 to 99 will reside in the next file.

The following provides some command examples:

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
1	100	0	10	0	1	16	55

This command continuously writes 10 words starting at address 100 to 109 in the MVI71-MCM internal database to addresses 55 to 64 at slave address 1.

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
2	0	0	1	0	5	6	1000

This command conditionally writes 1 word from address 0 in the MVI71-MCM internal database to address 1000 in slave 5. This means that the command is issued only when the value at address 0 changes.

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
1	0	15	1	0	5	6	1000

This command continuously writes 1 word from address 0 in the MVI71-MCM internal database to address 1000 in slave 5 every 15 seconds.

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
1	320	0	20	0	5	15	1600

This command continuously writes 20 bits from bit address 320 to 339 (word 20, bit 0) in the MVI71-MCM internal database to bit address 1600-1619 (word 100, bit 0) in slave 5.

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
1	320	15	1	0	5	5	1600

This command continuously writes 1 bit from bit address 320 (word 20, bit 0) in the MVI71-MCM internal database to bit address 1600 (word 100, bit 0) in slave 5 every 15 seconds.

Enable	Internal Address	Poll	Count	Swap	Slave Address	Function	Destination Address
1	850	0	10	3	5	3	700

This command continuously reads 10 holding registers from addresses 700 to 709 in slave 5 to addresses 850 to 859 in MVI71-MCM internal database after swapping the bytes in each word.

### Status Data

This data area views the status of the module. Refer to [MVI71-MCM Status Data Definition](#) (page 85) for a complete listing of the data stored in this object. When the side-connect interface is used, this data is automatically updated in the configuration file (for example, N10:) starting at offset 200 approximately every second and does not include the first two registers. For the block transfer interface, the module generates blocks with a BTR block identification code of -1. Ladder logic must transfer this information into a user data file.

### **2.2.5 User Data**

Data in the module's internal database in the register range of 0 to 4999 is available to the processor. The parameters set in the configuration determine the set of data that is transferred from the module to the processor (read data) and that transferred from the processor to the module (write data). If the block transfer interface is used, ladder logic is required to handle the transfer of data between the processor and the module. BTR messages are required to handle data read from the module, and BTW messages are required to handle data written to the module. When the side-connect interface is used, the data is directly transferred between the module and the user data files without the ladder logic requirement.

### **2.2.6 Slave Polling Control and Status**

The status data can be used to determine which slaves are currently active on the port, are in communication error, or have their polling suspended and disabled. Special blocks (block transfer interface) or control command (side-connect interface) are required to interface with this data. Using block (command) 3000 or 3100, slaves can be disabled for polling. They can be enabled using block (command) 3001 or 3101. Blocks 3002 to 3006 or 3102 to 3106 request the current status of each slave in the module.

### **2.2.7 Using Side-Connect (Requires Side-Connect Adapter)**

If the side-connect interface is used, make sure the file SC\_DATA.TXT on the Compact Flash Disk contains the correct configuration file number. You can run the setdnpvc.exe program to set the configuration file number to be used with your application.

Install the module in the rack and turn on the power. Connect the terminal server to the module's debug/configuration port and exit the program by pressing the Esc key followed by the "Y" key. This causes the program to exit and remain at the operating system prompt. Run the setdnpvc.exe program with a command line argument of the file number to use for the configuration file. For example, to select N10: as the configuration file, enter the following:

#### **SETDNPVC 10**

**Note:** The SETDNPVC.EXE utility will only set the N file number between 10 and 933.

The program will build the SC\_DATA.TXT on the Compact Flash Disk (C: drive in the root directory).

The next step in module setup is to define the data files to be used with the application. If the block transfer interface is used, define the data files to hold the configuration, status, and user data. Enter the module's configuration in the user data files. Enter the ladder logic to handle the blocks transferred between the module and the PLC. Download the program to the PLC and test the program with the module.

If the side-connect interface is used, no ladder logic is required for data transfer. The user data files to interface with the module must reside in contiguous order in the processor. The first file to be used by the interface is the configuration file. This is the file number set in the SC\_DATA.TXT file using the SETDNPSC.EXE program. The following table lists the files used by the side-connect interface:

File Number	Example	Size	Description
Cfg File	N10	300	Configuration/Control/Status File
Cfg File+1	N11	to 1000	Port 1 commands 0 to 99
Cfg File+2	N12	to 1000	Port 2 commands 0 to 99
Cfg File+5	N15	to 1000	Data transferred from the module to the processor. Other files for read data.
Cfg File+5+n	N16	to 1000	Data transferred from the processor to the module.
Cfg File +5+n+m			Other files for write data.

n is the number of read data files minus one. Each file contains up to 1000 words.  
 m is the number of write data files minus one. Each file contains up to 1000 words.

Even if both files are not required for a port's commands, they are still reserved and should only be used for that purpose. The read and write data contained in the last set of files possess the data transferred between the module and the processor. The number of files required for each is dependent on the number of registers configured for each operation. Two examples follow:

**Example of 240 words of read and write data (cfg file=10)**

Data Files	Description
N15:0 to 239	Read Data
N16:0 to 239	Write Data

**Example of 2700 read and 3500 write data registers (cfg file=10)**

Data Files	Description
N15:0 to 999	Read data words 0 to 999
N16:0 to 999	Read data words 1000 to 1999
N17:0 to 299	Read data words 2000 to 2299
N18:0 to 999	Write data words 2300 to 3299
N19:0 to 999	Write data words 3300 to 4299
N20:0 to 999	Write data words 4300 to 4999

Special care must be taken when defining the files for the side-connect interface. Because the module directly interacts with the PLC processor and its memory, any errors in the configuration may cause the processor to fault and it may even lose its configuration program. After defining the files and populating them with the correct data, download the program to the processor, and place the processor in Run mode. If everything is configured properly, the module should start its normal operation.

If all the configuration parameters are set correctly, and the module is attached to a Modbus network, the modules application LED (OK LED) should remain off and the backplane activity LED (BP ACT) should blink rapidly. Refer to the Diagnostics and Troubleshooting of this manual if you encounter errors. Attach a terminal to Port 1 on the module and look at the status of the module using the Configuration/Debug Menu in the module.





### 3 Ladder Logic

Ladder logic is required for application of the MVI71-MCM module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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



## 4 Diagnostics and Troubleshooting

### *In This Chapter*

- Reading Status Data From the Module ..... 35
- LED Status Indicators ..... 50

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 MVI71-MCM module returns a 29-word Status Data block that can be used to determine the module's operating status. This data is located in the module's database at registers 7600 to 7628 and at the location specified in the configuration. This data is transferred to the PLC processor continuously with each read block. For a complete listing of the status data object, refer to the **Module Set Up** section.

#### **4.1.1 The Configuration/Debug Menu**

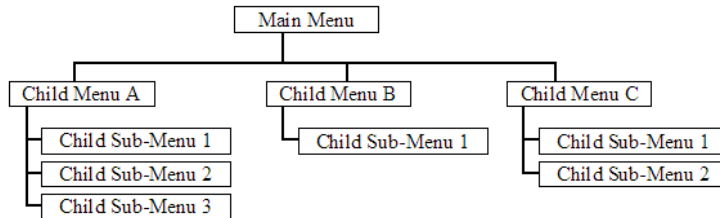
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

#### Navigation

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

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



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

### Keystrokes

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

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

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

### **4.1.2 Required Hardware**

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

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

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one serial communications port available
- A null modem serial cable.

### **4.1.3 Required Software**

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

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

---

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

---

#### 4.1.4 Using the Configuration/Debug Port

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

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

---

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

---

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

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

- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 Verify that RSLinx is not controlling the COM port. Refer to [Disabling the RSLinx Driver for the Com Port on the PC](#) (page 79).
- 3 Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
- 4 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

### 4.1.5 Main Menu

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

```
MODBUS MASTER/SLAVE COMMUNICATION MODULE MENU
?=Display Menu
A=Data Analyzer
B=Block Transfer Statistics
C=Module Configuration
D=Modbus Database View
Master Command Errors : E=Port 1 F=Port 2
Master Command List  : I=Port 1 J=Port 2
Slave Status List     : O=Port 1 P=Port 2
U=Version Information
W=Warn Boot Module
V=Transfer Module Cfg to Processor
Communication Status  : 1=Port 1 2=Port 2
Port Configuration   : 6=Port 1 7=Port 2
Z=Method Used
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.

#### Opening the Data Analyzer Menu

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

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

#### Viewing Block Transfer Statistics

Press **[B]** from the Main Menu to view the Block Transfer Statistics screen.

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

**Tip:** To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

#### Viewing Module Configuration

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

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

#### Opening the Database Menu

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

#### Opening the Command Error List Menu

Press **[I]** to open the Command Error List. This list consists of multiple pages of command list error/status data. Press **[?]** to view a list of commands available on this menu.

#### Opening the Command List Menu

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

#### Viewing the Slave Status List (Port 1 and 2)

Press **[O]** (port 1) or **[P]** (port 2) to view the 256 slave status values associated with the ports. The slave status values are defined as follows:

- 0 = slave is not used,
- 1 = slave being actively polled,
- 2 = slave suspended and
- 3 = slave disabled.

#### 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.

Transferring Module Configuration to the Processor

Press **[Y]** to transfer the module's configuration data to the processor. Ladder logic is required in the processor to receive and implement the updated configuration. You will be prompted to confirm the transfer.

If the operation is not successful, an error code will be returned.

Code	Description
0	Transfer successful
-1	Error transferring module configuration data (block -9000)
-2	Error transferring device definition data (blocks -9100 to -9103)
-3	Error transferring master command list data (blocks -6000 to -6007)

After successful data transfer, the module will perform a warm-boot operation to read in the new data.

Viewing Port Communication Status

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

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

Viewing Port Configuration

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

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



Exiting the Program

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

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

**4.1.6 Data Analyzer**

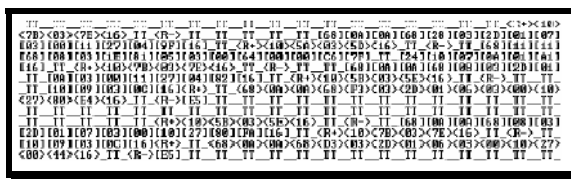
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

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

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

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

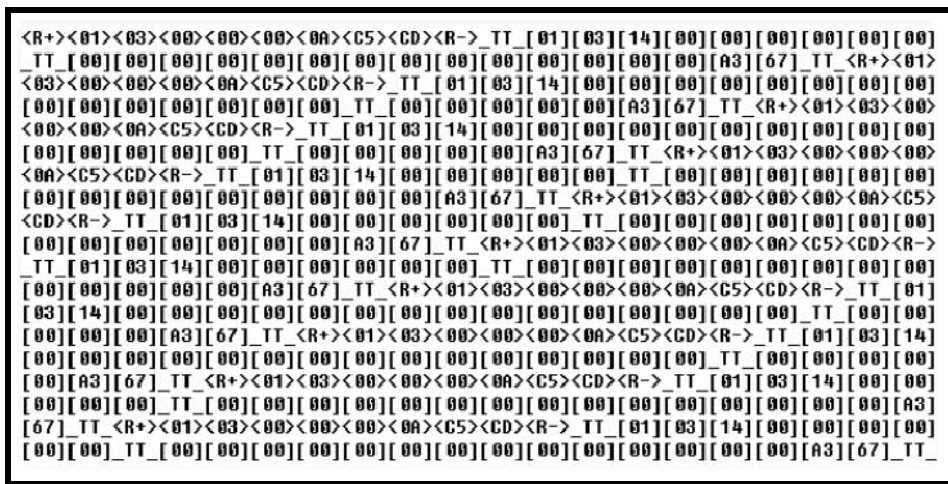
Press **[H]** to display the data on the current page in hexadecimal format.

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. An example display is shown below:



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.

#### **4.1.7 Data Analyzer Tips**

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

```
Data Analyzer Mode Selected
```

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

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

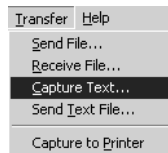
Port = 1, Format=HEX, Tick=10
```

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

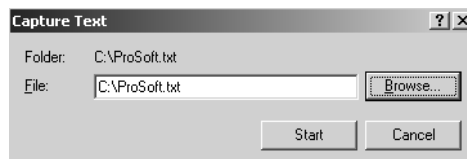
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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

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



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



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

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

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

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

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

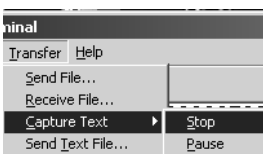
All characters shown in <> brackets are characters being sent out by the module.

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

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

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

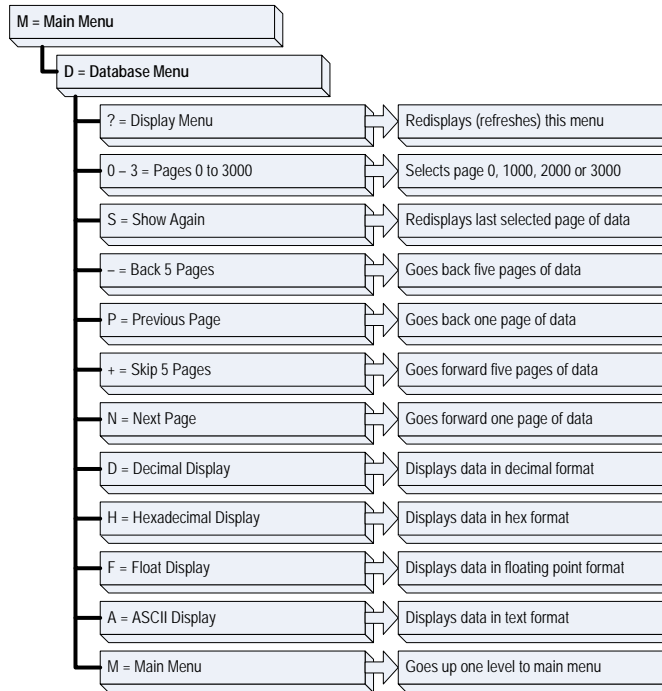
When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



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

### 4.1.8 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.



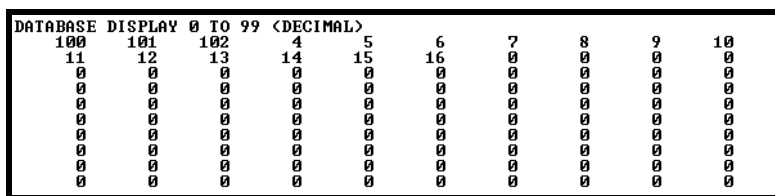
#### Viewing Register Pages

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

Command	Description
<b>[0]</b>	Display registers 0 to 99
<b>[1]</b>	Display registers 1000 to 1099
<b>[2]</b>	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



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*

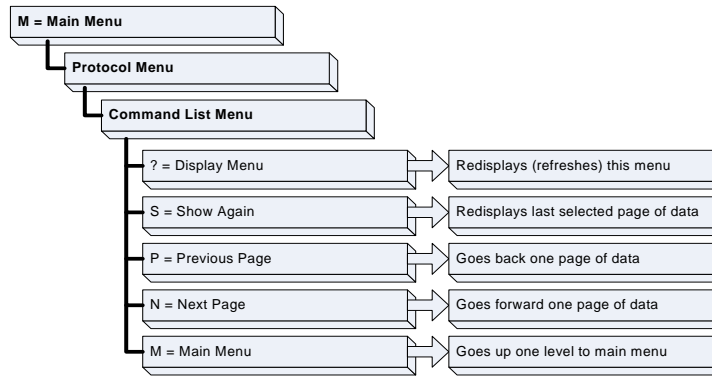
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

*Returning to the Main Menu*

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

### 4.1.9 Master Command Error List Menu

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



#### Redisplaying the Current Page

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

#### Viewing the Previous 20 Commands

Press **[-]** to display data for the previous 20 commands.

#### Viewing the Previous Page of Commands

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

#### Viewing the Next 20 Commands

Press **[+]** to display data for the next 20 commands.

#### Viewing the Next Page of Commands

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

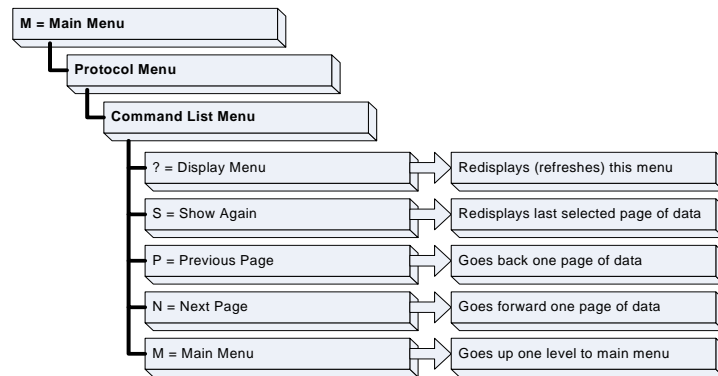
#### Returning to the Main Menu

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



#### 4.1.10 Master Command List Menu

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



##### Redisplaying the Current Page

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

##### Viewing the Previous 50 Commands

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

##### Viewing the Previous Page of Commands

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

##### Viewing the Next 50 Commands

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

##### Viewing the Next Page of Commands

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

##### Returning to the Main Menu

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

## 4.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the MODBUS 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 MODBUS network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI71-MCM is working normally.
		On	The MVI71-MCM 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. Turn off the power to the rack, remove the card from the rack re-insert the card and then restore power 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.

During module configuration, the OK LED will be red and the APP and BP ACT LEDs will be on. If the LEDs are latched in this mode for a long period of time, look at the configuration error words in the configuration request block.

Correct any invalid data in the configuration for proper module operation. When the configuration contains a valid parameter set, all the bits in the configuration words will be clear. This does not indicate that the configuration is valid for the user application. Make sure each parameter is set correctly for the specific application.

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 PLC 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"> <li>▪ The processor is in Run mode.</li> <li>▪ The backplane driver is loaded in the module.</li> <li>▪ The module is configured for read and write block data transfer.</li> <li>▪ The ladder logic handles all read and write block situations.</li> <li>▪ The module is configured in the processor.</li> </ul>
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..... 53
- Functional Overview..... 55
- Cable Connections..... 79
- MVI71-MCM Database Definition..... 85
- MVI71-MCM Status Data Definition ..... 85
- MVI71-MCM Configuration Data Definition ..... 87
- MVI71-MCM Command Error Codes ..... 94
- Configuration Error Codes ..... 95
- Frequently Asked Questions ..... 96

### 5.1 Product Specifications

The MV71 Modbus Master/Slave Communication Module allows PLC processors to interface easily with other Modbus protocol compatible devices.

Compatible devices include not only Modicon PLCs (which all support the Modbus protocol) but also a wide assortment of end devices. The module acts as an input/output module between the Modbus network and the Rockwell Automation backplane. The data transfer from the processor is asynchronous from the actions on the Modbus network. A 5000-word register space in the module exchanges data between the processor and the Modbus network.

#### **5.1.1 Features and Benefits**

The inRAX Modbus Master/Slave Communication Module is designed to allow PLC processors to interface easily with Modbus protocol-compatible devices and hosts.

The MVI71-MCM module acts as an input/output module between the Modbus network and the Rockwell Automation backplane. The data transfer from the PLC processor is asynchronous from the actions on the Modbus network. A 5000-word register space in the module exchanges data between the processor and the Modbus network.

Many host SCADA software applications support the Modbus protocol, while devices commonly supporting the protocol include several PLCs, as well as many other third party devices in the marketplace. (For a partial list of devices that speak Modbus, please visit the ProSoft Tested section of the ProSoft Technology web site).

**5.1.2 General Specifications**

- Single Slot - 1771 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.
- Configuration data obtained through user-defined ladder. Sample ladder file included

**5.1.3 Hardware Specifications**

Specification	Description
Form Factor	Single Slot 1771 chassis compatible BTR/BTW data transfer Local or remote rack
Backplane current load	800 mA @ 5 V
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 10150 Hz
Relative humidity	5% to 95% (non-condensing)
LED Indicators	Module status Backplane transfer status Application status Serial activity and error LED status
<b>Debug/Configuration port (CFG)</b>	
CFG Port (P1)	RJ45 (DB-9M with supplied cable) RS-232 only
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
<b>Application Ports</b>	
Application Serial port (P2, P3) (Serial Modules)	Two RJ45 RS-232/422/485 Application ports

### **5.1.4 Functional Specifications**

- Support for the storage and transfer of up to 5000 registers to/from the PLC processor's data files
- User-definable module memory usage
- Two ports to emulate any combination of Modbus master or slave device
- Supports Enron version of Modbus protocol for floating point data transactions

#### Slave Specifications

The MVI71-MCM module accepts Modbus function code commands of 1, 2, 3, 4, 5, 6, 8, 15, 16, 17, 22 and 23 from an attached Modbus master unit. A port configured as a Modbus slave permits a remote master to interact with all data contained in the module. This data can be derived from other Modbus slave devices on the network, through a master port, or from the PLC processor.

#### Master Specifications

A port configured as a virtual Modbus master device on the MVI71-MCM module actively issues Modbus commands to other nodes on the Modbus network. One hundred commands are supported on each port. Additionally, the master ports have an optimized polling characteristic that polls slaves with communication problems less frequently. The PLC processor can be programmed to control the activity on the port by actively selecting commands from the command list to execute or issuing commands directly from the ladder logic.

## **5.2 Functional Overview**

This section provides an overview of how the MVI71-MCM module transfers data using the MCM protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

### **5.2.1 General Concepts**

The MVI71-MCM is a module that allows the communication between the PLC and a Modbus network. The module has 2 ports that can be individually configured as a Modbus Master or as a Modbus Slave.

The module uses the rack backplane in order to transfer data to the PLC. Ladder logic is used for different tasks including:

- Downloading configuration data
- Updating the module's internal database (read or write)
- Executing control blocks

The module uses block transfer instructions in order to read and write data to the PLC. The MVI71-MCM also supports side connect transfer (requires a side connect adapter) which allows a faster data transfer and does not require ladder logic to transfer data from/to the PLC.

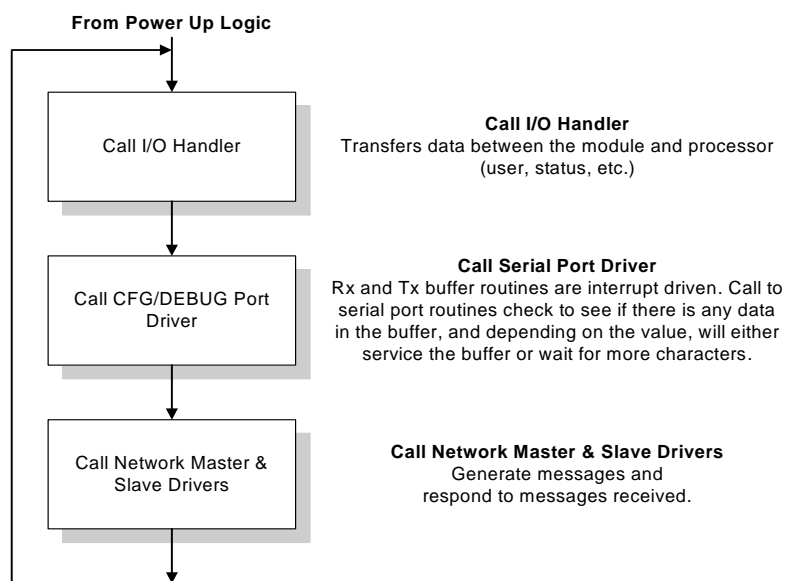
On power up the module begins performing the following logical functions:

- 1 Initialize hardware components
  - Initialize PLC backplane driver or side-connect drivers
  - Test and Clear all RAM
  - Initialize the serial communication ports
- 2 Determine the interface to the backplane (side-connect or block transfer).
- 3 Wait for Module Configuration from PLC processor
- 4 Initialize Module Register space
- 5 Enable Slave Driver on selected ports
- 6 Enable Master Driver on selected ports

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

### 5.2.2 Main Logic Loop

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



### 5.2.3 PLC Processor Not in Run

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



### 5.2.4 Backplane Data Transfer

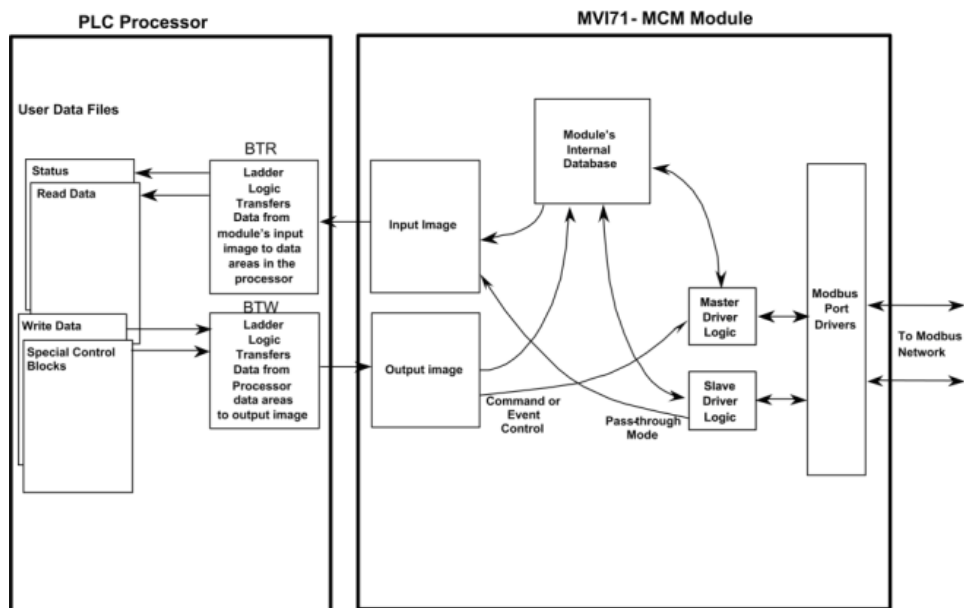
The MVI71-MCM module communicates directly over the PLC backplane. Data is paged between the module and the PLC processor across the backplane using the module's input and output images or directly to the processor using the side-connect interface (requires a side-connect adapter). The update frequency 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 2 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 data registers in the processor by the ladder logic. The input image for the module is set to 64 words. This large data area permits fast throughput of data between the module and the processor.

The processor inserts data to 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 set to 64 words. This large data area permits fast throughput of data from the processor to the module.

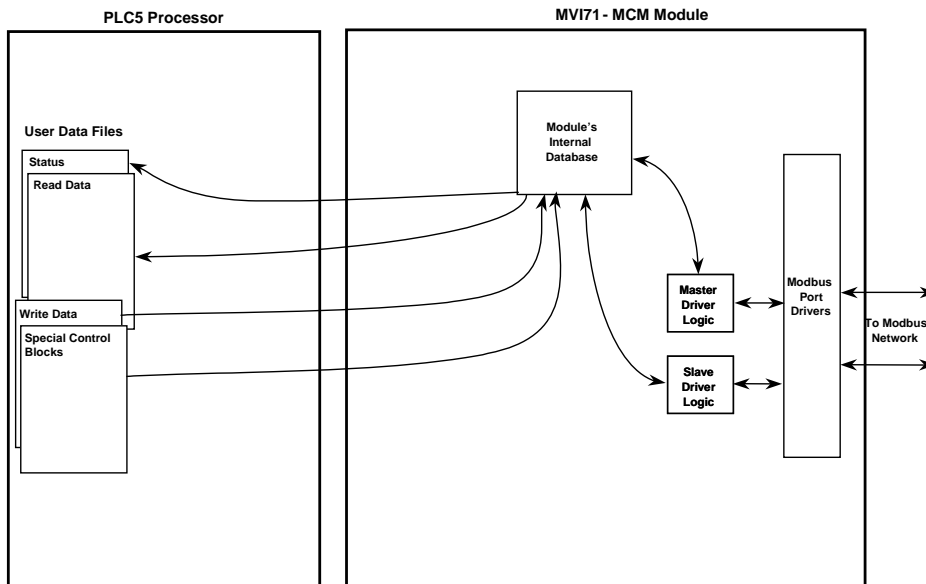
The following illustration shows the data transfer method used to move data between the PLC processor, the MVI71-MCM module and the Modbus Network.

#### Block Transfer



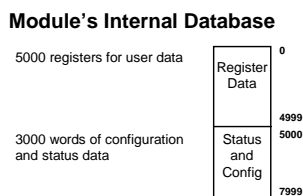
The following illustration shows the data transfer operations used when using the side-connect interface (requires the side-connect adapter):

Side-Connect



When the side connect interface is used, data is transferred directly between the processor and the module. The module's program interfaces directly to the set of user data files established in the PLC to pass all data between the two devices. No ladder logic is required for data transfer, only the establishment of the data files.

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 PLC processor to interface the input and output image data with data defined in the data registers. All data used by the module is stored in its internal database. This database is defined as a virtual Modbus data table with addresses from 0 (40001 Modbus) to 6999 (47000 Modbus). The following illustration shows the layout of the database:



Data contained in this database is paged through the input and output images by coordination of the PLC ladder logic and the MVI71-MCM module's program. Up to 60 words of data can be transferred from the module to the processor at a time. Up to 60 words of data can be transferred from the processor to the module. Each image has a defined structure depending on the data content and the function of the data transfer as defined below.

### 5.2.5 Modbus Addressing

It is important to familiarize yourself with the Modbus addressing concepts when using the MVI71-MCM module. The module supports the following address ranges for each Modbus function:

Function	Description	Modbus Address Range	MVI71-MCM Addressing
1	Read Output Status	0001 to 9999	bit
2	Read Input Status	10001 to 29999	bit
3	Read Holding Registers	40001 to 49999	word (16 bits)
4	Read Input Registers	30001 to 39999	word (16 bits)
5	Force Single Coil	0001 to 9999	bit
6	Preset Single Register	40001 to 49999	word (16 bits)
15	Force Multiple Coils	0001 to 9999	bit
16	Preset Multiple Registers	40001 to 49999	word (16 bits)

The table above also shows the MVI71-MCM addressing for each function. Depending on which function is used, the database address should be interpreted in bits or words.

#### Examples

##### MVI71-MCM as a Slave

- When a Modbus master device connected to a MVI71-MCM slave port sends a command function 5 to Modbus address 32, it will write to bit 32 in the MVI71-MCM database (bit 0 from word 2).
- When a Modbus master device connected to a MVI71-MCM slave port sends a command function 6 to Modbus address 40032, it will write to word 32 in the MVI71-MCM database.

##### MVI71-MCM as a Master

- When an MVI71-MCM Master Port sends a command with the following parameters:

Internal DB Address: 10

Count: 3

Function: 15

It will write 3 coils to the specified slave using the source bits 10, 11, 12 in the MVI71-MCM database.

- When an MVI71-MCM Master Port sends a command with the following parameters:

Internal DB Address: 10

Count: 3

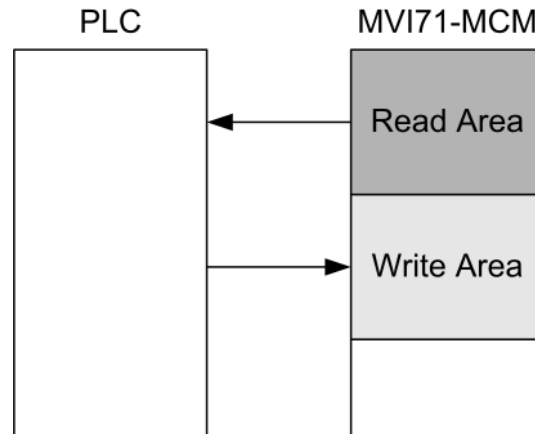
Function: 16

It will write 3 holding registers to the specified slave using as the source words 10, 11, 12 in the MVI71-MCM database.

### 5.2.6 Using the Read and Write Data Areas

The MVI71-MCM module separates user data into two regions; the Read Data area and the Write Data area. These areas must be defined inside the User Data area. The user can set up both areas when transferring the configuration data to the module.

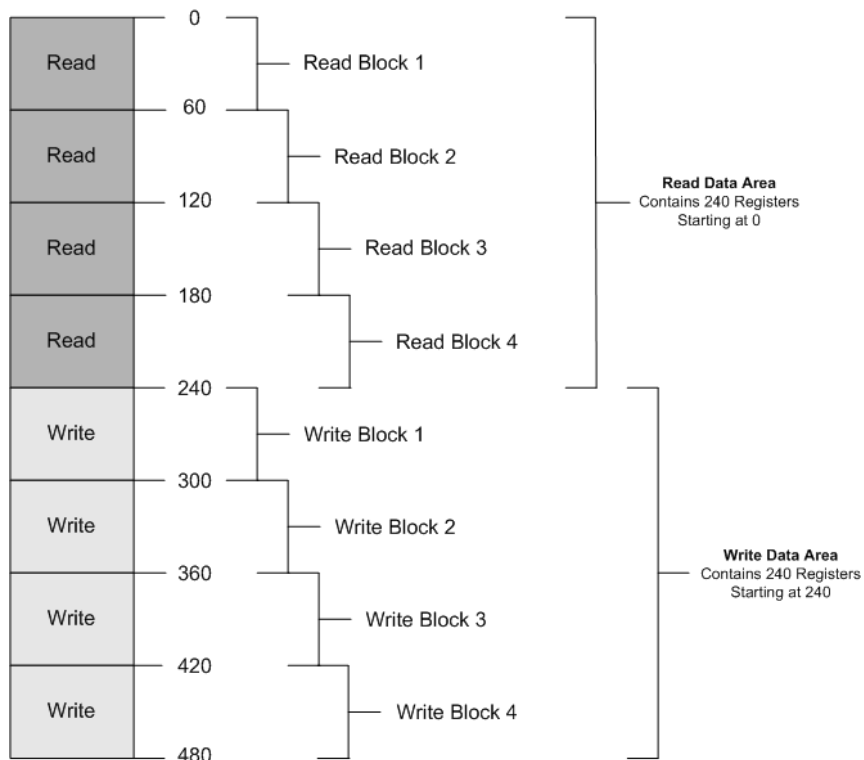
The Read Data area is constantly transferred from the MVI71-MCM to the PLC using BTR instructions while the Write Data area is constantly transferred from the PLC to the MVI71-MCM module using BTW instructions:



For example, if the user sets the Read and Write areas as:

- Read Register Start : 0
- Read Register Count : 240
- Write Register Start : 240
- Write Register Count : 240

This configuration would set the module internal database as:



The configuration first creates four possible blocks for each Read and Write area since every block contains 60 registers. Therefore, the Read and Write Block ID generated would be:

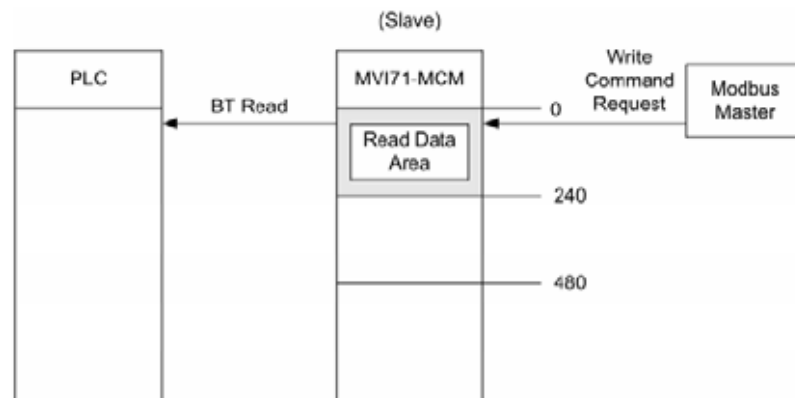
Read Block ID	Write Block ID
0	3
1	4
2	1
3	2
4	3
-1	4
0	1
1	2
2	3

If the ladder logic takes too much time to send a BTW instruction, the module sends a new BTR instruction requesting the same Write Block ID.

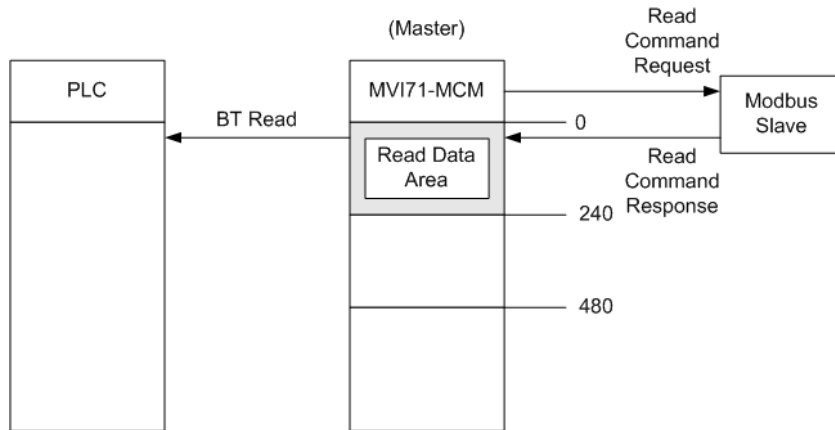
The Read Block ID 0 is a null block to guarantee ladder logic consistency if the user sets an empty Read Data area. The Read Block ID -1 transfers the configuration data.

#### Read Data Area Application Examples

**Ex. 1:** The following example shows a Read Data Area application; A Modbus Master device sends a Modbus Write Command to an MVI71-MCM slave port. The command destination address must be located inside the Read Data Region (between 0 and 239).

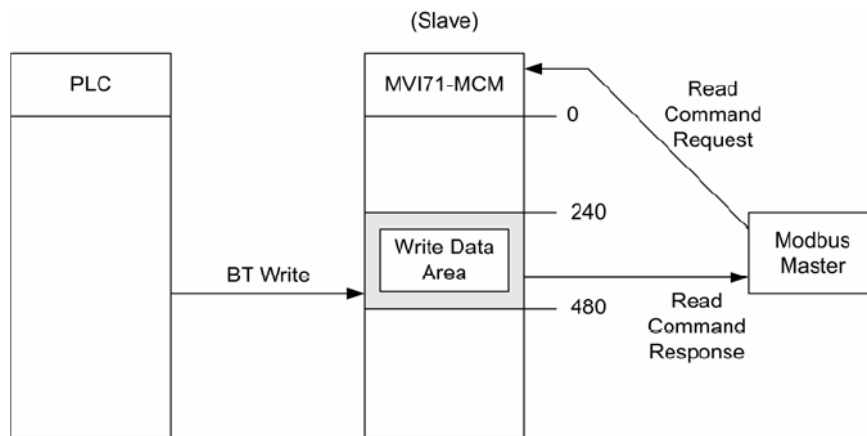


**Ex. 2:** The next figure shows another example of how to use the Read Data Area; an MVI71-MCM port configured as a Master sends a Modbus Read Command to a Slave device. The Modbus Read Command destination address must be located inside the Read Data Area (between addresses 0 and 239).

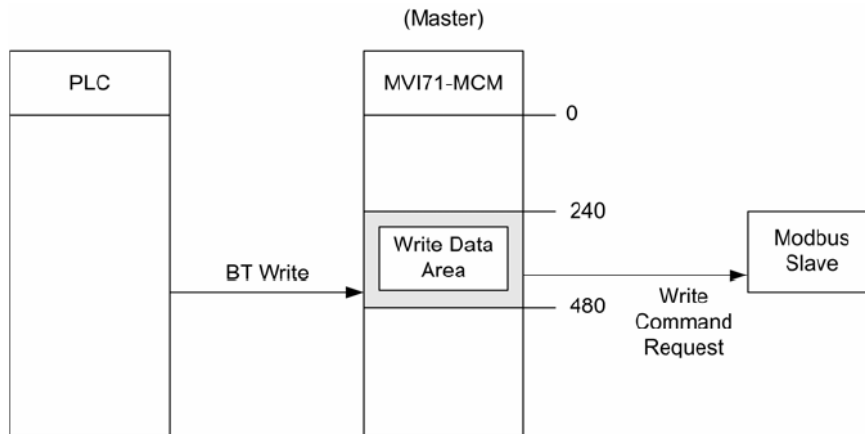


Write Data Area Application Examples

**Ex. 1:** The following figure shows a Write Data Area application; a Modbus Master device sends a Modbus Read Command to an MVI71-MCM slave port. The command source address (in the MVI71-MCM database) must be located inside the Read Data Region (between 240 and 479).



**Ex. 2:** The next figure shows another example of how to use the Write Data Area; an MVI71-MCM port configured as a Master sends a Modbus Write Command to a slave device. The Modbus Write Command source address in the MVI71-MCM database must be located inside the Write Data Area (between addresses 240 and 479).



### 5.2.7 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 Set Up** section 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 below.

#### Read Block

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

Offset	Description	Length
0	Read Block ID	1
1	Write Block ID	1
2 to 61	Read Data	60
62 to 63	Spare	2

The Read Block ID is an index value used to determine the location of where the data will be placed in the PLC processor user data table. Each transfer can move up to 60 words (block offsets 2 to 61) of data.

The Write Block ID associated with the block requests data from the PLC 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 Modbus network or operator control through the module's Configuration/Debug port.

If the ladder logic does not send a BTW instruction to the module quickly enough, it is possible for the MVI71-MCM module to send a new BTR instruction requesting the same write block ID.

Write Block

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

Offset	Description	Length
0	Write Block ID	1
1 to 60	Write Data	60
61 to 63	Spare	3

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 60 words (block offsets 1 to 60) of data.

Status Data Block (Read Block ID = -1)

After the last Read Block is sent, the module builds a BTR block (ID = -1) to transfer the module's status information to the processor. This information can be used by the PLC program to determine the current status of the module. Ladder logic should be constructed to transfer the information in this block to a user data file. The structure of this block is shown in the following table:

Offset	Content	Description
0	Read Block ID	Block identification code -1 to indicate a status block.
1	Write Block ID	Block requested from the processor by the module.
2	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
3 to 4	Product Code	These two registers contain the product code of "MCM"
5 to 6	Product Version	These two registers contain the product version for the currently running software.
7 to 8	Operating System	These two registers contain the month and year values for the program operating system.
9 to 10	Run Number	These two registers contain the Run Number value for the currently running software.
11	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
12	Port 1 Command List Response	This field contains the number of slave response messages received on the port.
13	Port 1 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
14	Port 1 Requests	This field contains the total number of messages sent out of the port.



Offset	Content	Description
15	Port 1 Responses	This field contains the total number of messages received on the port.
16	Port 1 Errors Sent	This field contains the total number of message errors sent out of the port.
17	Port 1 Errors Received	This field contains the total number of messages errors received on the port.
18	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
19	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
20	Port 2 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
21	Port 2 Requests	This field contains the total number of messages sent out the port.
22	Port 2 Responses	This field contains the total number of messages received on the port.
23	Port 2 Errors Sent	This field contains the total number of message errors sent out of the port.
24	Port 2 Errors Received	This field contains the total number of message errors received on the port
25	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
26	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
27	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
28	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
29	Command Block Count	This field contains the total number of command blocks received from the processor.
30	Error Block Count	This field contains the total number of block errors recognized by the module.
31	Port 1 Current Error	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the currently executing command.
32	Port 1 Last Error	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with an error.
33	Port 2 Current Error	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the currently executing command.
34	Port 2 Last Error	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with an error.

Master Command Blocks

Each port on the module can be configured as a Modbus master device containing its own list of one hundred commands.

Each command has the following structure:

Offset	Parameter
0	Enable
1	Internal Address
2	Poll Interval
3	Count
4	Swap
5	Slave Address
6	Function
7	Destination Address
8	Spare
9	Spare

For information about these parameters, refer to the **Configuring the Module** section under Modbus Master Commands.

The commands are read from the processor using the following Write Block ID's: Modbus Port 1 - 6000 to 6016 and Modbus Port 2 - 6100 to 6116. The module will sequentially poll for each block from the processor. Ladder logic must handle each and every one of the data transfers (the sample ladder logic already handles this). The structure of each block is shown in the following table.

Offset	Description	Length
0	6000 to 6016 and 6100 to 6116	1
1 to 10	Command Definition	10
11 to 20	Command Definition	10
21 to 30	Command Definition	10
31 to 40	Command Definition	10
41 to 50	Command Definition	10
51 to 60	Command Definition	10
61 to 63	Command Definition Spare	3

Each block contains six commands. Because there are 100 possible commands, the last block (6016 or 6116) should contain up to four commands.

As the list is read in from the processor and as the commands are processed, an error value is maintained in the module for each command. The errors generated by the module are listed in MVI71-MCM Command Error Codes (page 94).

Transferring the Command Error List to the Processor

You can transfer the command error list to the processor from the module database. To place the table in the database, set the Command Error Pointer parameter to the database location desired.

To transfer this table to the processor, make sure that the Command Error table is in the database area covered by the Read Data.

### 5.2.8 Slave Status Blocks

Slave status blocks send status information of each slave device on a master port. Slaves attached to the master port can have one of the following states:

State	Description
0	The slave is inactive and not defined in the command list for the master port.
1	The slave is actively being polled or controlled by the master port and communications is successful.
2	The master port has failed to communicate with the slave device. Communications with the slave is suspended for a user defined period based on the scanning of the command list.
3	Communications with the slave has been disabled by the ladder logic. No communication will occur with the slave until this state is cleared by the ladder logic.

Slaves are defined to the system when the module initializes the master command list. Each slave defined will be set to a state of one in this initial step. If the master port fails to communicate with a slave device (retry count expired on a command), the master will set the state of the slave to a value of 2 in the status table. This suspends communication with the slave device for a user specified scan count. Each time a command in the list is scanned that has the address of a suspended slave, the delay counter value will be decremented. When the value reaches zero, the slave state will be set to one. This will enable polling of the slave.

Block ID	Description
3002	Request for slave status values for Port 1 slaves 0 to 59
3003	Request for slave status values for Port 1 slaves 60 to 119
3004	Request for slave status values for Port 1 slaves 120 to 179
3005	Request for slave status values for Port 1 slaves 180 to 239
3006	Request for slave status values for Port 1 slaves 240 to 255
3102	Request for slave status values for Port 2 slaves 0 to 59
3103	Request for slave status values for Port 2 slaves 60 to 119
3104	Request for slave status values for Port 2 slaves 120 to 179
3105	Request for slave status values for Port 2 slaves 180 to 239
3106	Request for slave status values for Port 2 slaves 240 to 255

The format of these blocks is as shown in the following table:

Offset	Description	Length
0	3002 to 3006 or 3102 to 3106	1
1 to 63	Spare	63

The module will recognize the request by receiving the special write block code and respond with a read block with the following format:

Offset	Description	Length
0	3002 to 3006 or 3102 to 3106	1
1	Write Block ID	1
2 to 61	Slave Poll Data Status	60
62 to 63	Spare	2

Ladder logic can be written to override the value in the slave status table. It can disable (state value of 3) by sending a special block of data from the processor to the slave. Port 1 slaves are disabled using block 3000, and Port 2 slaves are disabled using block 3100. Each block contains the slave node addresses to disable. The structure of the block is displayed in the following table:

Offset	Description	Length
0	3000 or 3100	1
1	Number of Slaves in Block	1
2 to 61	Slave indexes	60
62 to 63	Spare	2

The module will respond with a block with the same identification code received and indicate the number of slaves acted on with the block. The format of this response block is displayed in the following table:

Offset	Description	Length
0	3000 or 3100	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to 63	Spare	61

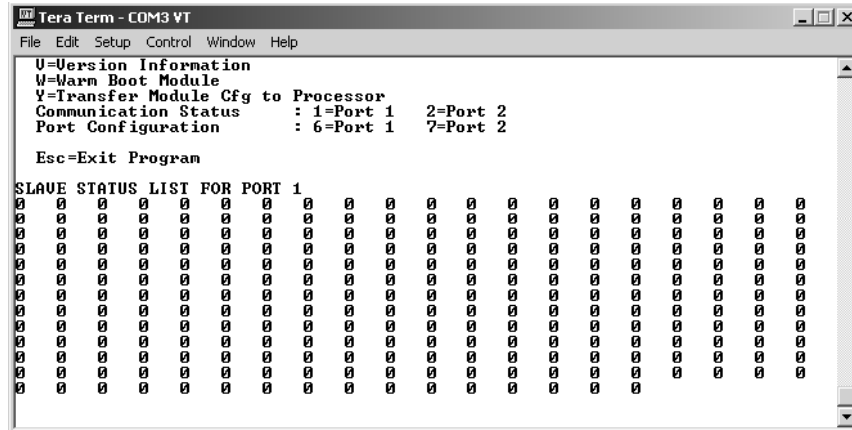
Ladder logic can be written to override the value in the slave status table to enable the slave (state value of 1) by sending a special block. Port 1 slaves are enabled using block 3001, and Port 2 slaves are enabled using block 3101. Each block contains the slave node addresses to enable. The following table describes the format for this block.

Offset	Description	Length
0	3001 or 3101	1
1	Number of Slaves in Block	1
2 to 61	Slave indexes	60
62 to 63	Spare	2

The module will respond with a block with the same identification code received and indicate the number of slaves acted on with the block. The format of this response block is displayed in the following table:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to 63	Spare	61

The user can monitor the slave status table using the Debug port .< option Slave Status List>:



### 5.2.9 Command Control Blocks

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

#### Using the Command Control Blocks

Ladder logic must be considered in order to use command control blocks. The ladder logic moves the control block data to the correct location in order for the data to be read by the module.

#### Block Transfer

The ladder logic should copy the command control block to the write block transfer data file. For example, if using N7:310 as the write block transfer data file, the structure would look as follows:

Offset	Data File
0	N7:310
1	N7:311
2	N7:312
3	N7:313
4	N7:314
5	N7:315
...	...
63	N7:363

**Side-Connect (Requires Side-Connect Adapter)**

The ladder logic should copy the command control block to the side-connect data file starting at register 80.

For example, if the user selected N10 as the side connect data file, the ladder logic should move the command control block according to the following structure:

Offset	Data File
0	N10:80
1	N10:81
2	N10:82
3	N10:83
4	N10:84
5	N10:85
...	...
63	N10:143

**Event Command**

Event command control blocks send Modbus commands directly from the ladder logic to one of the master ports. This control block should be only be used under special circumstances. Normally, the user should use the configuration command list to send commands. The format for these blocks is displayed in the following table:

**Write Block**

Offset	Description	Length
0	1000 to 1255 or 2000 to 2255	1
1	Internal DB Address	1
2	Point Count	1
3	Swap Code	1
4	Function	1
5	Device DB Address	1
6 to 63	Spare	57

The block number defines the Modbus port to be considered and the slave node to be accessed. For example, if sending a command to Slave 1, Block ID 1001 should be used (from Port 1). Blocks in the 1000 range are directed to Modbus Port 1, and blocks in the 2000 range are directed to Modbus Port 2. The slave address is represented in the block number in the range of 0 to 255. The sum of these two values determines the block number. The other parameters passed with the block construct the command. The **Internal DB Address** parameter specifies the module's database location to associate with the command. The **Point Count** parameter defines the number of points or registers for the command (100 registers or 800 coils). The **Swap Code** is used with Modbus function 3 requests to change the word or byte order. The **Modbus Function Code** has one of the following values 1, 2, 3, 4, 5, 6, 15 or 16. The **Device Database Address** is the Modbus register or point in the remote slave device to be associated with the command. When the command receives the block, it will process it and place it in the command queue. The module will respond to each event command block with a read block with the following format:

#### Read Block

Offset	Description	Length
0	1000 or 2000	1
1	Write Block ID	1
2	0 = Fail, 1 = Success	1
3 to 63	Spare	61

Word two of the block can be used by the ladder logic to determine if the command was added to the command queue of the module. The command will only fail if the command queue for the port is full (100 commands for each queue).

#### Command Control

Command control blocks place commands in the command list into the command queue. Each port has a command queue of up to 100 commands. The module services commands in the queue before the master command list. This gives high priority to commands in the queue. Commands placed in the queue through this mechanism must be defined in the master command list. Under normal command list execution, the module will only execute commands with the Enable parameter set to one or two. If the value is set to zero, the command is skipped. Commands may be placed in the command list with an Enable parameter set to zero. These commands can then be executed using the command control blocks.

One to six commands can be placed in the command queue with a single request. The following table describes the format for this block.

#### Write Block

Offset	Description	Length
0	5001 to 5006 or 5101 to 5106	1
1	Command index	1
2	Command index	1

Offset	Description	Length
3	Command index	1
4	Command index	1
5	Command index	1
6	Command index	1
7 to 63	Spare	57

Blocks in the range of 5001 to 5006 are used for Modbus Port 1, and blocks in the range of 5101 to 5106 are used for Modbus Port 2. The last digit in the block code defines the number of commands to process in the block. For example, a block code of 5003 contains 3 command indexes that are to be used with Modbus Port 1. The Command index parameters in the block have a range of 0 to 99 and correspond to the master command list entries.

The module responds to a command control block with a block containing the number of commands added to the command queue for the port. The following table describes the format for this block.

**Read Block**

Offset	Description	Length
0	5001 to 5006 or 5101 to 5106	1
1	Write Block ID	1
2	Number of commands added to command queue	1
3 to 63	Spare	61

Read Current Configuration

This block is sent from the PLC processor to the module to force the module to write its current configuration back to the processor. This function is used when the module's configuration has been altered remotely using database write operations. The write block contains a value of -9000 in the first word. The module will respond with blocks containing the module configuration data. Ladder logic must handle the receipt of these blocks. The blocks transferred from the module are as follows:

Block -9000, General Configuration Data:

Offset	Description	Length
0	-9000	1
1	-9000	1
2 to 7	Backplane Setup	6
8 to 32	Port 1 Configuration	25
33 to 57	Port 2 Configuration	25
58 to 63	Spare	6



Blocks -6000 to -6003 and -6100 to 6103, Master Command List Data for ports 1 and 2, respectively:

Offset	Description	Length
0	-6000 to 6016 and -6100 to 6116	1
1	-6000 to 6016 and -6100 to 6116	1
2 to 11	Command Definition	10
12 to 21	Command Definition	10
22 to 31	Command Definition	10
32 to 41	Command Definition	10
42 to 51	Command Definition	10
52 to 61	Command Definition	10
62 to 63	Spare	2

Each of these blocks must be handled by the ladder logic for proper module operation. The processor can request the module's configuration by sending a configuration read request block, block code 9997, to the module. The format of this request block is as follows:

Offset	Description	Length
0	9997	1
1 to 63	Spare	63

When the module receives this command block, it transfers the module's current configuration to the processor. If the block transfer interface is used, the blocks defined in the previous tables (-9000 and -6000 series blocks) will be sent from the module. If the side-connect interface is used, the user data files will be updated directly by the module.

#### Warm Boot

This block is sent from the PLC processor to the module (output image) 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 data registers. This will force the module to read the new configuration information and to restart. The structure of the control block is shown in the following table:

Offset	Description	Length
0	9998	1
1 to 63	Spare	63

#### Cold Boot

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

Offset	Description	Length
0	9999	1
1 to 63	Spare	63

### 5.2.10 Pass-Through Control Blocks

The pass-through feature allows a Master device to write commands directly to the PLC ladder logic.

#### Formatted Pass-Through Control Blocks

If one or more of the slave ports on the module are configured for the formatted pass-through mode, the module will pass blocks with identification codes of 9996 to the processor for each received write command. Any Modbus function 5, 6, 15 or 16 commands will be passed from the port to the processor using this block identification number. Ladder logic must handle the receipt of all Modbus write functions to the processor and to respond as expected to commands issued by the remote Modbus master device. The structure of the formatted pass-through control block is shown in the following tables:

#### Function 5

Offset	Description	Length
0	0	1
1	9958	1
2	1	1
3	Bit Address	1
4	Data	1
5 to 58	Modbus message received	59

The ladder logic will be responsible for parsing and copying the received message and performing the proper control operation as expected by the master device. The processor must then respond to the pass-through control block with a write block with the following format.

Offset	Description	Length
0	9958	1
1 to 63	Spare	63

This will inform the module that the command has been processed and can be cleared from the pass-through queue.

#### Function 6 and 16

Offset	Description	Length
0	0	1
1	9956/9957 (Floating-point)	1
2	Number of data words	1
3	Data Address	1
4 to 63	Data	60

The ladder logic will be responsible for parsing and copying the received message and performing the proper control operation as expected by the master device. The processor must then respond to the pass-through control block with a write block with the following format.

Offset	Description	Length
0	9956/9957	1
1 to 63	Spare	63

This will inform the module that the command has been processed and can be cleared from the pass-through queue.

#### Function 15

When the module receives a function code 15 when in pass-through mode, the module will write the data using block ID 9959 for multiple-bit data. First the bit mask clears the bits to be updated. This is accomplished by ANDing the inverted mask with the existing data. Next the new data ANDed with the mask is ORed with the existing data. This protects the other bits in the INT registers from being affected.

Offset	Description	Length
0	0	1
1	9959	1
2	Number of Words	1
3	Word Address	1
4 to 63	Data	60

The ladder logic will be responsible for parsing and copying the received message and performing the proper control operation as expected by the master device. The processor must then respond to the pass-through control block with a write block with the following format.

Offset	Description	Length
0	9959	1
1 to 63	Spare	63

This will inform the module that the command has been processed and can be cleared from the pass-through queue.

#### 5.2.11 Remote Command Control

Command Control data is received from other nodes on the network that can control the MVI71-MCM module. Specific values are written to regions of this block to control the module. Currently, the module is programmed to handle the receipt of the following requests: write configuration to processor, warm boot and cold boot.

The remote node controls the module by writing one of the following values to register 7800 (Modbus address 47801):

Value	Description
9997	Write configuration in database to the processor and warm boot the module.
9998	Warm boot the module.
9999	Cold boot the module.

The control register is cleared (a value of 0) after the operation is executed with the exception of the 9997 command. If the module fails to successfully transfer the configuration to the processor, an error code will be returned in the control register as follows:

Code	Description
0	No error, transfer successful
-1	Error transferring general configuration information.
-2	Error transferring Modbus Port 1 master command list
-3	Error transferring Modbus Port 2 master command list

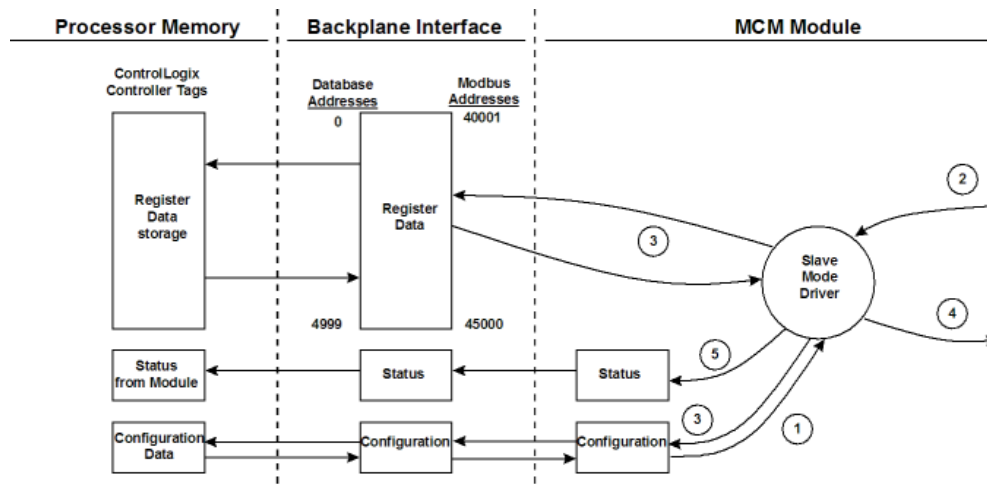
Ladder logic must handle the 9997 command. No ladder logic is required when using the warm or cold boot commands.

### 5.2.12 Data Flow Between MVI71-MCM Module and PLC Processor

The following topics describe the flow of data between the two pieces of hardware (PLC processor and MVI71-MCM module) and other nodes on the Modbus network under the module's different operating modes. Each port on the module is configured to emulate a Modbus master device or a Modbus slave device. The operation of each port is dependent on this configuration. The sections below discuss the operation of each mode.

#### Slave Driver

The Slave Driver Mode allows the MVI71-MCM module to respond to data read and write commands issued by a master on the Modbus network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Modbus slave port driver receives the configuration information from the PLC processor. This information configures the serial port and define the slave node characteristics. Additionally, the configuration information contains data that can be used to offset data in the database to addresses requested in messages received from master units.

Step	Description
2	A Host device, such as a Modicon PLC or an HMI application, 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 command, 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 2, 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 **Configuring the Module (page 22)** for a complete list of the parameters that must be defined for a slave port.

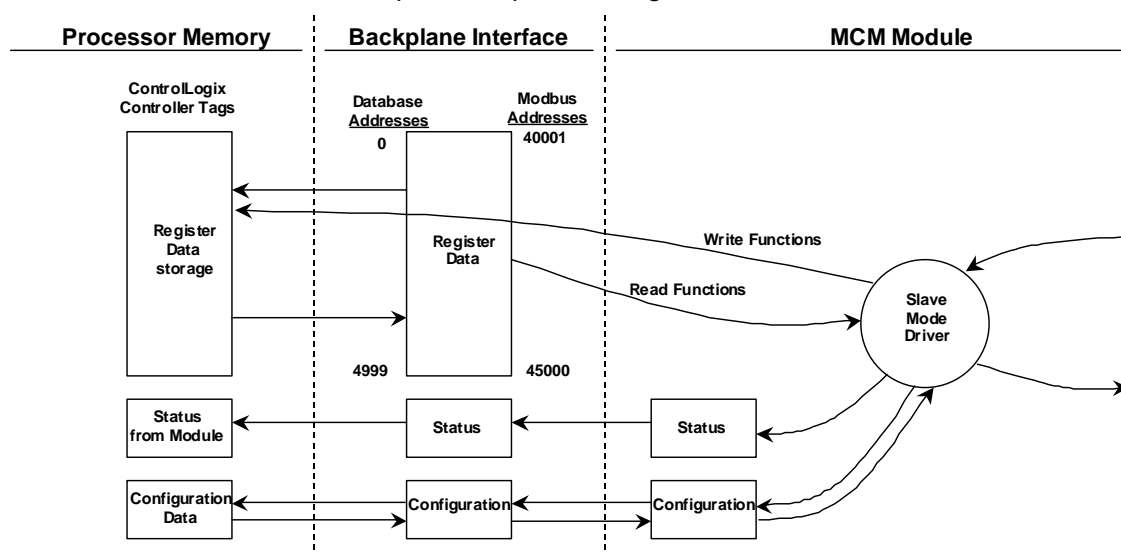
### Pass-Through

Each port can have the type parameter configured for pass-through mode where:

Swap Code	Description
2	Pass-Through Formatted
3	Pass-Through Formatted with Byte Swap

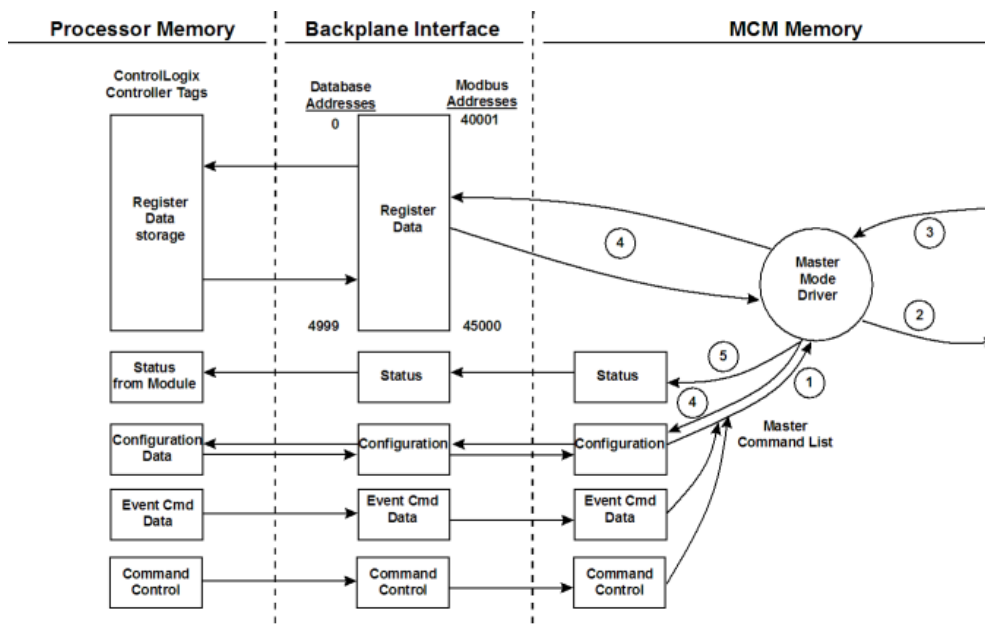
Option 3 swaps all bytes in the data received from a Modbus Master command.

In this mode, all write requests will be passed directly to the processor and will not be placed in the database. This permits direct, remote control of the processor without the intermediate database. This mode is especially useful for Master devices that do not send both states of control. For example, a SCADA system may only send an on command to a digital control point and never send the clear state. The SCADA system expects the local logic to reset the control bit. Pass-through must be used to simulate this mode. The following diagram shows the dataflow for a slave port with pass-through enabled:



Master Driver Mode

In Master Mode, the MVI71-MCM module issues read or write commands to slave devices on the Modbus network. These commands are user configured in the module via the Master Command List received from the PLC processor or issued directly from the PLC processor (event command control). Command status is returned to the processor for each individual command in the command list status block. The location of this status block in the module's internal database is user defined. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Master driver obtains configuration data from the PLC processor. The configuration data obtained includes the number of commands and the Master Command List. These values are used by the Master driver to determine the type of commands to be issued to the other nodes on the Modbus network (Refer to the Module Set Up section).
2	After configuration, the Master driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from the module's internal database to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the Master driver for processing.
4	Data received from the node on the network is passed into the module's internal database, assuming a read command.
5	Status is returned to the PLC processor for each command in the Master Command List.

Refer to **Configuring the Module (page 22)** for a complete description of the parameters required to define the virtual Modbus master port. Refer to the **MCM Driver** documentation for a complete discussion of the structure and content of each command. Care must be taken in constructing each command in the list for predictable operation of the module. If two commands write to the same internal database address of the module, the results will not be as desired. All commands containing invalid data are ignored by the module.

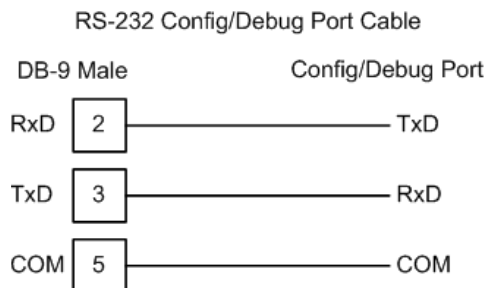
### 5.3 Cable Connections

The application ports on the MVI71-MCM module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

#### 5.3.1 RS-232 Configuration/Debug Port

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

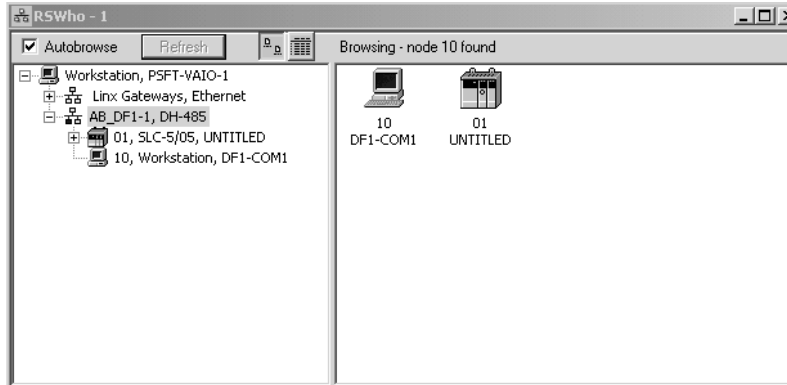


#### Disabling the RSLinx Driver for the Com Port on the PC

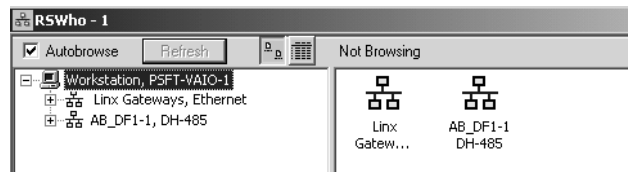
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable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



- 1 Open RSLinx and go to Communications>RSWho

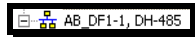
- 2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



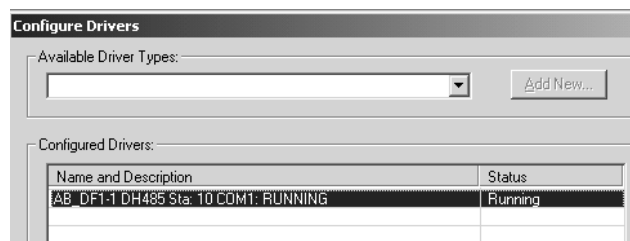
- 3 Notice how the DF1 driver is opened, and the driver is looking for a processor on node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your RSWho screen should look like this:



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



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

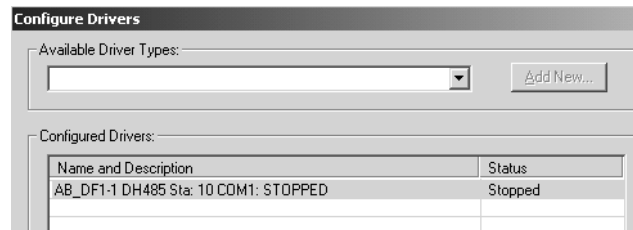




If you see the status as running, you will not be able to use this com port for anything other than communication to the processor. To stop the driver press the "Stop" on the side of the window:



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

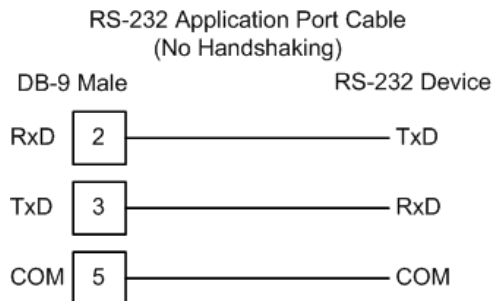


- 6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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

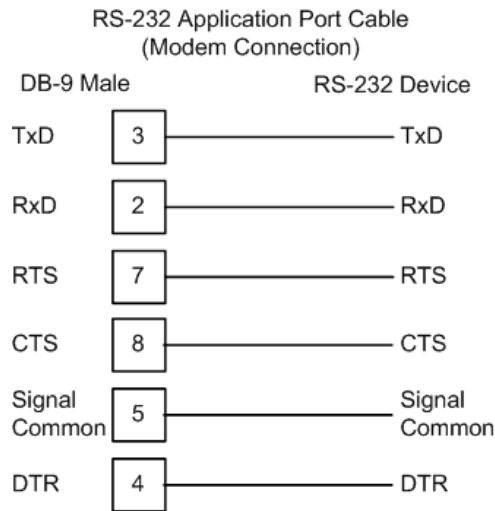
### 5.3.2 RS-232

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



RS-232: Modem Connection

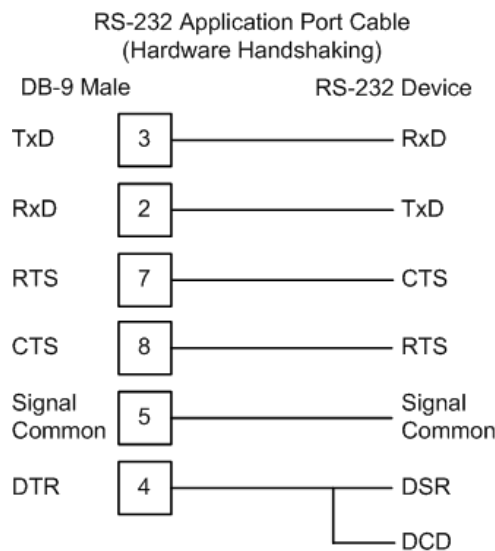
This type of connection is required between the module and a modem or other communication device.



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

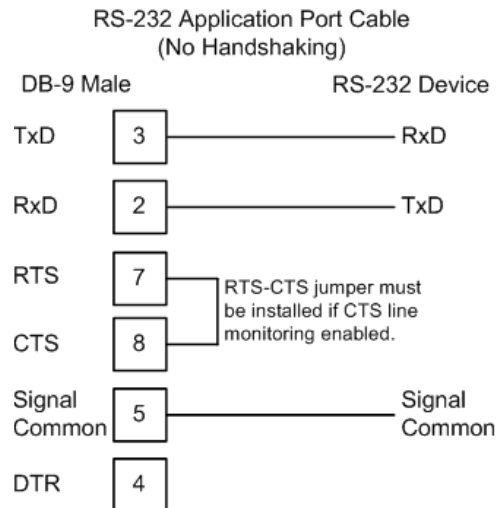
RS-232: Null Modem Connection (Hardware Handshaking)

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

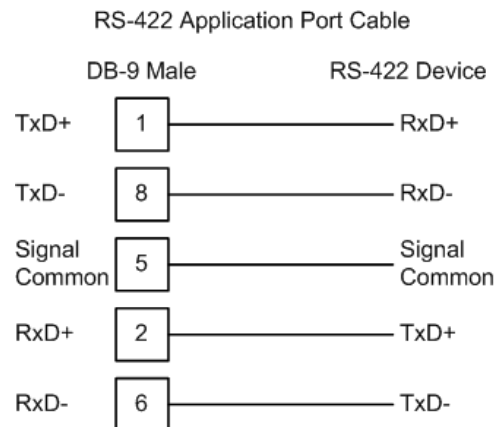


RS-232: Null Modem Connection (No Hardware Handshaking)

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

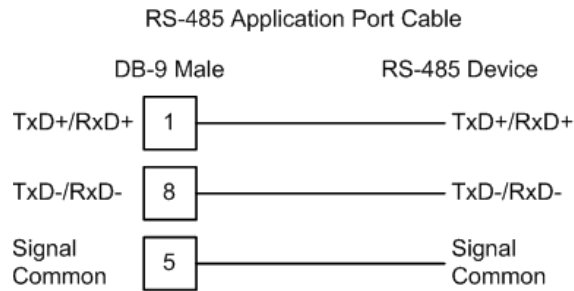


**Note:** If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

**5.3.3 RS-422**

### 5.3.4 RS-485

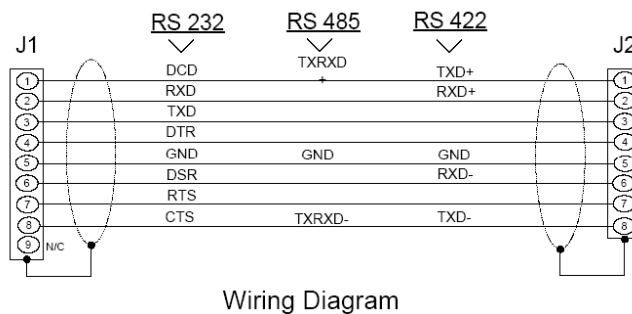
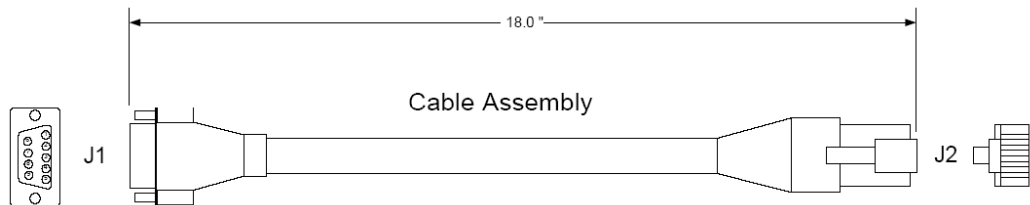
The RS-485 interface requires a single two or three wire cable. The Common connection is optional and dependent on the RS-485 network. The cable required for this interface is shown below:



#### RS-485 and RS-422 Tip

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

### 5.3.5 DB9 to RJ45 Adaptor (Cable 14)



## 5.4 MVI71-MCM Database Definition

This section contains a listing of the internal database of the MVI71-MCM module. This information can be used to interface other devices to the data contained in the module.

Register Range	Modbus Low	Modbus High	Content	Size
0 to 4999	40001	45000	User Data	5000
5000 to 5006	45001	45007	Backplane Configuration	10
5010 to 5039	45011	45040	Port 1 Setup	30
5040 to 5069	45041	45070	Port 2 Setup	30
5200 to 6199	45901	46900	Port 1 Commands	1000
6400 to 7399	46401	47400	Port 2 Commands	1000
7600 to 7632	47601	47633	Misc. Status Data	33
7800 to 7999	47801	4800	Command Control	200

The User Data area holds data collected from other nodes on the network (master read commands) or data received from the processor (write blocks).

Additionally, this data area is used as a data source for the processor (read blocks) or other nodes on the network (write commands).

Detailed definition of the miscellaneous status data area can be found in [MVI71-MCM Status Data Definition](#) (page 85).

Definition of the configuration data areas can be found in the data definition section of this document and in [MVI71-MCM Configuration Data Definition](#) (page 87).

[MVI71-MCM Command Error Codes](#) (page 94) contains a discussion of the command control section of the database.

[Configuration Error Codes](#) (page 95) contains a table of Configuration Error Codes.

## 5.5 MVI71-MCM Status Data Definition

This section contains a description of the data contained in the status data block. This data is transferred from the module to the processor either through a BTR block -1 (block transfer interface) or directly into the configuration file (side-connect interface).

Offset	Register	Content	Description
0	7600	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
1 to 2	7601 to 7602	Product Code	These two registers contain the product code of "DFCM"
3 to 4	7603 to 7604	Product Version	These two registers contain the product version for the current running software.
5 to 6	7605 to 7606	Operating System	These two registers contain the month and year values for the program operating system.

Offset	Register	Content	Description
7 to 8	7607 to 7608	Run Number	These two registers contain the run number value for the currently running software.
9	7609	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
10	7610	Port 1 Command List Response	This field contains the number of slave response messages received on the port.
11	7611	Port 1 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
12	7612	Port 1 Requests	This field contains the total number of messages sent out of the port.
13	7613	Port 1 Responses	This field contains the total number of messages received on the port.
14	7614	Port 1 Errors Sent	This field contains the total number of message errors sent out of the port.
15	7615	Port 1 Errors Received	This field contains the total number of message errors received on the port.
16	7616	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
17	7617	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
18	7618	Port 2 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
19	7619	Port 2 Requests	This field contains the total number of messages sent out the port.
20	7620	Port 2 Responses	This field contains the total number of messages received on the port.
21	7621	Port 2 Errors Sent	This field contains the total number of message errors sent out of the port.
22	7622	Port 2 Errors Received	This field contains the total number of message errors received on the port.
23	7623	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
24	7624	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
25	7625	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
26	7626	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
27	7627	Command Block Count	This field contains the total number of command blocks received from the processor.
28	7628	Error Block Count	This field contains the total number of block errors recognized by the module.
29	7629	Port 1 Current Error	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the currently executing command.

Offset	Register	Content	Description
30	7630	Port 1 Last Error	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with an error.
31	7631	Port 2 Current Error	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the currently executing command.
32	7632	Port 2 Last Error	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with an error.

## 5.6 MVI71-MCM Configuration Data Definition

This section contains listings of the MVI71-MCM module's database related to the module's configuration. This data is available to any node on the network and is read from the PLC processor when the module first initializes.

### 5.6.1 Backplane Setup

Write Block Offset	Internal Database Register	Content	Description
1	5000	Write Start Reg	This parameter specifies the starting register in the module where the data transferred from the processor will be placed. Valid range for this parameter is 0 to 4999.
2	5001	Write Reg Count	This parameter specifies the number of registers to transfer from the processor to the module. Valid entry for this parameter is 0 to 5000.
3	5002	Read Start Reg	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 4999.
4	5003	Read Reg Count	This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 5000.
5	5004	Backplane Fail	This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to zero, the communication ports will continue to operate under all conditions. If the value is set larger than 0 (1 to 65535), communications will cease if the specified number of failures occur.
6	5005	Error Status Pointer	This parameter specifies the register location in the module's database where module status data will be stored. If a value less than zero is entered, the data will not be stored in the database. If the value specified in the range of 0 to 4940, the data will be placed in the user data area.

**5.6.2 Port 1 Setup**

Write Block Offset	Internal Database Register	Content	Description
7	5010	Enable	This parameter defines if this Modbus port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
8	5011	Type	This parameter specifies if the port will emulate a Modbus master device (0), a Modbus slave device without pass-through (1), or a Modbus slave device with unformatted pass-through (2), or a Modbus slave device with formatted pass-through and byte swapping (3).
9	5012	Float Flag	This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to 1, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.
10	5013	Float Start	This parameter defines the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled.
11	5014	Float Offset	This parameter defines the start register for floating-point data in the internal database. This parameter is only used if the Float Flag is enabled.
12	5015	Protocol	This parameter specifies the Modbus protocol to be used on the port. Valid protocols are: 0 = Modbus RTU and 1 = Modbus ASCII.
13	5016	Baud Rate	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entries are 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 384 (38400), 576 (57600), 115 (115200).
14	5017	Parity	This is the parity code to be used for the port. Values are None, Odd, Even.
15	5018	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Valid entries for this field are 5 through 8.
16	5019	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Valid entries are 1 and 2.
17	5020	RTS On	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Valid values are in the range of 0 to 65535 milliseconds.
18	5021	RTS Off	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low. Valid values are in the range of 0 to 65535.
19	5022	Minimum Response Time	This parameter specifies the minimum number of milliseconds to delay before responding to a request message. This pre-send delay is applied before the RTS on time. This may be required when communicating with slow devices.



Write Block Offset	Internal Database Register	Content	Description
20	5023	Use CTS Line	This parameter specifies if the CTS modem control line is to be used. If the parameter is set to 0, the CTS line will not be monitored. If the parameter is set to 1, the CTS line will be monitored and must be high before the module will send data. This parameter is normally only required when half-duplex modems are used for communication (2-wire).
21	5024	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. All requests received by the port with this address are processed by the module. Verify that each device has a unique address on a network. Valid range for this parameter is 1 to 255 (247 on some networks).
22	5025	Bit Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus Function 2 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.
23	5026	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network request for Modbus function 4 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.
24	5027	Bit Output Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus function 1,5, or 15 commands. For example, if the value is set to 100, an address request of 0 will correspond to register 100 in the database.
25	5028	Holding Reg Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus function 3, 6, or 16 commands. For example, if a value of 50 is entered, a request for address 0 will correspond to the register 50 in the database.
26	5029	Command Count	This parameter specifies the number of commands to be processed by the Modbus master port.
27	5030	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between issuing each command. This delay value is not applied to retries.
28	5031	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error will be placed. If the value is set to -1, the data will not be transferred to the database. The valid range of values for this parameter is -1 to 4999.
29	5032	Response Timeout	This parameter represents the message response timeout period in 1-millisecond increments. This is the time that a port configured as a master will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending upon the communication network used and the expected response time of the slowest device on the network.

Write Block Offset	Internal Database Register	Content	Description
30	5033	Retry Count	This parameter specifies the number of times a command will be retried if it fails. If the master port does not receive a response after the last retry, the slave devices communication will be suspended on the port for Error Delay Counter scans.
31	5034	Error Delay Counter	This parameter specifies the number of polls to skip on the slave before trying to re-establish communications. After the slave fails to respond, the master will skip commands to be sent to the slave the number of times entered in this parameter.
59	5068	Guard Band	Timeout. A value of 0 uses the default baud rate or you can set a timeout value in milliseconds (0 to 65535)

### 5.6.3 Port 2 Setup

Write Block Offset	Internal Database Register	Content	Description
32	5040	Enable	This parameter defines if this Modbus port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
33	5041	Type	This parameter specifies if the port will emulate a Modbus master device (0), a Modbus slave device without pass-through (1), or a Modbus slave device with unformatted pass-through (2), or a Modbus slave device with formatted pass-through and byte swapping (3).
34	5042	Float Flag	This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to 1, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.
35	5043	Float Start	This parameter defines the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled.
36	5044	Float Offset	This parameter defines the start register for floating-point data in the internal database. This parameter is only used if the Float Flag is enabled.
37	5045	Protocol	This parameter specifies the Modbus protocol to be used on the port. Valid protocols are: 0 = Modbus RTU and 1 = Modbus ASCII.
38	5046	Baud Rate	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entries are 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 384 (for 38400bps), 576 (for 57600bps), and 115 (for 115,200bps).
39	5047	Parity	This is the parity code to be used for the port. Values are None, Odd, Even.

<b>Write Block Offset</b>	<b>Internal Database Register</b>	<b>Content</b>	<b>Description</b>
40	5048	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Valid entries for this field are 5 through 8.
41	5049	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Valid entries are 1 and 2.
42	5050	RTS On	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Valid values are in the range of 0 to 65535 milliseconds.
43	5051	RTS Off	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low. Valid values are in the range of 0 to 65535.
44	5052	Minimum Response Time	This parameter specifies the minimum number of milliseconds to delay before responding to a request message. This pre-send delay is applied before the RTS on time. This may be required when communicating with slow devices.
45	5053	Use CTS Line	This parameter specifies if the CTS modem control line is to be used. If the parameter is set to 0, the CTS line will not be monitored. If the parameter is set to 1, the CTS line will be monitored and must be high before the module will send data. This parameter is normally only required when half-duplex modems are used for communication (2-wire).
46	5054	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. All requests received by the port with this address are processed by the module. Verify that each device has a unique address on a network. Valid range for this parameter is 1 to 255 (247 on some networks).
47	5055	Bit Input Offset	This parameter specifies the offset word address in the internal Modbus database that is to be used with network requests for Modbus Function 2 commands. For example, if the value is set to 150, an address request of 0 will return the value at the first bit in register 150 in the database.
48	5056	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network request for Modbus function 4 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.
49	5057	Bit Output Offset	This parameter specifies the offset word address in the internal Modbus database that is to be used with network requests for Modbus function 1,5, or 15 commands. For example, if the value is set to 100, an address request of 0 will correspond to the first bit at register 100 in the database.
50	5058	Holding Reg Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus function 3, 6, or 16 commands. For example, if a value of 50 is entered, a request for address 0 will correspond to the register 50 in the database.
51	5059	Command Count	This parameter specifies the number of commands to be processed by the Modbus master port.

Write Block Offset	Internal Database Register	Content	Description
52	5060	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between issuing each command. This delay value is not applied to retries.
53	5061	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error will be placed. If the value is set to -1, the data will not be transferred to the database. The valid range of values for this parameter is -1 to 4999.
54	5062	Response Timeout	This parameter represents the message response timeout period in 1-millisecond increments. This is the time that a port configured as a master will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending upon the communication network used and the expected response time of the slowest device on the network.
55	5063	Retry Count	This parameter specifies the number of times a command will be retried if it fails. If the master port does not receive a response after the last retry, the slave devices communication will be suspended on the port for Error Delay Counter scans.
56	5064	Error Delay Counter	This parameter specifies the number of polls to skip on the slave before trying to re-establish communications. After the slave fails to respond, the master will skip commands to be sent to the slave the number of times entered in this parameter.
60	5069	Guard Band	Timeout. A value of 0 uses the default baud rate or you can set a timeout value in milliseconds (0 to 65535)

Guard Band Parameter

With multiple nodes in the network (multidrop), the module must be able to define the time period used to properly recognize the received data sequence as a valid Modbus message. This time period is defined through the guardband timer parameter. Failure to set the Guard Band functionality may cause successive communication errors in the Modbus network.

This functionality is only used for Modbus RTU mode, when the module is configured as a slave device in a multidrop network (RS-422 or RS-485 wiring). In RTU mode, when the module receives sequence of data from another node, it will recognize the end of the message when it detects a 3.5 character gap. The next byte received after a 3.5 character gap will be recognized as the start of a new Modbus message.

The 3.5 character gap depends on the port communication settings, especially the baud rate. The higher the baud rate, the lower the corresponding time period equivalent to the 3.5 character delay.

To use the default Guard Band Timeout parameter, set the Guard Band Time as follows:

Guard Band Time = 0

This configuration will set the module to use a pre-defined time interval for Modbus message recognition calculated for each baud rate. This should provide optimal performance for most applications.

The following guard band timer values will be automatically used by the module with the default parameter value (Guard Band Timer = 0) is selected:

Baud	Guard Band Timer (ms)
110	350
150	256
300	128
600	64
1200	32
2400	16
4800	8
9600	4
19200	2
28800	2
38400	2
57600	1
115200	1

If you still observe a considerable number of communication errors, try to increase the Guard Band Time value (in milliseconds) until the network performance is improved.

#### 5.6.4 Port 1 Commands

Register	Content	Description
5200 to 5207	Command #1	This set of registers contains the parameters for the first command in the master command list. The structure of this data area is as described in the data object section of the documentation.
5210 to 5217	Command #2	Command #2 data set
-	-	-
6190 to 6197	Command #100	Command #100 data set

#### 5.6.5 Port 2 Commands

Register	Content	Description
6400 to 6407	Command #1	This set of registers contains the parameters for the first command in the master command list. The structure of this data area is as described in the data object section of the documentation.
6410 to 6417	Command #2	Command #2 data set
-	-	-
7390 to 7397	Command #100	Command #100 data set

### 5.6.6 Command Control

Register	Content	Description
7800	Command Code	Enter one of the valid control command codes in this register to control the module (9997, 9998, or 9999). Refer to <a href="#">MVI71-MCM Command Error Codes</a> (page 94) for more information.
7801	Command Data	Not Used
-	-	-
7999	Command Data	Not Used

## 5.7 MVI71-MCM Command Error Codes

The following tables list the MVI71-MCM Command Error Codes:

### 5.7.1 Standard Modbus Protocol Errors

Code	Description
1	Illegal Function
2	Illegal Data Address
3	Illegal Data Value
4	Failure in Associated Device
5	Acknowledge
6	Busy, Rejected Message

### 5.7.2 Module Communication Error Codes

Code	Description
-1	CTS modem control line not set before transmit
-2	Timeout while transmitting message
-11	Timeout waiting for response after request
253	Incorrect slave address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

### 5.7.3 Command List Entry Errors

Code	Description
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (< 0 or > 255)
-44	Count parameter set to 0
-45	Invalid function code
-46	Invalid swap code

## 5.8 Configuration Error Codes

The configuration update procedure starts when the module receives a configuration block (Write Block ID = 9000):

Offset	Description	Length
0	9000	1
1 to 6	Backplane Setup	6
7 to 31	Port 1 Configuration Errors	25
32 to 56	Port 2 Configuration Errors	25
57 to 63	Spare	7

The module then responds with a Read Block (ID = -2) with any possible errors:

### 5.8.1 BTR

Offset	Description	Length
0	-2	1
1	9000	1
2	Module Configuration Errors	1
3	Port 1 Configuration Errors	1
4	Port 2 Configuration Errors	1
5 to 63	Spare	58

The bits in each configuration word are shown below. The module configuration error word has the following definition:

Bit	Description	Value
0	Write block start value is greater than the database size.	0x0001
1	Write block start value is less than zero.	0x0002
2	Write block count value is less than zero.	0x0004
3	Write block count + start is greater than the database size.	0x0008
4	Read block start value is greater than the database size.	0x0010
5	Read block start value is less than zero.	0x0020
6	Read block count value is less than zero.	0x0040
7	Read block count + start is greater than the database size.	0x0080
8		0x0100
9		0x0200
10		0x0400
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

The port configuration error words have the following definitions:

Bit	Description	Value
0	Type code is not valid. Enter a value from 0 (master) to 1 (slave).	0x0001
1	The float flag parameter is not valid.	0x0002
2	The float start parameter is not valid.	0x0004
3	The float offset parameter is not valid.	0x0008
4	Protocol parameter is not valid.	0x0010
5	Baud rate parameter is not valid.	0x0020
6	Parity parameter is not valid.	0x0040
7	Data bits parameter is not valid.	0x0080
8	Stop bits parameter is not valid.	0x0100
9	Slave ID is not valid.	0x0200
10	Input bit or word, output word and/or holding register offset(s) are not valid.	0x0400
11	Command count parameter is not valid.	0x0800
12	Spare	0x1000
13	Spare	0x2000
14	Spare	0x4000
15	Spare	0x8000

## 5.9 Frequently Asked Questions

### 5.9.1 What is the MVI71-MCM?

The MVI71-MCM is a single slot solution that allows Modbus devices to interface easily with a PLC processor. The module has 2 ports individually configurable either as a Modbus Master or Modbus Slave.

The module has an internal database with 5000 registers that are accessible to external Modbus devices as well as accessible to the PLC through the rack backplane. Ladder logic is required, among other tasks, to access the MVI71-MCM internal database.

### 5.9.2 Why does the module have to break down the data in blocks of 64 words?

Because of a limitation of the PLC backplane that allows only 64 words at a time.

### 5.9.3 How do I read and write data between the internal MVI71-MCM database and the PLC?

Ladder logic is required to read and write blocks from the MVI71-MCM. Initially you must configure the Read and Write Data areas inside the MVI71-MCM database. It creates 2 regions: the Read Data area which is constantly read from the module database to the PLC, and the Write Data area which is constantly written from the PLC ladder logic to the MVI71-MCM database.



Block Transfer instructions transfer data: the BTR instruction reads data from the module and the BTW instruction writes data to the module.

If using side connect (requires side connect adapter) option, ladder logic is not required to transfer data to the MVI71-MCM.

#### ***5.9.4 What is the procedure to change the module's configuration?***

A simple boot operation will force the module to request configuration data. The ladder logic can force the module to perform a Cold Boot or Warm Boot.

#### ***5.9.5 How do I set a Port Modbus Master command?***

You can create Modbus commands using a PLC data file that is copied to the BTW data file by the ladder logic. The sample ladder logic uses:

N11: Port 1 Master Commands

N12: Port 2 Master Commands

Ladder logic is necessary to copy the commands to the MVI71-MCM during the boot operation (the sample ladder already handles it).

#### ***5.9.6 The Modbus Master Command is not working, what should I do?***

Look at the Command Error Menu in the debug port and write down the code associated with the command index. Look for the meaning of the error code in [MVI71-MCM Command Error Codes](#) (page 94) and ensure that the parameters entered are valid.

If there is no error code in the Command Error Menu, the Modbus command could refer to a different location than the one considered.

#### ***5.9.7 What is the Pass-Through feature?***

An MVI71-MCM slave port can be configured for pass-through mode. This causes every write command to write directly to the PLC, without modifying the MVI71-MCM internal database. In order to accomplish this, every time a slave pass-through port receives a write command, the module generates specific blocks that allows ladder logic to handle each write command internally (refer to the Pass-Through section).

#### ***5.9.8 How can I check a Slave's status connected to an MVI71-MCM Master port?***

The debug port allows the user to monitor every slave attached to a master port.

### **5.9.9 *Should I use the sample ladder logic?***

Yes, the sample ladder should always be used as the source for your application. You may copy and paste the ladder logic to your current application.

### **5.9.10 *Is it possible to control the Master Modbus Commands from ladder logic?***

Yes, the ladder logic can create and send Modbus commands (event command) or enable commands that are currently disabled in the Master Command List (command control).

Refer to the User Manual for more information.

### **5.9.11 *How is the MVI71-MCM database divided into Discrete Output, Discrete Input, Input Registers, and Holding Registers?***

The MVI71-MCM uses the same database (0 to 4999) for all of these data types. The only difference is how you address each point. For Discrete Output or Discrete Input, use a bit addressing (1 bit) and for Input Registers or Holding Registers, use a word addressing (16 bits).

## 6 Support, Service & Warranty

### *In This Chapter*

- How to Contact Us: Sales and Support..... 100
- Return Material Authorization (RMA) Policies and Conditions101
- Procedures for Return of Units Under Warranty..... 101
- Procedures for Return of Units Out of Warranty..... 102
- LIMITED WARRANTY ..... 103

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

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

---

<b>Internet</b>	Web Site: <a href="http://www.prosoft-technology.com/support">http://www.prosoft-technology.com/support</a> E-mail address: <a href="mailto:support@prosoft-technology.com">support@prosoft-technology.com</a>
-----------------	---

---

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

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

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

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

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

## 6.1 How to Contact Us: Sales and Support

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

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

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

### **Asia Pacific (office in Malaysia)**

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

### **China Pacific (office in China)**

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

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

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

### **Brasil (office in Sao Paulo)**

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

## 6.2 Return Material Authorization (RMA) Policies and Conditions

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

### 6.2.1 All Product Returns

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

The following policy applies for Non-Warranty Credit Returns:

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

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

## 6.3 Procedures for Return of Units Under Warranty

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

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

## 6.4 Procedures for Return of Units Out of Warranty

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

### 6.4.1 Un-repairable Units

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

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

### **6.4.2 Purchasing Warranty Extension**

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

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

## **6.5 LIMITED WARRANTY**

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

### **6.5.1 What Is Covered By This Warranty**

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

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

### **6.5.2 What Is Not Covered By This Warranty**

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



### **6.5.3 DISCLAIMER REGARDING HIGH RISK ACTIVITIES**

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

### **6.5.4 DISCLAIMER OF ALL OTHER WARRANTIES**

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

### **6.5.5 LIMITATION OF REMEDIES\*\***

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

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

### **6.5.6 Time Limit for Bringing Suit**

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

### **6.5.7 No Other Warranties**

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

### **6.5.8 Intellectual Property**

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

### **6.5.9 Additional Restrictions Relating To Software And Other Intellectual Property**

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

**6.5.10 Allocation of risks**

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

**6.5.11 Controlling Law and Severability**

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



# Index

## A

Additional Restrictions Relating To Software  
And Other Intellectual Property • 106  
All Product Returns • 101  
Allocation of risks • 107  
Analyzing Data for the first application port •  
41  
Analyzing Data for the second application  
port • 41

## B

Backplane Data Transfer • 57  
Backplane Setup • 87  
Block Transfer • 69  
BTR • 95

## C

Cable Connections • 79  
Changing Parameters During Operation • 24  
Clearing a Fault Condition • 51  
Cold Boot • 73  
Command Control • 71, 94  
Command Control Blocks • 69  
Command List Entry Errors • 94  
Configuration Data • 24  
Configuration Data Transfer • 23  
Configuration Error Codes • 24, 85, 95  
Configuring RSLinx • 15  
Configuring the Module • 22, 77, 79  
Connect your PC to the Module • 17  
Connect your PC to the Processor • 13  
Controlling Law and Severability • 107

## D

Data Analyzer • 41  
Data Analyzer Tips • 43  
Data Flow Between MVI71-MCM Module  
and PLC Processor • 76  
Data Transfer Parameter • 25  
Database View Menu • 46  
DB9 to RJ45 Adaptor (Cable 14) • 84  
Diagnostics and Troubleshooting • 7, 35  
Disabling the RSLinx Driver for the Com Port  
on the PC • 37, 79  
DISCLAIMER OF ALL OTHER  
WARRANTIES • 105  
DISCLAIMER REGARDING HIGH RISK  
ACTIVITIES • 105

Displaying the Current Page of Registers  
Again • 46  
Displaying Timing Marks in the Data  
Analyzer • 42  
Download the Sample Program to the  
Processor • 14

## E

Error Codes • 39, 48, 66, 94, 95  
Event Command • 70  
Exiting the Program • 41

## F

Features and Benefits • 53  
Formatted Pass-Through Control Blocks • 74  
Frequently Asked Questions • 96  
Function 15 • 75  
Function 5 • 74  
Function 6 and 16 • 74  
Functional Overview • 7, 55  
Functional Specifications • 55

## G

General Concepts • 55  
General Specifications • 54  
Guard Band Parameter • 92  
Guide to the MVI71-MCM User Manual • 7

## H

Hardware Specifications • 54  
How can I check a Slave's status connected  
to an MVI71-MCM Master port? • 97  
How do I read and write data between the  
internal MVI71-MCM database and the  
PLC? • 96  
How do I set a Port Modbus Master  
command? • 97  
How is the MVI71-MCM database divided  
into Discrete Output, Discrete Input, Input  
Registers, and Holding Registers? • 98  
How to Contact Us  
Sales and Support • 100

## I

Install the Module in the Rack • 12  
Installing and Configuring the Module • 19  
Intellectual Property • 106  
Is it possible to control the Master Modbus  
Commands from ladder logic? • 98

## K

Keystrokes • 36

## L

Ladder Logic • 33  
LED Status Indicators • 50  
LIMITATION OF REMEDIES\*\* • 105  
LIMITED WARRANTY • 103

## M

Main Logic Loop • 56  
Main Menu • 38  
Master Command Blocks • 66  
Master Command Error List Menu • 48  
Master Command List Menu • 49  
Master Driver Mode • 78  
Modbus Addressing • 59  
Modbus Master Commands • 26  
Modbus Port Parameters • 26  
Module Communication Error Codes • 94  
Module Configuration • 23  
Module Data Files • 24  
Moving Back Through 5 Pages of Registers • 47  
MVI71-MCM Command Error Codes • 66, 85, 94, 97  
MVI71-MCM Configuration Data Definition • 23, 26, 85, 87  
MVI71-MCM Database Definition • 85  
MVI71-MCM Status Data Definition • 29, 85

## N

Navigation • 35  
No Other Warranties • 106  
Normal Data Transfer • 63

## O

Opening the Command Error List Menu • 39  
Opening the Command List Menu • 39  
Opening the Data Analyzer Menu • 38  
Opening the Database Menu • 39

## P

Package Contents • 10  
Pass-Through • 77  
Pass-Through Control Blocks • 74  
PLC Processor Not in Run • 56  
Please Read This Notice • 2  
Port 1 Commands • 93  
Port 1 Setup • 88  
Port 2 Commands • 93  
Port 2 Setup • 90  
Power Up • 22  
Procedures for Return of Units Out of Warranty • 102  
Procedures for Return of Units Under Warranty • 101

Product Specifications • 7, 53  
Purchasing Warranty Extension • 103

## Q

Quick Start • 19

## R

Read Block • 63  
Read Current Configuration • 72  
Read Data Area Application Examples • 61  
Reading Status Data From the Module • 7, 35  
Redisplaying the Current Page • 48, 49  
Reference • 7, 53  
Remote Command Control • 75  
Removing Timing Marks in the Data Analyzer • 42  
Required Hardware • 36  
Required Software • 36  
Return Material Authorization (RMA) Policies and Conditions • 101  
Returning to the Main Menu • 43, 47, 48, 49  
RS-232 • 81  
    Modem Connection • 82  
    Null Modem Connection (Hardware Handshaking) • 82  
    Null Modem Connection (No Hardware Handshaking) • 83  
RS-232 Configuration/Debug Port • 79  
RS-422 • 83  
RS-485 • 84  
RS-485 and RS-422 Tip • 84

## S

Setting Jumpers • 11  
Should I use the sample ladder logic? • 98  
Side-Connect (Requires Side-Connect Adapter) • 70  
Skipping 500 Registers of Data • 47  
Slave Driver • 76  
Slave Polling Control and Status • 29  
Slave Status Blocks • 67  
Standard Modbus Protocol Errors • 94  
Start Here • 7, 9  
Starting the Data Analyzer • 42  
Status Data • 29  
Status Data Block (Read Block ID = -1) • 64  
Stopping the Data Analyzer • 43  
Support, Service & Warranty • 7, 99  
System Requirements • 9

## T

The Configuration/Debug Menu • 35  
The Modbus Master Command is not working, what should I do? • 97

Time Limit for Bringing Suit • 105  
Transferring Module Configuration to the  
Processor • 40  
Transferring the Command Error List to the  
Processor • 66  
Troubleshooting • 51

**Y**

Your Feedback Please • 3

**U**

Un-repairable Units • 102  
User Data • 29  
Using Side-Connect (Requires Side-Connect  
Adapter) • 29  
Using the Command Control Blocks • 69  
Using the Configuration/Debug Port • 37  
Using the Read and Write Data Areas • 59

**V**

Viewing Block Transfer Statistics • 38  
Viewing Data in ASCII (Text) Format • 42, 47  
Viewing Data in Decimal Format • 47  
Viewing Data in Floating Point Format • 47  
Viewing Data in Hexadecimal Format • 42,  
47  
Viewing Module Configuration • 39  
Viewing Port Communication Status • 40  
Viewing Port Configuration • 40  
Viewing Register Pages • 46  
Viewing the Next 100 Registers of Data • 47  
Viewing the Next 20 Commands • 48  
Viewing the Next 50 Commands • 49  
Viewing the Next Page of Commands • 48,  
49  
Viewing the Previous 100 Registers of Data •  
47  
Viewing the Previous 20 Commands • 48  
Viewing the Previous 50 Commands • 49  
Viewing the Previous Page of Commands •  
48, 49  
Viewing the Slave Status List (Port 1 and 2) •  
39  
Viewing Version Information • 39

**W**

Warm Boot • 73  
Warm Booting the Module • 40  
What Is Covered By This Warranty • 103  
What Is Not Covered By This Warranty • 104  
What is the MVI71-MCM? • 96  
What is the Pass-Through feature? • 97  
What is the procedure to change the  
module's configuration? • 97  
Why does the module have to break down  
the data in blocks of 64 words? • 96  
Write Block • 64  
Write Data Area Application Examples • 62