inRAx

MVI46-MBP

SLC Platform

Modbus Plus Communication Module

User Manual

February 19, 2008

ProSoft TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation SLC hardware, the MVI46-MBP Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

Your Feedback Please

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

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

MVI46-MBP User Manual
February 19, 2008
PSFT.MBP.MVI46.UM.08.02.19

Contents

Guide to the MVI46-MBP User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 7)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 34) Diagnostics and Troubleshooting (page 21)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 37) Functional Overview (page 39) Product Specifications (page 37)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 89)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI46-MBP module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI46-MBP Module and the application in which they will be used.

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

1.1 System Requirements

The MVI46-MBP module requires the following minimum hardware and software components:

- Rockwell Automation SLC 5/02 M0/M1 capable processors (or newer), with compatible power supply and one free slot in the rack, for the MVI46-MBP module. The module requires 800mA of available power.
- Rockwell Automation RSLogix 500 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® II 500 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft® Windows 98
  - Windows NT® (version 4 with SP4 or higher)
  - Windows 2000
  - Windows XP
- 32 Mbytes of RAM minimum, 64 Mbytes of RAM recommended
- 50 Mbytes of free hard disk space (or more based on application requirements)
- 16-color VGA graphics adapter, 640 x 480 minimum resolution (256 Color 800 × 600 recommended)

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

1.2 Package Contents

The following components are included with your MVI46-MBP module, and are all required for installation and configuration.

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

Qty.	Part Name	Part Number	Part Description
1	MVI46-MBP Module	MVI46-MBP	Modbus Plus Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
1	Cable	Cable #14, RJ45 to DB9 Male Adapter	For DB9 Connection to the CFG Port
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI46-MBP module.

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

1.3 Install the Module in the Rack

If you have not already installed and configured your SLC processor and power supply, please do so before installing the MVI46-MBP module. Refer to your Rockwell Automation product documentation for installation instructions.

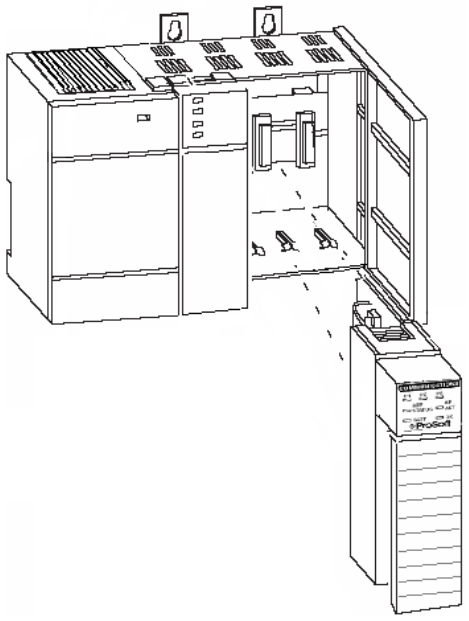
Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify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

After you have checked the placement of the jumpers, insert MVI46-MBP into the SLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install SLC™ modules.

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

1 Turn power OFF.

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

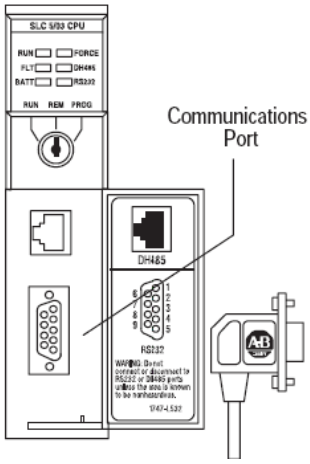


3 With a firm but steady push, snap the module into place.
4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample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

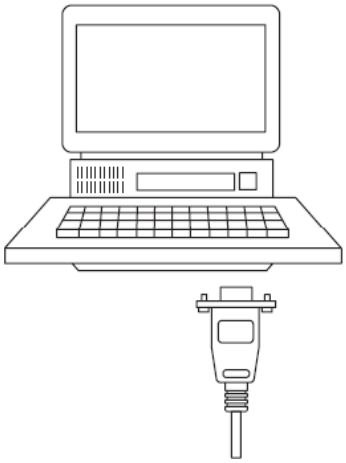
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.4 Connect your PC to the Processor

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



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

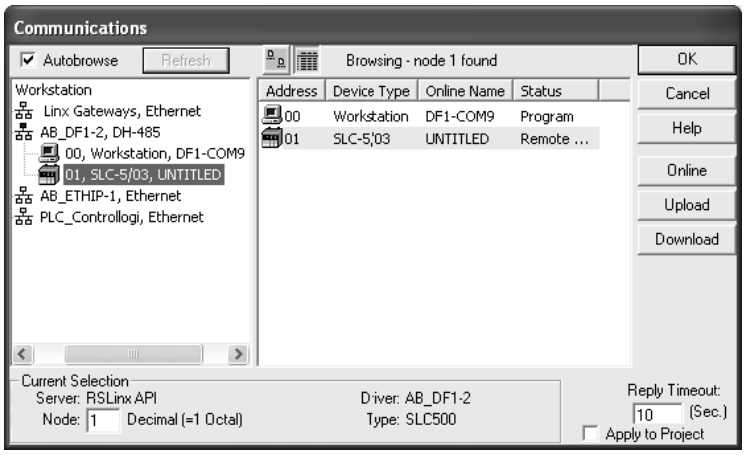


1.5 Download the Sample Program to the Processor

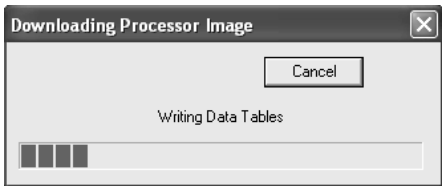
To download the sample program from RSLogix 500 to the SLC processor:

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

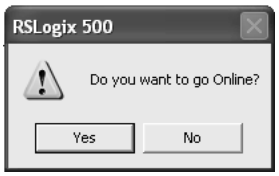
1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



2 Click the Download button to transfer the sample program to the processor.
3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



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

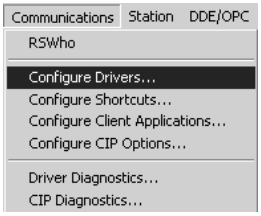


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

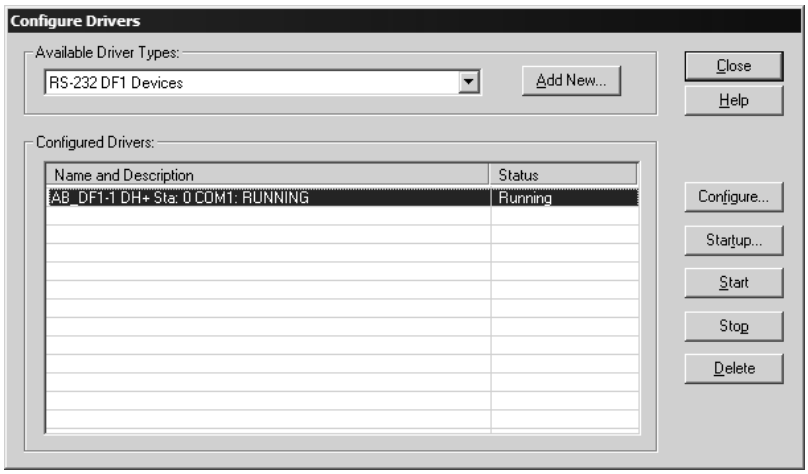
1.5.1 Configuring RSLinx

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

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

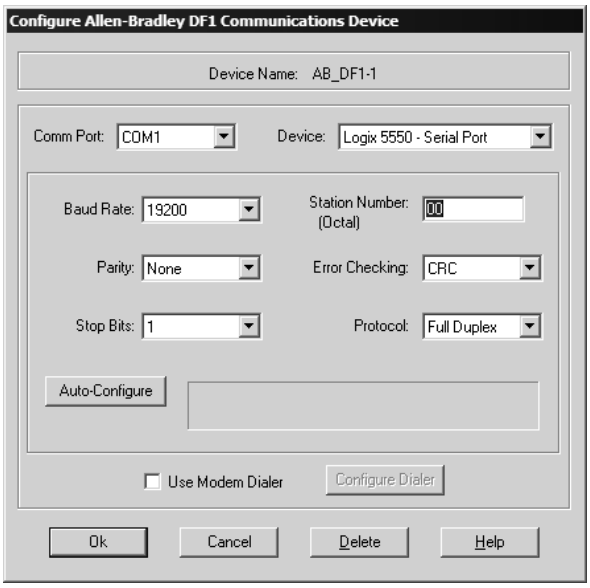


This action opens the Configure Drivers dialog box.



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

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



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

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

1.6 Connect your PC to the Module

With the module securely mounted, connect your PC to the Configuration/Debug port using the RJ45-DB-9 Serial Adapter Cable and the Null Modem Cable included in the package with the MVI46-MBP module.

1 Connect the RJ45-DB-9 Serial Adapter Cable to the Null Modem Cable.
2 Insert the RJ45 cable connector from the RJ45-DB-9 cable into the Configuration/Debug port of the module.
3 Attach the other end to the serial port on your PC or laptop.

2 Installing and Configuring the Module

In This Chapter

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

1 Use RSLogix 500 to identify the module to the processor and add the module to a project.

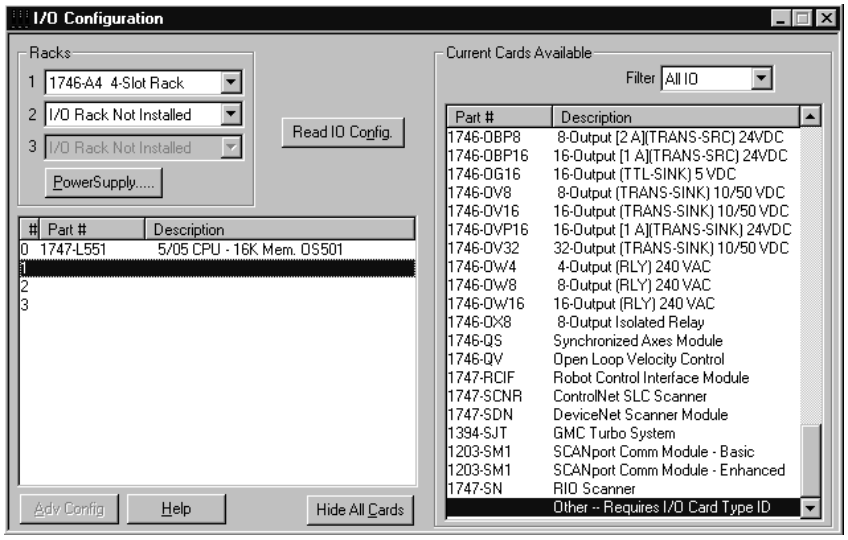
Note: The RSLogix 500 software must be in "offline" mode to add the module to a project.

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

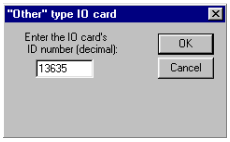
Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

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

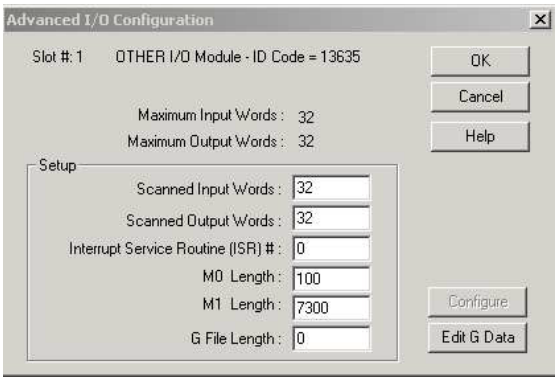
The first step in installing and configuring the module is to define the module to the system. Select the I/O Configuration option from the program screen. This displays the following dialog box:



Select the Other module from the list. This action opens the following dialog box.



Enter the module I/O card ID number as 13635, and then click OK. Double-click the mouse on the module just added to the rack. Fill in the following dialog box as shown in the following screen example:



Click OK to apply these settings to the module. Then, close the I/O Configuration dialog box.

The next step is to define the user-defined files to hold status and read/write database areas.

The last step 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 use with your application. Insert the module in the rack, then attach the serial communication cable to the debug port and the cable from the application port to the Modbus Plus network. 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 network, the module's Application LED (APP LED) should blink (6 times per second) and the backplane activity LED (BP ACT) should blink very rapidly. If you encounter errors, refer to the **Diagnostics and Troubleshooting** (page 21) section for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

2.1 Module Data

All data related to the MVI46-MBP module is stored in user defined data files and the module's M1 and M0 files. Files should be defined for each data type to be used with the module. Additionally, a file should be defined to hold the module status data. The status data should be copied from the M1 file and placed in the assigned status file. Read (monitor) data should be copied to the user files from the M1 file and write (control) data should be copied from the user files to the M1 file.

2.2 Module Configuration

In order for the MVI46-MBP module to function in any of its possible modes, 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
4370 to 4409	Global In Global Out Slave Master	General Module Configuration	This section of the configuration data contains the generic module configuration data, and must be configured for the module to operate.
4410 to 4449	Global In	Input File Map	If the module's Input File is to be used to transfer data from the module to the processor, then this section of configuration data must be set up.
4450 to 5089	Global In Master	Device Definition	If the module's Global Input or Master Mode functionality is to be used, then this section of configuration data must be set up.
5090 to 7089	Master	Master Command List	If the module's Master Mode functionality is to be used, then he Master Command List must be set up.

Refer to the Reference chapter of this manual for a description of the configuration of the module.

Important: The module will not function correctly until the Module Configuration Data is received from the processor with at least the Local Modbus Plus Node Address set to a valid value.

The MVI46-MBP module must be configured at least once when the card is first powered, and any time thereafter when the parameters must be changed.

Power Up

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

Changing Parameters During Operation

Changing values in the configuration table can be done at any time. Because the module is operating using the live data in the internal database, any changes made to the database become immediately active. This permits remote programming of the module by any node on the Modbus Plus network. Care must be taken when altering the parameters in an order that will not disturb any running processes. New configuration data can be downloaded from the database to the SLC processor by executing a write command with a value of 9997 to register 4370 in the module's internal database or M1 file. To force the module to perform a warm-boot or cold-boot operation, write to register 4370 with values of 9998 and 9999, respectively.

The only parameters that must be set through a restart of the module are related to the read and write data sizes and registers for data transfer. These parameters cannot be changed while the module is operating as it could place the module in an inoperable mode.

3 Ladder Logic

Ladder logic is required for application of the MVI46-MBP module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

4 Diagnostics and Troubleshooting

In This Chapter

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

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

4.1 Reading Status Data from the Module

The MVI46-MBP module returns a 365-word Status Data area that can be used to determine the module's operating status. This data is located in the module's database and the M1 file at registers 4000 to 4365. Nodes on the Modbus Plus network read this data area through the issuance of read commands to the module. For a complete listing of the status data objects, Refer to the Reference chapter.

The Configuration/Debug port provides the following functionality:

- Full view of the modules database
- View of the module's status data
- View of the module's configuration
- Version Information
- Control over the module (that is, cold boot)

4.1.1 The Configuration/Debug Menu

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

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

4.1.2 Required Hardware

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

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

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

4.1.3 Required Software

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

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

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

4.1.4 Using the Configuration/Debug Port

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

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

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

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

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

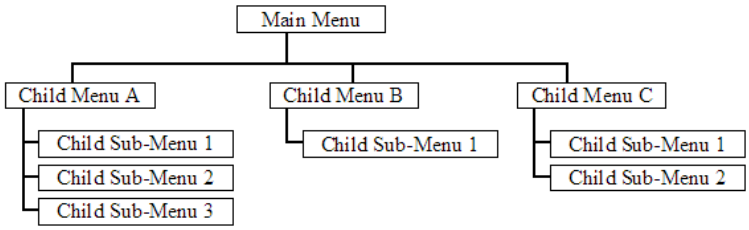
1. Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2. Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC (page 57).
3. Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
4. On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

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



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

Keystrokes

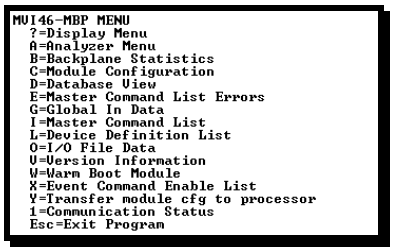
The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

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

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

4.1.5 Main Menu

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



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

Opening the Data Analyzer Menu

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

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

Viewing Backplane Diagnostic Information

Press **[B]** from the Configuration/Debug Menu to view the Backplane Diagnostic Information screen.

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

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

Viewing Module Configuration

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

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

Opening the Database Menu

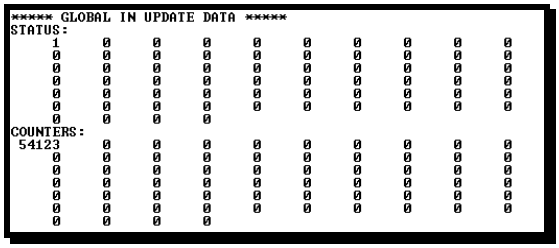
Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Opening the Command Error List Menu

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

Viewing Global Input Status and Counter Data

Press **[G]** to view the Global Input Status and Counter data for each Modbus Plus node defined in the Device Definition Table.



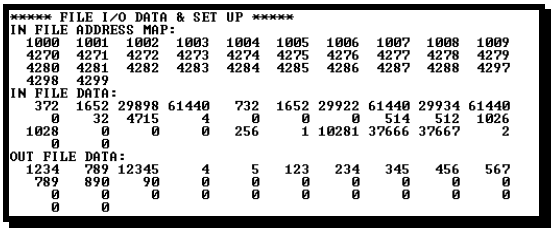
The Status data area displays the current status of each node on the network. Refer to the Module Set Up section for a complete listing and definition of the status codes. The Counters section displays the number of Global Input data messages received from the device.

Opening the Device Definition List Menu

Press **[L]** to open the Device Definition List. This list consists of multiple pages of device definition data. Press **[?]** to view a list of commands available on this menu.

Viewing I/O File Data

Press [O] to display the module's global input and output data.



The **In File Address Map**, values represent the data set for selecting the registers in the module's database to transfer to the In File Data area. The **In File Data** section displays the current values transferred from the module to the SLC processor. The **Out File Data** section contains the values used for Global Output Data to be transferred from the module to the Modbus Plus network.

Viewing Version Information

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

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

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

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

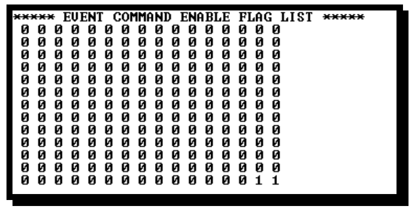
Warm Booting the Module

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

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

Viewing the Event Command Enable List

Press **[X]** to view the Event Command Enable List. Use this command to display the status of each of the event command enable bits received from the SLC processor.



```
***** EVENT COMMAND ENABLE FLAG LIST *****
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
```

If the event command bit is set, a value of 1 will be displayed. A value of 0 indicates the command event bit is clear. Each bit in the table corresponds to an associated command in the master command table. The bits displayed are shown with the high-bits in the word on the left side and the least-significant bit on the right side. Therefore, bit 15 (command 16) is the first bit displayed in the upper-right of the list and bit 0 (command 1) is the last bit of the first row of data.

Transferring Module Configuration to the Processor

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

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

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

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

Viewing Communication Status

Press **[1]** to view the communication status and statistics of the Modbus Plus Network for the module's node address. This command is useful for troubleshooting purposes.

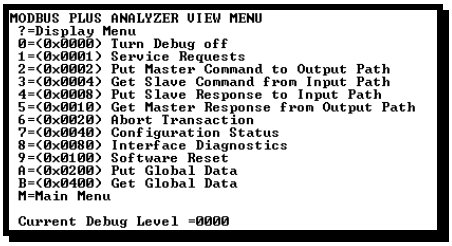
Exiting the Program

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

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

4.1.6 Data Analyzer

Use this menu to display Modbus Plus messages generated and received by the module. This tool is extremely useful in determining the operation of the module and nodes on the network. Press **[?]** to view the list of commands available on this menu. In the following illustration, master command status errors can be resolved.



Use the commands on this menu to choose the type of data to view. You can view more than one type of data at the same time. For example, to view the master command processing of messages from the MVI46-MBP module to other nodes on the network, press **[2]**, and then press **[5]**. All master Put and Get messages handled by the module will be displayed on the screen.

The **Current Debug Level** parameter displayed at the bottom of the display shows the current debug level being monitored by the analyzer. A value of 0000 indicates that the analyzer is not monitoring any messages.

(0x0000) Turn Debug off

Press **[0]** (zero) to turn the analyzer off. This is useful to freeze the analyzer screen with the last values displayed. If you are using a terminal emulation program that buffers previously received data, you can scroll through the acquired data to view the transactions that occurred on the Modbus Plus interface by the module.

(0x0001) Service Requests

Press **[1]** to view all service request operations performed. After selecting the option, the screen should scroll very rapidly as these messages are trapped. An example display is shown below:

(0x0002) Put Master Command to Output Path

Press **[2]** to display master command requests sent from the module to the network. Example output of this option is shown below:

```
MBP Put Master Command :  4 FF  2 24 11 22  0  4  0  0  2  0  0  0  0 10  2 8A
0  A 14  0  1  0  2  0  3  0  4  0  5  0  6  0  7  0  8  0 64  0 65
MBP Put Master Command Response (4)...MBP Put Master Command :  5 FF  2 24 11  D
  0  5  0  0  2  0  0  0  0  3  2 8A  0  A
MBP Put Master Command Response (5)...MBP Put Master Command :  6 FF  2 24 11  D
  0  6  0  0  2  0  0  0  0  3  2 8A  0  A
```

(0x0004) Get Slave Command from Input Path

Press **[3]** to display slave request messages received for the module. Example output of this option is shown below:

```
MBP Get Slave Command (72)....MBP Get Slave Command Response : 48 FF 24  2 11  D
  0  8  0 C2 24  0  0  0  0  3  0  0  0 14
MBP Get Slave Command (72)....MBP Get Slave Command Response : 48 FF 24  2 11 36
  0  5  0 EE 24  0  0  0 10  2 BC  0 14 28  0  1  0  2  0  3  0  4  0  5  0
 6  0  7  0  8  0 64  0 65  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0
```

(0x0008) Put Slave Response to Input Path

Press **[4]** to display slave response messages sent to requests made by other nodes on the network to the module. Example output of this option is shown below:

```
MBP Put Slave Response : 48 FF  2 24 16  D  0  5  1 12 24  0  0  0  0 10  2 BC
0 14
MBP Put Slave Command Response (72)...MBP Put Slave Response : 48 FF  2 24 16 32
  0  8  1 E6 24  0  0  0  0  3 28  0  1  0  2  0  3  0  4  0  5  0  6  0  7  0
8  0 64  0 65  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
MBP Put Slave Command Response (72)...MBP Put Slave Response : 48 FF  2 24 16  D
  0  5  1 13 24  0  0  0  0 10  2 BC  0 14
```

(0x0010) Get Master Response from Output Path

Press **[5]** to display master response messages received from other nodes on the network in response to command requests made by the module. Example output of this option is shown below:

```
MBP Get Master Response (6)....MBP Get Master Response :  6 FF 24  2 96  D  0  6
  1  3  2  0  0  0  0 10  2 8A  0  A
MBP Get Master Response (7)....MBP Get Master Response :  7 FF 24  2 96 1E  0  7
  1  3  2  0  0  0  0  3 14  0  1  0  2  0  3  0  4  0  5  0  6  0  7  0  8  0  6
4  0 65
MBP Get Master Response (8)....MBP Get Master Response :  8 FF 24  2 96 1E  0  8
  1  3  2  0  0  0  0  3 14  0  1  0  2  0  3  0  4  0  5  0  6  0  7  0  8  0  6
4  0 65
```

(0x0020) Abort Transaction

Press **[6]** to display any abort transaction messages processed on the Modbus Plus network by the module.

(0x0040) Configuration Status

Press **[7]** to display any configuration status messages processed by the module.

(0x0080) Interface Diagnostics

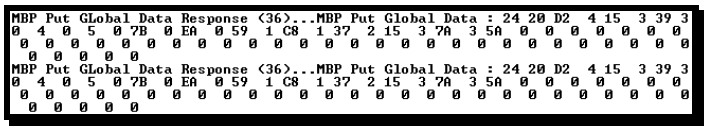
Press **[8]** to display any interface diagnostic messages processed by the module.

(0x0100) Software Reset

Press **[9]** to display any software-reset commands sent from the module to the Modbus Plus Chipset.

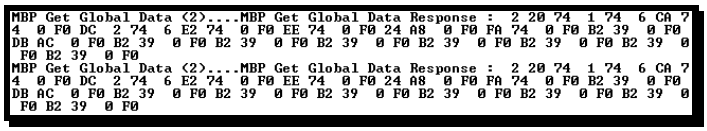
(0x0200) Put Global Data

Press **[A]** to display any global output messages processed by the module. The following illustration shows an example of the output.



(0x0400) Get Global Data

Press **[B]** to display any global input messages processed by the module. The following illustration shows an example of the output.

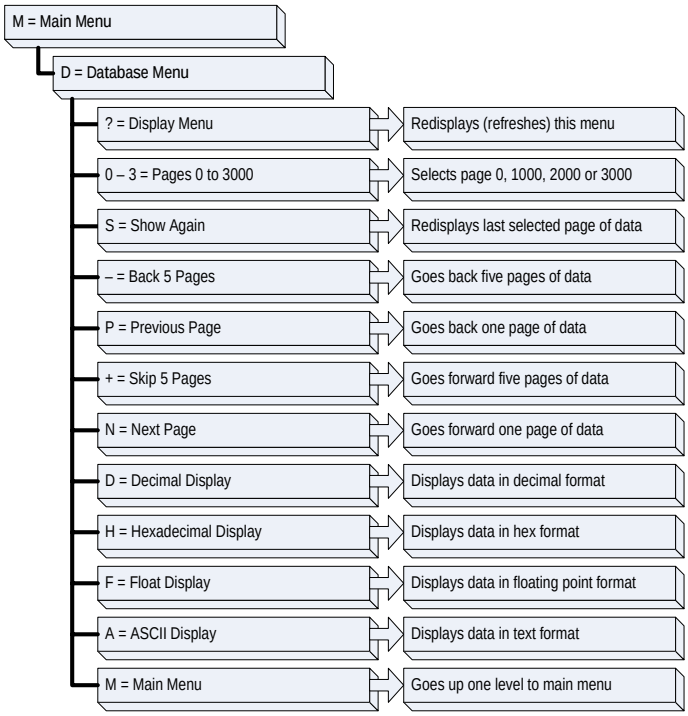


Returning to the Main Menu

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

4.1.7 Database View Menu

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



Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again

```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
   100   101   102     4     5     6     7     8     9    10
    11    12    13    14    15    16     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

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

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

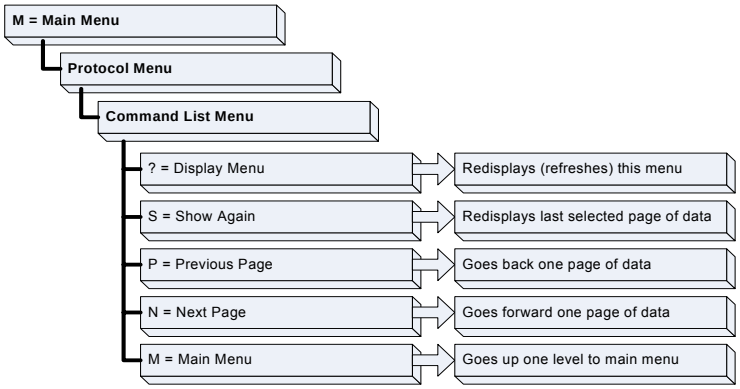
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.1.8 Master Command Error List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 20 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 20 Commands

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

Viewing the Next Page of Commands

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

Returning to the Main Menu

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

4.2 LED Status Indicators

The LEDs indicate the module's operating status as shown in the following table:

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	Off	This LED should always be off, as port 2 on the module is not used.
		On	Hardware problem or you have connected to the wrong port. Connect your terminal to port 1.
P2	Green	Off	This LED should always be off, as port 3 on the module is not used.
		On	Hardware problem or you have connected to the wrong port. Connect your terminal to port 1.
APP (Modbus Plus Status)	Amber	6 flashes per second	The MVI46-MBP is working normally in that it is successfully receiving and passing the token. All nodes on the link should be flashing this pattern.
		1 flash per second	This node is off-line after just being powered up, or after exiting the four flashes per second mode. In this state, the node monitors the network and builds a table of active nodes and token-holding nodes. It remains in this state for five seconds, then attempts to go to its normal operating state.
		Two Flashes then OFF for two sec	The node is hearing the token being passed among other nodes, but is never receiving the token. Check the network for an open circuit or defective termination.
		Three Flashes then OFF for 1.7 sec	The node is not hearing any other nodes. It is periodically claiming the token but finding no other node to which to pass it. Check the network for an open circuit or defective termination.
		Four Flashes then OFF for 1.4 sec	The node has heard a valid message from another node that is using the same address as this node. The node remains in this state as long as it continues to hear the duplicate address. If the duplicate address is not heard for five seconds, the node then changes to the pattern of one flash every second.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not plugged securely into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

4.2.1 Troubleshooting

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

Processor Errors

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

Module Errors

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI46 Modbus Plus Communication Module allows Rockwell Automation SLC I/O compatible processors to interface easily with other Modbus Plus protocol compatible devices.

The Modbus Plus module is a powerful module designed with both Master and Slave support, enabling easy connection to other Modbus devices (Modicon processors and many others).

5.1.1 Features and Benefits

Compatible devices include not only Modicon PLCs (which all support the Modbus Plus protocol) but also a wide assortment of end devices licensed through the ModConnect Program. The MVI46-MBP module acts as an input/output module between the Modbus Plus network and the SLC backplane. The data transfer from the SLC processor is asynchronous from the actions on the Modbus Plus network. A 4000-word register space in the module exchanges data between the processor and the Modbus Plus network.

These modules allow Rockwell Automation platforms to connect directly on Modbus Plus networks as a peer. In addition, the modules act as slaves to processors that must read/write data from the module's memory.

Crossing all industrial boundaries, potential applications include the connection of Rockwell Automation processors to Modicon processors, and the connection of Modbus Plus speaking devices such as drives, relays, and power monitor hardware to the SLC backplanes.

5.1.2 General Specifications

- Single Slot - 1746 backplane compatible (Local or extended I/O rack only. Remote rack not supported)
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module using M0/M1 files
- Ladder Logic is used for data transfer between module and processor
- Configuration data obtained through user-defined ladder. Sample ladder file included

5.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 ma @ 5V (from backplane)
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Shock	30g operational, 50g non-operational
Relative Humidity	5% to 95% (non-condensing)
Vibration	5 g from 10150 Hz
Processor	Compatible with Rockwell Automation SLC 5/02 M0/M1 capable processors or newer
LED indicators	Module status, Backplane transfer status, Application status, Serial activity and error LED status
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Modbus Plus ports	Dual DB9 Application ports for redundant operation

5.1.4 Functional Specifications

- Communication parameters
  - Baud: 57,600 (fixed)
  - Parity: none (fixed)
  - Stop: 1 (fixed)
- Modbus Plus ports
  - User-definable module memory usage
  - Support for the storage and transfer of up to 4,000 registers across the backplane
  - 100 word reads and writes (max. allowed)
  - Supports all five levels of Modbus Plus routing
  - Software configurable parameters Node address: 1 to 64
  - Global out size: 0 to 32 words
  - Global in size: 0 to 32 words
  - Module data transfer: 0 to 4,000 words
  - Master command count: 0 to 100 commands

- Function codes accepted (as a Slave): 1, 2, 3, 4, 5, 6, 15, 16
- Function codes transmitted (as a Master): 3, 16

Global Data Specifications

The MBP module actively exchanges global in (32 words max. per node) and global out (32 words max.) data on the Modbus Plus network. Priority is given to these data types to provide a high speed mechanism for the transfer of control data.

Modbus Slave Mode Specifications

- Supports broadcast commands from host
- Communication error codes returned to ladder logic

Modbus Master Mode Specifications

- Command list support of up to 100 commands
- Conditional and continuous command list polling
- Each command list entry is fully configurable for function register to/from addressing and word/bit count/word and byte swap
- Event driven bit and register write commands (ladder logic controlled)
- Supports sending of broadcast commands
- Communication status error codes returned to ladder logic on a per command basis

5.2 Functional Overview

This section provides an overview of how the MVI46-MBP module transfers data using the MBP protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI46-MBP module.

Module Power Up

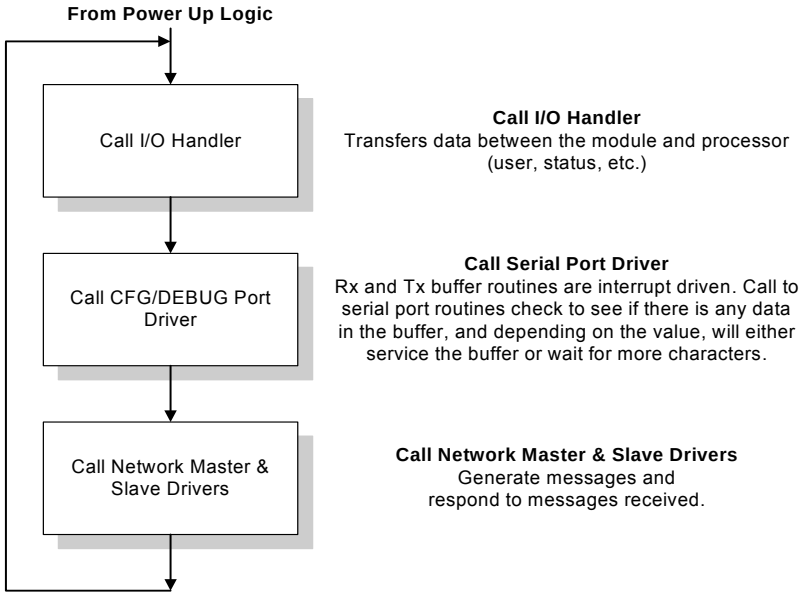
On power up the module begins performing the following logical functions:

1. Initialize hardware components
   - Initialize SLC backplane driver
   - Test and Clear all RAM
   - Reset Modbus Plus Chipset
2. Wait for Module Configuration from SLC processor
3. Initialize Module Register space
4. Initialize Modbus Plus Chipset
5. Enable Global Input task
6. Enable Global Output task
7. Enable Slave Driver
8. Enable Master Driver

After the module receives the module configuration from the processor, the Modbus Plus chipset is enabled (presuming valid configuration values were received), and begins communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

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



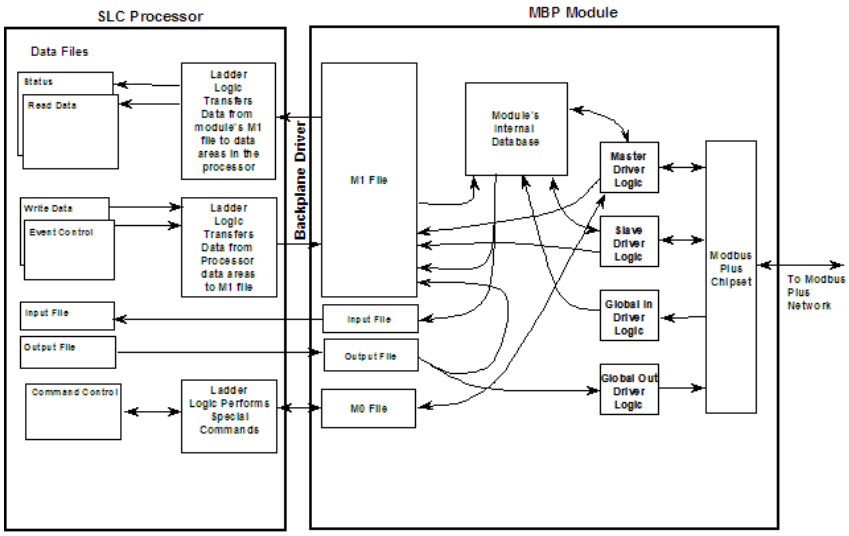
Processor Not in Run

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

Backplane Data Transfer

The MVI46-MBP module communicates directly over the SLC backplane. All data for the module is contained in the module's M0 and M1 files and the input and output files. 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 M1 file. The COP instruction can be used to move data between user data files and the module's M0 file and the M1 file and the user files. All register data travels between the module and the processor using the M1 file. The M0 file is used for special command control of the module from the ladder logic. High-speed data transfer between the processor and the module is accomplished using the module's input and output files.

The following illustration shows the data transfer method used to move data between the SLC processor, the MVI46-MBP module, and the Modbus Plus network:



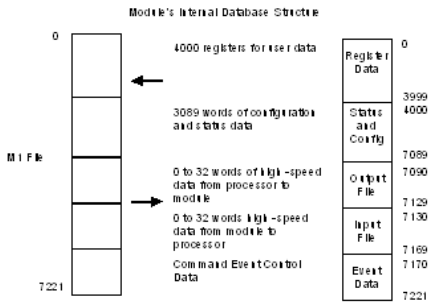
As shown in the data flow diagram, all register data transferred between the module and the processor over the backplane is through the M1 file. Ladder logic must be written in the SLC processor to interface the M1 file data with data defined in the user-defined data files in the SLC. All data used by the module is stored in its internal database. The data transferred from the module to the M1 file is updated when new data becomes available from the Modbus Plus network.

The ladder logic in the SLC can control the module by using M files for command and module control.

File	Offset	Description
M0	0	Command Control
M1	4370	Module Control

A value of 9999 in this word (4370 of the M1 file) causes the module to perform a cold-boot operation.

All data used by the module is stored in its internal database. This database is defined as a virtual Modbus data table with addresses from 0 (40001 Modbus) to 7221 (4222 Modbus). The following illustration shows the layout of the database:



This database is stored in the M1 file of the module and the internal database of the module. Data contained in this database is paged from the M1 file to the module. When data is received on the network for the database, it is copied to both the M1 file and the internal database. If this data is overwritten in the M1 file, it is also overwritten in the internal database. To move data from a user file to the M1 file, use the COP instruction. This data is then copied to the module's database. To read data from the module, use the COP instruction to copy data from the M1 file to the user data file.

Configuration Data Transfer

Writing the Configuration to the MVI46-MBP

When the module performs a restart operation, it requests configuration information from the SLC processor. This data is transferred to the module in the M1 file. When the ladder logic recognizes a value of 9000 in register 4370 of the M1 file, it should copy all configuration information into the M1 file. After the operation is complete, the ladder logic should place a value of 9001 into the control register (4370) of the M1 file. The module will recognize the configuration completion code (9001) and configure the module using the data contained in the M1 file.

The Reference chapter contains a complete listing of the configuration information of the M1 file, and design forms to aid in the module configuration process.

Reading the Configuration from the MVI46-MBP

Option "Y" (Transfer Module Cfg to Processor) when using the Debug port, allows the transfer of configuration data from the MVI module to the SLC. When the user presses the "Y" key, the module forms a block (M1:x.4370 = -9000) with the configuration. Ladder logic should copy the data from the M1 file to the SLC data file. The ladder logic should then move 0 to M1:x.4370 in order to clear the module.

Status Blocks

Status data is placed in the M1 file to determine the "health" of the module and the activity of the Modbus Plus network. The format of this data area is shown in the Reference chapter. Registers 4000 to 4369 in the M1 file contain the module's status data.

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 eight command control blocks:

- User Command Block (9002)
- Command Execution Block (9003)
- Command Enable/Disable Blocks (9010, 9011, 9012)
- M0 Pass-Through Control (9956, 9958, 9959)
- Write Configuration (9997)
- Warm Boot (9998)
- Cold Boot (9999)

User Command Block

This block is sent from the SLC processor to the module to execute up to six commands generated from the ladder logic using the M0 file. These commands are placed in the command queue and executed at a high priority in the module. The format of the block used for this control process is shown in the following table:

Offset	Description	Length
0	9002	1
1	Number of commands to add	1
2 to 11	User Command 1	10
12 to 21	User Command 2	10
22 to 31	User Command 3	10
32 to 41	User Command 4	10
42 to 51	User Command 5	10
52 to 61	User Command 6	10

Word 1 of the block sets the number of commands present in the block. This word should be set to a value from 1 to 6. The format of each command in the block is shown in the following table:

Word Offset	Parameter
0	Module's Database Register Number
1	Register Count
2	Swap Code
3	Device Index
4	Function Code
5	Register Address in Device
6	Spare
7	Spare
8	Spare
9	Spare

The definition of each parameter is that given in the command list description in this manual. After the module completes processing the command block, the M0 file is set with the following data:

Offset	Description	Length
0	Done Flag (0)	1
1	9002	1
2	Number of commands added to the queue	1

Command Execution Block

This command block inserts commands from the command list into the module's command queue. The command queue is executed at a high priority. Commands with enable codes set to zero, can be executed by the module using this feature. The format of the block sent by the processor to the module in the M0 file has the following format:

Offset	Description	Length
0	9003	1
1	Number of commands to add	1
2 to 61	List of command indexes	60

Word 1 of the block sets the number of commands listed in the block. Up to 60 commands can be sent to the command queue using this block. Words 2 to 61 contain the command indexes to be added to the queue. After the module has processed the block, the M0 file is set with the following data:

Offset	Description	Length
0	Done Flag (0)	1
1	9003	1
2	Number of commands added to the queue	1

Command Enable/Disable Blocks

Block codes 9010, 9011 and 9012 alter the enable code for a set of commands in the module's command list. Word 1 in each block defines the number of commands to be considered by the module in the list of command indexes provided. This data is passed to the module in the M0 file.

Block 9010 disables one or more commands in the command list by setting the enable code to a value of zero. The format of the block is as follows:

Offset	Description	Length
0	9010	1
1	Number of commands to disable	1
2 to 61	List of command indexes	60

Block 9011 sets the enable code for the commands in the list to a value of one. The format of the block is as follows:

Offset	Description	Length
0	9011	1
1	Number of commands to enable	1
2 to 61	List of command indexes	60

Block 9012 sets the enable code for the commands in the list to a value of two. The format of the block is as follows:

Offset	Description	Length
0	9012	1
1	Number of commands to enable	1
2 to 61	List of command indexes	60

After the command blocks are processed by the module, word zero of the M0 file will be set to a value of zero.

You can look at the enable code to determine if the enable code has changed or to see the value of the code by using the Debug port. You can restore the original command enable code forcing the module to perform a cold boot and thereby copying the new commands to the MVI46-MBP module.

M0 Pass-Through Control

M0 PASS_THROUGH CONTROL (REGISTER 100)	
Block Code	**Descriptions**
9956	Formatted pass-through block from function 6 or 16 with word data.
9958	Formatted pass-through block from function 5.
9959	Formatted pass-through block from function 15.

Formatted Pass-Through Command Blocks (Read Block)		
Offset	**Description**	**Length**
100	9956 or 9958	1
101	Number of word registers in Modbus data set	1
102	Starting address for Modbus data set	1
103 to 366	Modbus data set	264

Formatted Pass-Through Response (Write Block)		
Offset	**Description**	**Length**
100	Place a zero in this word after the SLC has processed the pass-through message block.	1

Formatted Pass-Through Command Blocks (Read Block)		
Offset	**Description**	**Length**
100	9959	1
101	Number of word registers in Modbus data set	1
102	Starting word address for Modbus data set	1
103 to 152	Modbus data set	50
153 to 202	Bit mask to use with the data set. Each bit to be considered with the data set will have a value of 1 in the mask. Bits to ignore in the data set will have a value of 0 in the mask.	50

Formatted Pass-Through Response (Write Block)		
Offset	**Description**	**Length**
100	Place a zero in this word after the SLC has processed the pass-through message block.	1

Write Configuration

This command is sent from the SLC processor to the module to force the module to write its current configuration back to the processor. This function is used when the module's configuration has been altered remotely using database write operations. The ladder logic generates the request by placing a value of 9997 in register 4370 (control register) of the M1 file. The module places all configuration data in the M1 file and places a value of 0 in the control register after completion. The ladder logic copies the configuration data in the M1 file to the user data files.

Warm Boot

This command is sent from the SLC processor to the module's control register (4370) in the M1 file by inserting a value of 9998 into the register. When the module recognizes this value in the control register it performs the warm-boot operation and sets the register to a value of zero.

Cold Boot

This command is sent from the SLC processor to the module's control register (4370) in the M1 file by inserting a value of 9999 into the register. When the module recognizes this value in the control register it performs the cold-boot operation and sets the register to a value of zero.

5.2.2 Pass-Through Control Blocks

Formatted Pass Through Control Blocks

If the port on the module is configured for the formatted pass-through mode, the module will pass blocks with identification codes of 9956 = FC16, and 9958 = FC5, and 9959 = FC15 to the processor for each received write command. Any Modbus Function 5, 15, or 16 commands will be passed from the port to the processor using this block identification number. Ladder logic must handle the receipt of all Modbus write functions to the processor and to respond as expected to commands issued by the remote Modbus Master device. The format of the formatted pass-through control block is shown in the following tables:

Function 5

Offset	Description	Length
100	9958	1
101	Number if word registers in Modbus data set	1
102	Starting Address fro Modbus data set	1
103 to 366	Modbus data set	264

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

Offset	Description	Length
100	Place a zero in this word after the SLC has processed the pass-through message block	1

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

Function 6 and 16

Offset	Description	Length
100	9956	1
101	Number if word registers in Modbus data set	1
102	Starting Address fro Modbus data set	1
103 to 366	Modbus data set	264

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

Offset	Description	Length
100	Place a zero in this word after the SLC has processed the pass-through message block	1

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

Function 15

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

Offset	Description	Length
100	9959	1
101	Number of word registers in Modbus data set	1
102	Starting word address for Modbus data set	1
103 to 152	Modbus data set	50
153 to 202	Bit mask to use with the data set. Each bit to be considered with the data set will have a value of 1 in the mask. Bits to ignore in the data set will have a value of 0 in the mask.	50

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

Offset	Description	Length
100	Place a zero in this word after the SLC has processed the pass-through message block.	1

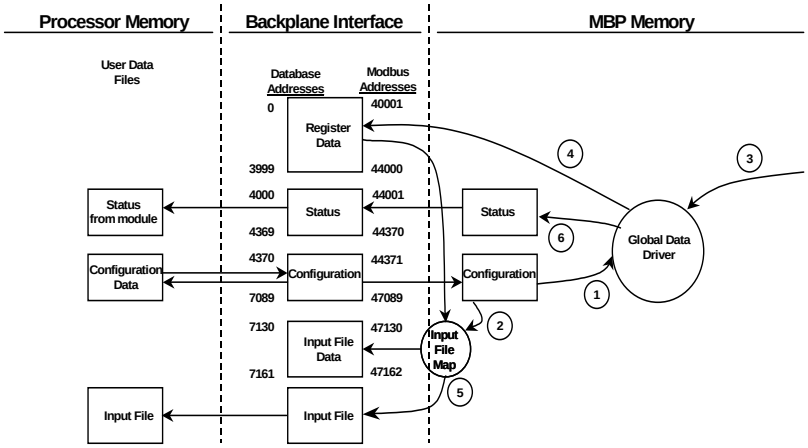
This will inform the module that the command has been processed and can be cleared from the pass-through queue.

5.2.3 Data Flow between MVI46-MBP Module and SLC Processor

The following topics describe the flow of data between the two pieces of hardware (SLC processor and MVI46-MBP module) and other nodes on the Modbus Plus network under the module's different operating modes. Note that all four modes can operate effectively simultaneously if desired. Under most likely operating cases, the Global Input and Global Output tasks operate in conjunction with either the Master or the Slave driver.

Global Data In Mode

When the Global Data In mode is operational, the MVI46-MBP module is receiving Global Input data from up to 64 other nodes on the Modbus Plus network. Each node is capable of transferring up to 32 words, and therefore, the MVI46-MBP module is capable of accepting up to 2048 words in this manner. The amount of data, and which slaves to collect it from, are all user-defined through the Device Definition File. The following flow chart and associated table show the flow of data into and out of the module.

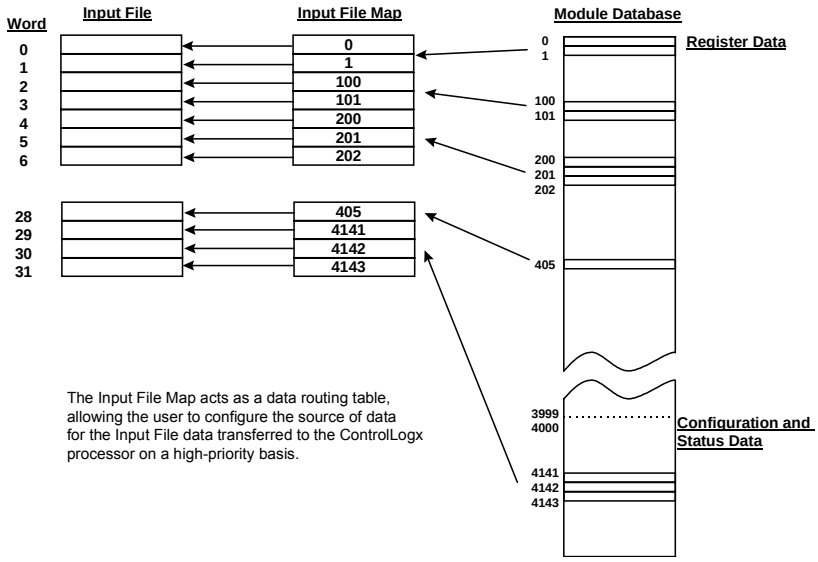


Step	Description
1	The Global In driver reads configuration data from the processor. This data includes the Device Definition File that includes the node address data, the number of Global In words and where to put this data in the module's internal database.
2	During the configuration process, the Input File Map is updated out of the configuration file. The Input File Map informs the module which data registers in the internal database to feed into the module's input image. This operational mode is independent of the Global In mode but is commonly used to transfer global input data from other nodes directly to the processor.
3	The Global In Driver monitors Global In data from other nodes on the network. If the data matches one of the node addresses in the Device Definition File and is qualified in terms of length, etc. the data is accepted.
4	After the data is accepted, the data is transferred into the module's internal database. The user via configuration in the Device Definition File determines the location of the data.
5	As data is read from the other nodes on the network into the module, an asynchronous process moves the data from the database into the module's input image. The values to be moved are user determined via configuration of the Input File Map. Up to 32 words of data can be transferred in this fashion.

Step	Description
6	Status is monitored for each device in the Device Definition File that is expected to return Global In data to the module. This status is updated on an on-going basis and is transferred to the SLC processor for processing. This data includes the node status value and a counter incremented each time global input data is received.

In order for the Global Data In mode to operate, the minimum configuration includes setting the Device Definition File and the Global Input Timeout values. If this or other data is to transfer to the SLC processor using the Input File, the Input File Map, Input File Size, and Input File Update parameters must also be set.

It is important to understand how the Input File Map determines what data is transferred from the module to the processor. The Input File Map is a 32-word data block that selects the module's internal data registers to transfer to the Input File. The Input File Map is copied to the module during module configuration. The structure of this data block is as follows:

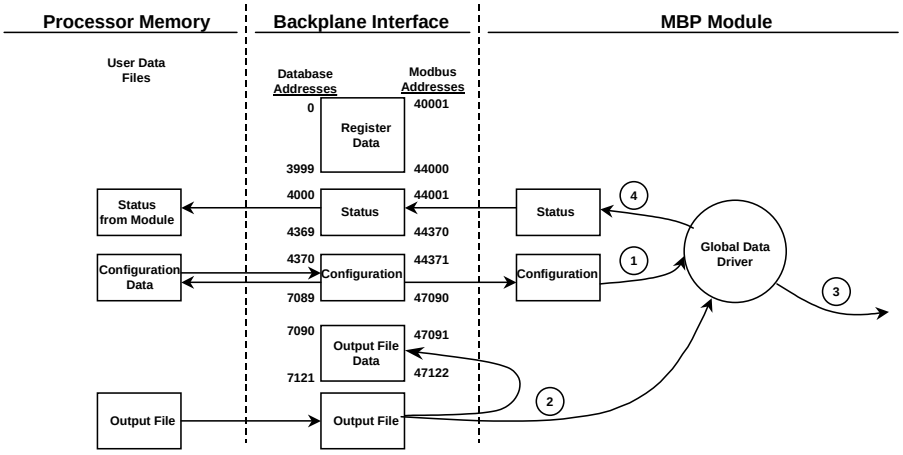


Each scan of the module's program places the data into the input image using this Input File Map to select data out of the module's database. Any data in the module's database can be assigned to the high-speed Input File data using this map.

Global Data Out Mode

When the MVI46-MBP module's Global Output capability is enabled, up to 32 words of data can be transferred onto the Modbus Plus network by the module. This data, typically reserved for high-speed data such as for application control data, is transmitted each time the module receives the network token.

The number of words transferred to the Modbus Plus network is user determined through the Module Configuration Block described in the Reference chapter. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Global Output driver reads configuration data from the SLC processor. This data consists of the number of words to be transmitted by the module each time the module has the token. In addition, timing data on the update rate for the Global Out transmission is also obtained from the configuration data.
2	The Global Out data image is updated from the processor through the module's output image. Based on the update rate configured by the user, the Global Out image in the Modbus Plus chipset will also be updated.
3	The Global Output driver in the Modbus Plus chipset will transmit the Global Out data each time the token is received by the module.
4	The Global Output driver status is updated in the module's database.

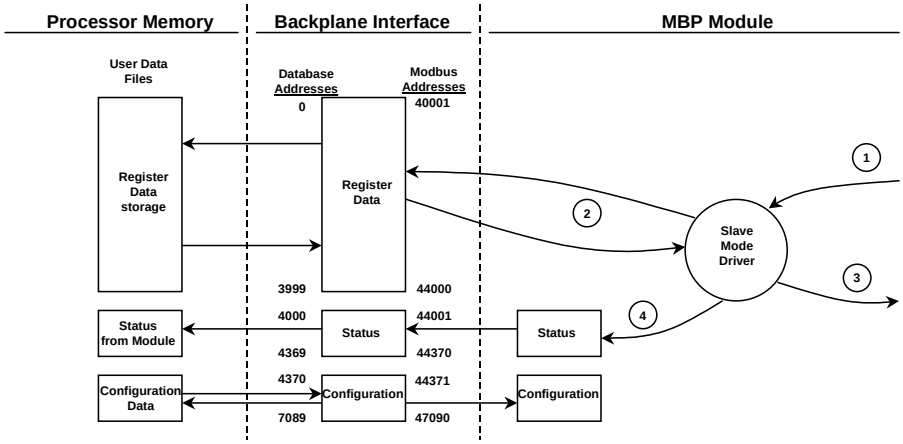
To enable the Global Output Mode, set the Global Output Length parameter to a value between 1 and 32. To disable this feature, set the parameter to a value of zero. Status information about the global output data is found in the status block transferred from the module to the SLC processor.

Slave Driver Mode

Slave Driver Mode allows the MVI46-MBP module to respond to data read and write commands issued by other nodes on the Modbus Plus network. Two aspects of the module's operation must be kept in mind when considering using this mode:

1. The module supports MSTR Type 1 and Type 2 commands issued from a Modicon processor or another device acting in a similar capacity.
2. The module is a Modbus Plus Host type of node, therefore any device wishing to read or write data from the module **must** be able to define a Data Slave Input Path in the Routing Path. The module supports all 8 Data Slave Input paths, **but a Data Slave Path of 0 (zero) will cause the command to be rejected**.

The following flow chart and associated table show the flow of data into and out of the module.

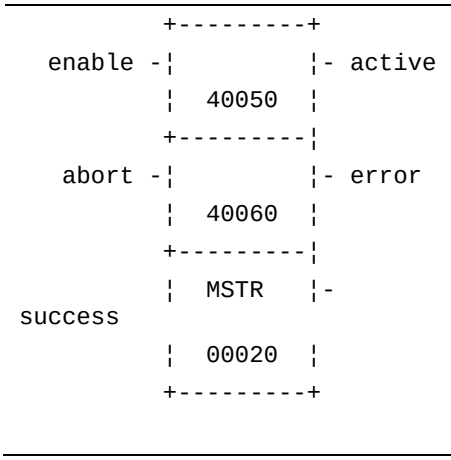


Step	Description
1	A Host device, such as a Modicon PLC or an HMI application issues a read or write command to the module's node address. The Modbus Plus chipset qualifies the message before accepting it into the module.
2	After the module accepts the command, the data is immediately transferred to or from the internal database in the module. If the command is a Read command, the data is read out of the database and appended to the response, and if the command is a Write command, the data is written directly into the database.
3	After the data processing has been completed in Step 2, the response is issued to the originating node.
4	Several counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

There are no special module configuration requirements to place the module in the Slave Operating Mode. When the module is operating in the slave mode, external devices act as masters by polling for data from the module or writing to the module. As such, the module needs to only respond to read and write commands, transferring data to/from the module's database depending on the command type.

In order for a Modicon PLC to read data from the MVI46-MBP module, a MSTR Type 2 instruction must be entered in the Modicon's ladder program. This instruction initiates a Modbus Plus network transaction between the PLC and the module. In the configuration of the command, the programmer can specifically choose the location and amount of data to be read from the module and returned to the Modicon's memory.

The following illustration shows an example configuration for a MSTR Type 2 command.

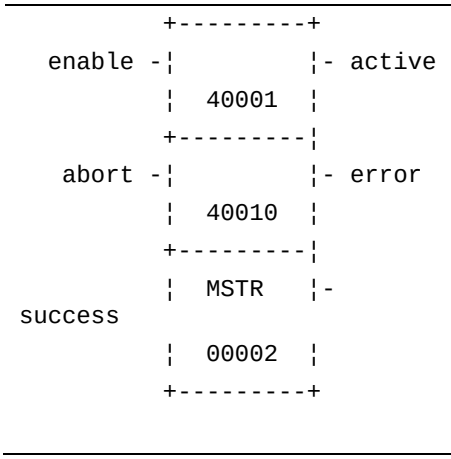
	Contents of registers in the control block 40050 = 2 Read instruction 40051 = 0 Error code 40052 = 20 Length of the read 40053 = 50 slave register to read (module address 50) 40054 = 6 MVI46-MBP Node address to retrieve data from 40055 = 1 Slave Input Path for routing 40056 = 0 Routing Address 3 40057 = 0 Routing Address 4 40058 = 0 Routing Address 5 40060 = Destination address in the Modicon PLC for the data from the MVI46-MBP module

The MSTR 2 instruction shown in the previous example reads 20 words from the MVI46-MBP module beginning at address 50 and places the data in the Modicon PLC beginning at address 40060.

Note that the Slave Input Path value must be entered in order for the command to execute successfully. Valid values are from 1 to 8. Any other values will cause the command to fail.

In order for a Modicon PLC to write data to the module, a MSTR Type 1 instruction must be entered in the Modicon's ladder program. This instruction initiates a Modbus Plus network transaction between the PLC and the module. In the configuration of the command, the programmer can specifically choose the destination address and the amount of data to be written to the module from the Modicon's memory.

The following illustration shows an example configuration for a MSTR Type 1 command.

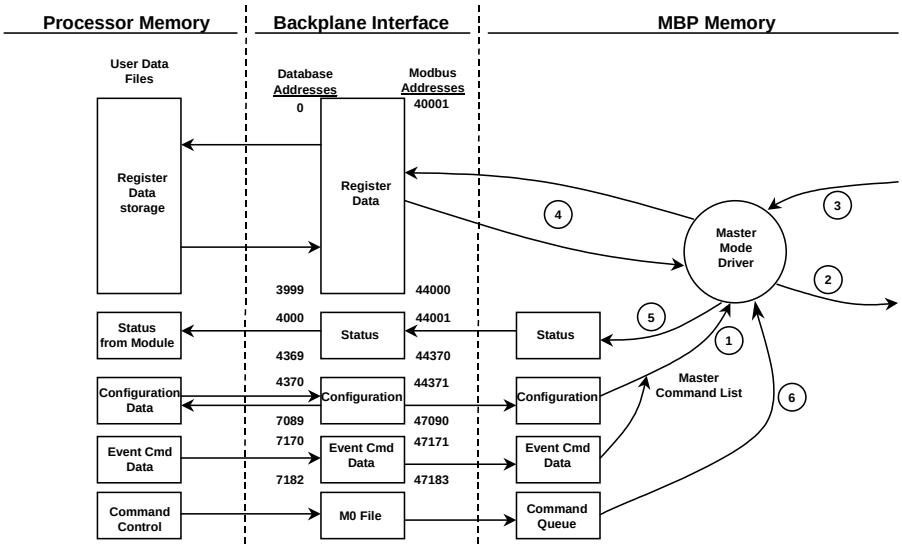
	Contents of registers in the control block 40001 = 1 Write instruction 40002 = 0 Error code 40003 = 30 Length of the write 40004 = 100 register to Write to in module 40005 = 6 MVI46-MBP Node address to write to 40006 = 1 Slave Input Path for routing 40007 = 0 Routing Address 3 40008 = 0 Routing Address 4 40009 = 0 Routing Address 5 40010 = Source of the data in the Modicon PLC to send to the MVI46-MBP module

This instruction will write 30 words (from 40010 to 40039) in the PLC to the MVI46-MBP module's database beginning at Data Register address 100.

Note that the Slave Input Path value must be entered in order for the command to execute successfully. Valid values are from 1 to 8. Any other values will cause the command to fail.

Master Driver Mode

In the Master mode, the MVI46-MBP module issues read or write commands to other devices on the Modbus Plus network. These commands are user-configured in the module via the Master Command List received from the SLC processor or using the command control blocks. Command status is returned to the processor for each individual command in the command list status block. The module emulates the MSTR Type 1 and Type 2 commands in terms of data read and write functionality. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Master driver obtains configuration data from the SLC processor. The configuration data obtained includes the number of commands, the Device Definition File, and the Master Command List. These values are used by the Master driver to determine the type of commands to issue to the other nodes on the Modbus Plus network. In addition, Event Command control bits are available to control individual commands from ladder logic.
2	After configuration, the Master driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from the module's internal database to build the command.
3	Presuming successful processing by the other nodes, responses are received into the Master driver for processing.
4	Data received from the other nodes on the network is passed into the module's internal database, assuming a read command.
5	Status is returned to the SLC processor for each command in the Master Command List.
6	Ladder logic can alter the enable code for commands or cause the master driver to execute user commands or commands in the list by placing the commands in the command queue.

In order for the MVI46-MBP module to operate in the Master Mode (actively reading/writing data with other nodes on the network), several user configurable parameters must be received from the SLC processor. The following topics detail these configuration requirements.

Device Definition File

An entry must be made in the Device Definition File for each node on the network that is to be addressed by the module's Master Command List. The entry in the Device Definition File must consist of at least the Routing Path, **including a valid Data Slave Input path if the device is not a Modicon PLC**.

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 200 individual entries, with each entry containing the information required to construct a valid command. This includes the following:

- Command enable mode (disabled, continuous or event control)
- Node Route Path: From Device Definition File (index in Device Definition File)
- Command Type: Read or Write up to 100 words per command
- Source and Destination Register Address: Determines where data will be placed and/or obtained
- Count: Select the number of words to be transferred - 1 to 100

Event Command Control

Commands can be entered in the command list to operate either continuously or under ladder logic control. The operating mode is selected when entering the command in the Master Command List.

If a command is configured as an Event Command, the module looks to the Event Command Control words received from the SLC processor to determine when to execute set in the M1 file. The following table describes the structure of the Control words:

Event Command Control

Database Address	Bit Offset for Event Control	Name	Description
7170	0 to 15	Event Commands #1 to 16	This block of data consists of 13 words of bits that are mapped to individual commands in the Master Command List. Setting a bit will trigger the corresponding command and will set a one-shot bit for the command in the module. This bit must be reset before being set again in order for the command to execute again. This data is transferred continuously to the module in the module's output image.
7171	16 to 31	Event Commands #17 to 32	
7172	32 to 47	Event Commands #33 to 48	
7173	48 to 63	Event Commands #49 to 64	
7174	64 to 79	Event Commands #65 to 80	
7175	80 to 95	Event Commands #81 to 96	
7176	96 to 111	Event Commands #97 to 112	
7177	112 to 127	Event Commands #113 to 128	
7178	128 to 143	Event Commands #129 to 144	
7179	144 to 159	Event Commands #145 to 160	
7180	160 to 175	Event Commands #161 to 176	
7181	176 to 191	Event Commands #177 to 192	
7182	192 to 199	Event Commands #193 to 200	

There is a one-to-one relationship between each bit in the table and a command in the Master Command List based on the bit position in the table. The following table shows this relationship:

Bit Position	Name
0	Event Commands #1
1	Event Commands #2
2	Event Commands #3
3	Event Commands #4
4	Event Commands #5
5	Event Commands #6
-	-
-	-
-	-
199	Event Commands #200

Execution status can be monitored in the same fashion as other commands in the Master Command List, via the Master Command Status block.

Successful Execution

If the command was successful (Status Code = 0x01), the status will be maintained as long as the Event Command bit is set. After the enable bit is cleared, the Status Code field will go to zero (0).

Unsuccessful Execution

If the command was unsuccessful (Status Code > 0x01), the status will be maintained until the command executes successfully. The ladder logic must 're-submit' the command (clear the enable bit and set it again) in order for the command to execute again.

5.2.4 Dual Port Operation

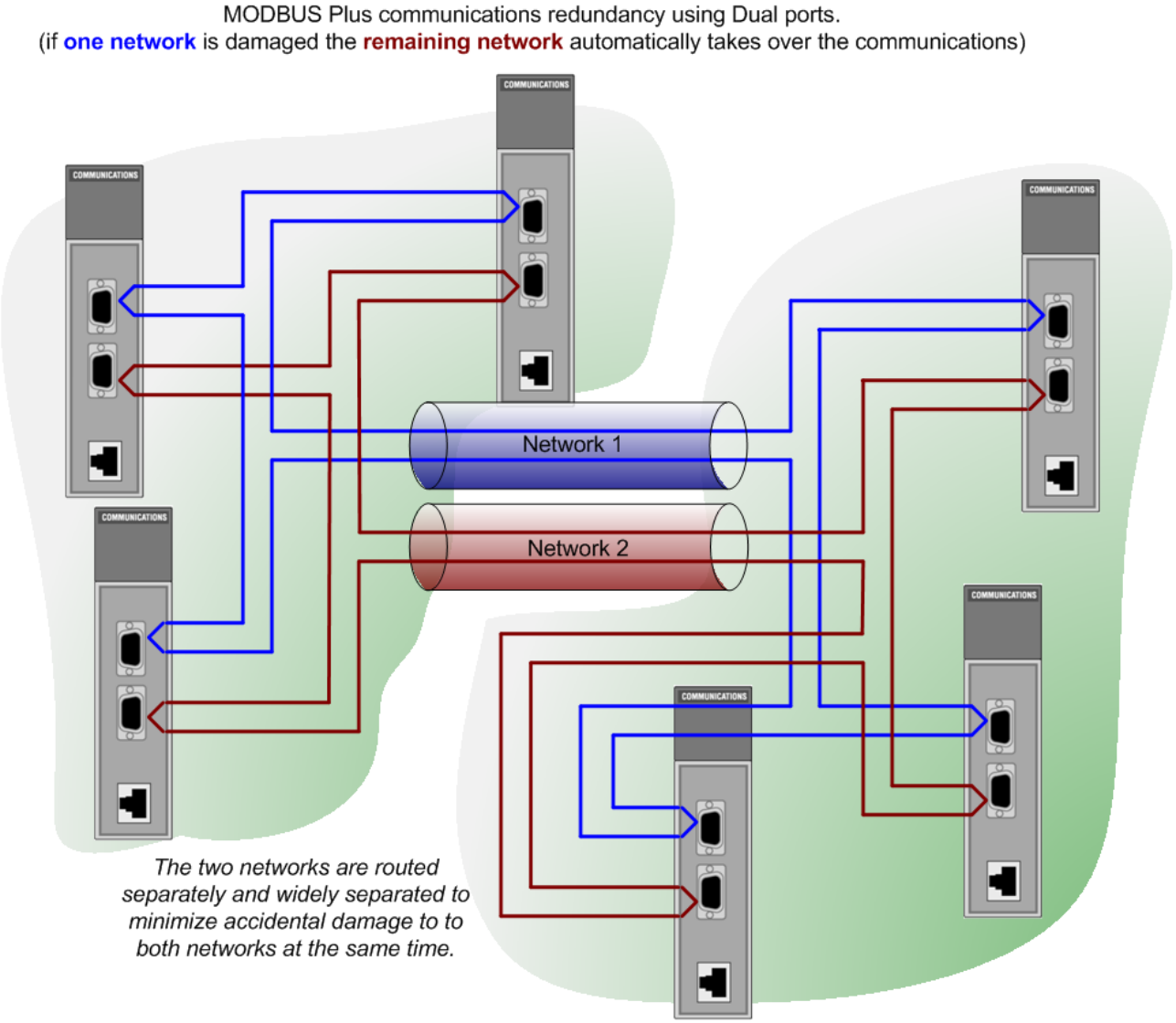
Some Modbus Plus modules are equipped with a second, redundant Modbus Plus port. A Dual Port Modbus Plus network is implemented in the following way:

- The primary network connects to Port 1 on all modules.
- The secondary (backup) network connects to Port 2 on all modules.

If the primary network connected through Port 1 fails (for example, if a cable is cut or disconnected), the network connected through Port 2 will become active, maintaining the connection between devices.

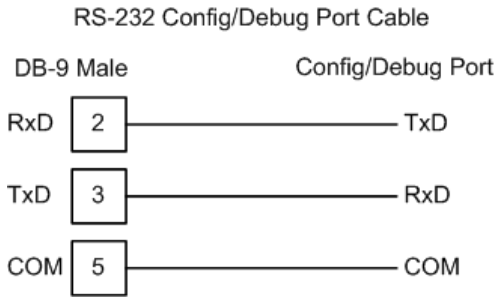
Important: All Modbus Plus nodes on a network must be equipped with dual ports in order to implement redundant port operation.

Refer to the following illustration for an example of redundant port operation on a Modbus Plus network.



5.3 RS-232 Configuration/Debug Port

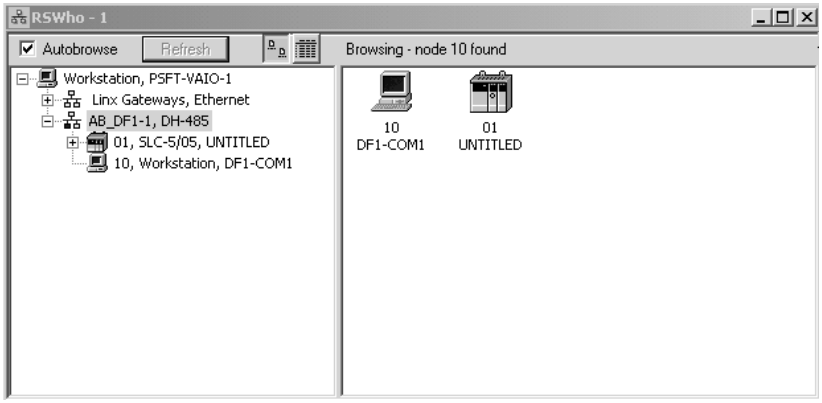
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following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



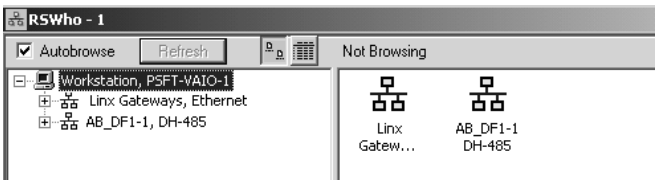
5.3.1 Disabling the RSLinx Driver for the Com Port on the PC

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

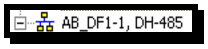
1. Open RSLinx and go to Communications>RSWho
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

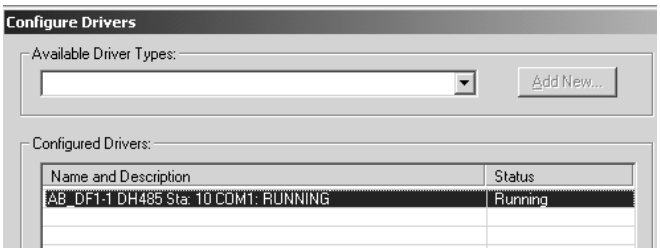


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



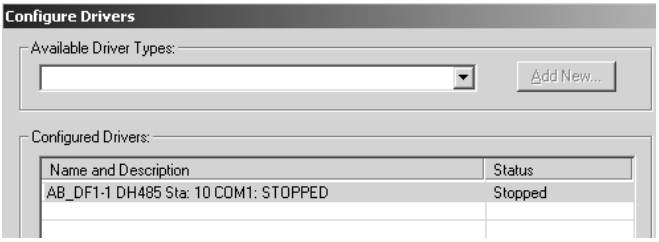
4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:



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

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



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

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

5.4 Modbus Plus Connections

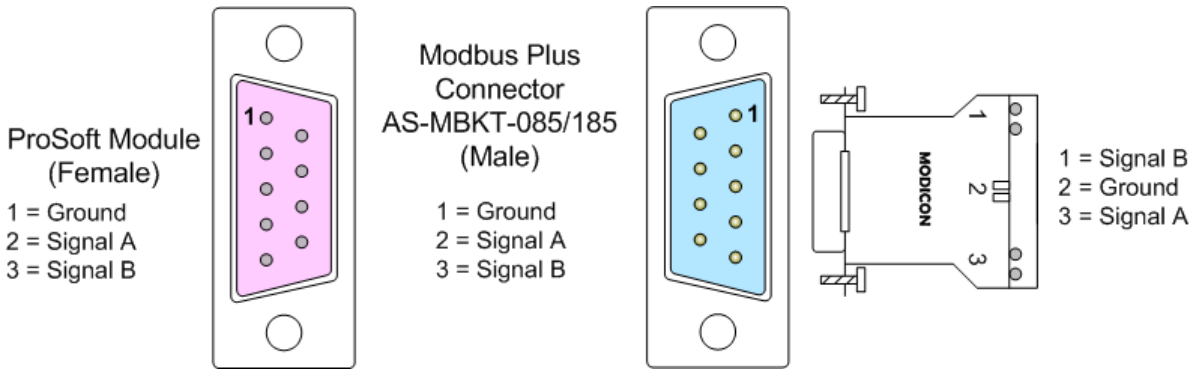
The MVI46-MBP module has one or two physical Modbus Plus connectors (depending on hardware configuration) with a DB-9 Female plug located on the front of the module.

Modicon provides two different Modbus Plus connectors to ease installation. These connectors are as follows:

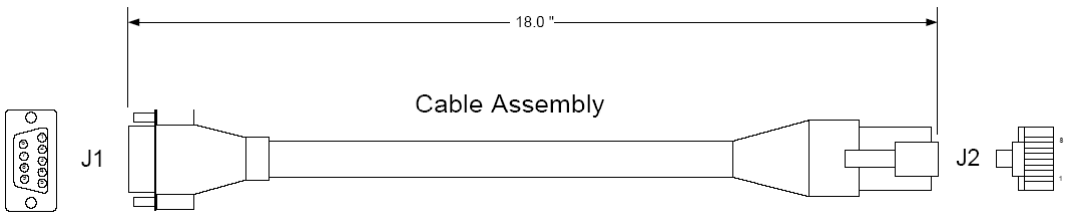
Modicon Part Number	Description
AS-MBKT-085	Inline Connector
AS-MBKT-185	Terminating Connector

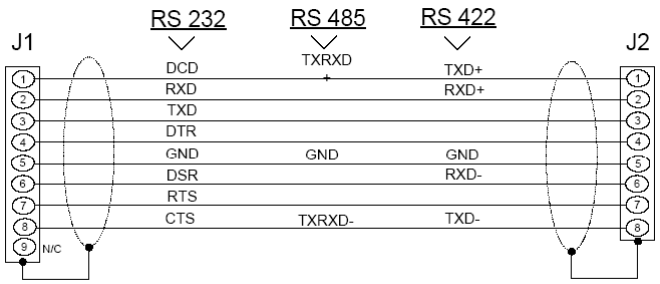
The actual cable installation and the wiring of the cable to the connectors is fully documented in the Modicon publication *Modicon Modbus Plus Network Planning and Installation Guide - Pub No. GM-Modbus Plus L-001*.

If the Modicon connectors are not available during installation, the following pin out applies to the DB-9 Modbus Plus port connections:



5.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.6 Modbus Plus Communication Port

The MVI46-MBP module has two physical Modbus Plus connectors with two DB-9 Female plugs located on the front of the module.

5.7 Database Definition

This section contains a listing of the internal database of the MVI46-MBP module. This information can be used to interface other devices to the data contained in the module.

5.7.1 Module Memory Map

Overview

Access	Module Address Range		Modbus Address Range		Description	Block
	Low	High	Low	High		Size
R/W	0	3999	40001	44000	Data Registers	4000
R	4000	4069	44001	44070	Global In Update Status	70
R	4070	4269	44071	44270	Master Command Status	200
R	4270	4299	44271	44300	Misc Module Status	30
R	4300	4369	44301	44370	Global In Update Counters	70
R/W	4370	4409	44371	44410	Module Configuration Block	40
R/W	4410	4449	44411	44450	Global In File Map	40
R/W	4450	5089	44451	45090	Device Definition	640
R/W	5090	7089	45091	47090	Master Command List	2000
O File	7090	7129	47091	47130	Global Out Image	40
I File	7130	7169	47131	47170	Global In Image	40
R	7170	7221	47171	47222	Command Event Data	52

The Data Registers area holds data collected from other nodes on the network (master read commands or global input data) 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 (master write commands).

Detailed definition of the status data area can be found in Status Data Definition (page 61). These areas include the following: Global In Update Status, Master Command Status, Misc Module Status and Global In Update Counters.

Definition of the configuration data areas can be found in Configuration Data Definition. These data areas include the following: Module Configuration Block, Global In File Map, Device Definition, Master Command List.

The Global Out Image is a data area of 40 words. Only the first 32 words are filled with data. This data is transferred from the processor to the module on each write block command and placed on the Modbus Plus network. This data area is used for high-speed data that must be passed to other nodes on the network at a high frequency.

The Global In Image is a data area of 40 words. Only the first 32 words are filled with data. This data is transferred from the module to the processor on each block read command. Data in the block is determined by the Global In File Map data set. This data area is used for high-speed data that must be passed to the processor at the highest frequency.

The Command Event Data area controls master commands under processor control. This data area is updated on each write block transfer from the processor to the module. Each bit in the data area corresponds to a command in the master command list. When the bit is set high, the command will be issued. When the bit is clear, the command will not execute. Event command control bits apply only to commands that have a value of two for their **Enable** parameter in the master command list

5.7.2 Status Data Definition

This section contains a description of the members present in the **MBPStat** object. This data is transferred from the module to the processor approximately every second.

Global In Update Status

Element	DB Address	Description
0	4000	Global In Update Stat - Device #1
1	4001	Global In Update Stat - Device #2
2	4002	Global In Update Stat - Device #3
3	4003	Global In Update Stat - Device #4
4	4004	Global In Update Stat - Device #5
5	4005	Global In Update Stat - Device #6
6	4006	Global In Update Stat - Device #7
7	4007	Global In Update Stat - Device #8
8	4008	Global In Update Stat - Device #9
9	4009	Global In Update Stat - Device #10
10	4010	Global In Update Stat - Device #11
-	-	-
-	-	-
-	-	-
60	4060	Global In Update Stat - Device #61

Element	DB Address	Description
61	4061	Global In Update Stat - Device #62
62	4062	Global In Update Stat - Device #63
63	4063	Global In Update Stat - Device #64
64	4064	Spare
65	4065	Spare
66	4066	Spare
67	4067	Spare
68	4068	Spare
69	4069	Spare

Status Code Definitions

Code	Definition
0x01	Updating: All okay
0x02	
0x04	
0x08	Global Update Timeout in milliseconds
0x10	Global Data Not Configured
0x20	
0x40	Invalid Global Data Length
0x80	

Master Command Status

Element	DB Address	Description
0	4070	Master Command Status: #1
1	4071	Master Command Status: #2
2	4072	Master Command Status: #3
3	4073	Master Command Status: #4
4	4074	Master Command Status: #5
5	4075	Master Command Status: #6
6	4076	Master Command Status: #7
7	4077	Master Command Status: #8
8	4078	Master Command Status: #9
9	4079	Master Command Status: #10
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
195	4265	Master Command Status: #196
196	4266	Master Command Status: #197
197	4267	Master Command Status: #198
198	4268	Master Command Status: #199
199	4269	Master Command Status: #200

Error Code Definition

High Byte	Low Byte	Description
0x00	0x00	Normal - Driver will reset when processing command
	0x01	Command has completed successfully
0x02		Routing Errors
	0x01	No response received - Is addressed unit online
	0x02	program access denied
	0x04	exception response received
	0x08	invalid node type in routing path
	0x10	slave rejected the Modbus command - invalid input path??
	0x20	initiated transaction forgotten by slave
	0x40	unexpected master output path received
	0x80	unexpected response received
0x04		Configuration Errors
	0x01	Invalid Configuration for Command
	0x02	Invalid Command Type - 1 = Write, 2 = Read , All other error
	0x04	
	0x08	
	0x10	
	0x20	
	0x40	
	0x80	

Miscellaneous Module Status

Element	DB Address	Description
0	4270	Module Status - Software Reset Response
1	4271	Peer Status
2	4272	Token Pass Counter
3	4273	Token Rotation Time in milliseconds
4	4274-H	Communication Failed Error Counter
4	4274-L	Communication Retry Counter
5	4275-H	No Response Received Error Counter
5	4275-L	Good Received Packet Success Counter
6	4276-H	Unexpected Path Error Counter
6	4276-L	Exception Response Received Error Counter
7	4277	Data master output path 1 & 2 counter
8	4278	Data master output path 3 & 4 counter
9	4279	Data master output path 5 & 6 counter
10	4280	Data master output path 7 & 8 counter
11	4281	Data slave output path 1 & 2 counter
12	4282	Data slave output path 3 & 4 counter
13	4283	Data slave output path 5 & 6 counter
14	4284	Data slave output path 7 & 8 counter
15	4285	Global Out Update Status
16	4286	Global Out Update Counter
17	4287	Data transfer read counter

Element	DB Address	Description
18	4288	Data transfer write counter
19	4289	Data parse counter
20	4290	Spare
21	4291	Spare
22	4292	Data transfer error counter
23	4293	Product ID
24	4294	Product ID
25	4295	Revision Level
26	4296	Batch Number
27	4297	MBP Update Processing Time
28	4298	Global In Update Time
29	4299	Global Out Update Time

Module Status Values

Normal Operation

Code	Description
0x00	Interface operational

Interface Crash Codes

Code	Description
0x01	2-second interface timeout
0x02	Bad interface opcode
0x03	Interface data error
0x04	Interface test error
0x05	Interface x-fer done error
0x06	Bad interface path
0x07	Bad interface state
0x08	Bad interface length
0x09	Global data length error
0x0A	Global data address error
0X0B	Global data not present

Fatal Crash Codes

Code	Description
0x81	PROM checksum error
0x82	Internal RAM data test error
0x83	External RAM data test error
0x84	External RAM address test error
0x85	Bad confidence test index
0x86	External Int0 event error
0x87	External Int1 event error
0x88	DMA Int0 event error
0x89	Comm. Int event error
0x8A	Xmit-no good event error
0X8B	No response timeout MAC State
0X8C	No response timeout MAC idle
0X8D	Receive OK MAC state

Code	Description
0X8E	Transmit OK MAC state
0X8F	No receive buffer free
0X90	Bad input transfer length
0X91	Reserved rev buffer error
0X92	Bad trans control state
0X93	Bad word request bit
0X94	Node queue overflow
0X95	Bad data queue error
0X96	Empty data path error
0X97	Bad path search index
0X98	Bad data slave path

Peer Status Codes

Code	Description
0	Monitor link operation
32	Normal link operation
64	Never getting token
96	Sole station
128	Duplicate station

Global Out Update Status

Code	Description
0x01	Updating all OK
0x02	Not assigned
0x04	Not assigned
0x08	Global update timeout (not transmitted in 500 milliseconds)
0x10	Global data not configured (length set to zero)
0x20	Not assigned
0x40	Invalid Global Data Length (Length > 32 words)
0x80	Not assigned

Global In Update Counters

Element	DB Address	Description
0	4300	Global In Update Counter - Device #1
1	4301	Global In Update Counter - Device #2
2	4302	Global In Update Counter - Device #3
3	4303	Global In Update Counter - Device #4
4	4304	Global In Update Counter - Device #5
5	4305	Global In Update Counter - Device #6
6	4306	Global In Update Counter - Device #7
7	4307	Global In Update Counter - Device #8
8	4308	Global In Update Counter - Device #9
9	4309	Global In Update Counter - Device #10
10	4310	Global In Update Counter - Device #11
-	-	-
-	-	-
-	-	-
60	4360	Global In Update Counter - Device #61

Element	DB Address	Description
61	4361	Global In Update Counter - Device #62
62	4362	Global In Update Counter - Device #63
63	4363	Global In Update Counter - Device #64
64	4364	Spare
65	4365	Spare
66	4366	Spare
67	4367	Spare
68	4368	Spare
69	4369	Spare

5.7.3 Configuration Data

This section contains listings of the MVI46-MBP module's database that is related to the module's configuration. This data is available to any node on the network.

General Configuration

DB Address	Modbus Address	Parameter	Description
4370	44371	Module Command Word	This register controls the module from a remote device. The value placed in the register will be processed and then set to zero after the function requested has been performed. The following function codes are currently supported: 9997=transfer module configuration to processor, 9998=warm boot module and 9999=cold boot module.
4371	44372	Local Modbus Plus Node Address (1 to 64)	This value defines the Modbus Plus Node Address for the module. A valid node address must be entered for the module to operate, and the address must be unique on the network. Valid values are between 1 and 64, inclusive.
4372	44373	Global Output File Length (0 or 1 to 32)	This value defines the number of Global Output words to be placed on the Modbus Plus network. If the value is set to 0, no global output data will be placed on the network. Values of 1 to 32 represent the number of words to be used by the module. This data must be transferred from the processor to the module.
4373	44374	Global Out Update Timing	This parameter determines the frequency of update of the Global Output Data from the processor to the Modbus Plus Chipset. This value should be set to zero to provide the fastest possible update of this data from the processor to the network. Valid values for this parameter are 0 to 65535 milliseconds.
4374	44375	Input File Length (0 or 1 to 32)	This value defines the number of words to be transferred by the module to the processor in the In File section of the input image. Valid values for this parameter are 0 to 32. For most applications, a value of 32 is used.

DB Address	Modbus Address	Parameter	Description
4375	44376	Input File Update Timing	This parameter is not used by the MVI46-MBP module as the In File data is updated with each new input image sent to the processor.
4376	44377	Global In Update Timeout	This value defines the timeout period (0 to 65535 milliseconds) for receiving Global Input Data from other nodes on the network. After the timeout period has been exceeded, the Timeout Error will be returned in the Global In Update Status data set. A value of 0 will result in the default value of 1000 milliseconds (1 second).
4377	44378	Spare	Not used
4378	44379	Spare	Not used
4379	44380	Spare	Not used
4380	44381	Number of Nodes Defined in the Device Definition File	This value specifies the number of devices to consider in the Device Definition Table. Valid range for this parameter is 0 to 64 inclusive. These node definitions are used by several modes of operation of the module.
4381	44382	Number of Master Commands	This parameter specifies the number of commands to be processed in the Master Command List. The module will only process this many commands starting with the first one in the list. If the value is set to zero, the Master Command Mode will be disabled. Valid values for this parameter are 0 to 200.
4382	44383	Number of Master Data Paths Maximum	This value defines the number of Master Data Paths that will be made available to the module's Master Mode Driver by the Modbus Plus Chipset. Valid values for this parameter are 1 to 8. A value of zero will result in the default value of 8.
4383	44384	Master Command Timeout Preset	This parameter defines the number of milliseconds (0 to 65535) to wait for a response to a master command issued by the module on the network. If a timeout condition exists for a command, it will be reflected in the Master Command List Status data area. A value of 0 will result in a value of 1000 (1 second) for the parameter.
4384	44385	Read Block Start Register	This parameter specifies the module's start address where data will be read from and transferred to the processor. Valid values for this parameter are 0 to 3999. This is a read-only value and can only be set in the processor.
4385	44386	Read Block Register Count	This parameter specifies the number of registers to transfer from the module's database to the processor. This is a read-only value and can only be set in the processor.
4386	44387	Read Block Maximum Count	This read-only value specifies the number of blocks of 200-word data are to be transferred from the module to the processor.
4387	44388	Write Block Start Register	This read-only parameter specifies the starting register in the module's database where data will be transferred from the processor to the module.

DB Address	Modbus Address	Parameter	Description
4388	44389	Write Block Register Count	This read-only parameter specifies the number of registers to set in the module's database from the processor.
4389	44390	Write Block Maximum Count	This read-only value specifies the number of blocks of 200-word data are to be transferred from the processor to the module.
4390	44391	Block Transfer Failure Count	This read-only parameter specifies the number of successive transfer errors must occur before the Modbus Plus network communications ceases. If the value is set to zero, communications will never be disabled. A value from 1 to 65535 indicates the number of successive errors that will result in the communications shutdown.
4391	44392	Spare	These registers are not assigned for use in the MVI46-MBP database
-	-	-	-
4409	44410	Spare	

Global In File Map

DB Address	Modbus Address	Parameter	Description
4410	44411	Word 0	Module's database register address for Word 0 of Input File Data.
4411	44412	Word 1	Module's database register address for Word 1 of Input File Data.
4412	44413	Word 2	Module's database register address for Word 2 of Input File Data.
4413	44414	Word 3	Module's database register address for Word 3 of Input File Data.
4414	44415	Word 4	Module's database register address for Word 4 of Input File Data.
-	-		
4439	44440	Word 29	Module's database register address for Word 29 of Input File Data.
4440	44441	Word 30	Module's database register address for Word 30 of Input File Data.
4441	44442	Word 31	Module's database register address for Word 31 of Input File Data.
4442	44443	Spare	
-	-		
4449	44450	Spare	

Device Definition

The Device Definition table consists of 64 blocks of data with each block containing the information for a single device to interface with on the network. The following table describes the starting addresses of each block of data.

DB Address	Modbus Address	Parameter	Description
4450	44451	Device Def ID 0	Start register of device definition block for device index 0
4460	44461	Device Def ID 1	Start register of device definition block for device index 1
4470	44471	Device Def ID 2	Start register of device definition block for device index 2
4480	44481	Device Def ID 3	Start register of device definition block for device index 3
4490	44491	Device Def ID 4	Start register of device definition block for device index 4
4500	44501	Device Def ID 5	Start register of device definition block for device index 5
4510	44511	Device Def ID 6	Start register of device definition block for device index 6
4520	44521	Device Def ID 7	Start register of device definition block for device index 7
4530	44531	Device Def ID 8	Start register of device definition block for device index 8
4540	44541	Device Def ID 9	Start register of device definition block for device index 9
4550	44551	Device Def ID 10	Start register of device definition block for device index 10
4560	44561	Device Def ID 11	Start register of device definition block for device index 11
4570	44571	Device Def ID 12	Start register of device definition block for device index 12
4580	44581	Device Def ID 13	Start register of device definition block for device index 13
4590	44591	Device Def ID 14	Start register of device definition block for device index 14
4600	44601	Device Def ID 15	Start register of device definition block for device index 15
4610	44611	Device Def ID 16	Start register of device definition block for device index 16
4620	44621	Device Def ID 17	Start register of device definition block for device index 17
4630	44631	Device Def ID 18	Start register of device definition block for device index 18
4640	44641	Device Def ID 19	Start register of device definition block for device index 19
4650	44651	Device Def ID 20	Start register of device definition block for device index 20
4660	44661	Device Def ID 21	Start register of device definition block for device index 21
4670	44671	Device Def ID 22	Start register of device definition block for device index 22

DB Address	Modbus Address	Parameter	Description
4680	44681	Device Def ID 23	Start register of device definition block for device index 23
4690	44691	Device Def ID 24	Start register of device definition block for device index 24
4700	44701	Device Def ID 25	Start register of device definition block for device index 25
4710	44711	Device Def ID 26	Start register of device definition block for device index 26
4720	44721	Device Def ID 27	Start register of device definition block for device index 27
4730	44731	Device Def ID 28	Start register of device definition block for device index 28
4740	44741	Device Def ID 29	Start register of device definition block for device index 29
4750	44751	Device Def ID 30	Start register of device definition block for device index 30
4760	44761	Device Def ID 31	Start register of device definition block for device index 31
4770	44771	Device Def ID 32	Start register of device definition block for device index 32
4780	44781	Device Def ID 33	Start register of device definition block for device index 33
4790	44791	Device Def ID 34	Start register of device definition block for device index 34
4800	44801	Device Def ID 35	Start register of device definition block for device index 35
4810	44811	Device Def ID 36	Start register of device definition block for device index 36
4820	44821	Device Def ID 37	Start register of device definition block for device index 37
4830	44831	Device Def ID 38	Start register of device definition block for device index 38
4840	44841	Device Def ID 39	Start register of device definition block for device index 39
4850	44851	Device Def ID 40	Start register of device definition block for device index 40
4860	44861	Device Def ID 41	Start register of device definition block for device index 41
4870	44871	Device Def ID 42	Start register of device definition block for device index 42
4880	44881	Device Def ID 43	Start register of device definition block for device index 43
4890	44891	Device Def ID 44	Start register of device definition block for device index 44
4900	44901	Device Def ID 45	Start register of device definition block for device index 45
4910	44911	Device Def ID 46	Start register of device definition block for device index 46
4920	44921	Device Def ID 47	Start register of device definition block for device index 47
4930	44931	Device Def ID 48	Start register of device definition block for device index 48

DB Address	Modbus Address	Parameter	Description
4940	44941	Device Def ID 49	Start register of device definition block for device index 49
4950	44951	Device Def ID 50	Start register of device definition block for device index 50
4960	44961	Device Def ID 51	Start register of device definition block for device index 51
4970	44971	Device Def ID 52	Start register of device definition block for device index 52
4980	44981	Device Def ID 53	Start register of device definition block for device index 53
4990	44991	Device Def ID 54	Start register of device definition block for device index 54
5000	45001	Device Def ID 55	Start register of device definition block for device index 55
5010	45011	Device Def ID 56	Start register of device definition block for device index 56
5020	45021	Device Def ID 57	Start register of device definition block for device index 57
5030	45031	Device Def ID 58	Start register of device definition block for device index 58
5040	45041	Device Def ID 59	Start register of device definition block for device index 59
5050	45051	Device Def ID 60	Start register of device definition block for device index 60
5060	45061	Device Def ID 61	Start register of device definition block for device index 61
5070	45071	Device Def ID 62	Start register of device definition block for device index 62
5080	45081	Device Def ID 63	Start register of device definition block for device index 63

Device Definition Block Format

The structure of the data in each block is described in the following table:

DB Address	Modbus Address	Parameter	Description
4450	44451	Route 1	These values determine the network route that a message will use to get from the module to a node on the network. The values are entered directly into the Modbus Plus message. Note the following: **PLCs:** When addressing these devices, the last non-zero byte in the routing list specifies the network node address. For example, to reach node 7 on the network, enter 7, 0, 0, 0, 0. **ModConnect Type Devices**: When addressing these devices (including other MVI46-MBP modules), the next-to-last non-zero values specifies the node address and the last non-zero value specifies the slave data path to use (1 to 8). If the slave path is set incorrectly the message will fail. For example, to reach slave path 1 on node 6, enter values of 6, 1, 0, 0, 0.
4451	44452	Route 2	
4452	44453	Route 3	
4453	44454	Route 4	
4454	44455	Route 5	

DB Address	Modbus Address	Parameter	Description
4455	44456	Device Type	This parameter is ignored in the current version of the software. It may be required in future releases to customize communication for non-compliant devices.
4456	44457	Global In Length	This parameter sets the length of global data accepted from the specified node. If a value of zero is entered, no global data will be accepted from this node. A value of 1 to 32 indicate that global data is to be expected from the module and the length of data to accept is set in this parameter.
4457	44458	Global In Storage Address	This parameter specifies the starting address in the module's database where the received global data will be placed. Valid locations are from 0 to 3999.
4458	44459	Spare	Not used
4459	44460	Spare	Not used

Master Command List

The next data set present in the module's database is the master command list. Each command requires a block of ten registers in the database to describe the command. There are 200 commands that can be defined in the module. Therefore, this data occupies 2000 registers. The starting locations for the commands is shown in the following table:

DB Address	Modbus Address	Parameter	Description
5090	45091	Cmd # 0	Start of command block for command # 0
5100	45101	Cmd # 1	Start of command block for command # 1
5110	45111	Cmd # 2	Start of command block for command # 2
5120	45121	Cmd # 3	Start of command block for command # 3
5130	45131	Cmd # 4	Start of command block for command # 4
5140	45141	Cmd # 5	Start of command block for command # 5
-	-		
7060	47061	Cmd # 197	Start of command block for command # 197
7070	47071	Cmd # 198	Start of command block for command # 198
7080	47081	Cmd # 199	Start of command block for command # 199

The data structure associated with each command block is described in the following table:

DB Address	Modbus Address	Parameter	Description
5090	45091	Enable	This parameter is used define if the command will be executed or will be 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 event control bit for the command is set.
5091	45092	Module's Database Register Number	This parameter specifies the starting internal register address to be associated with the command. Valid entry for this parameter is 0 to 3999.
5092	45093	Polling Interval	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.
5093	45094	Register Count	This parameter defines the number of registers to be considered by the command. Valid entry for this parameter is 1 to 100.
5094	45095	Swap Code	This parameter is not implemented in the current version of the software but will be used in the future to swap the bytes of word and double-word values when floating-point data is required.
5095	45096	Device Index	This parameter associates the command with a device defined in the device definition table (Device[ ] in the MBPlusModuleDef object). The index defined in this parameter sets the route path defined for the device in the Modbus Plus message.
5096	45097	Function Code	This parameter specifies the function to be performed by the command. Valid entries are 3= Read register data from a node and 16= Write register data to a node.
5097	45098	Register Address in Device	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.
5098	45099	Spare	Not used
5099	45100	Spare	Not used

5.7.4 Global Output Data

Global Output Data transmitted from the MVI46-MBP module to the network is located at registers 7090 to 7121 in the module's database. This data is globally broadcast to all active nodes on the network.

Destination nodes can be configured to accept or ignore incoming data from specific source nodes. Data is transferred from the processor to the module in each new output image block. The following table shows the database registers used by this data.

Global Out Image

DB Address	Modbus Address	O*x* Output Table Offset*	Parameter	Description
7090	47091	O:*x*.0	Global Out 0	Global output data for module word # 0
7091	47092	O:*x*.1	Global Out 1	Global output data for module word # 1
7092	47093	O:*x*.2	Global Out 2	Global output data for module word # 2
7093	47094	O:*x*.3	Global Out 3	Global output data for module word # 3
7094	47095	O:*x*.4	Global Out 4	Global output data for module word # 4
-	-	-		
7121	47122	O:*x*.31	Global Out 31	Global output data for module word # 31
7122	47123		Spare	Not used
7123	47124		Spare	Not used
7124	47125		Spare	Not used
7125	47126		Spare	Not used
7126	47127		Spare	Not used
7127	47128		Spare	Not used
7128	47129		Spare	Not used
7129	47130		Spare	Not used

* *x* = the slot number in the chassis where the module is installed.

5.7.5 Input File Data

The MVI46-MBP module transfers up to 32 words of data in each new input image block sent to the processor. Incoming data from each source node can be indexed into up to eight fields for delivery into separate data destinations in the receiving node. Nodes configured for global input can request up to 32 words from each node up to a maximum of 500 words.

The data used to construct is block is determined by the Input File Map data set. The data transferred to the processor is located in the module's database at registers 7130 to 7161. The following table describes the register locations for this data in the module's database:

Global In Image

DB Address	Modbus Address	I*x* Input Table Offset*	Parameter	Description
7130	47131	I:*x*.0	In File Word 0	Data from Input File Map index 0 for word 0
7131	47132	I:*x*.1	In File Word 1	Data from Input File Map index 1 for word 1
7132	47133	I:*x*.2	In File Word 2	Data from Input File Map index 2 for word 2
7133	47134	I:*x*.3	In File Word 3	Data from Input File Map index 3 for word 3
-	-	-		
7161	47162	I:*x*.31	In File Word 31	Data from Input File Map index 31 for word 31
7162	47163		Spare	Not used
7163	47164		Spare	Not used
7164	47165		Spare	Not used
7165	47166		Spare	Not used
7166	47167		Spare	Not used
7167	47168		Spare	Not used
7168	47169		Spare	Not used
7169	47170		Spare	Not used

* *x* = the slot number in the chassis where the module is installed.

5.7.6 Command Event Data

The command event data controls the master command list. If a command is set to be controlled by events, then the status of the bits contained in this data set enable the command for a single instance. Each bit in the event data set is associated with an individual command. When the bit is set, the command will be issued once. The bit must be cleared and set again for the command to be issued again. The first thirteen words of data area associated with the event command data set contain these control bits. The remainder of the data area holds data associated with each event. The following table describes the event data area:

Command Event Control

DB Address	Modbus Address	Parameter	Description
7170	47171	Event Control Word #0	Event control bits for commands 0 to 15
7171	47172	Event Control Word #1	Event control bits for commands 16 to 31
7172	47173	Event Control Word #2	Event control bits for commands 32 to 47
7173	47174	Event Control Word #3	Event control bits for commands 48 to 63
7174	47175	Event Control Word #4	Event control bits for commands 64 to 79
7175	47176	Event Control Word #5	Event control bits for commands 80 to 95
7176	47177	Event Control Word #6	Event control bits for commands 96 to 111
7177	47178	Event Control Word #7	Event control bits for commands 112 to 127
7178	47179	Event Control Word #8	Event control bits for commands 128 to 143
7179	47180	Event Control Word #9	Event control bits for commands 144 to 159
7180	47181	Event Control Word #10	Event control bits for commands 160 to 175

DB Address	Modbus Address	Parameter	Description
7181	47182	Event Control Word #11	Event control bits for commands 176 to 191
7182	47183	Event Control Word #12	Event control bits for commands 192 to 199
7183	47184	Event Done Word #0	Event done bits for commands 0 to 15
7184	47185	Event Done Word #1	Event done bits for commands 16 to 31
7185	47186	Event Done Word #2	Event done bits for commands 32 to 47
7186	47187	Event Done Word #3	Event done bits for commands 48 to 63
7187	47188	Event Done Word #4	Event done bits for commands 64 to 79
7188	47189	Event Done Word #5	Event done bits for commands 80 to 95
7189	47190	Event Done Word #6	Event done bits for commands 96 to 111
7190	47191	Event Done Word #7	Event done bits for commands 112 to 127
7191	47192	Event Done Word #8	Event done bits for commands 128 to 143
7192	47193	Event Done Word #9	Event done bits for commands 144 to 159
7193	47194	Event Done Word #10	Event done bits for commands 160 to 175
7194	47195	Event Done Word #11	Event done bits for commands 176 to 191
7195	47196	Event Done Word #12	Event done bits for commands 192 to 199
7196	47197	Event Err Word #0	Event error bits for commands 0 to 15
7197	47198	Event Err Word #1	Event error bits for commands 16 to 31
7198	47199	Event Err Word #2	Event error bits for commands 32 to 47
7199	47200	Event Err Word #3	Event error bits for commands 48 to 63
7200	47201	Event Err Word #4	Event error bits for commands 64 to 79
7201	47202	Event Err Word #5	Event error bits for commands 80 to 95
7202	47203	Event Err Word #6	Event error bits for commands 96 to 111
7203	47204	Event Err Word #7	Event error bits for commands 112 to 127
7204	47205	Event Err Word #8	Event error bits for commands 128 to 143
7205	47206	Event Err Word #9	Event error bits for commands 144 to 159
7206	47207	Event Err Word #10	Event error bits for commands 160 to 175
7207	47208	Event Err Word #11	Event error bits for commands 176 to 191
7208	47209	Event Err Word #12	Event error bits for commands 192 to 199
7209	47210	Event One Shot #0	Event one shot bits for commands 0 to 15
7210	47211	Event One Shot #1	Event one shot bits for commands 16 to 31
7211	47212	Event One Shot #2	Event one shot bits for commands 32 to 47
7212	47213	Event One Shot #3	Event one shot bits for commands 48 to 63
7213	47214	Event One Shot #4	Event one shot bits for commands 64 to 79
7214	47215	Event One Shot #5	Event one shot bits for commands 80 to 95
7215	47216	Event One Shot #6	Event one shot bits for commands 96 to 111
7216	47217	Event One Shot #7	Event one shot bits for commands 112 to 127
7217	47218	Event One Shot #8	Event one shot bits for commands 128 to 143
7218	47219	Event One Shot #9	Event one shot bits for commands 144 to 159
7219	47220	Event One Shot #10	Event one shot bits for commands 160 to 175
7220	47221	Event One Shot #11	Event one shot bits for commands 176 to 191
7221	47222	Event One Shot #12	Event one shot bits for commands 192 to 199

5.8 Configuration Forms

This section contains forms that aid in the design of applications using the MVI46-MBP module:

5.8.1 Module Configuration

Global Input File Map

N20 Word	Description	Value	Word	Description	Value
0	Module Command Word		40	Global In Source Addr	
1	Local Modbus Plus Node Address (1 to 64)		41	Global In Source Addr	
2	Global Output File Length (0 or 1 to 32)		42	Global In Source Addr	
3	Global Out Update Timing		43	Global In Source Addr	
4	Input File Length (0 or 1 to 32)		44	Global In Source Addr	
5	Input File Update Timing		45	Global In Source Addr	
6	Global In Update Timeout		46	Global In Source Addr	
7	Spare		47	Global In Source Addr	
8	Spare		48	Global In Source Addr	
9	Spare		49	Global In Source Addr	
10	Number of Nodes Defined in the Device Definition File		50	Global In Source Addr	
11	Number of Master Commands		51	Global In Source Addr	
12	Number of Master Data Paths Maximum		52	Global In Source Addr	
13	Master Command Timeout Preset		53	Global In Source Addr	
14	Spare		54	Global In Source Addr	
15	Spare		55	Global In Source Addr	
16	Spare		56	Global In Source Addr	
17	Spare		57	Global In Source Addr	
18	Spare		58	Global In Source Addr	
19	Spare		59	Global In Source Addr	
20	Block Transfer Failure Count		60	Global In Source Addr	
21	Spare		61	Global In Source Addr	
22	Spare		62	Global In Source Addr	
23	Spare		63	Global In Source Addr	
24	Spare		64	Global In Source Addr	
25	Spare		65	Global In Source Addr	
26	Spare		66	Global In Source Addr	
27	Spare		67	Global In Source Addr	

N20 Word	Description	Value	Word	Description	Value
28	Spare		68	Global In Source Addr	
29	Spare		69	Global In Source Addr	
30	Spare		70	Global In Source Addr	
31	Spare		71	Global In Source Addr	
32	Spare		72	Not Used	
33	Spare		73	Not Used	
34	Spare		74	Not Used	
35	Spare		75	Not Used	
36	Spare		76	Not Used	
37	Spare		77	Not Used	
38	Spare		78	Not Used	
39	Spare		79	Not Used	

	Module Command Word
	9000: Module Configuration Request
	9001: Module Configuration Ready
	9997: Read Configuration from M1 File
	9998: Warm Boot Module
	9999: Cold Boot Module

Device Definition Configuration

N21		0	1	2	3	4	5	6	7	8	9
SLC Address	Device Number	Route 1	Route 2	Route 3	Route 4	Route 5	Device Type	Global Length	Global Dest	Not Used	Not Used
0 to 9	0										
10 to 19	1										
20 to 29	2										
30 to 39	3										
40 to 49	4										
50 to 59	5										
60 to 69	6										
70 to 79	7										
80 to 89	8										
90 to 99	9										
100 to 109	10										
110 to 119	11										
120 to 129	12										
130 to 139	13										
140 to 149	14										
150 to 159	15										
160 to 169	16										
170 to 179	17										

N21		0	1	2	3	4	5	6	7	8	9
SLC Address	Device Number	Route 1	Route 2	Route 3	Route 4	Route 5	Device Type	Global Length	Global Dest	Not Used	Not Used
180 to 189	18										
190 to 199	19										
200 to 209	20										
210 to 219	21										
220 to 229	22										
230 to 239	23										
240 to 249	24										

N22		0	1	2	3	4	5	6	7	8	9
SLC Address	Device Number	Route 1	Route 2	Route 3	Route 4	Route 5	Device Type	Global Length	Global Dest	Not Used	Not Used
0 to 9	25										
10 to 19	26										
20 to 29	27										
30 to 39	28										
40 to 49	29										
50 to 59	30										
60 to 69	31										
70 to 79	32										
80 to 89	33										
90 to 99	34										
100 to 109	35										
110 to 119	36										
120 to 129	37										
130 to 139	38										
140 to 149	39										
150 to 159	40										
160 to 169	41										
170 to 179	42										
180 to 189	43										
190 to 199	44										
200 to 209	45										
210 to 219	46										
220 to 229	47										
230 to 239	48										
240 to 249	49										

N23		0	1	2	3	4	5	6	7	8	9
SLC Address	**Device Number**	**Route 1**	**Route 2**	**Route 3**	**Route 4**	**Route 5**	**Device Type**	**Global Length**	**Global Dest**	**Not Used**	**Not Used**
0 to 9	50										
10 to 19	51										
20 to 29	52										
30 to 39	53										
40 to 49	54										
50 to 59	55										
60 to 69	56										
70 to 79	57										
80 to 89	58										
90 to 99	59										
100 to 109	60										
110 to 119	61										
120 to 129	62										
130 to 139	63										

Master Command List

N24		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	0										
10 to 19	1										
20 to 29	2										
30 to 39	3										
40 to 49	4										
50 to 59	5										
60 to 69	6										
70 to 79	7										
80 to 89	8										
90 to 99	9										
100 to 109	10										
110 to 119	11										
120 to 129	12										
130 to 139	13										
140 to 149	14										
150 to 159	15										
160 to 169	16										
170 to 179	17										
180 to 189	18										
190 to 199	19										
200 to 209	20										

N24		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
210 to 219	21										
220 to 229	22										
230 to 239	23										
240 to 249	24										

N25		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	25										
10 to 19	26										
20 to 29	27										
30 to 39	28										
40 to 49	29										
50 to 59	30										
60 to 69	31										
70 to 79	32										
80 to 89	33										
90 to 99	34										
100 to 109	35										
110 to 119	36										
120 to 129	37										
130 to 139	38										
140 to 149	39										
150 to 159	40										
160 to 169	41										
170 to 179	42										
180 to 189	43										
190 to 199	44										
200 to 209	45										
210 to 219	46										
220 to 229	47										
230 to 239	48										
240 to 249	49										

N26		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	50										
10 to 19	51										
20 to 29	52										
30 to 39	53										
40 to 49	54										
50 to 59	55										
60 to 69	56										
70 to 79	57										
80 to 89	58										
90 to 99	59										
100 to 109	60										
110 to 119	61										
120 to 129	62										
130 to 139	63										
140 to 149	64										
150 to 159	65										
160 to 169	66										
170 to 179	67										
180 to 189	68										
190 to 199	69										
200 to 209	70										
210 to 219	71										
220 to 229	72										
230 to 239	73										
240 to 249	74										

N27		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	75										
10 to 19	76										
20 to 29	77										
30 to 39	78										
40 to 49	79										
50 to 59	80										
60 to 69	81										
70 to 79	82										
80 to 89	83										
90 to 99	84										
100 to 109	85										

N27		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
110 to 119	86										
120 to 129	87										
130 to 139	88										
140 to 149	89										
150 to 159	90										
160 to 169	91										
170 to 179	92										
180 to 189	93										
190 to 199	94										
200 to 209	95										
210 to 219	96										
220 to 229	97										
230 to 239	98										
240 to 249	99										

N28		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	100										
10 to 19	101										
20 to 29	102										
30 to 39	103										
40 to 49	104										
50 to 59	105										
60 to 69	106										
70 to 79	107										
80 to 89	108										
90 to 99	109										
100 to 109	110										
110 to 119	111										
120 to 129	112										
130 to 139	113										
140 to 149	114										
150 to 159	115										
160 to 169	116										
170 to 179	117										
180 to 189	118										
190 to 199	119										
200 to 209	120										
210 to 219	121										

N28		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
220 to 229	122										
230 to 239	123										
240 to 249	124										

N29		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	125										
10 to 19	126										
20 to 29	127										
30 to 39	128										
40 to 49	129										
50 to 59	130										
60 to 69	131										
70 to 79	132										
80 to 89	133										
90 to 99	134										
100 to 109	135										
110 to 119	136										
120 to 129	137										
130 to 139	138										
140 to 149	139										
150 to 159	140										
160 to 169	141										
170 to 179	142										
180 to 189	143										
190 to 199	144										
200 to 209	145										
210 to 219	146										
220 to 229	147										
230 to 239	148										
240 to 249	149										

N30		0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	150										
10 to 19	151										
20 to 29	152										
30 to 39	153										
40 to 49	154										

N30	0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
50 to 59	155										
60 to 69	156										
70 to 79	157										
80 to 89	158										
90 to 99	159										
100 to 109	160										
110 to 119	161										
120 to 129	162										
130 to 139	163										
140 to 149	164										
150 to 159	165										
160 to 169	166										
170 to 179	167										
180 to 189	168										
190 to 199	169										
200 to 209	170										
210 to 219	171										
220 to 229	172										
230 to 239	173										
240 to 249	174										

N31	0	1	2	3	4	5	6	7	8	9
SLC Address	**Cmd**	**Enable Code**	**DB Address**	**Poll Interval**	**Reg Count**	**Swap Code**	**Device Index**	**Func Code**	**Dev Reg**	**Spare**	**Spare**
0 to 9	175										
10 to 19	176										
20 to 29	177										
30 to 39	178										
40 to 49	179										
50 to 59	180										
60 to 69	181										
70 to 79	182										
80 to 89	183										
90 to 99	184										
100 to 109	185										
110 to 119	186										
120 to 129	187										
130 to 139	188										
140 to 149	189										
150 to 159	190										

N31		0	1	2	3	4	5	6	7	8	9
SLC Address	Cmd	Enable Code	DB Address	Poll Interval	Reg Count	Swap Code	Device Index	Func Code	Dev Reg	Spare	Spare
160 to 169	191										
170 to 179	192										
180 to 189	193										
190 to 199	194										
200 to 209	195										
210 to 219	196										
220 to 229	197										
230 to 239	198										
240 to 249	199										

User Command List

		0	1	2	3	4	5	6	7	8	9
SLC Address	Cmd	DB Reg	Reg Count	Swap Code	Dev Index	Func Code	Dev Reg	Spare	Spare	Spare	Spare
32											
0 to 9	0										
10 to 19	1										
20 to 29	2										
30 to 39	3										
40 to 49	4										
50 to 59	5										
60 to 69	6										
70 to 79	7										
80 to 89	8										
90 to 99	9										
100 to 109	10										
110 to 119	11										
120 to 129	12										
130 to 139	13										
140 to 149	14										
150 to 159	15										
160 to 169	16										
170 to 179	17										
180 to 189	18										
190 to 199	19										
200 to 209	20										
210 to 219	21										
220 to 229	22										
230 to 239	23										
240 to 249	24										

5.9 Frequently Asked Questions

5.9.1 How is data transferred between the SLC and the MVI46-MBP?

The transfer of data is performed using the module M0 and M1 files across the backplane. So the M1 file is a copy of the module internal database and the M0 file is used for module control from ladder logic.

5.9.2 How many registers can be used for user data?

The first 4000 registers (0 to 3999) can be used to store user data.

5.9.3 How is new configuration data written to the MVI46-MBP module?

The SLC data files should be used to store the configuration data. When the module performs a restart operation (either through a cold boot or warm boot) it will request configuration data to the SLC by moving a value of 9000 to register 4370 of M1 file. When this happens, the ladder logic should copy the configuration data to the module (general configuration, global in file map, device definition and master command list). After the configuration data is copied, the ladder logic should place a value of 9001 to register 4370 of M1 file in order to flag the end of the operation.

5.9.4 How is the current module configuration copied to the SLC memory?

The user should select option 'Y' (Transfer module cfg to processor) using the debug terminal to initiate the transfer. At this point, the module sets a value of -9000 to word 4370 of the M1 file. Ladder logic must recognize this value and the move the configuration data from the M1 file to the SLC data file (general configuration, global in file map, device definition and master command list). The order of the registers is the same as defined the Reference chapter - Memory Definition. After the data is copied to the SLC memory, the ladder logic must move a value of 0 to word 4370 of the M1 file in order to flag the end of the operation.

5.9.5 Why do I get an error code 3 every time I try to send configuration data from the module to the processor and no data is moved?

Look at the ladder logic and make sure the ladder logic moves a value of 0 to word 4370 of the M1 file in order to flag the end of the operation.

5.9.6 Is it possible to create and send commands from the ladder logic?

Yes. You can create your own commands in the SLC memory and send these commands using block 9002. Up to 6 commands can be sent at each scan using block 9902.

5.9.7 Is it possible to disable and enable commands in ladder?

Yes. The user can disable or enable commands that are configured in the Master Command List using ladder logic. Block 9010 disable commands (enable code =0), block 9011 enable commands (enable code =1) and block 9012 enable commands (enable code = 2). Up to 60 commands may be enabled/disabled using these blocks.

Another way to enable commands is by using the ladder logic to send a command execution block (block 9003). This block puts commands that are configured in the Master Command List on a high priority basis in the command queue even if these commands are disabled.

Refer to the user manual for information on how to use blocks 9010, 9011, 9012 and 9003.

5.9.8 After I disable (or enable) a specific command, how do I check the status of the command?

Use the debug port to view the Master Command List to determine if the command is currently disabled or enabled.

5.9.9 What happens if the MVI46-MBP module performs a COLDBOOT or WARMBOOT after commands have been disabled by the ladder?

The new ENABLE code for each command will be sent to the module.

5.9.10 Does the MVI46-MBP operate as a Master and Slave simultaneously?

Yes. If the module starts receiving commands on the MBP port, it starts to operate as a Slave. If the MVI46-MBP sends commands, it operates as a Master. Therefore, no specific configuration is required to set the module as a Master or Slave.

6 Support, Service & Warranty

In This Chapter

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

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

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

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

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

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

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

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

6.1 How to Contact Us: Sales and Support

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

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

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

Asia Pacific Sales (office in Malaysia)

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

Asia Pacific Support (office in China)

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

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

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

Brasil (office in Sao Paulo)

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

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns

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

The following policy applies for Non-Warranty Credit Returns:

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

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

6.3 Procedures for Return of Units Under Warranty

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

a) If no defect is found, a credit is issued.

b) If a defect is found and is determined to be customer generated or if the defect is otherwise not covered by ProSoft's Warranty, or if the module is not repairable, a credit is not issued and payment of the replacement module is due.

6.4 Procedures for Return of Units Out of Warranty

1 Customer sends unit in for evaluation.

2 If no defect is found, Customer will be charged the equivalent of US $100 plus shipping, duties and taxes that may apply. A new Purchase Order will be required for this evaluation fee.

If the unit is repaired the charge to the Customer will be 30%* of the list price plus any shipping, duties and taxes that may apply. A new Purchase Order will be required for a product repair.

3 For an immediate exchange, a new module may be purchased and sent to Customer while repair work is being performed. Credit for purchase of the new module will be issued when the new module is returned in accordance with ProSoft's shipping instructions and subject to ProSoft's policy on non-warranty returns. This is in addition to charges for repair of the old module and any associated charges to Customer.

4 If, upon contacting ProSoft Customer Service, the Customer is informed that unit is believed to be unrepairable, the Customer may choose to send unit in for evaluation to determine if the repair can be made. Customer will pay shipping, duties and taxes that may apply. If unit cannot be repaired, the Customer may purchase a new unit.

6.4.1 Un-repairable Units

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

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

6.4.2 Purchasing Warranty Extension

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

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

6.5 LIMITED WARRANTY

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

6.5.1 What Is Covered By This Warranty

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

B *Warranty On Services*: Material and labor used by ProSoft to repair a verified malfunction or defect are warranted on the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

C The Warranty Period for RadioLinx accessories (such as antennas, cables, brackets, etc.) are the same as for RadioLinx modules, that is, three years from the date of shipment.

6.5.2 What Is Not Covered By This Warranty

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

B With the exception of RadioLinx accessories referenced in paragraph 1(c) this Warranty does not cover any product, components, or parts not manufactured by ProSoft.

C This Warranty also does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3 programming languages, or "C") not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges outside of the limits indicated on the product specifications; or (viii) disasters such as fire, flood, earthquake, wind or lightning.

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

6.5.3 DISCLAIMER REGARDING HIGH RISK ACTIVITIES

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

6.5.4 DISCLAIMER OF ALL OTHER WARRANTIES

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

6.5.5 LIMITATION OF REMEDIES**

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

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

6.5.6 Time Limit for Bringing Suit

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

6.5.7 No Other Warranties

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

6.5.8 Intellectual Property

A Any documentation included with Product purchased from ProSoft is protected by copyright and may not be photocopied or reproduced in any form without prior written consent from ProSoft.

B ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.

C Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.

D Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.

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

6.5.9 Additional Restrictions Relating To Software And Other Intellectual Property

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

6.5.10 Allocation of risks

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

6.5.11 Controlling Law and Severability

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

Index

Y