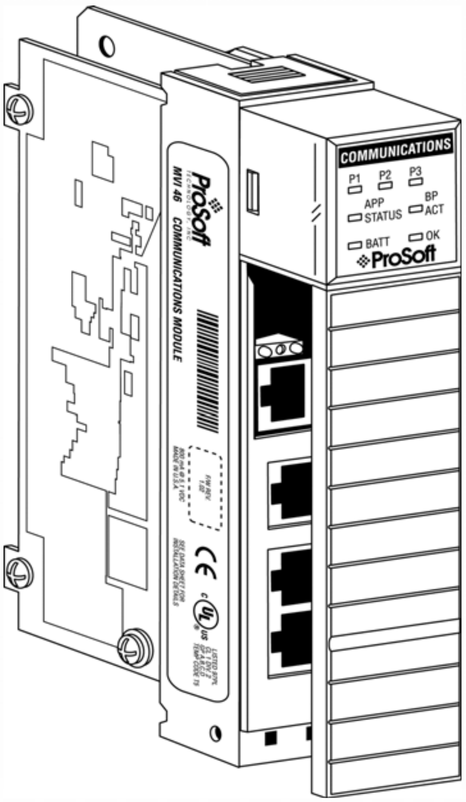
inRAx



MVI46-MCM

SLC Platform

Modbus Interface Module

User Manual

December 27, 2004

ProSoft
TECHNOLOGY

Please Read This Notice

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

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

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

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

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

ProSoft Technology, Inc.

1675 Chester Avenue, 2nd Floor

Bakersfield, CA 93301

(661) 716-5100

(661) 716-5101 (Fax)

www.prosoft-technology.com

MVI46-MCM User Manual

December 27, 2004

Table of Contents

1 Product Specifications

The MVI46-MCM (“Modbus Communication Module”) product allows Allen-Bradley SLC I/O compatible processors to easily interface with other Modbus protocol compatible devices. Compatible devices include not only Modicon PLC's (which all support the Modbus protocol) but also a wide assortment of end devices.

1.1 General Specifications

The MVI46-MCM module acts as a gateway between the Modbus network and the Allen-Bradley backplane. The data transfer from the SLC processor is asynchronous from the actions on the Modbus network. A 5000-word register space in the module is used to exchange data between the processor and the Modbus network.

Some of the general specifications include:

- Support for the storage and transfer of up to 5000 registers to/from the SLC processor's data files
- Module memory usage that is completely user definable
- Two ports to emulate any combination of Modbus master or slave device
- Configurable parameters include:

Protocol	:	RTU or ASCII
Baud Rate	:	110 to 115,200
Parity	:	None, Odd and Even
Data Bits	:	5 to 8
Stop Bits	:	1 or 2
RTS On and Off Timing	:	0 to 65535 milliseconds
Minimum Response Delay	:	0 to 65535 milliseconds
Use of CTS Modem Line	:	Yes or No
Device Routing Paths	:	1 to 64
Floating-Point Support		

1.1.1 Slave Functional Specifications

The MVI46-MCM module accepts Modbus function code commands of 1, 2, 3, 4, 5, 6, 8, 15 and 16 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 SLC processor.

1.1.2 Modbus Master Functional Specifications

A port configured as a virtual Modbus master device on the MVI46-MCM module will actively issue 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 will poll slaves with communication problems less frequently.

The SLC 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. The SLC processor also has the ability to control the scanning of slaves on the port. Polling of individual slaves can be selectively controlled (enabled/disabled) through the ladder logic.

1.1.3 Physical

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

- SLC Form Factor - Single Slot
- Connections:

  2– RJ45 connectors for support of RS-232, RS-422 or RS-485 interfaces

  1– RJ45 RS-232 Configuration Tool Connector

1.1.4 SLC Interface

- Operation via simple ladder logic
- Complete set up and monitoring of module through RSLogix 500 software
- SLC backplane interface via I/O access
- SLC 5/02 (M0/M1 compatible) or higher is required
- Module configuration and communication configuration data is transferred to the MVI46-MCM from the M1 file.

1.2 Hardware Specifications

The MVI46-MCM module is designed by ProSoft Technology and incorporates licensed technology from Allen-Bradley (SLC backplane technology).

• Current Loads:	800 ma @ 5V (from backplane)
• Operating Temperature:	0 to 60 Deg C (32 to 140 Deg F)
• Storage Temperature:	-40 to 85 Deg C (-40 to 185 Def F)
• Relative Humidity:	5-95% (w/o condensation)

- Modbus Port Connector: Two RJ45 Connectors (RJ45 to DB9 cable shipped with unit) supporting RS-232, RS-422 and RS-485 interfaces (RJ45 to DB9 cables shipped with unit)
- Configuration Connector: RJ45 RS-232 Connector (RJ45 to DB9 cable shipped with unit)

2 Functional Overview

This section gives the reader a functional overview of the MVI46-MCM module. Details associated with the ladder logic and the memory-map are not covered in this section (refer to the **Module Set Up** section). A thorough understanding of the information contained in this document is required for successful implementation of the module in a user application. If you are not familiar with the data transfer and Modbus protocol operations, read this section before setting up the module.

2.1 General Concepts

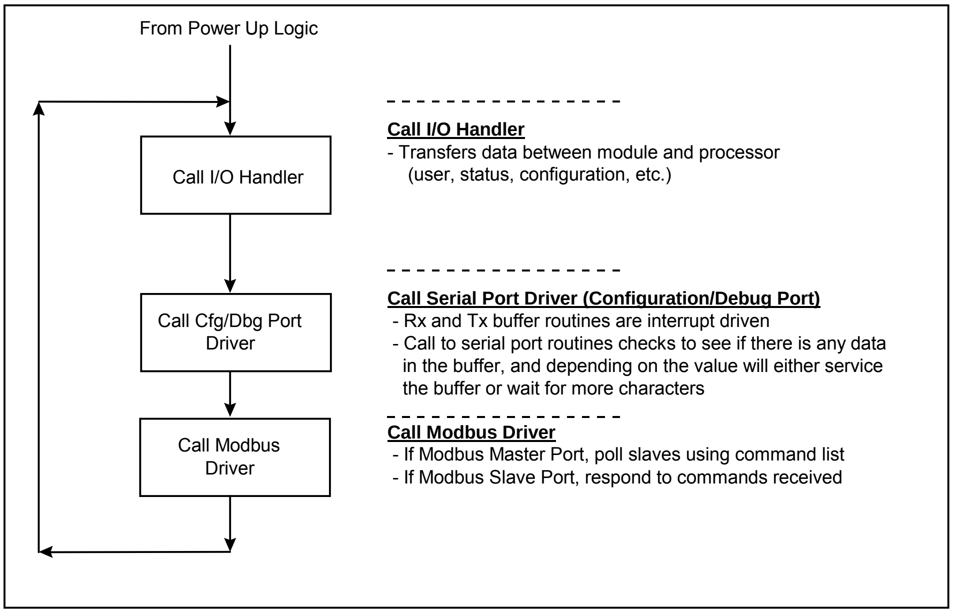
The following discussion covers several concepts that are key to understanding the operation of the MVI46-MCM module.

1. On power up the module begins performing the following logical functions:
    a. Initialize hardware components
    b. Initialize SLC backplane driver
    c. Test and Clear all RAM
    d. Initialize the serial communication ports
2. Waits for module configuration from the SLC processor
3. Allocate and initialize Module Register space
4. Enable Slave Driver on selected ports
5. Enable Master Driver on selected port if configured

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

2.1.1 Main Logic Loop

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


From Power Up Logic
Call I/O Handler
Call Cfg/Dbg Port Driver
Call Modbus Driver
Call I/O Handler
- Transfers data between module and processor (user, status, configuration, etc.)
Call Serial Port Driver (Configuration/Debug Port)
- Rx and Tx buffer routines are interrupt driven
- Call to serial port routines checks to see if there is any data in the buffer, and depending on the value will either service the buffer or wait for more characters
Call Modbus Driver
- If Modbus Master Port, poll slaves using command list
- If Modbus Slave Port, respond to commands received

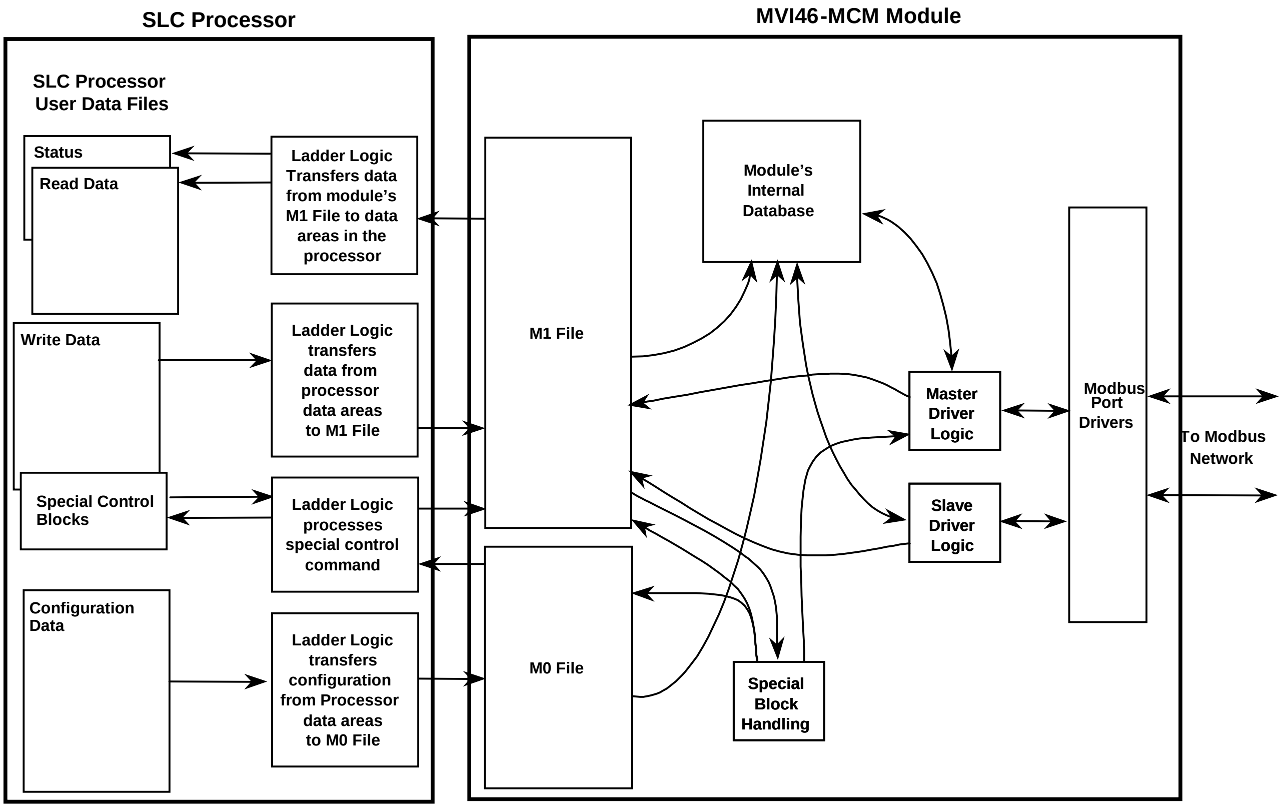
2.1.2 SLC Processor Not in Run

Anytime the module detects that the processor has gone out of the Run mode (i.e., Fault or PGM), the Modbus ports will be shut down. When the processor is returned to a running state, the module will resume communications on the network. No backplane data transfers occur when the processor is not in run mode.

2.1.3 Backplane Data Transfer

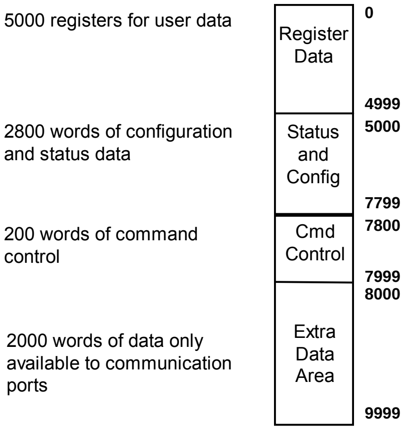
The MVI46-MCM module is unique in the way that the SLC backplane is used. All data for the module is contained in the module’s M1 file. Data is moved between the module and the SLC processor across the backplane using the module's M-files. The SLC scan rate and the communication load on the module determine the update frequency of the M-files. The COP instruction can be used to move data between user data files and the module’s M1 file.

The following diagram displays the data transfer method used to move data between the SLC processor, the MVI46-MCM module and the MCM network.


SLC Processor
MVI46-MCM Module
SLC Processor User Data Files
Status
Read Data
Ladder Logic Transfers data from module's M1 File to data areas in the processor
Write Data
Ladder Logic transfers data from processor data areas to M1 File
Special Control Blocks
Ladder Logic processes special control command
Configuration Data
Ladder Logic transfers configuration from Processor data areas to M0 File
M1 File
M0 File
Module's Internal Database
Master Driver Logic
Slave Driver Logic
Special Block Handling
Modbus Port Drivers
To Modbus Network

As shown in the previous diagram, all data transferred between the module and the processor over the backplane is through the M0 and M1 files. Ladder logic must be written in the SLC processor to interface the M-file data with data defined in the user-defined data files in the SLC. All data used by the module is stored in its internal database. The following diagram displays the layout of the database:

Module's Internal Database Structure – M1 File


5000 registers for user data
Register Data
0
4999
2800 words of configuration and status data
Status and Config
5000
7799
200 words of command control
Cmd Control
7800
7999
2000 words of data only available to communication ports
Extra Data Area
8000
9999

User data contained in this database is continuously read from the M1 file. The configuration data is only updated in the M1 file after each configuration request by the module to the SLC. All data in the M1 file is available to devices on the Modbus networks. This permits data to be transferred from these devices to the SLC using the user data area. Additionally, remote devices can alter the module's configuration, read the status data, and issue control commands. Block identification codes are used to define specific functions to the module.

The block identification codes used by the module are listed in the following table:

Block Range	Descriptions
1000	Event Port 1
2000	Event Port 2
3000 to 3001	Port 1 slave polling control
3100 to 3101	Port 2 slave polling control
5000 to 5006	Port 1 command control
5100 to 5106	Port 2 command control
9000	Configuration request from module
9001	Configuration ready from controller
9997	Write configuration to controller
9998	Warm-boot control block
9999	Cold-boot control block

Each block has a defined structure depending on the data content and the function of the data transfer as defined in the following sections.

2.2 Normal Data Transfer

This version of the module provides for direct access to the data in the module. All data related to the module is stored in the module's M1 file. To read data from the module, simply use the COP instruction to copy data from the module's M1 file to a user data file. To write data to the module, use the COP instruction to copy data from a user file to the module's M1 file. Registers 0 to 4999 are those that should be used for user data. All other registers are reserved for other module functions.

2.3 Configuration Data Transfer

When the module performs a restart operation, it will request configuration information from the SLC processor. This data is transferred to the module in a specially formatted write block in the M0 file. The module will poll for this information by placing the value 9000 in word 0 of the M0 file. The ladder logic must construct the requested block in order to configure the module. Refer to the **Module Set Up** section for a description of the data objects used with the blocks and the ladder logic required. The format of the block for configuration is given in the following section.

2.3.1 Module Configuration Data

This block is used to send configuration information from the processor to the module. The data is transferred in a block with an identification code of 9001. The structure of the block is displayed in the following table:

M0 Offset	Description	Length
0	9001	1
1-6	Backplane Setup	6

M0 Offset	Description	Length
10-39	Port 1 Configuration	30
40-69	Port 2 Configuration	30
70-79	Port 1 Command # 0 Definition	10
80-89	Port 1 Command # 1 Definition	10
90-1059	Port 1 Command # 2 to # 98	980
1060-1069	Port 1 Command # 99 Definition	10
1070-1079	Port 2 Command # 0 Definition	10
1080-1089	Port 2 Command # 1 Definition	10
1090-2059	Port 2 Command # 2 to # 98	980
2060-2069	Port 2 Command # 99 Definition	10

If there are any errors in the configuration, the bit associated with the error will be set in one of the two configuration error words. The error must be corrected before the module starts its normal mode of operation.

2.4 Slave Status Data

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

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.
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 (**Error Delay Counter** value in the module configuration for each port). 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.

In order to read the slave status table, ladder logic must be written and the slave status data must be located in the user data area. The module will constantly update the user defined data area with the slave data for each Modbus master port. This data can be transferred to a user-defined file in the processor using the COP instruction.

Ladder logic can be written to override the value in the slave status table to disable slaves (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:

M1 Offset	Description	Length
7800	3000 or 3100	1
7801	Number of Slaves in Block	1
7802-7927	Slave Indexes	126

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 shown in the following table:

M0 Offset	Description	Length
0	3000 or 3100	1
1	Number of Slaves Processed	1

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 format of the block is shown in the following table:

M0 Offset	Description	Length
7800	3001 or 3101	1
7801	Number of Slaves in Block	1
7802-7927	Slave Indexes	126

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 shown in the following table:

M0 Offset	Description	Length
0	3001 or 3101	1
1	Number of Slaves Processed	1

2.5 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.

2.5.1 Event Command

Event command control blocks are used to send Modbus commands directly from the ladder logic to one of the master ports. The format for these blocks is shown in the following table:

M1 Offset	Description	Length
7800	1000 or 2000	1
7801	Internal DB Address	1
7802	Point Count	1
7803	Swap Code	1
7804	Node Address	1
7805	Function Code	1
7806	Device Address	1

The block number defines the Modbus port to be considered. Block 1000 commands are directed to Port 1, and block 2000 commands are directed to Port 2. The parameters passed with the block are used to 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 registers for the command. The **Swap Code** is used to change the word or byte order. The **Node Address** parameter is used to define the device on the Modbus network to consider. The **Function Code** parameter is one of those defined in the ProSoft Modbus Command Set documentation. When the block is received, the module will process it and place the command in the command queue. The module will respond to each event command block with a read block in the following format:

M0 Offset	Description	Length
0	1000 or 2000	1
1	0=Fail, 1=Success	1

The ladder logic can use Word one of the block 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) or the command requested is invalid.

2.5.2 Command Control

Command control blocks are used to place commands from 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 format of the block is shown in the following table:

M1 Offset	Description	Length
7800	5001-5006 or 5101-5106	1
7801	Command Index	1
7802	Command Index	1
7803	Command Index	1
7804	Command Index	1
7805	Command Index	1
7806	Command Index	1

Blocks in the range of 5001 to 5006 are used for Port 1, while blocks in the range of 5101 to 5106 are used for 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 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 format of the block is shown in the following table:

M0 Offset	Description	Length
0	5000-5006 or 5100-5106	1
1	Number of commands added to comm and queue	1

2.5.3 Write Configuration

This block is sent from the SLC processor to the module to force the module to write it's 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 9997 in the first word. The module responds with a block containing the module configuration data. Ladder logic must be written to handle the receipt of the block. The block transferred from the module is as follows:

M0 Offset	Description	Length
0	9001	1
1-9	Backplane Setup	9
10-39	Port 1 Configuration	30
40-69	Port 2 Configuration	30
70-79	Port 1 Command # 0 Definition	10
80-89	Port 1 Command # 1 Definition	10
90-1059	Port 1 Command # 2 to # 98	980
1060-1069	Port 1 Command # 99 Definition	10
1070-1079	Port 2 Command # 0 Definition	10
1080-1089	Port 2 Command # 1 Definition	10

M0 Offset	Description	Length
1090-2059	Port 2 Command # 2 to # 98	980
2060-2069	Port 2 Command # 99 Definition	10

Ladder logic must be written to process this block of information and place the data received in the correct data files in the SLC. The processor requests this block of information using the following write block:

M1 Offset	Description	Length
7800	9997	1

2.5.4 Warm Boot

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

M1 Offset	Description	Length
7800	9998	1

2.5.5 Cold Boot

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

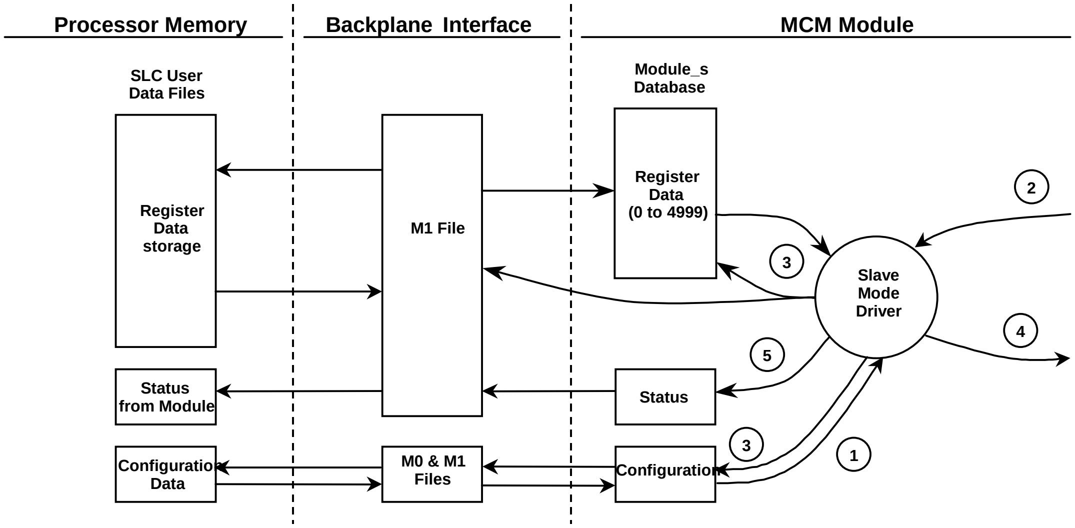
M1 Offset	Description	Length
7800	9999	1

Data Flow Between MVI46-MCM Module and SLC Processor

The following discussion describes the flow of data between the two pieces of hardware (SLC processor and MVI46-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 following sections discuss the operation of each mode.

2.5.6 Slave Driver

The Slave Driver Mode allows the MVI46-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 in and out of the module.

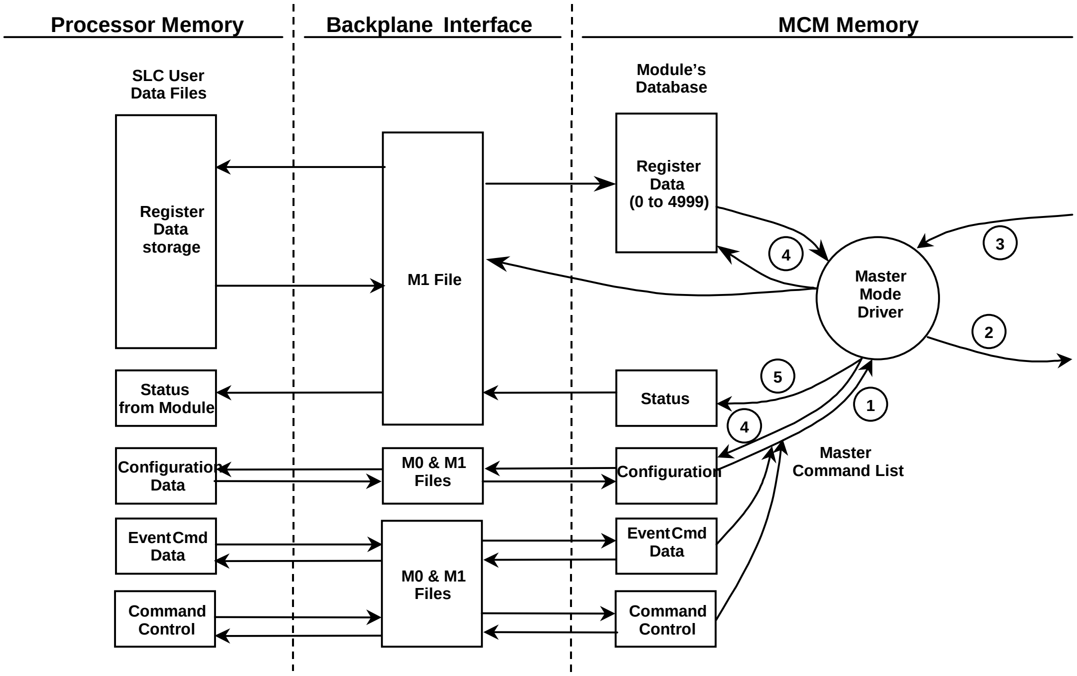


Step	Description
1	The MODBUS slave port driver receives the configuration information from the SLC processor. This information is used to configure the serial port and define the slave node characteristics.
2	A Host device, such as an Allen-Bradley PLC or an MMI package issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	Once 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 the M1 file and a response message is built.
4	Once the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

Review the **Module Set Up** section for a complete list of the parameters that must be defined for a slave port.

2.5.7 Master Driver Mode

In the Master Mode of operation, the MVI46-MCM module is responsible for issuing 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 SLC processor or issued directly from the SLC 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 in and out of the module.



Step	Description
1	The Master driver obtains configuration data from the SLC 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 (See the Module Set Up section).
2	Once configured, 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 and the M1 file, assuming a read command.
5	Status is returned to the SLC processor for each command in the Master Command List.

Refer to the **Module Set Up** section 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.

2.5.7.1. Master Command List

In order to function in the Master Mode, the module's Master Command List must be defined. This list contains up to 100 individual entries, with each entry containing the information required to construct a valid command. This includes the following:

- Command Enable Code ((0) disabled, (1) continuous or (2) conditional)
- Internal Address – Module's internal address associated with the command.

- Poll Interval Time – Minimum number of seconds between issuance of command (0 to 65535)
- Count – The number of words to be transferred.
- Swap Code – Swap code used with the command ((1) swap words, (2) swap bytes and words, (3) swap bytes)
- Node Address – Device ID
- Function Code – Function code for the command.
- Device Address – Address in the device associated with the command.

Modbus Command Structure

Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Function Code	Device Address
Fc1	Code	Bit	Seconds	Count	0	Node	1	Register
Fc2	Code	Bit	Seconds	Count	0	Node	2	Register
Fc3	Code	Register	Seconds	Count	Code	Node	3	Register
Fc4	Code	Register	Seconds	Count	Code	Node	4	Register
Fc5	Code	Bit	Seconds	Count	0	Node	5	Register
Fc6	Code	Register	Seconds	Count	Code	Node	6	Register
Fc15	Code	Bit	Seconds	Count	0	Node	15	Register
Fc16	Code	Register	Seconds	Count	Code	Node	16	Register

Node Address = Destination Address for Message

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. This error list can be transferred to the processor. The errors generated by the module are displayed in the following tables:

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

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

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

3 Module Configuration

In order for the MVI46-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 will require, depending on the operating modes to be supported.

Module Register Address	Functional Modes Affected	Name	Description
5000-5009	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 SLC processor.
5010-5039 and 5040-5069	Master and Slave	Port Configuration	These sections are used to define the characteristics of each of the Modbus serial communication ports on the module. These parameters must be set correctly for proper module operation.
5070-5869 and 5870-6669	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 the Setting Up the Module section for a description of the configuration of the module. The MVI46-MCM module must be configured at least once when the card is first powered, and any time thereafter when the parameters must be changed.

3.1 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.

3.2 Changing Parameters During Operation

A copy of the module's configuration data is mapped in the module's M1 file as displayed in the previous table. These values are initialized when the module first receives its configuration from the SLC 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 7800 (N46:0) 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 SLC processor. Ladder logic must exist in the processor to accept the block sent by the module. If everything is configured correctly, the module can receive its configuration from a remote device.

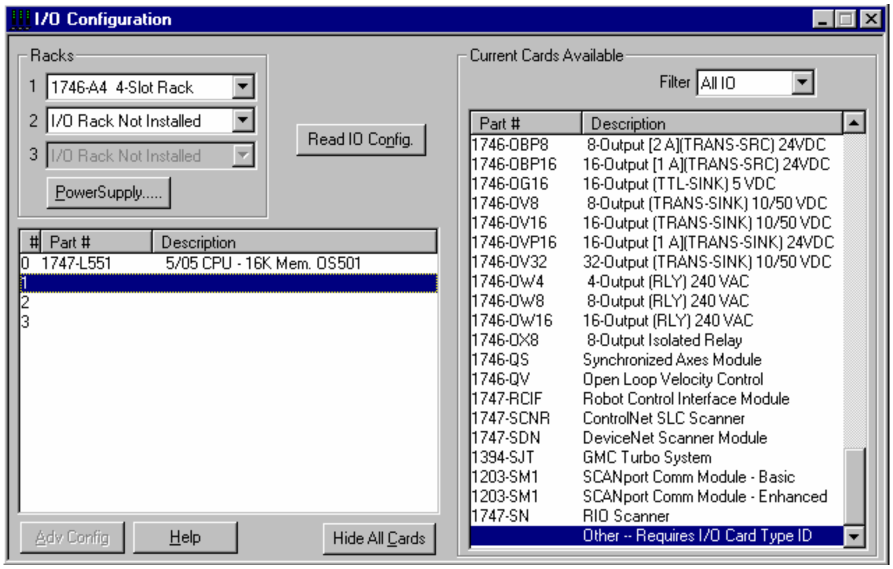
The ladder logic requirement section defines the minimum ladder logic required to apply the module in a user application. The logic is simple to understand and implement. Go to the **Ladder Logic** section for a complete discussion of the ladder logic requirements.

3.3 Setting Up the Module

Setup of the MVI46-MCM module only requires software configuration using the RSLogix 500 program. The easiest method to implement the module is to start with the example provided with the module (MVI46MCM.RSS). When installing this module in an existing application, simply copy the elements required from the example ladder logic to the application.

NOTE: The module can only be added to a project using the software in offline mode.

The first step in setting up the module is to define the module to the system. Select the IO Configuration option from the program screen. This displays the following dialog box:



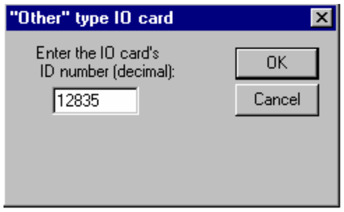
Racks:
1 1746-A4 4-Slot Rack
2 I/O Rack Not Installed
3 I/O Rack Not Installed

#	Part #	Description
0	1747-L551	5/05 CPU - 16K Mem. OS501
1		
2		
3		

Filter: All IO

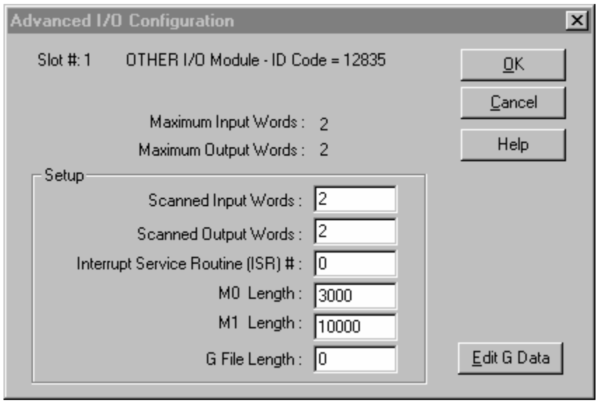
Part #	Description
1746-OBP8	8-Output [2 A](TRANS-SRC) 24VDC
1746-OBP16	16-Output [1 A](TRANS-SRC) 24VDC
1746-OG16	16-Output (TTL-SINK) 5 VDC
1746-OV8	8-Output (TRANS-SINK) 10/50 VDC
1746-OV16	16-Output (TRANS-SINK) 10/50 VDC
1746-OVP16	16-Output [1 A](TRANS-SINK) 24VDC
1746-OV32	32-Output (TRANS-SINK) 10/50 VDC
1746-OW4	4-Output (RLY) 240 VAC
1746-OW8	8-Output (RLY) 240 VAC
1746-OW16	16-Output (RLY) 240 VAC
1746-OX8	8-Output Isolated Relay
1746-QS	Synchronized Axes Module
1746-QV	Open Loop Velocity Control
1747-RCIF	Robot Control Interface Module
1747-SCNR	ControlNet SLC Scanner
1747-SDN	DeviceNet Scanner Module
1394-SJT	GMC Turbo System
1203-SM1	SCANport Comm Module - Basic
1203-SM1	SCANport Comm Module - Enhanced
1747-SN	RIO Scanner
	Other -- Requires I/O Card Type ID

Select the Other module from the list. This causes the program to display the following dialog box:



Enter the IO card's ID number (decimal): 12835

Enter the module IO card ID number as 12835, and then select the OK command button. Double-click the mouse on the module just added to the rack. Fill in the following dialog box presented as shown in the example:



Slot #: 1 OTHER I/O Module - ID Code = 12835

Maximum Input Words : 2
Maximum Output Words : 2

Scanned Input Words : 2
Scanned Output Words : 2
Interrupt Service Routine (ISR) # : 0
M0 Length : 3000
M1 Length : 10000
G File Length : 0

Select the OK command button to apply these settings to the module. Then, close the IO Configuration dialog box.

The next step in the module's setup is to define the user defined data areas to hold the configuration, port command lists, status and read and write database areas.

At this point, take the time to fill in the configuration values in the MCM configuration data table. Refer to the Module Data section of this document for information on configuring the module.

The last step in the module setup is to add the ladder logic. If the example ladder logic is used, adjust the ladder to fit the application. When the ladder example is not used, copy the example ladder logic to your application and alter as necessary.

The module is now set up and ready to be used with your application. Insert the module in the rack and attach the Modbus serial communication cables. Download the new application to the controller and place the processor in run mode. If all the configuration parameters are set correctly and the module is attached to a Modbus network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink very rapidly. Refer to the **Diagnostics and Troubleshooting** section of this manual if you encounter errors. Attach a computer or terminal to Debug/Configuration port on the module and check the status of the module using the resident debugger in the module.

3.4 Module Data

All data related to the MVI46-MCM module is stored in user defined data areas. The user is responsible for setting up the data areas to match the specific application for which the module is used. Each data area is discussed in the following sections.

3.4.1 Configuration Data

Configuration of the module is performed by filling in a user defined data table. In the example ladder logic, file N10 is used to store the general module configuration information. N11 is used to store the command list for port 1. N12 is used to store the command list for port 2. Each register in the files has an associated symbol and description to aid in filling in the data. Appendix E lists the items that must be configured for the module and their associated location in the M0 file.

3.4.1.1. Backplane Parameters

In this revision of the module, all data to be transferred between the SLC processor and the module is held in the module's M1 file. This simplifies the ladder logic. In order to read data in the module, copy the specific data area in the M1 file into a user defined file. Repeat this operation for each data area. Remember, the maximum data area that can be copied with the COP instruction is 128 words. In order to write data to the module, copy the data in the user defined file to the specific data area in the M1 file. The read and write data operations should be limited to the M1 words 0 to 4999. The modules status data can be read from the M1 file starting at register 7200.

Only one parameter is used in this section of the configuration.

The Error Status Pointer parameter is used to define the location in the module's database where the error/status data is 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.

3.4.1.2. Port Parameters

These parameters are used to define the operation of each of the Modbus ports on the module. Refer to the appendix for a definition of each parameter.

3.4.1.3. Master Commands

These records are used to define the commands in the master command list. Each parameter is described in the following table:

Parameter	**Description**
Enable	This parameter is used to define if the command is executed or disregarded. The following values are valid: 0=Disables the command and it will not execute. 1=The command will be considered for execution each scan of the command list and will be controlled by the PollInt parameter. And 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.
IntAddress	This parameter specifies the starting internal register address to be associated with the command. Valid entry for this parameter is 0 to 4999 for Register Address (0 to 65535 for Bit Address).
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 entry for this parameter is 0 to 65535.
Count	This parameter defines the number of registers to be considered by the command. Valid entry for this parameter is dependent on the Modbus specification for the command.

Parameter	Description
Swap	This parameter is used to specify if the data used in the command must be altered when a Modbus function code 3 is used to read data from a node on the network. Values that can be assigned are as follows: 0=no swapping of data, 1=swap word values, 2=swap word and byte values and 3=swap byte values. This option is used when interfacing the module with ASCII and floating-point data on other devices.
Node	This parameter is used to assign the Modbus slave node address for the module to reach with the command on the Modbus network. This parameter can be assigned values from 0 to 255. Most Modbus networks limit the upper value to 247.
Func	This parameter specifies the Modbus function to be performed by the command. Valid entries are 1, 2, 3, 4, 5, 6, 15 and 16.
DevAddress	This parameter defines the starting address in the device being considered by the command. Values entered in this field are dependent on the node's database definition. Refer to the specific manufacture's database definition for the device to determine the location of the data to be interfaced.

3.4.1.4. Status Data

This data area is used to view the status of the module. This data can be used to monitor the state of the module at a "real-time rate". Refer to the Appendix for a complete listing of the data stored in this object. This data can be read from the module’s M1 file starting at register 7600. User data file N30 is defined in the example ladder logic for this purpose.

3.4.1.5. User Data

All user data is stored in the module’s M1 file in registers 0 to 4999. This 5000-word area is directly accessible from the ladder logic. The COP instruction should be used to move blocks of data between the user data files and the module’s M1 file. This limits the number of accesses to the M1 data area and provides faster system response. In the example ladder logic, N31 is used to hold data read from the module, and N32 is used to store data to write to the module.

3.4.1.6. Slave Polling Control and Status

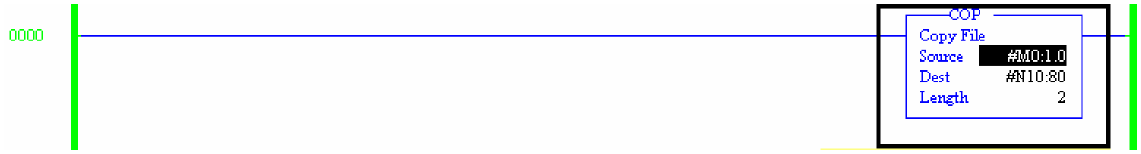
Two data areas can be allocated in the SLC to hold the polling status of each slave on the master ports. This status data can be used to determine which slaves are currently active on the port, are in communications error, or have their polling suspended and disabled. If the configuration supplies an address where this data resides, copy the data from the M1 file to the reserved files in the SLC. Using special blocks, the processor can enable or disable the polling of selected slaves.

4 Ladder Logic

Ladder logic is required for application of the MVI46-MCM module. Tasks that must be handled by the ladder logic are module configuration, data transfer, special block handling and status data transfer. This section discusses each aspect of the ladder logic as required by the module. Additionally, a power-up handler should be written to handle the initialization of the module's data and to clear any processor fault conditions.

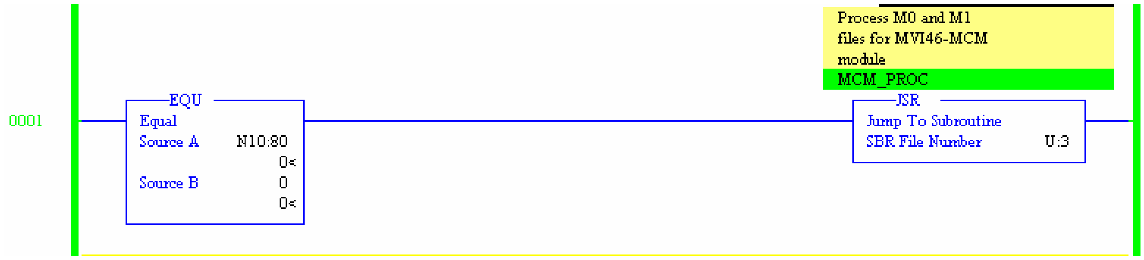
4.1 MCM Main

The MCM Main routine is used to recognize the presence of configuration requests, special command request and response messages and data transfer between the module and the processor. The following rung is used to save the current control word found in the M0 file:



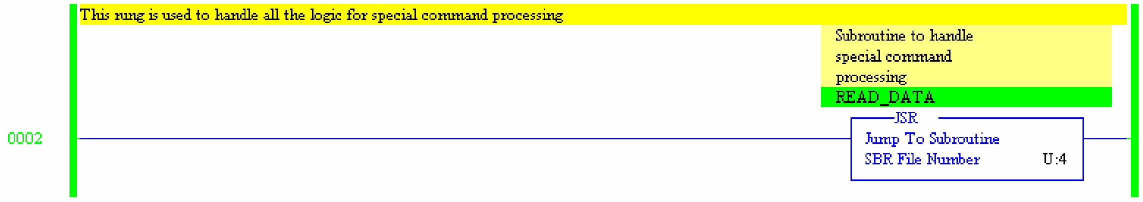
```
0000                                  COP
                                      Copy File
                                      Source      #M0:1.0
                                      Dest        #N10:80
                                      Length            2
```

Word 0 of the M0 file is used by the module to indicate the response to a special command instruction or to request the module's configuration. During normal program execution, this register should have a value of zero. If any other value is present, the data transfer function will not be executed. The following rung is used to execute the data transfer function (MCM Data, U:3) when the control word is set to zero.



```
                                      Process M0 and M1
                                      files for MVI46-MCM
                                      module
                                      MCM_PROC
0001  EQU                             JSR
      Equal                           Jump To Subroutine
      Source A    N10:80              SBR File Number     U:3
                      0<
      Source B         0
                      0<
```

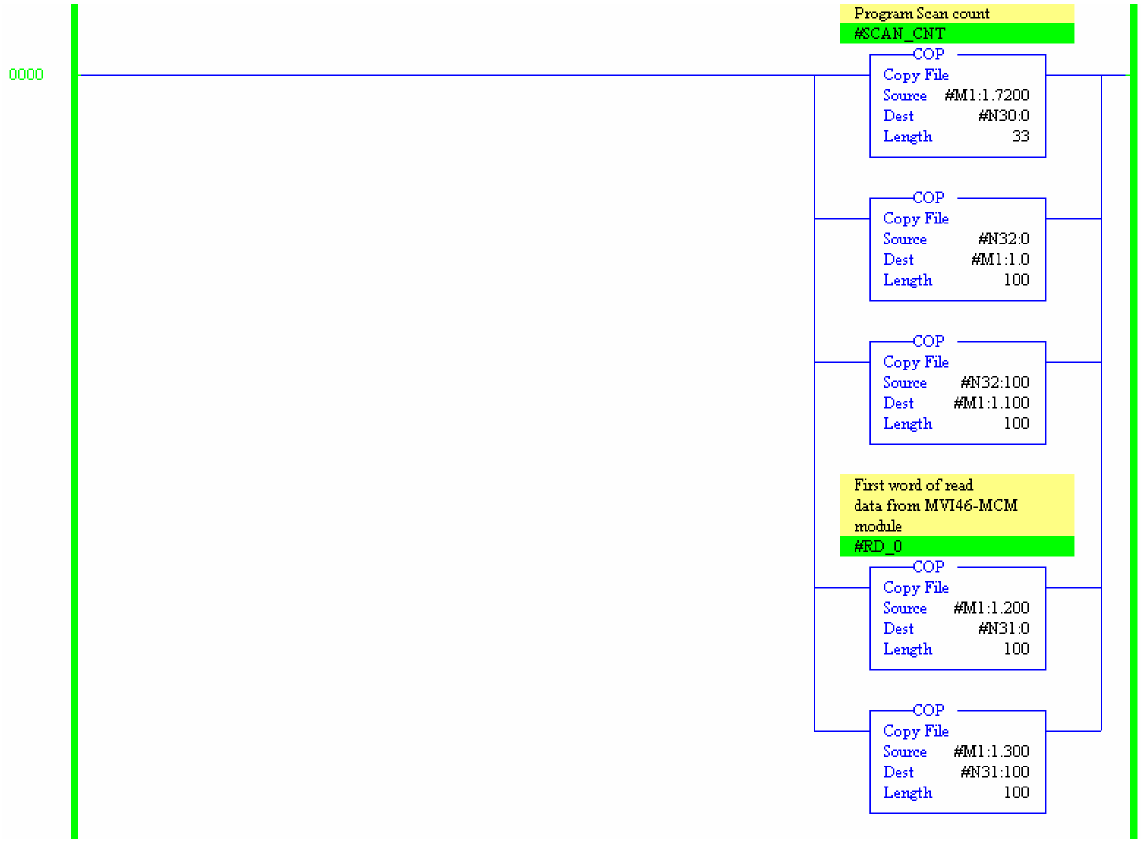
The following rung is called each scan to process any special command request and response messages.



```
This rung is used to handle all the logic for special command processing
                                      Subroutine to handle
                                      special command
                                      processing
                                      READ_DATA
0002                                  JSR
                                      Jump To Subroutine
                                      SBR File Number     U:4
```

4.2 MCM Data

The MCM Data task is responsible for handling the transfer of data between the processor user defined files and the module's M1 file. In this simple example, the status data is copied from M1:1.7200 to N30:0; 200-words of read data are copied from the M1 file to N31; and 200-words of data are written from N32 to the M1 file. The logic to perform these tasks is shown in the following rung:

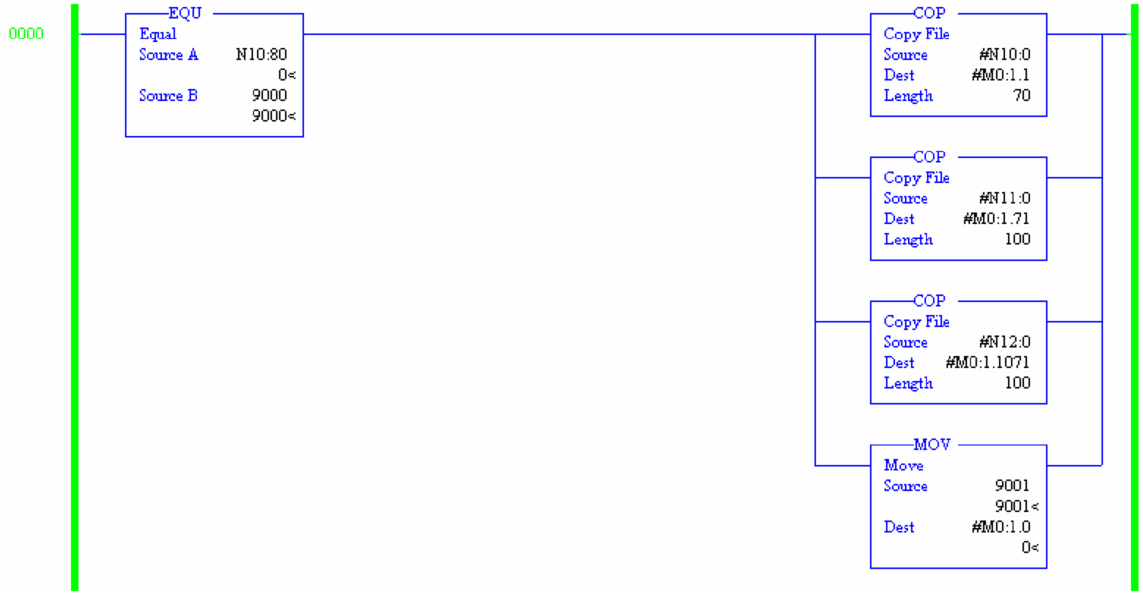


```
0000
Program Scan count
#SCAN_CNT
COP
Copy File
Source   #M1:1.7200
Dest        #N30:0
Length          33

COP
Copy File
Source      #N32:0
Dest        #M1:1.0
Length         100

COP
Copy File
Source    #N32:100
Dest     #M1:1.100
Length         100

First word of read
data from MVI46-MCM
module
#RD_0
COP
Copy File
Source   #M1:1.200
Dest        #N31:0
Length         100

COP
Copy File
Source   #M1:1.300
Dest      #N31:100
Length         100
```

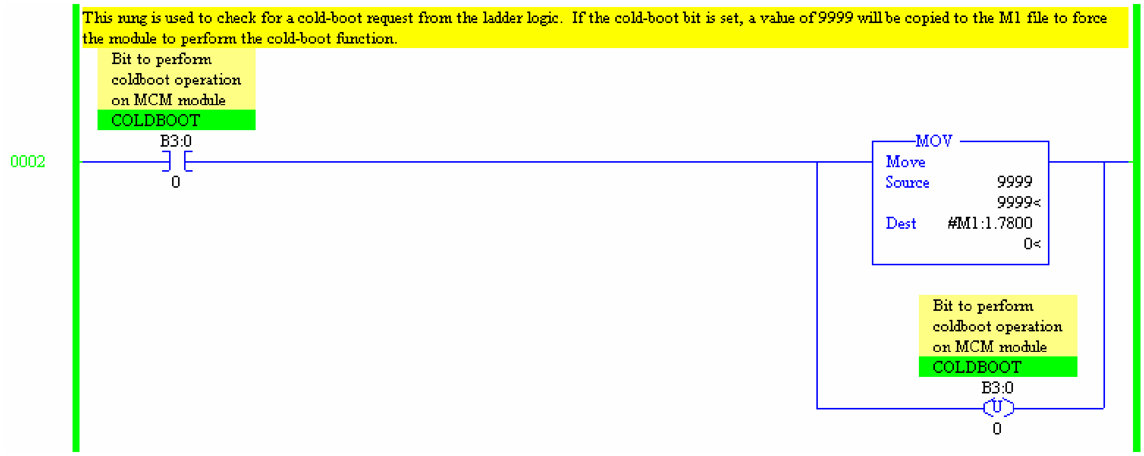
4.3 MCM Cmds

The MCM CMDS subroutine is used to handle special block processing. This includes module configuration requests from the module and processor induced commands. Each time the module perform a restart or warm-boot operation, the module will request its configuration from the SLC processor. This request is made by placing a value of 9000 in the M0:1.0 register.

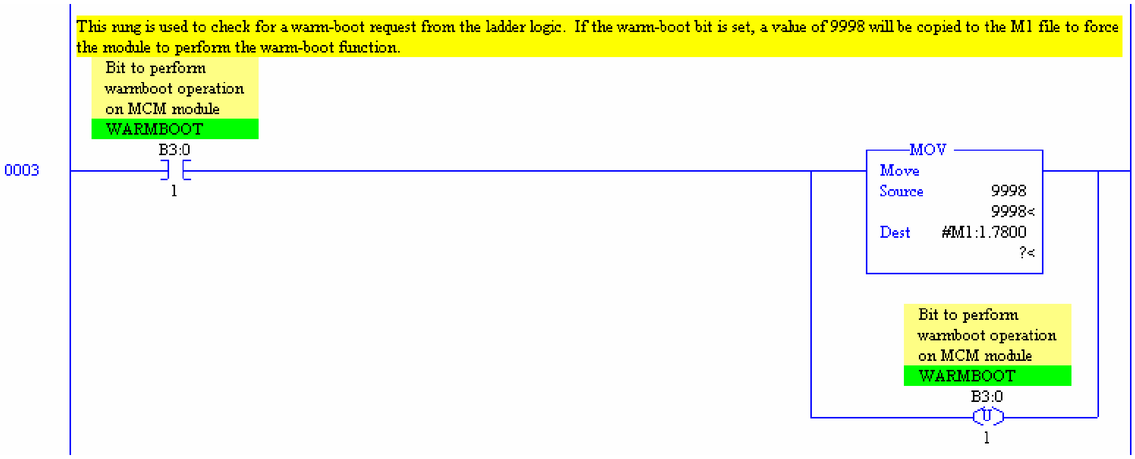
When the ladder logic recognizes the value of 9000 in the control register, it copies the configuration information from the N10 file and the master port commands from the N11 and N12 file. After completing this task, the processor places a value of 9001 into the control register. This signals to the module that the configuration is ready to process. The module will continually wait for the value of 9001 in the control register. After the module is successfully configured, it will place a value of 0 in the control register so that normal data transfer can occur. Example ladder logic to configure the module is shown in the following rung.



The SLC processor can request a cold-boot operation of the module by placing a value of 9999 in the M1 register 7800 (command control register). Ladder logic to perform this task is shown in the following example:


This rung is used to check for a cold-boot request from the ladder logic. If the cold-boot bit is set, a value of 9999 will be copied to the M1 file to force the module to perform the cold-boot function.

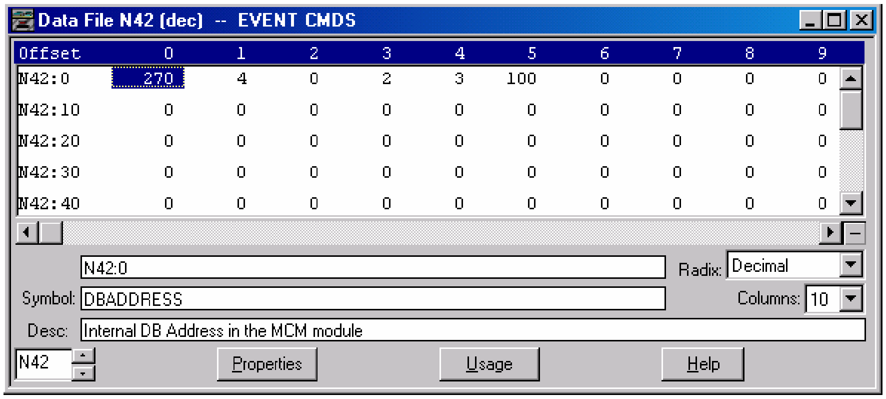
The SLC processor can request a warm boot operation of the module by placing a value of 9998 in the M1 register 7800 (command control register). Ladder logic to perform this task is shown in the following rung.



This rung is used to check for a warm-boot request from the ladder logic. If the warm-boot bit is set, a value of 9998 will be copied to the M1 file to force the module to perform the warm-boot function.

User defined commands can be constructed and sent to a Modbus master port on the module under processor control. In order to perform this task, the command must be constructed or stored in a user-defined file. In the example ladder logic, file N42 is used to store a single command. Under program control, this command is sent to MODBUS master port 1 on the module.

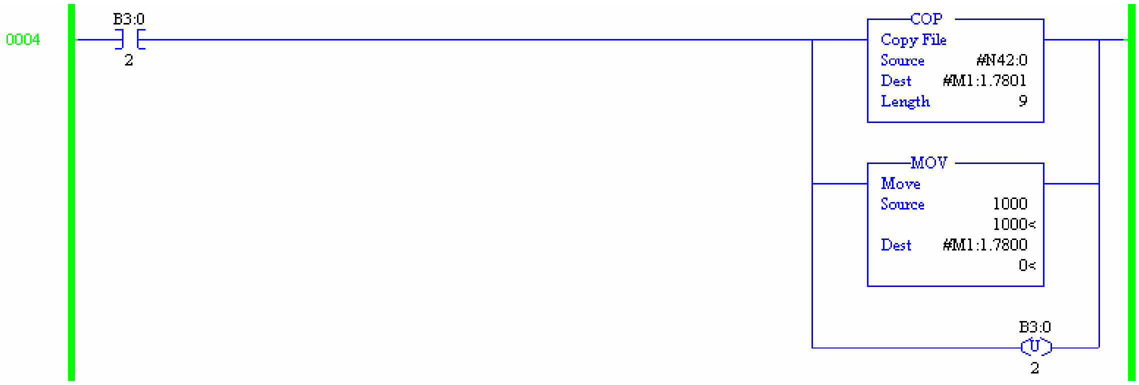
The contents of N42 is shown in the following example:



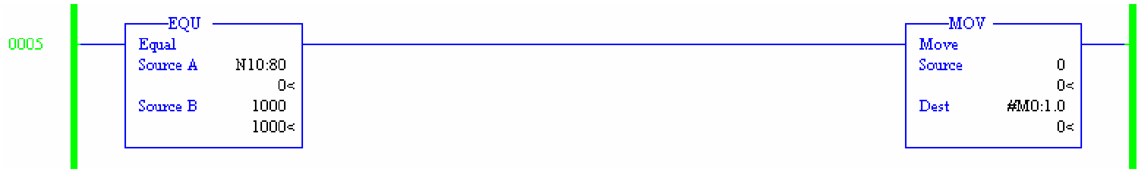
Data File N42 (dec) -- EVENT CMDS

Offset	0	1	2	3	4	5	6	7	8	9
N42:0	270	4	0	2	3	100	0	0	0	0
N42:10	0	0	0	0	0	0	0	0	0	0
N42:20	0	0	0	0	0	0	0	0	0	0
N42:30	0	0	0	0	0	0	0	0	0	0
N42:40	0	0	0	0	0	0	0	0	0	0

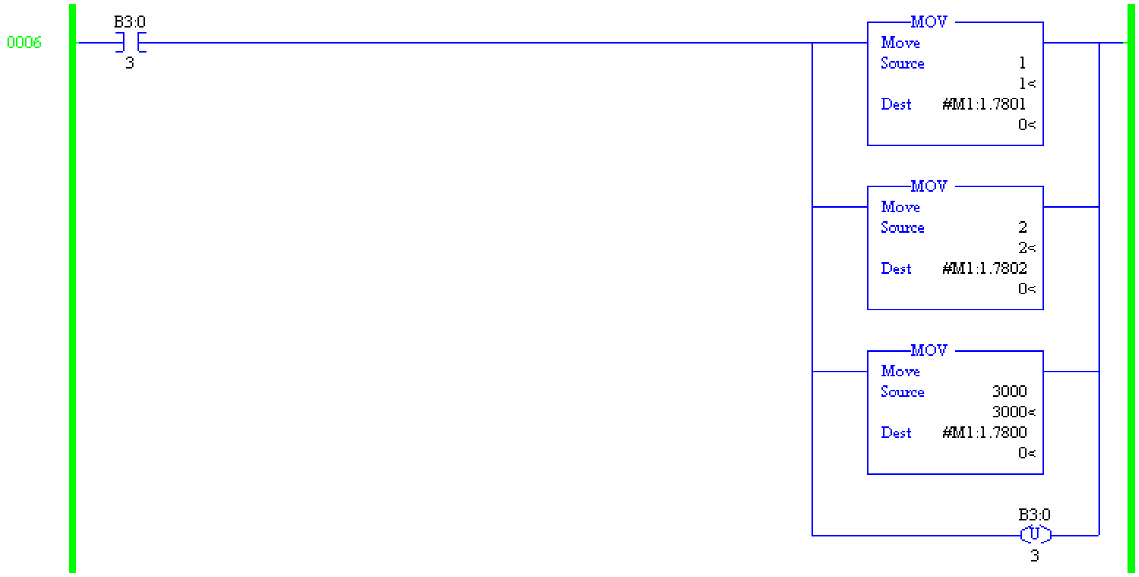
N42:0
Radix: Decimal
Symbol: DBADDRESS
Columns: 10
Desc: Internal DB Address in the MCM module
N42



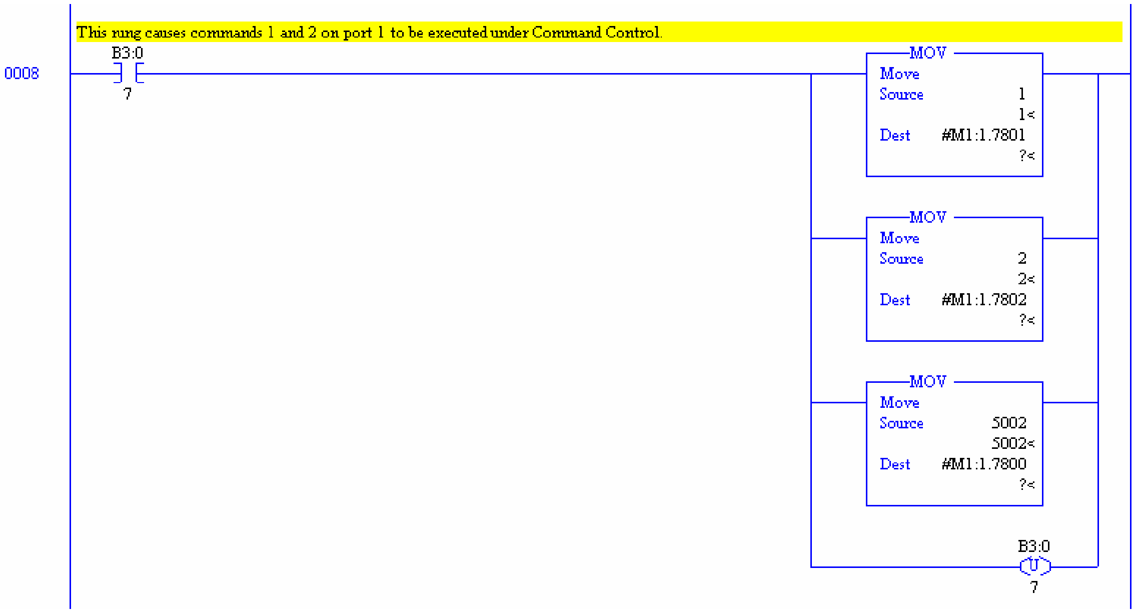
After the module processes the command event request, it will set the value in M1:1.7800 to zero and place the command event request code in M0:1.0. The ladder logic can use the information returned from the module in M0:1.1 to perform more processing or simply ignore the data (as shown in the following rung).



The following rung is used to control the slave poll list for the MODBUS master port 1. The processor can enable or disable a slave in the poll list by placing the slave node address into logic like that shown. The logic shown is used to disable polling for slave unit 2. Similar logic is given in the example ladder logic to enable the same slave unit.



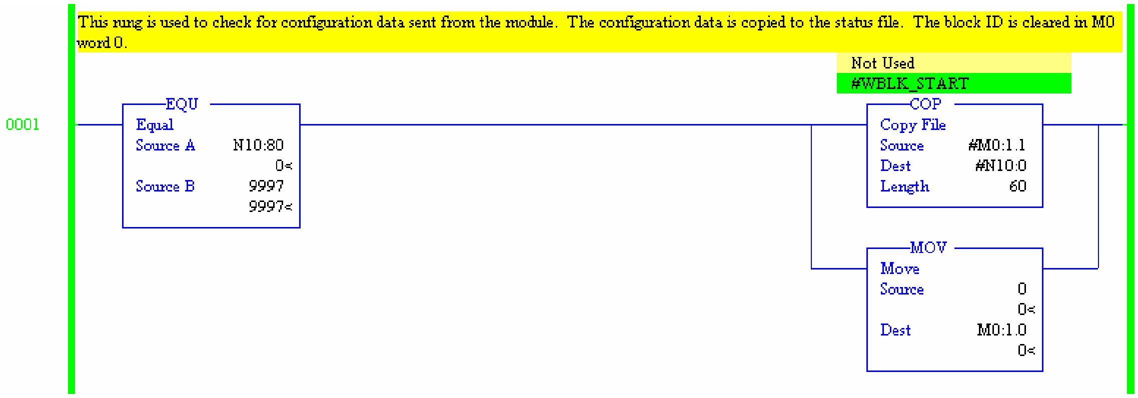
During normal program execution in the module, the command list is processed from the first enabled command to the last. The processor can place commands in the command list (enabled or disabled commands) in the command queue. The queue has a higher priority and all commands in the queue will execute before the normal command list is processed. Ladder logic to place commands 1 and 2 into the command queue is shown in the following example:



This rung causes commands 1 and 2 on port 1 to be executed under Command Control.

```
0008  B3:0 ] [ 7

MOV  Move
     Source          1
                     1<
     Dest   #M1:1.7801
                     ?<

MOV  Move
     Source          2
                     2<
     Dest   #M1:1.7802
                     ?<

MOV  Move
     Source       5002
                  5002<
     Dest   #M1:1.7800
                     ?<

B3:0 (U) 7
```

The configuration information can be transferred from the module to the processor with block 9997. This rung copies the configuration information from M0 to the N:10 configuration file.



This rung is used to check for configuration data sent from the module. The configuration data is copied to the status file. The block ID is cleared in M0 word 0.

```
0001

EQU  Equal
     Source A    N10:80
                     0<
     Source B      9997
                  9997<

Not Used
#WBLK_START
COP  Copy File
     Source   #M0:1.1
     Dest     #N10:0
     Length       60

MOV  Move
     Source          0
                     0<
     Dest       M0:1.0
                     0<
```

5 Diagnostics and Troubleshooting

The module provides diagnostic information in three forms to the user. 1) Status Data values are transferred from the module to the controller tags in the SLC processor. 2) All data contained in the module can be viewed through the configuration/debug port to an attached terminal emulator. 3) LED status indicators on the front of the module yield information on the modules status.

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

5.1 Reading Status Data From the Module

The MVI46-MCM module returns a 33-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 7200 to 7232 and at the location specified in the configuration. This data is transferred to the SLC processor continuously with each read block. For a complete listing of the status data object, refer to the **Module Set Up** section.

5.1.1 Required Hardware

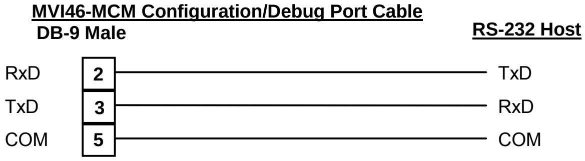
The hardware requirements to interface with the configuration/debugger port are not too stringent. A personal computer with a standard serial port should suffice. For optimal performance, the minimum is required:

80468 based processor (Pentium preferred)

1 megabyte of memory

At least one serial communications port available

Additionally, a null-modem cable is required between your PC and the port. The module's port has a DB-9 male connector at the end of a RJ-45 to DB-9 pigtail. The RJ-45 end of the cable is to be placed in the MVI46-MCM port 1 connector (top port). The cable required is shown in the following diagram:



5.1.2 Required Software

The software required on your personal computer to interface with the configuration/debugger port is operating system dependent. Tested software includes the following:

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

Any ASCII terminal emulation software package provided with your operating system should work as long as it can be configured as follows:

Baud Rate	57,600
Parity	None
Data Bits	8
Stop Bits	1
File Transfer Protocol	Zmodem

5.1.3 Using the Port

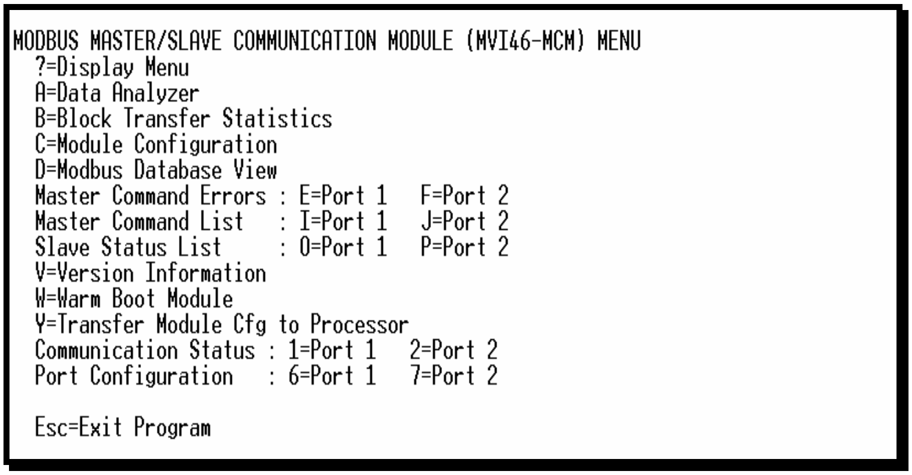
The following steps are required to interface with the configuration/debugger port:

1. Connect your computer to the module's port using a null-modem cable.
2. Start the terminal emulation program on your computer and configure the communication parameters to those shown in the Required Software section (57,600K, N, 8, 1).
3. Enter the '?' character on your computer. If everything is set up correctly, the port's menu is displayed.

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

5.1.4 Menu Options

Features available through the use of the configuration/debug port on the MVI46-MCM module are all reached using single keystrokes on your computer. There is a single main menu and several sub-menus presented on the port. To view the current selections available, press the '?' key on your computer. If you are in main menu mode, the following menu is displayed:

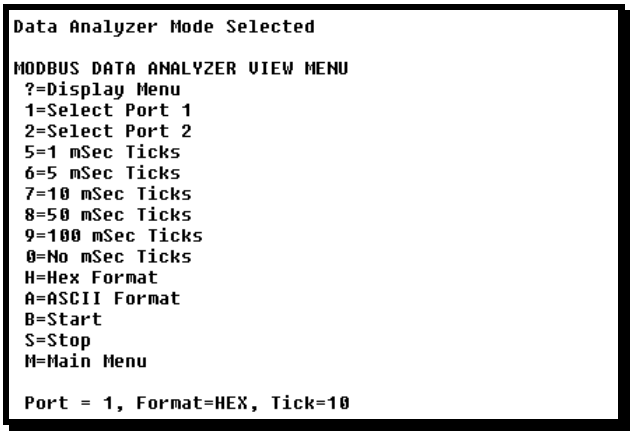


```
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MVI46-MCM) MENU
 ?=Display Menu
 A=Data Analyzer
 B=Block Transfer Statistics
 C=Module Configuration
 D=Modbus Database View
 Master Command Errors : E=Port 1   F=Port 2
 Master Command List   : I=Port 1   J=Port 2
 Slave Status List     : O=Port 1   P=Port 2
 V=Version Information
 W=Warm Boot Module
 Y=Transfer Module Cfg to Processor
 Communication Status : 1=Port 1   2=Port 2
 Port Configuration   : 6=Port 1   7=Port 2

 Esc=Exit Program
```

If this menu is not displayed, press the 'M' key to display the main menu. All facilities offered by the configuration/debugger are shown on the main menu. Each option is discussed in the following sections:

5.1.4.1. A=Data Analyzer

Selection of this menu option places the program in analyzer menu mode. This mode of operation is used to display Modbus messages generated and received by the module. To view the menu options available in this mode, press the '?' key and the following menu is displayed:



```
Data Analyzer Mode Selected

MODBUS 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
```

This tool is extremely useful in determining the operation of the module and nodes on the network of each port. The parameters shown at the bottom of the display show the current analyzer settings. Each of the menu options is discussed in the following sections:

5.1.4.1.1. 1=Select Port 1

This option is used to select Modbus Port 1 for analysis. Data displayed when in analyzer mode will relate to this port.

5.1.4.1.2. 2=Select Port 2

This option is used to select the Modbus Port 2 for analysis. Data displayed when in analyzer mode will relate to this port.

5.1.4.1.3. 5=1 mSec Ticks

This option is used to generate 1-millisecond timing marks on the display. This may help when determining communication-timing characteristics.

5.1.4.1.4. 6=5 mSec Ticks

This option is used to generate 5-millisecond timing marks on the display. This may help when determining communication-timing characteristics.

5.1.4.1.5. 7=10 mSec Ticks

This option is used to generate 10-millisecond timing marks on the display. This may help when determining communication-timing characteristics.

5.1.4.1.6. 8=50 mSec Ticks

This option is used to generate 50-millisecond timing marks on the display. This may help when determining communication-timing characteristics.

5.1.4.1.7. 9=100 mSec Ticks

This option is used to generate 100-millisecond timing marks on the display. This may help when determining communication-timing characteristics.

5.1.4.1.8. 0=No mSec Ticks

This option is used to turn the display of timing marks off.

5.1.4.1.9. H=Hex Format

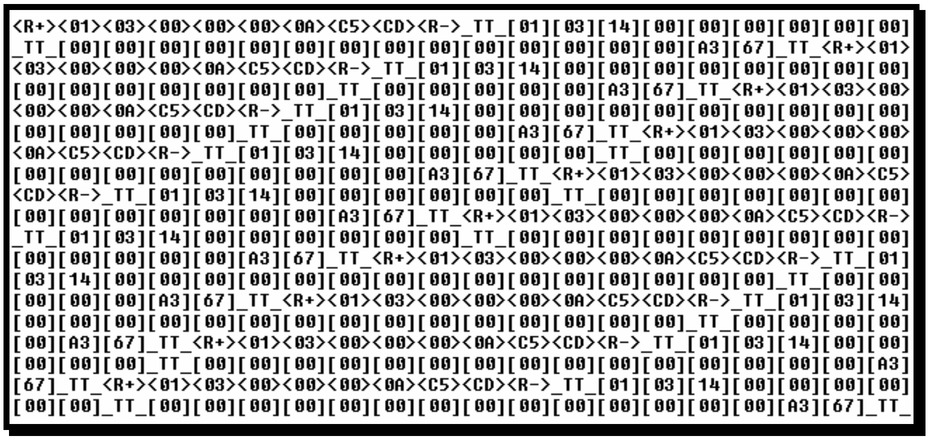
This option is used to select the display of the data in hexadecimal format. This format is most useful when viewing Modbus RTU protocol messages.

5.1.4.1.10. A=ASCII Format

This option is used to select the display of the data in ASCII format. This format is most useful when viewing Modbus ASCII protocol messages.

5.1.4.1.11. B=Start

This option is used to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port is displayed. An example display is shown:



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

The following table describes the special characters used in the display:

[ ]	**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.

5.1.4.1.12. S=Stop

This option is used to stop the analyzer. Use this option to freeze the display so the data can be analyzed. To restart the analyzer, press the 'B' key.

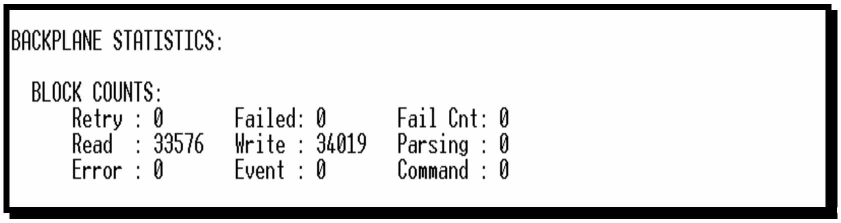
Warning -- When in analyzer mode, program execution will slow down. Only use this tool during a trouble-shooting session. Disable the analyzer before leaving the module to run in its normal mode.

5.1.4.1.13. M = Main Menu

This menu option is used to return to the main menu mode.

5.1.4.2. B=Block Transfer Statistics

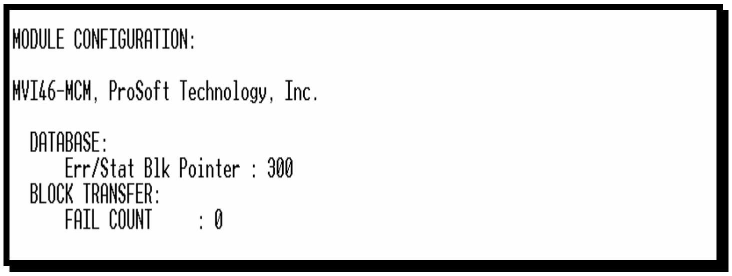
This menu option is used to display the configuration and statistics of the backplane data transfer operations. After selecting this option, the following is displayed. Selecting this option at one-second intervals can be used to determine the number of blocks transferred each second.



```
BACKPLANE STATISTICS:

  BLOCK COUNTS:
      Retry : 0      Failed: 0      Fail Cnt: 0
      Read  : 33576  Write : 34019  Parsing : 0
      Error : 0      Event : 0      Command : 0
```

5.1.4.3. C=Module Configuration

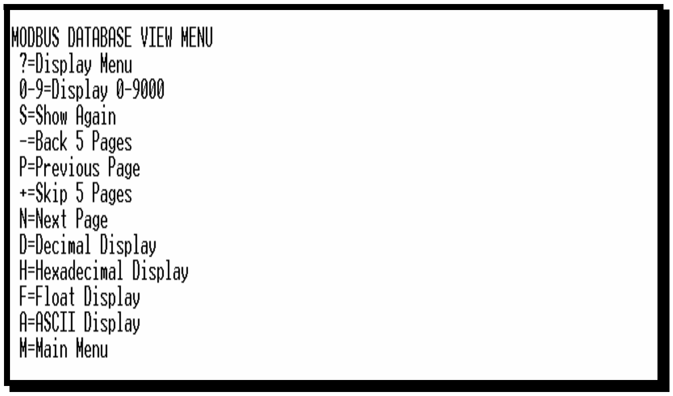
This option displays the general module configuration information for the MVI46-MCM module. After selecting the option, the following screen is displayed.



```
MODULE CONFIGURATION:

MVI46-MCM, ProSoft Technology, Inc.

  DATABASE:
     Err/Stat Blk Pointer : 300
  BLOCK TRANSFER:
     FAIL COUNT     : 0
```

5.1.4.4. D=Modbus Database View

Selection of this menu option places the program in database view menu mode. This mode of operation is used to display the module's internal database values. To view the menu options available in this mode, press the '?' key and the following menu is displayed.



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

All data contained in the module's database is available for viewing using the menu options. Each option available on the menu is discussed in the following sections.

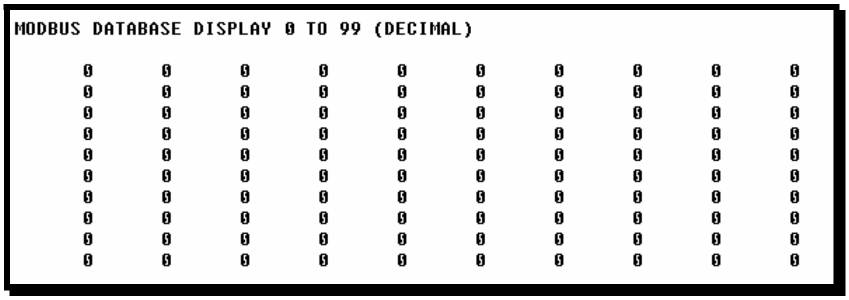
5.1.4.5. 0-9 Register Pages 0-9000

This menu option is used to jump to a specific set of registers in the database and display the data. The keys perform the following functions.

Key	FUNCTION
0	Display registers 0 to 99
1	Display registers 1000 to 1099
2	Display registers 2000 to 2099
3	Display registers 3000 to 3099
4	Display registers 4000 to 4099
5	Display registers 5000 to 5099
6	Display registers 6000 to 6099
7	Display registers 7000 to 7099
8	Display registers 8000 to 8099
9	Display registers 9000 to 9099

5.1.4.6. S=Show Again

This menu option is used to display the current page of 100 registers in the database. Example output of the database display is shown:



```
MODBUS DATABASE DISPLAY 0 TO 99 (DECIMAL)

       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
       0      0      0      0      0      0      0      0      0      0
```

5.1.4.6.1. - = Back 5 Pages

This menu option is used to skip the previous 500 registers of data for viewing and display the data.

5.1.4.6.2. P = Previous Page

This menu option is used to select and display the previous 100 registers of data.

5.1.4.6.3. + = Skip 5 Pages

This menu option is used to skip 500 registers of data and to display the new page of data.

5.1.4.6.4. N = Next Page

This menu option is used to select the next 100 registers of data for viewing and displays the data.

5.1.4.6.5. D = Decimal Display

This menu option is used to display the data on the current page in decimal format.

5.1.4.6.6. H = Hexadecimal Display

This menu option is used to display the data on the current page in hexadecimal format.

5.1.4.6.7. F = Float Display

This menu option is used 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 will not be displayed properly.

5.1.4.6.8. A = ASCII Display

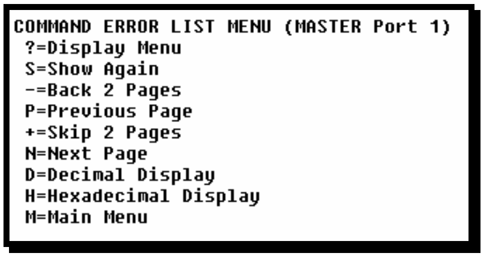
This menu option is used to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

5.1.4.6.9. M = Main Menu

This menu option is used to return to the main menu mode.

5.1.4.7. <u>E and F=Master Command Errors (Ports 1 and 2)</u>

Selection of these menu options places the program in master command error menu mode for the specified port. This mode of operation is used to display multiple pages of master command list error/status data. To view the menu options available in this mode, press the '?' key and the following menu is displayed.

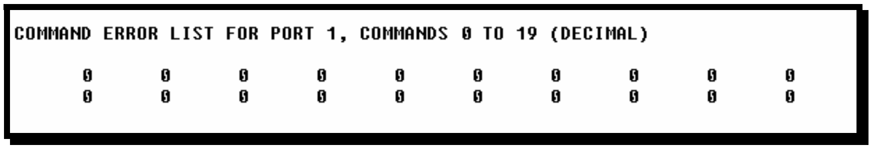


```
COMMAND ERROR LIST MENU (MASTER Port 1)
 ?=Display Menu
 S=Show Again
 -=Back 2 Pages
 P=Previous Page
 +=Skip 2 Pages
 N=Next Page
 D=Decimal Display
 H=Hexadecimal Display
 M=Main Menu
```

Each menu option is discussed in the following sections:

5.1.4.7.1. S = Show Again

This option is used to display the current page of master command error/status data. After selecting the option, the following screen is displayed.



```
COMMAND ERROR LIST FOR PORT 1, COMMANDS 0 TO 19 (DECIMAL)

      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
```

Each value displayed on the screen corresponds to the error/status code for the associated master command list index. Refer to the **Module Set Up** section for complete listing and interpretation of the codes displayed.

5.1.4.7.2. - = Back 2 Pages

This option is used to skip back 20 commands and display the data.

5.1.4.7.3. P = Previous Page

This option is used to display the previous page of data.

5.1.4.7.4. + = Skip 2 Pages

This option is used to skip past the next 20 commands and display the data.

5.1.4.7.5. N = Next Page

This option is used to display the next page of master command list error/status data.

5.1.4.7.6. D = Decimal Display

This option is used to change the display of the data to decimal format.

5.1.4.7.7. H = Hexadecimal Display

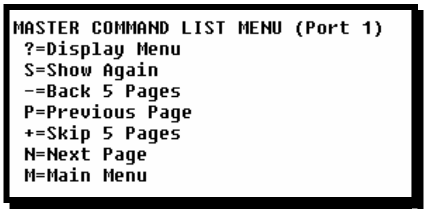
This option is used to change the display of error/status data to hexadecimal format.

5.1.4.7.8. M = Main Menu

This option is used to return the program to main menu mode.

5.1.4.8. I and J=Master Command List (Ports 1 and 2)

Selection of these menu options places the program in master command list menu mode for the specified port. This mode of operation is used to display multiple pages of master command list data. To view the menu options available in this mode, press the '?' key and the following menu is displayed.

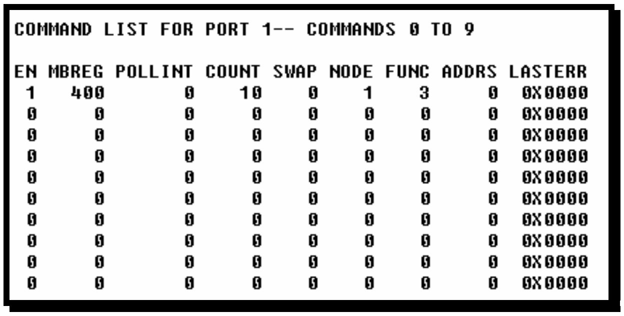


```
MASTER COMMAND LIST MENU (Port 1)
 ?=Display Menu
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 M=Main Menu
```

Each option on the menu is discussed in the following sections:

5.1.4.8.1. S = Show Again

This option is used to display the current page of master commands. Ten commands are displayed on each page as shown in the following example:



```
COMMAND LIST FOR PORT 1-- COMMANDS 0 TO 9

EN MBREG POLLINT COUNT SWAP NODE FUNC ADDRS LASTERR
 1   400       0    10    0    1    3     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
 0     0       0     0    0    0    0     0  0X0000
```

5.1.4.8.2. - = Back 5 Pages

This menu option is used to display the master command list data after skipping the previous 50 commands.

5.1.4.8.3. P = Previous Page

This menu option is used to display the previous page of master command list data.

5.1.4.8.4. + = Skip 5 Pages

This menu option is used to display the master command list data after skipping the next 50 commands.

5.1.4.8.5. N = Next Page

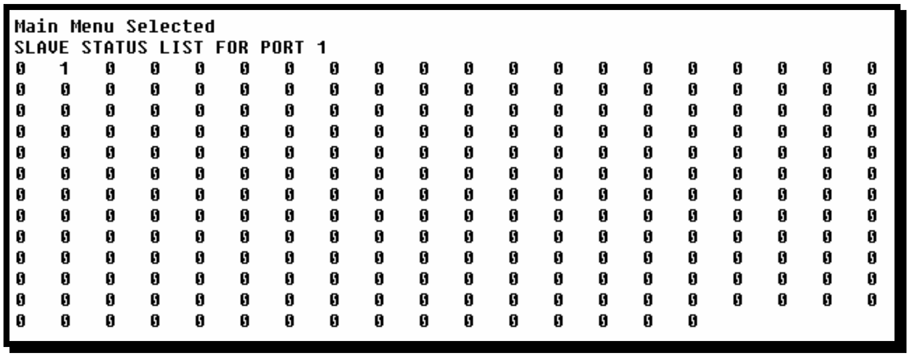
This menu option is used to display the next page of master command list data.

5.1.4.8.6. M = Main Menu

This option is used to return to the main menu mode of operation.

5.1.4.9. O and P=Slave Status List (Port 1 and 2)

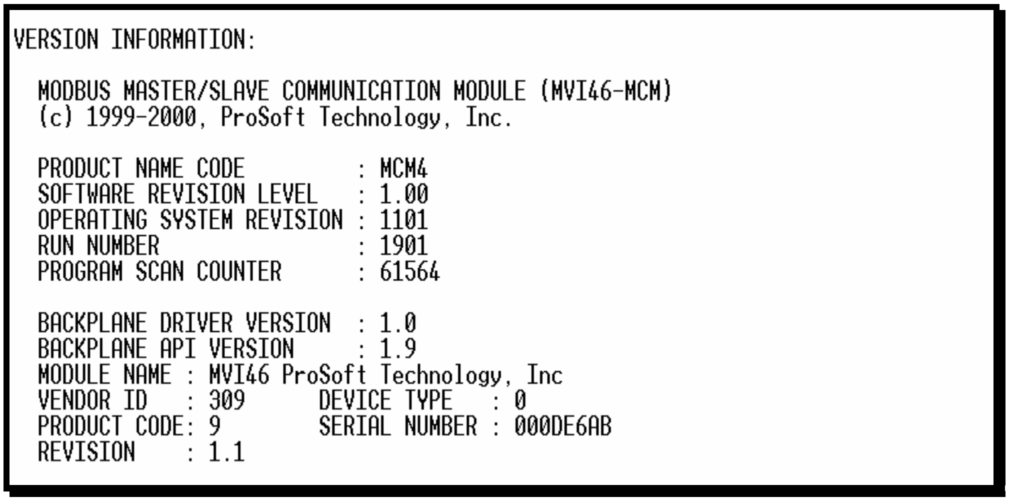
Selection of these menu options displays the 256 slave status values associated with the ports. Values shown have the following definitions: 0 = slave is not used, 1 = slave being actively polled, 2 = slave suspended and 3 = slave disabled.



```
Main Menu Selected
SLAVE STATUS LIST FOR PORT 1
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

5.1.4.10. V=Version Information

This option is used to view the current version of the software for the module and other important values. After selecting the option, the following is displayed.

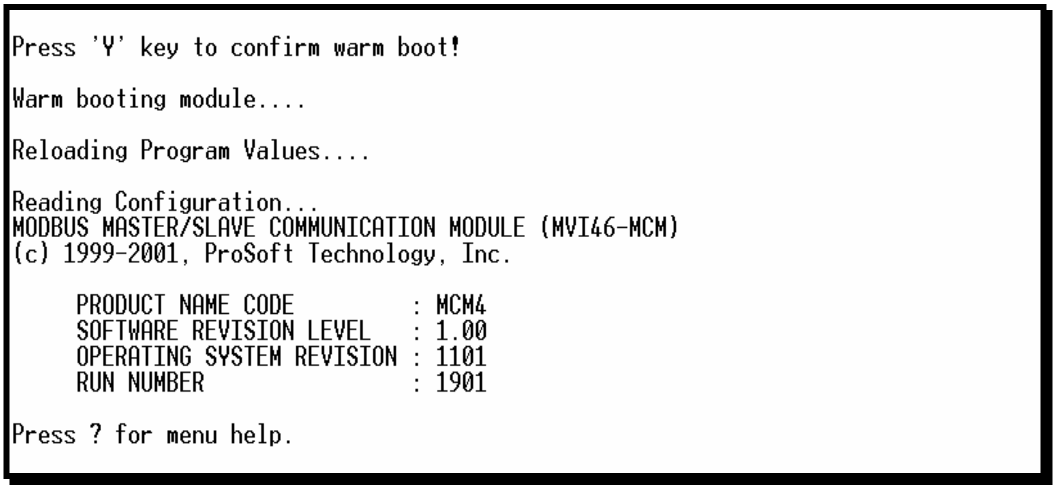


```
VERSION INFORMATION:

  MODBUS MASTER/SLAVE COMMUNICATION MODULE (MVI46-MCM)
  (c) 1999-2000, ProSoft Technology, Inc.

  PRODUCT NAME CODE        : MCM4
  SOFTWARE REVISION LEVEL  : 1.00
  OPERATING SYSTEM REVISION : 1101
  RUN NUMBER               : 1901
  PROGRAM SCAN COUNTER     : 61564

  BACKPLANE DRIVER VERSION  : 1.0
  BACKPLANE API VERSION     : 1.9
  MODULE NAME : MVI46 ProSoft Technology, Inc
  VENDOR ID   : 309      DEVICE TYPE   : 0
  PRODUCT CODE: 9        SERIAL NUMBER : 000DE6AB
  REVISION    : 1.1
```

This information may be requested 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. This value can be used to determine the frequency of program execution by pressing the 'V' key at one-second intervals.

5.1.4.11. W=Warm Boot Module

This option is selected when a warm-boot operation is required of the module. This request is usually made after configuration changes are set in the SLC processor's Controller Tags data area to implement the changes. After selecting the option, the following is displayed.



```
Press 'Y' key to confirm warm boot!

Warm booting module....

Reloading Program Values....

Reading Configuration...
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MVI46-MCM)
(c) 1999-2001, ProSoft Technology, Inc.

     PRODUCT NAME CODE        : MCM4
     SOFTWARE REVISION LEVEL  : 1.00
     OPERATING SYSTEM REVISION : 1101
     RUN NUMBER               : 1901

Press ? for menu help.
```

5.1.4.12. Y=Transfer Module Cfg to Processor

This option is used to transfer the current configuration data in the module to the SLC processor. Ladder logic must exist in the processor to successfully implement this option. After selecting the option, the following is displayed for successful operation.

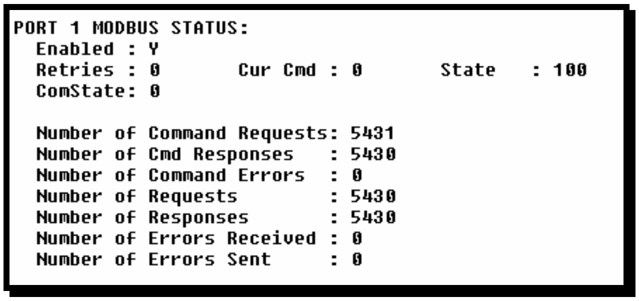
If the operation is not successful, an error code is returned. Error codes returned are as follows:

Code	Description
0	Transfer Successful
-1	Error transferring module configuration data (block –9000)

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

5.1.4.13. 1 and 2=Communication Status (Ports 1 and 2)

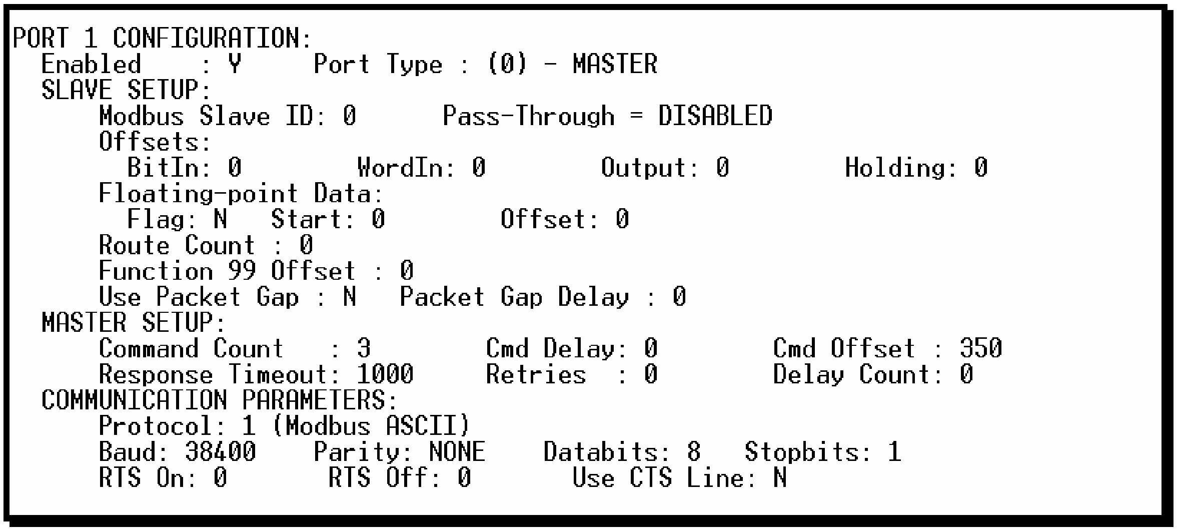
These options are used to display the communication status and statistics of the specified Modbus port. This information can be informative when trouble-shooting network problems. After selecting the option, the following information is displayed.



```
PORT 1 MODBUS STATUS:
  Enabled : Y
  Retries : 0       Cur Cmd : 0       State   : 100
  ComState: 0

  Number of Command Requests: 5431
  Number of Cmd Responses   : 5430
  Number of Command Errors  : 0
  Number of Requests        : 5430
  Number of Responses       : 5430
  Number of Errors Received : 0
  Number of Errors Sent     : 0
```

5.1.4.14. 6 and 7=Port Configuration (Ports 1 and 2)

These options are used to display the configuration information for the selected Modbus port. After selecting the option, the following information is displayed.



```
PORT 1 CONFIGURATION:
  Enabled    : Y     Port Type : (0) - MASTER
  SLAVE SETUP:
      Modbus Slave ID: 0      Pass-Through = DISABLED
      Offsets:
        BitIn: 0       WordIn: 0       Output: 0       Holding: 0
      Floating-point Data:
        Flag: N   Start: 0       Offset: 0
      Route Count : 0
      Function 99 Offset : 0
      Use Packet Gap : N   Packet Gap Delay : 0
  MASTER SETUP:
      Command Count   : 3       Cmd Delay: 0       Cmd Offset : 350
      Response Timeout: 1000    Retries  : 0       Delay Count: 0
  COMMUNICATION PARAMETERS:
      Protocol: 1 (Modbus ASCII)
      Baud: 38400   Parity: NONE    Databits: 8   Stopbits: 1
      RTS On: 0      RTS Off: 0      Use CTS Line: N
```

5.1.4.15. Esc=Exit Program

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

5.2 LED Status Indicators

The LED's will indicate the module’s operating status as follows:

ProSoft Module	Color	Status	Indication
P1	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.
P2	Green	On	Data is being transferred between the module and the Modbus network on its Modbus Port 1.
		Off	No data is being transferred on the port.
P3	Green	On	Data is being transferred between the module and the Modbus network on its Modbus Port 2.
		Off	No data is being transferred on the port.
APP	Amber	On	The MVI46-MCM is working normally.
		Off	The MVI46-MCM module program has recognized a communication error on one of its Modbus 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.

ProSoft Module	Color	Status	Indication
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or the battery is not present. Replace the battery on the module.

During module configuration, the ACT/FLT LED is red and the APP and BP ACT LED’s will be on. If the LED’s are latched in this mode for a long period of time, check the configuration error words in the configuration request block. The structure of the block is shown in the following table.

M0 Offset	Description	Length
0	9000	1
1	Spare	1
2	Port 1 Configuration Errors	1
3	Port 2 Configuration Errors	1

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	Protocol parameter is not valid.	0x0002
2	Termination type parameter is not valid.	0x0004
3	Baud rate parameter is not valid.	0x0008
4	Parity parameter is not valid.	0x0010
5	Data bits parameter is not valid.	0x0020
6	Stop bits parameter is not valid.	0x0040
7	Command count parameter is not valid.	0x0080
8	Retry count parameter is not valid.	0x0100
9	Spare	0x0200
10	Spare	0x0400
11	Spare	0x0800
12	Spare	0x1000
13	Spare	0x2000
14	Spare	0x4000
15	Spare	0x8000

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 ACT/FLT LED’s blink at a rate of every one-second, call ProSoft Technology, Inc. support. There is a serious problem with the module, and it will have to be sent back to ProSoft.

5.2.1 Clearing a Fault Condition

Typically, if the ACT/FAULT LED on the front of the module becomes illuminated red for over ten seconds, a hardware problem has been detected in the module or the program has exited. To attempt to clear the condition:

1. Remove the card from the rack and re-insert the card in the rack
2. Verify the configuration data being transferred to the module from the SLC processor

If the module's ACT/FAULT LED does not turn green, make sure the module is inserted completely into the rack. If this does not cure the problem, contact the factory.

5.2.2 Troubleshooting

In order to assist in the troubleshooting of the module, the following table has been put together to assist you. Please use the following to help in using the module, but if you have additional questions or problems, please do not hesitate to contact us.

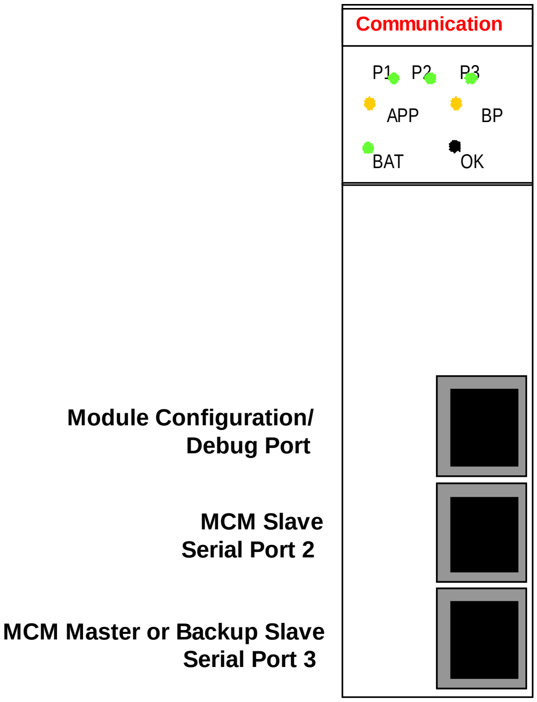
The entries in this section have been placed in the order in which the problems would most likely occur after powering up the module.

Problem Description	**Steps to take**
Processor Fault	Be sure that the module is plugged into the slot that has been configured for the MVI46-MCM module. Assure that the slot in the rack configuration has been set up correctly:
Processor I/O LED flashes	This indicates there is a problem with backplane communications. Be certain this and all modules in the rack are configured in the processor.
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Use the Configuration/Debug port facility to check this. To establish backplane communications make sure of the following: The backplane driver is loaded in the module. The module is configured for read and write block data transfer. The ladder logic handles all read and write block situations. The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, remove the card from the rack and re-insert the card in the rack.

6 Cable Connections

The MVI46-MCM module has the following communication connections on the module:

- Two Modbus communication ports (RJ45 connector)
- One RS-232 Configuration/Debug port (RJ45 connector)



6.1 Modbus Communication Ports

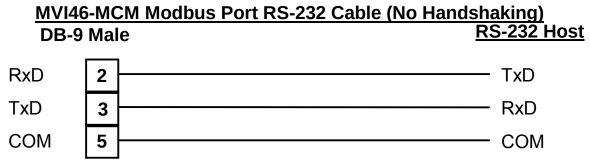
The MVI46-MCM module has two physical Modbus connectors with a RJ45 plug located on the front of the module.

6.1.1 Connecting the Cable to the Connector

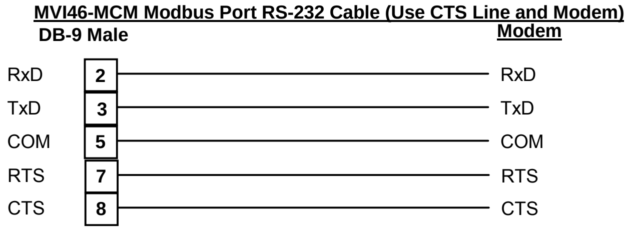
ProSoft provides two RJ45 to male DB-9 pigtails to permit simpler interfacing to other devices. The module's MCM ports can be configured to operate in RS-232, RS-422 or RS-485 mode. The interface to be associated with a port is set with jumpers on the module. There is a jumper for each of the two ports (*See* Setting the Jumpers). Additionally, the use of the modem control lines is user definable. Each interface is described in the following sections.

6.1.1.1. RS-232

When the RS-232 interface is selected, the use of the modem control lines is user definable. If no modem control lines are used, the cable to connect to the port is as shown in the following example:

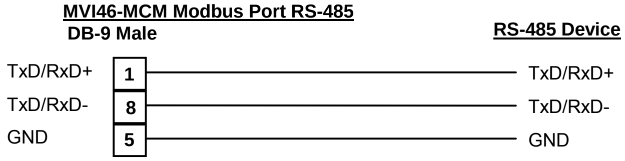


The RTS line is controlled by the RTS on and off parameters set for the port. If the CTS line is used (usually only required for half-duplex modems), the RTS and CTS lines must either be connected together or connected to the modem. The following diagram displays the cable required when connecting the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



6.1.1.2. RS-485

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



6.1.1.3. RS-422

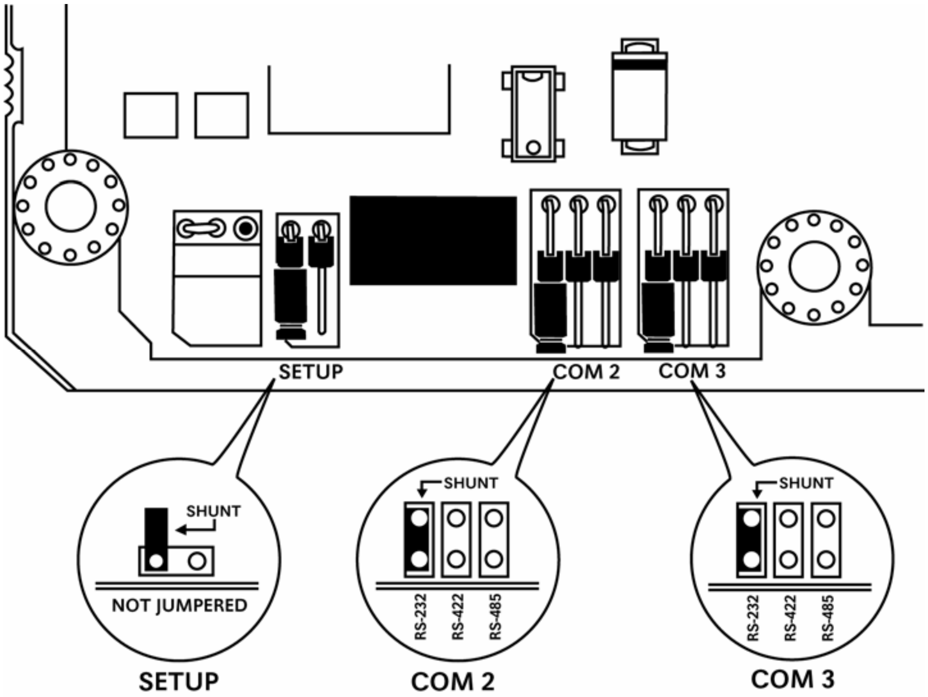
When the RS-422 interface is used, a four or five wire cable is required. The use of the ground is optional and dependent on the RS-422 network. The cable required for this interface is shown in the following diagram:

MVI46-MCM Modbus Port RS-422 Cable

	DB-9 Male	RS-422 Device
TxD+	1	RxD+
TxD-	8	RxD-
COM	5	COM
RxD+	2	TxD+
RxD-	6	TxD-

6.1.2 Setting the Jumpers

If using an interface other than RS-232 (default), you must change the jumpers on the module to the appropriate interface. The following diagram shows the jumper configurations for the MVI46-MCM module:



6.1.2.1. *RS-232 Configuration/Debug Port*

This port is physically a RJ-45 connection. A RJ-45 to DB-9 pigtail cable is shipped with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:

MVI46-MBP Configuration/Debug Port Cable

	DB-9 Male	RS-232 Host
RxD	2	TxD
TxD	3	RxD
COM	5	COM

Appendix A - MVI46-MCM Database Definition

This appendix contains a listing of the internal database of the MVI46-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
0-4999	40001	45000	User Data	5000
5000-5009	45001	45010	Backplane Configuration	10
5010-5039	45011	45040	Port 1 Setup	30
5040-5069	45041	45070	Port 2 Setup	30
5070-6069	45071	46070	Port 1 Commands	1000
6070-7069	46071	47070	Port 2 Commands	1000
7200-7232	47201	47233	Misc. Status Data	33
7600-7799	47601	48000	Command Control	200

The User Data area is used to hold 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 Appendix B.

Definition of the configuration data areas can be found in the data definition section of this document and in Appendix C.

Appendix D contains a discussion of the command control section of the database.

Appendix B – MVI46-MCM Status Data Definition

This appendix contains a description of the data present in the status data area. This data is constantly transferred from the module to the M1 data area.

M1 File	Content	Description
7200	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
7201-7202	Product Code	These two registers contain the product code of “MCM4”.
7203-7204	Product Version	These two registers contain the product version for the currently running software.
7205-7206	Operating System	These two registers contain the month and year values for the program operating system.
7207-7208	Run Number	These two registers contain the run number value for the currently running software.
7209	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
7210	Port 1 Command List Response	This field contains the number of slave response messages received on the port.
7211	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.
7212	Port 1 Requests	This field contains the total number of messages sent out of the port.
7213	Port 1 Responses	This field contains the total number of messages received on the port.
7214	Port 1 Errors Sent	This field contains the total number of message errors sent out of the port.
7215	Port 1 Errors Received	This field contains the total number of message errors received on the port.
7216	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
7217	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
7218	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.
7219	Port 2 Requests	This field contains the total number of messages sent out of the port.
7220	Port 2 Responses	This field contains the total number of messages received on the port.
7221	Port 2 Errors Sent	This field contains the total number of message errors sent out of the port.
7222	Port 2 Errors Received	This field contains the total number of message errors received on the port.
7223	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.

M1 File	Content	Description
7224	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
7225	Parse Block Count	Not used in this version of the module.
7226	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
7227	Command Block Count	This field contains the total number of command blocks received from the processor.
7228	Error Block Count	This field contains the total number of block errors recognized by the module.

Appendix C – MVI46-MCM Configuration Data Definition

This appendix contains listings of the MVI46-MCM module's database that are related to the module's configuration. This data is available to any node on the network and is read from the SLC processor when the module first initializes.

Group	Register	Content	Description
Backplane Setup	5000	Write Start Reg	Not used in this version of the software
	5001	Write Reg Count	Not used in this version of the software
	5002	Read Start Reg	Not used in this version of the software
	5003	Read Reg Count	Not used in this version of the software
	5004	Backplane Fail	Not used in this version of the software
	5005	Error Status Pointer	This parameter specifies the register location in the module’s database where module status data is stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 4940, the data is placed in the user data area.
	5006	Spare	
	5007	Spare	
	5008	Spare	
	5009	Spare	
Port 1 Setup	5010	Enable	This parameter is used to define if this port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
	5011	Type	This parameter defines if the port emulates a master or slave device. Enter 0 to emulate a master device and 1 to emulate a slave device.
	5012	Float Flag	This flag specifies if the floating-point data access functionality is to be used. If the float flag is set to Y, Modbus functions 3,6, and 16 will interpret floating point values for registers as specified by the two following parameters.
	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. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered floating-point data.

Group	Register	Content	Description
	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. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
	5015	Protocol	0=Modbus RTU, 1=Modbus ASCII
	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. Exceptions: 38400 baud, enter 384. 57600 enter 576. 115000 enter 115.
	5017	Parity	This is the parity code to be used on the port. The coded values are as follows: 0=None 1=Odd 2=Even
	5018	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Enter a value in the range of 5 to 8.
	5019	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Enter a value of 1 or 2.
	5020	RTS On Delay	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Enter a value in the range of 0 to 65535.
	5021	RTS Off Delay	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. Enter a value in the range of 0 to 65535.
	5022	Minimum Response Delay	This parameter sets the number of milliseconds to wait before a response message is sent out of the port. This parameter is required when interfacing to a slow responding device. Enter a value in the range of 0 to 65535.
	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. Normally, this parameter is required when half-duplex modems are used for communication (2-wire).

Group	Register	Content	Description
	5024	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. Any requests received by the port with this address will be processed by the module. Be certain each device has a unique address on the network.
	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.
	5026	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests 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.
	5027	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 return the value at register 100 in the database.
	5028	Holding Register Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.
	5029	Command Count	This parameter specifies the number of commands to be processed for the port. Enter a value of 0 to 100.
	5030	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid “flooding” commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized. Enter a value in the range of 0 to 65535.
	5031	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error data is placed. If the value is set to –1, the data will not be transferred to the database. Enter a value of 0 to 4999.

Group	Register	Content	Description
	5032	Response Timeout	This parameter represents the message response timeout period in 1 ms 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 on to communication network used and the expected response time of the slowest device on the network.
	5033	Retry Count	This parameter specifies the number of times a command is retried if it fails. Enter a value in the range of 0 to 10.
	5034	Error Delay Count	This parameter specifies the number of polls to be skipped 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 the parameter. Enter a value in the range of 0 to 65535.
	5035	Reserved	
	5036	Guard Band Timer	Use packet gap timeout for messages (Yes or No). Use only in multi-drop applications.
	5037	Guard Band Timeout	A value of 0 uses the default baud rate or you can set a timeout value in milliseconds.
Port 2 Setup	5040	Enable	This parameter is used to define if this port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
	5041	Type	This parameter defines if the port emulates a master or slave device. Enter 0 to emulate a master device and 1 to emulate a slave device.
	5042	Float Flag	This flag specifies if the floating-point data access functionality is to be used. If the float flag is set to Y, Modbus functions 3,6, and 16 will interpret floating point values for registers as specified by the two following parameters.
	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. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered floating-point data.
	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. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.

Group	Register	Content	Description
	5045	Protocol	0=Modbus RTU, 1=Modbus ASCII
	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. Exceptions: 38400 baud, enter 384. 57600 enter 576. 115000 enter 115.
	5047	Parity	This is the parity code to be used on the port. The coded values are as follows: 0=None 1=Odd 2=Even
	5048	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Enter a value in the range of 5 to 8.
	5049	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Enter a value of 1 or 2.
	5050	RTS On Delay	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Enter a value in the range of 0 to 65535.
	5051	RTS Off Delay	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. Enter a value in the range of 0 to 65535.
	5052	Minimum Response Delay	This parameter sets the number of milliseconds to wait before a response message is sent out of the port. This parameter is required when interfacing to a slow responding device. Enter a value in the range of 0 to 65535.
	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. Normally, this parameter is required when half-duplex modems are used for communication (2-wire).
	5054	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. Any requests received by the port with this address will be processed by the module. Be certain each device has a unique address on the network.
	5055	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.

Group	Register	Content	Description
	5056	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests 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.
	5057	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 return the value at register 100 in the database.
	5058	Holding Register Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.
	5059	Command Count	This parameter specifies the number of commands to be processed for the port. Enter a value of 0 to 100.
	5060	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid “flooding” commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized. Enter a value in the range of 0 to 65535.
	5061	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error data will be placed. If the value is set to –1, the data will not be transferred to the database. Enter a value of 0 to 4999.
	5062	Response Timeout	This parameter represents the message response timeout period in 1 ms 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 on to communication network used and the expected response time of the slowest device on the network.
	5063	Retry Count	This parameter specifies the number of times a command will be retried if it fails. Enter a value in the range of 0 to 10.

Group	Register	Content	Description
	5064	Error Delay Count	This parameter specifies the number of polls to be skipped 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 the parameter. Enter a value in the range of 0 to 65535.
	5065	Reserved	
	5066	Use Guard Band Timer	Use packet gap timeout for messages (Yes or No). Use only in multi-drop applications.
	5066	Guard Band Timeout	A value of 0 uses the default baud rate or you can set a timeout value in milliseconds (0 to 65535).
Port 1 Commands	5200-5209	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.
	5210-5219	Command #2	Command #2 dataset
	6190-6199	Command #100	Command #100 dataset
Port 2 Commands	6200-6209	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.
	6210-6219	Command #2	Command #2 dataset
	7190-7199	Command #100	Command #100 dataset
Misc. Status	7200	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
	7201	Product Code	These two registers contain the product code of “MCM4”.
	7202		
	7203	Product Version	These two registers contain the product version for the currently running software.
	7204		
	7205	Operating System	These two registers contain the month and year values for the program operating system.
	7206		
	7207	Run Number	These two registers contain the run number value for the currently running software.
	7208		
	7209	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
	7210	Port 1 Command List Response	This field contains the number of slave response messages received on the port.

Group	Register	Content	Description
	7211	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.
	7212	Port 1 Requests	This field contains the total number of messages sent out the port.
	7213	Port 1 Responses	This field contains the total number of messages received on the port.
	7214	Port 1 Errors Sent	This field contains the total number of message errors sent out the port.
	7215	Port 1 Errors Received	This field contains the total number of message errors received on the port.
	7216	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
	7217	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
	7218	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.
	7219	Port 2 Requests	This field contains the total number of messages sent out the port.
	7220	Port 2 Responses	This field contains the total number of messages received on the port.
	7221	Port 2 Errors Sent	This field contains the total number of message errors sent out the port.
	7222	Port 2 Errors Received	This field contains the total number of message errors received on the port.
	7223	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
	7224	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
	7225	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
	7226	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
	7227	Command Block Count	This field contains the total number of command blocks received from the processor.
	7228	Error Block Count	This field contains the total number of block errors recognized by the module.
	7229	Port 1 Current Error/Index	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.

Group	Register	Content	Description
	7230	Port 1 Last Error/Index	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.
	7231	Port 2 Current Error/Index	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.
	7232	Port 2 Last Error/Index	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.
Command Control	7800	Command Code	Enter one of the valid control command codes in this register to control the module (9997, 9998, or 9999). Refer to Appendix D for more information.
	7801	Command Data	Reserved for future use.
	…	…	…
	7999	Command Data	Reserved for future use.

Appendix D – MVI46-MCM Remote Configuration

Remote configuration data can be received from other nodes on the network that can control the MVI46-MCM module. Specific values are written to regions of this block to change the module's configuration parameters. 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):

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:

0	No error, transfer successful
-1	Error transferring general configuration information.

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

Appendix E – MVI46-MCM Configuration

This appendix contains listings of the MVI46-MCM configuration data required in the M0 file. The module requests data after a restart or warm-boot operation. The module will continually request the data until a valid configuration is received.

Group	File	M0 Offset	Content	Description
Backplane Setup	N10:0	1	Write Start Reg	Not used in this version of the software
	N10:1	2	Write Reg Count	Not used in this version of the software
	N10:2	3	Read Start Reg	Not used in this version of the software
	N10:3	4	Read Reg Count	Not used in this version of the software
	N10:4	5	Backplane Fail	Not used in this version of the software
	N10:5	6	Error Status Pointer	This parameter specifies the register location in the module's database where module status data is stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 4940, the data is placed in the user data area.
	N10:6	7	Spare	
	N10:7	8	Spare	
	N10:8	9	Spare	
	N10:9	10	Spare	
Port 1 Setup	N10:10	11	Enable	This parameter is used to define if this port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
	N10:11	12	Type	This parameter defines if the port emulates a master or slave device. Enter 0 to emulate a master device and 1 to emulate a slave device.
	N10:12	13	Float Flag	This flag specifies if the floating-point data access functionality is to be used. If the float flag is set to Y, Modbus functions 3,6, and 16 will interpret floating point values for registers as specified by the two following parameters.

Group	File	M0 Offset	Content	Description
	N10:13	14	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. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered floating-point data.
	N10:14	15	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. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
	N10:15	16	Protocol	0=RTU, 1=ASCII
	N10:16	17	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. Exceptions: 38400 baud, enter 384. 57600 enter 576. 115000 enter 115.
	N10:17	18	Parity	This is the parity code to be used on the port. The coded values are as follows: 0=None 1=Odd 2=Even
	N10:18	19	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Enter a value in the range of 5 to 8.
	N10:19	20	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Enter a value of 1 or 2.
	N10:20	21	RTS On Delay	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Enter a value in the range of 0 to 65535.
	N10:21	22	RTS Off Delay	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. Enter a value in the range of 0 to 65535.

Group	File	M0 Offset	Content	Description
	N10:22	23	Minimum Response Delay	This parameter sets the number of milliseconds to wait before a response message is sent out of the port. This parameter is required when interfacing to a slow responding device. Enter a value in the range of 0 to 65535.
	N10:23	24	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. Normally, this parameter is required when half-duplex modems are used for communication (2-wire).
	N10:24	25	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. Any requests received by the port with this address will be processed by the module. Be certain each device has a unique address on the network.
	N10:25	26	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.
	N10:26	27	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests 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.
	N10:27	28	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 return the value at register 100 in the database.
	N10:28	29	Holding Register Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.

Group	File	M0 Offset	Content	Description
	N10:29	30	Command Count	This parameter specifies the number of commands to be processed for the port. Enter a value of 0 to 100.
	N10:30	31	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized. Enter a value in the range of 0 to 65535.
	N10:31	32	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error data will be placed. If the value is set to –1, the data will not be transferred to the database. Enter a value of 0 to 4999.
	N10:32	33	Response Timeout	This parameter represents the message response timeout period in 1 ms 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 on to communication network used and the expected response time of the slowest device on the network.
	N10:33	34	Retry Count	This parameter specifies the number of times a command will be retried if it fails. Enter a value in the range of 0 to 10.
	N10:34	35	Error Delay Count	This parameter specifies the number of polls to be skipped 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 the parameter. Enter a value in the range of 0 to 65535.
	N10:35	36	Reserved	
	N10:36	37	Use Guard Band Timer	Use packet gap timeout for messages (Yes or No). Use only in multi-drop applications.
	N10:37	38	Guard Band Timeout	A value of 0 uses the default baud rate or you can set a value in milliseconds (0 to 65535)
	N10:38	39	Spare	
	N10:39	40	Spare	

Group	File	M0 Offset	Content	Description
Port 2 Setup	N10:40	41	Enable	This parameter is used to define if this port will be used. If the parameter is set to 0, the port is disabled. A value of 1 enables the port.
	N10:41	42	Type	This parameter defines if the port emulates a master or slave device. Enter 0 to emulate a master device and 1 to emulate a slave device.
	N10:42	43	Float Flag	This flag specifies if the floating-point data access functionality is to be used. If the float flag is set to Y, Modbus functions 3,6, and 16 will interpret floating point values for registers as specified by the two following parameters.
	N10:43	44	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. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered floating-point data.
	N10:44	45	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. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
	N10:45	46	Protocol	0=RTU, 1=ASCII
	N10:46	47	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. Exceptions: 38400 baud, enter 384. 57600 enter 576. 115000 enter 115.
	N10:47	48	Parity	This is the parity code to be used on the port. The coded values are as follows: 0=None 1=Odd 2=Even
	N10:48	49	Data Bits	This parameter sets the number of data bits for each word used by the protocol. Enter a value in the range of 5 to 8.
	N10:49	50	Stop Bits	This parameter sets the number of stop bits to be used with each data value sent. Enter a value of 1 or 2.

Group	File	M0 Offset	Content	Description
	N10:50	51	RTS On Delay	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted. Enter a value in the range of 0 to 65535.
	N10:51	52	RTS Off Delay	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. Enter a value in the range of 0 to 65535.
	N10:52	53	Minimum Response Delay	This parameter sets the number of milliseconds to wait before a response message is sent out of the port. This parameter is required when interfacing to a slow responding device. Enter a value in the range of 0 to 65535.
	N10:53	54	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. Normally, this parameter is required when half-duplex modems are used for communication (2-wire).
	N10:54	55	Slave ID	This parameter defines the virtual Modbus slave address for the internal database. Any requests received by the port with this address will be processed by the module. Be certain each device has a unique address on the network.
	N10:55	56	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.
	N10:56	57	Word Input Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests 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.

Group	File	M0 Offset	Content	Description
	N10:57	58	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 return the value at register 100 in the database.
	N10:58	59	Holding Register Offset	This parameter specifies the offset address in the internal Modbus database that is to be used with network requests for Modbus 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.
	N10:59	60	Command Count	This parameter specifies the number of commands to be processed for the port. Enter a value of 0 to 100.
	N10:60	61	Minimum Command Delay	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid “flooding” commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized. Enter a value in the range of 0 to 65535.
	N10:61	62	Command Error Pointer	This parameter sets the address in the internal Modbus database where the command error data will be placed. If the value is set to –1, the data will not be transferred to the database. Enter a value of 0 to 4999.
	N10:62	63	Response Timeout	This parameter represents the message response timeout period in 1 ms 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 on to communication network used and the expected response time of the slowest device on the network.
	N10:63	64	Retry Count	This parameter specifies the number of times a command will be retried if it fails. Enter a value in the range of 0 to 10.

Group	File	M0 Offset	Content	Description
	N10:64	65	Error Delay Count	This parameter specifies the number of polls to be skipped 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 the parameter. Enter a value in the range of 0 to 65535.
	N10:65	66	Reserved	
	N10:66	67	Use Guard Band Timer	Use packet gap timeout for messages (Yes or No). Use only in multi-drop applications.
	N10:67	68	Guard Band Timeout	A value of 0 uses the default baud rate or you can set a timeout value in milliseconds (0 to 65535).
	N10:68	69	Spare	
	N10:69	70	Spare	
Port 1 Commands	N11:0 to N11:09	71 to 80	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.
	N11:10 to N11:19	81 to 90	Command #2	Command #2 dataset
		1061 to 1070	Command #100	Command #100 dataset
Port 2 Commands	N12:0 to N12:09	1071 to 1080	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.
	N12:10 to N12:19	1081 to 1090	Command #2	Command #2 dataset
		2061 to 2070	Command #100	Command #100 dataset

Support, Service, and Warranty

Technical Support

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

Internet	Web Site: http://www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Phone	(661) 716-5100 (661) 716-5101 (fax)
Postal Mail	ProSoft Technology, Inc. 1675 Chester Avenue, Second Floor Bakersfield, CA 93301

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

1. Product Version Number
2. System hierarchy
3. Module configuration and contents of MCM.CFG file
4. Module Operation
   - Configuration/Debug status information
   - LED patterns
5. Information about the processor and data areas as viewed through RSLogix 500 and LED patterns on the processor
6. Details about the serial network

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

Module Service and Repair

The MVI46-MCM card is an electronic product, designed and manufactured to function under somewhat adverse conditions. As with any product, through age, misapplication, or any one of many possible problems the card may require repair.

When purchased from ProSoft Technology, the module has a one-year parts and labor warranty according to the limits specified in the warranty. Replacement and/or returns should be directed to the distributor from whom the product was purchased. If you need to return the card for repair, obtain an RMA number from ProSoft Technology. Please call the factory for this number and display the number prominently on the outside of the shipping carton used to return the card.

General Warranty Policy

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

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

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

Limitation of Liability

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

Where directed by State Law, some of the above exclusions or limitations may not be applicable in some states. This warranty provides specific legal rights; other rights that vary from state to state may also exist. This warranty shall not be applicable to the extent that any provisions of this warranty are prohibited by any Federal, State or Municipal Law that cannot be preempted.

Hardware Product Warranty Details

Warranty Period: ProSoft warranties hardware product for a period of one (1) year.

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

- - - - - - - - - - - - - - - - END OF MANUAL - - - - - - - - - - - - - - - - -