ProSoft®
TECHNOLOGY

Where Automation Connects.

MVI56-MBP

ControlLogix Platform

Modbus Plus Communication Module

June 20, 2018

USER MANUAL

Your Feedback Please

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

How to Contact Us

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

MVI56-MBP User Manual
Rev 1.0.0

June 20, 2018

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

Important Installation Instructions

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

WARNING - EXPLOSION HAZARD **-** SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;

WARNING - EXPLOSION HAZARD **-** WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES

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

Class 2 Power

MVI (Multi Vendor Interface) Modules

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

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

Warnings

North America Warnings

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

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

ATEX Warnings and Conditions of Safe Usage

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

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

Battery Life Advisory

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

Note: The battery is not user replaceable.

Markings

Electrical Ratings

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

Label Markings

ATEX

II 3 G

EEx nA IIC T6

0°C <= Ta <= 60°C

cULus

E183151

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

T6

-30°C <= Ta <= 60°C

Agency Approvals and Certifications

Agency	Applicable Standard
RoHS	
CE	EMC-EN61326-1:2006; EN61000-6-4:2007
ATEX	EN60079-15:2003
cULus	UL508; UL1604; CSA 22.2 No. 142 & 213
CB Safety	CA/10533/CSA IEC 61010-1 Ed.2; CB 243333-2056722 (2090408)
GOST-R	EN 61010
CSA	EN 61010
Korea KCC	KCC-REM-PFT-MVI56-AFC

243333

ME06

E183151

Contents

1 Start Here

In This Chapter

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

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

1.1 System Requirements

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

- Rockwell Automation ControlLogix™ processor, with compatible power supply and one free slot in the rack, for the MVI56-MBP module. The module requires 800 mA of available power.
- Rockwell Automation RSLogix 5000 programming software version 2.51 or higher
- Rockwell Automation RSLinx communication software
- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
    - Microsoft Windows 10
    - Microsoft Windows 7 Professional (32-or 64-bit)
    - Microsoft Windows XP Professional with Service Pack 1 or 2
    - Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3
    - Microsoft Windows Server 2003
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- ProSoft Configuration Builder, HyperTerminal or other terminal emulator program.

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

1.2 Package Contents

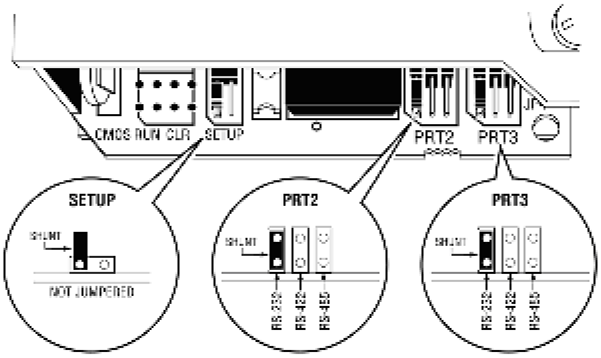
The following components are included with your MVI56-MBP module, and are all required for installation and configuration.

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

Qty.	Part Name	Part Number	Part Description
1	MVI56-MBP Module	MVI56-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

1.3 Setting Jumpers

The following illustration shows the MVI56-MBP jumper configuration:



Not used **Not used**

Do not change the position of the Setup jumper unless instructed to do so by ProSoft Technical Support.

1.4 Installing the Module in the Rack

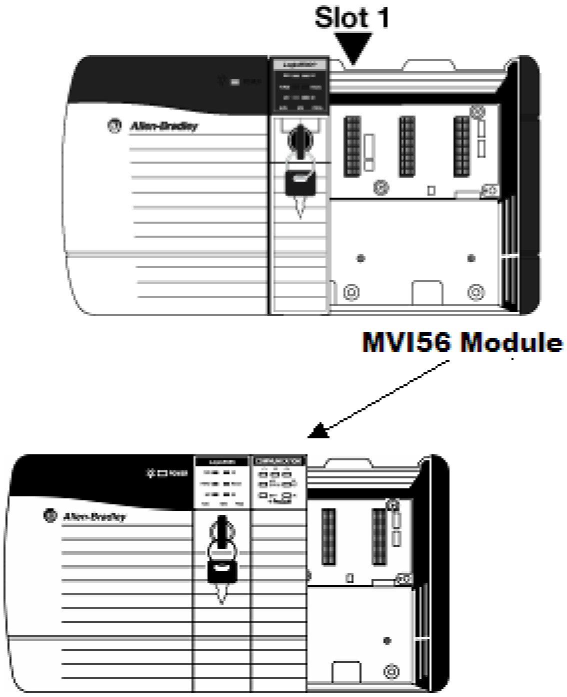
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-MBP module. Refer to your Rockwell Automation product documentation for installation instructions.

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 MVI56-MBP into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

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

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



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

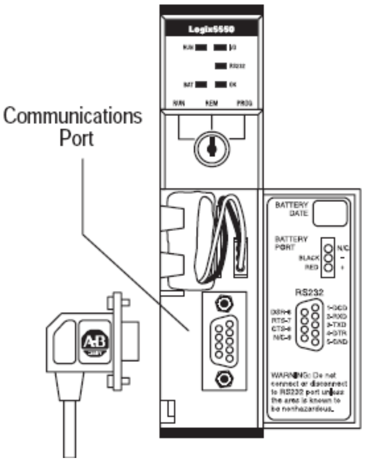
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.5 Connecting Your PC to the ControlLogix Processor

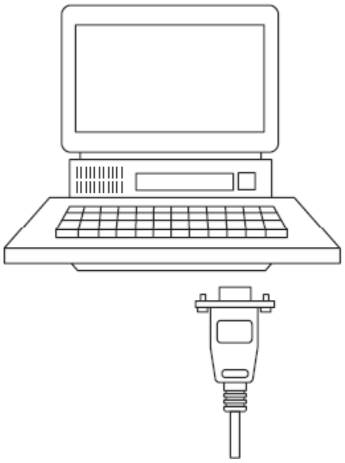
There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface.

> **Note:** It is not mandatory that you use the processor's serial interface. You may access the processor through any network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

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



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

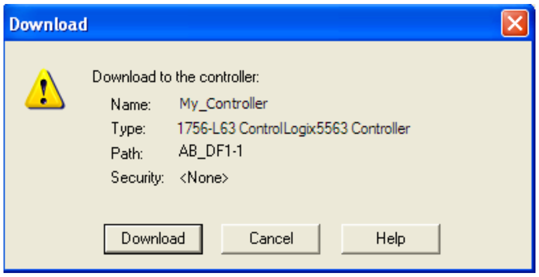


1.6 Downloading the Sample Program to the Processor

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

1 If you are not already online with the processor, in RSLogix 5000 open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.

2 When communication is established, RSLogix 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



Download

Download to the controller:

Name: My_Controller

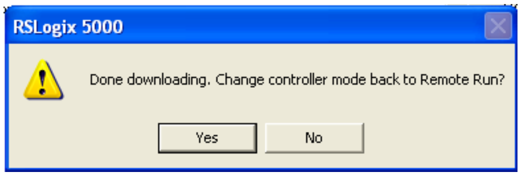
Type: 1756-L63 ControlLogix5563 Controller

Path: AB_DF1-1

Security: <None>

3 RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.

4 When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.



RSLogix 5000

Done downloading. Change controller mode back to Remote Run?

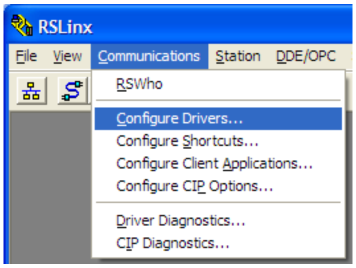
> **Note:** If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.6.1 Configuring the RSLinx Driver for the PC COM Port

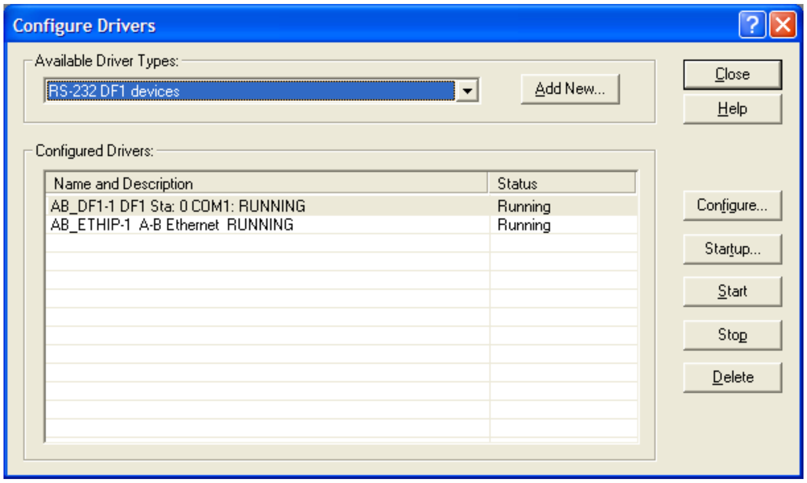
When trying to connect serially, if RSLogix is unable to establish communication with the processor, follow these steps.

1 Open *RSLinx*.

2 Open the **COMMUNICATIONS** menu, and click **CONFIGURE DRIVERS**.



This action opens the *Configure Drivers* dialog box.

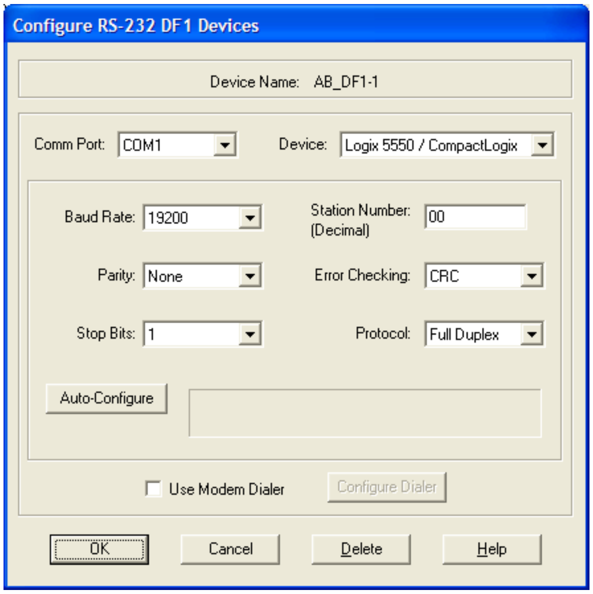


Available Driver Types: RS-232 DF1 devices

Name and Description	Status
AB_DF1-1 DF1 Sta: 0 COM1: RUNNING	Running
AB_ETHIP-1 A-B Ethernet RUNNING	Running

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

3 Click to select the driver, and then click **CONFIGURE**. This action opens the *Configure RS-232 DF1 Devices* dialog box.



Device Name: AB_DF1-1

Comm Port: COM1

Device: Logix 5550 / CompactLogix

Baud Rate: 19200

Station Number: (Decimal) 00

Parity: None

Error Checking: CRC

Stop Bits: 1

Protocol: Full Duplex

□ Use Modem Dialer

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

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

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

1.7 Connecting 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 MVI56-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.

2 MVI56-MBP Configuration

In This Chapter

2.1 Module Configuration

In order for the MVI56-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 describes the 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.

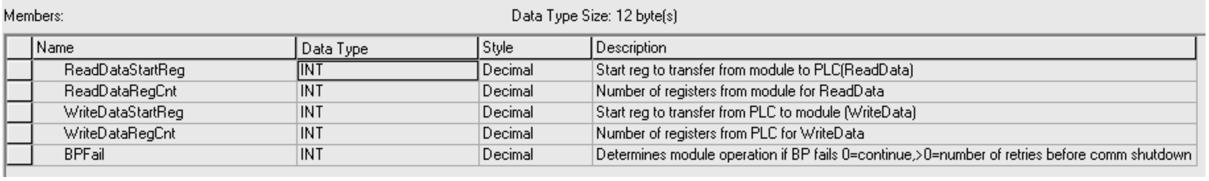
Important Note: The module will not function correctly until the Module Configuration Data is received from the processor with at least the Local Modbus Plus Node Address set to a valid value.

The MVI56-MBP module must be configured at least once when the card is first powered, and any time thereafter when the parameters must be changed.

2.2 Configuration Objects

Configuration of the module is performed by filling in the values in the module object defined in the Controller Tags Edit Tags dialog box. Each parameter required by the module has a defined location in the object. The tables and discussions below describe the parameters set in dialog box. You can view these tables by opening the data type under the User Defined Data Type option in the Controller Organization list.

2.2.1 Data Transfer Parameters



Members: Data Type Size: 12 byte(s)

Name	Data Type	Style	Description
ReadDataStartReg	INT	Decimal	Start reg to transfer from module to PLC(ReadData)
ReadDataRegCnt	INT	Decimal	Number of registers from module for ReadData
WriteDataStartReg	INT	Decimal	Start reg to transfer from PLC to module (WriteData)
WriteDataRegCnt	INT	Decimal	Number of registers from PLC for WriteData
BPFail	INT	Decimal	Determines module operation if BP fails 0=continue,>0=number of retries before comm shutdown

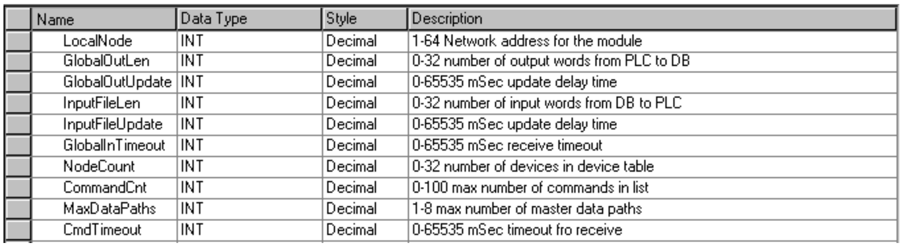
This object defines the parameters for data movement between the module and the processor. Values entered determine the ladder logic and data size required in the application. The ReadData and WriteData arrays must be sized to or larger than the count values entered. The ladder logic must process the number of blocks of data to be transferred. The number of blocks is computed as follows:

```
BlockCnt = INT(RegCnt/200) + if(MOD(RegCnt,200), 1,0)
```

If the register count is evenly divisible by 200, the number of blocks is easy to compute and the ladder is much simpler to write. If the number is not evenly divisible by 200, special handling of the last block of data must developed, as it must transfer fewer than 200 words. **It is recommended that the count values always be set to values evenly divisible by 200.**

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

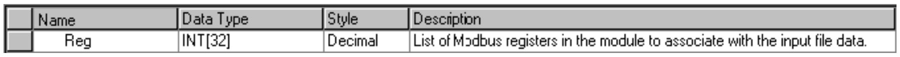
2.2.2 Modbus Plus Parameters



Name	Data Type	Style	Description
LocalNode	INT	Decimal	1-64 Network address for the module
GlobalOutLen	INT	Decimal	0-32 number of output words from PLC to DB
GlobalOutUpdate	INT	Decimal	0-65535 mSec update delay time
InputFileLen	INT	Decimal	0-32 number of input words from DB to PLC
InputFileUpdate	INT	Decimal	0-65535 mSec update delay time
GlobalInTimeout	INT	Decimal	0-65535 mSec receive timeout
NodeCount	INT	Decimal	0-32 number of devices in device table
CommandCnt	INT	Decimal	0-100 max number of commands in list
MaxDataPaths	INT	Decimal	1-8 max number of master data paths
CmdTimeout	INT	Decimal	0-65535 mSec timeout fro receive

This object defines the parameters for the operation of the Modbus Plus communication driver in the module.

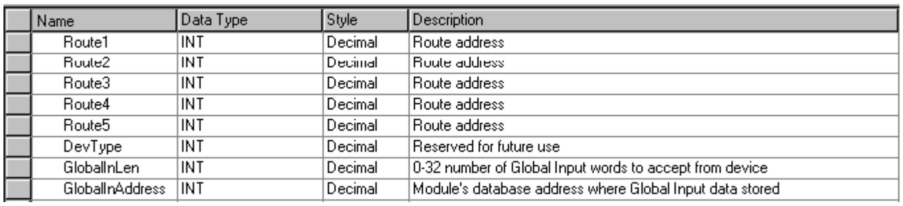
2.2.3 Input File Map



Name	Data Type	Style	Description
Reg	INT[32]	Decimal	List of Modbus registers in the module to associate with the input file data.

This object defines the list of registers in the module's internal database to be associated with the input image transferred from the module to the processor. The number of entries processed in the list is determined by the **InputFileLen** parameter in the **MBPConfig** object. Each position in the **Reg** array corresponds to the associated word in the input image and contains a register value valid for the module. Refer to the MVI56-MBP database layout chart in **Database Definition**.

2.2.4 Device Definition



Name	Data Type	Style	Description
Route1	INT	Decimal	Route address
Route2	INT	Decimal	Route address
Route3	INT	Decimal	Route address
Route4	INT	Decimal	Route address
Route5	INT	Decimal	Route address
DevType	INT	Decimal	Reserved for future use
GlobalInLen	INT	Decimal	0-32 number of Global Input words to accept from device
GlobalInAddress	INT	Decimal	Module's database address where Global Input data stored

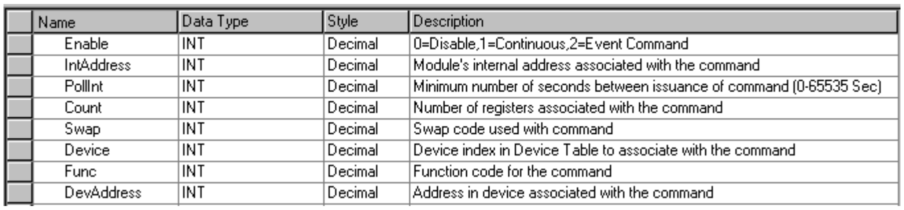
This object defines the parameters for each node to be considered on the network by the module. The route address entered in the Route1 to Route5 parameters correspond to the route a message must take to reach its final destination. These values will be entered directly into the Modbus Plus route portion of each message to the device. The route list will be processed until a value of zero is reached in the list. For example, to reach node 7 on the local segment, enter a value of 7 for Route1 and 0 for Route2 to Route5.

The **GlobalInLen** parameter defines how many words of global input data the module will be expecting from the device. If less data is returned than expected, an error code is returned for the device in the status table and the data is rejected. If more data is received than expected, the first **GlobalInLen** words will be accepted and no error condition will be reported.

The **GlobalInAddress** parameter defines the location in the module's internal database where the input data will be placed. Valid entry for this parameter is 0 to 3999.

An array of these objects is defined in the **MBPlusModuleDef** object to hold multiple device definitions for the module. The index of each device in the array is used in the master port command definitions to define the destination for the command. Therefore, a device definition should exist for each node on the network to receive global input data from and to interact with using the command list.

2.2.5 Modbus Plus Master Commands



Name	Data Type	Style	Description
Enable	INT	Decimal	0=Disable,1=Continuous,2=Event Command
IntAddress	INT	Decimal	Module's internal address associated with the command
PollInt	INT	Decimal	Minimum number of seconds between issuance of command (0-65535 Sec)
Count	INT	Decimal	Number of registers associated with the command
Swap	INT	Decimal	Swap code used with command
Device	INT	Decimal	Device index in Device Table to associate with the command
Func	INT	Decimal	Function code for the command
DevAddress	INT	Decimal	Address in device associated with the command

This object defines the parameters for each command in the master command list. The **MBPlusModuleDef** object contains an array of these objects that define the complete list. The parameter, **CommandCnt**, in the **MBPConfig** object determines the number of elements in the array to process. The definition of each parameter required for each command is given below:

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.

IntAddress: This parameter specifies the starting internal register address to be associated with the command. Valid entry for this parameter is 0 to 3999.

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

Count: This parameter defines the number of registers to be considered by the command. Valid entry for this parameter is 1 to 100.

Swap: This parameter is used to swap the bytes in the data read or written by the command. The codes used are as follows:

- 0 = No swapping
- 1 = Swap words
- 2 = Swap words and bytes
- 3 = Swap bytes

Device: 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.

Func: 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.

DevAddress: This parameter defines the starting address in the device being considered by the command. Values entered in this field are dependent on the node's database definition. Refer to the specific manufacture's database definition for the device to determine the location of the data to be interfaced.

3 Ladder Logic

In This Chapter

Ladder logic is required for the MVI56-MBP module to operate. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

3.1 Module Data Object

All data related to the MVI56-MBP is stored in a user defined data type. An instance of the data type is required before the module can be used. This is done by declaring a variable of the data type in the Controller Tags Edit Tags dialog box. The following table describes the structure of this object.

Name	Data Type	Description
ModDef	MBPModule	Backplane operation characteristics
Cfg	MBPConfig	MBPlus settings for module
InMap	MBPInputFileMap	Input file register map
Device	MBPDeviceDef[64]	Device Definition Table
Cmd	MBPCmd[200]	Master Command List
BP	MBPlusBackplane	Backplane runtime variables
Stat	MBPStat	Module Status Data
GlobalIn	INT[32]	Global Input Data
GlobalOut	INT[32]	Global Output Data
InStat	MBPInStat	Status information in each read block
ReadData	INT[600]	Data read from module
WriteData	INT[600]	Data to write to module
EventCtrl	INT[13]	Event Command Control Flags (commands per word)
UserCommands	MBPUserCmd[20]	Cmds transferred from ladder logic
SendUCmds	BOOL[32]	
SendCmds	BOOL	

Name	Data Type	Description
Enable1Cmds	BOOL	
Enable2Cmds	BOOL	
DisableCmds	BOOL	

The Module Data object contains objects that define the configuration, user data, status and command control data related to the module. Each of these object types is discussed in the following sections of the document.

3.1.1 MBP User Cmd

This object sends Modbus Plus commands directly from ladder logic to the MVI56-MBP.

Name	Data Type	Description
DB	INT	Internal database address in module
Count	INT	Number of registers for command
SwapCode	INT	Swap code (0 to 3)
Device	INT	Device index from configured list
Func	INT	Function Code 3 or 16
DevDB	INT	Register address in device
Spare_1	INT	
Spare_2	INT	
Spare_3	INT	
Spare_4	INT	

3.1.2 Status Objects

These objects store the status of the module. The **MBPInStat** object shown below is updated each time a read block is received by the processor. Use this data to monitor the state of the module at a "real-time rate".

Name	Data Type	Description
MBPErr	INT	MBP Error Status
Peer	INT	Peer Status
TokenPass	INT	Token Pass Counter
TokenTime	INT	Token Rotation Time
CFail	SINT	Communication Failed Error Counter
CRetry	SINT	Communication Retry Counter
NoResp	SINT	No Response Received Error Counter
GoodRec	SINT	Good Received Packet Success Counter
PathErr	SINT	Unexpected Path Error Counter
ExRec	SINT	Exception Response Received Error Counter
OPath1	SINT	Data master output path 1 counter
OPath2	SINT	Data master output path 2 counter
OPath3	SINT	Data master output path 3 counter

Name	Data Type	Description
OPath4	SINT	Data master output path 4 counter
OPath5	SINT	Data master output path 5 counter
OPath6	SINT	Data master output path 6 counter
OPath7	SINT	Data master output path 7 counter
OPath8	SINT	Data master output path 8 counter
IPath1	SINT	Data Slave input counter path 1
IPath2	SINT	Data Slave input counter path 2
IPath3	SINT	Data Slave input counter path 3
IPath4	SINT	Data Slave input counter path 4
IPath5	SINT	Data Slave input counter path 5
IPath6	SINT	Data Slave input counter path 6
IPath7	SINT	Data Slave input counter path 7
IPath8	SINT	Data Slave input counter path 8

In addition to the **MBPInStat** data, the module provides more detailed and complete status information in the **MBPStat** object. The following table describes the structure of the object. Refer to Database Definition (page 71) for a complete listing of the data stored in this object.

Name	Data Type	Description
InUpdate	INT[70]	Array of Global Input status data for devices
MstrCmd	INT[200]	Array of command list error codes
Misc	INT[30]	Miscellaneous status data
InCntr	INT[70]	Array of Global Input counters for devices

3.1.3 User Data Objects

These objects hold data to be transferred between the processor and the MVI56-MBP module. The first set of user data is the global input and output data.

Name	Data Type	Description
GlobalIn	INT[32]	Global Input Data
GlobalOut	INT[32]	Global Output Data

The global input data (**GlobalIn**) is transferred to the processor each time a new data read block is received. This data area is used for data that requires the highest update frequency for use in the ladder logic. Data placed in the input file image is determined by the values entered into the **InMap** data object (**MBPInputFileMap** object).

The global output data (**GlobalOut**) is transferred from the module to other nodes on the network each time it generates a Put Global Output command. This data is set in the ladder logic by the user and is transferred from the process to the module each time a new write block is constructed and sent to the module. This data area is used for data that requires the highest update frequency on the Modbus Plus Network.

The second set of user data is the read and write data transferred between the processor and the module as "pages" of data up to 200 words long.

Name	Data Type	Description
ReadData	INT[600]	Data read from module
WriteData	INT[600]	Data to write to module

The read data (**ReadData**) is an array set to match the value entered in the **ReadRegCnt** parameter of the **MBPModule** object. For ease of use, this array should be dimensioned as even increment of 200 words. This data is paged up to 200 words at a time from the module to the processor. The ReadData task places the data received into the proper position in the read data array. Use this data for status and control in the ladder logic of the processor.

The write data (**WriteData**) is an array set to match the value entered in the **WriteRegCnt** parameter of the **MBPModule** object. For ease of use, this array should be dimensioned as even increment of 200 words. This data is paged up to 200 words at a time from the processor to the module. The WriteData task places the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.

3.1.4 Command Control Object

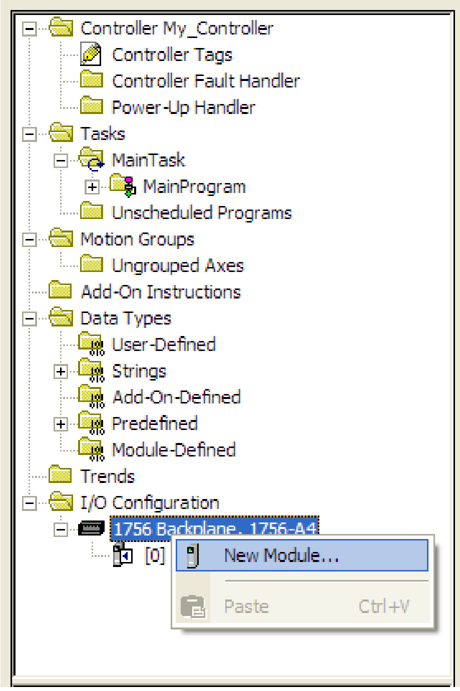
This object contains the current states of the event command control bits. Each command in the command list has an associated event bit in this object. These bits can be controlled by ladder logic to cause the module to issue a master port command on the Modbus Plus network. These bits are located in the **EventCtrl** member of the **MBPlusModuleDef** data type and should be displayed as shown below in the Controller Tags Edit Tags dialog box.

Name	Data Type	Description
EventCtrl	INT[13]	Event Command Control Flags (commands per word)

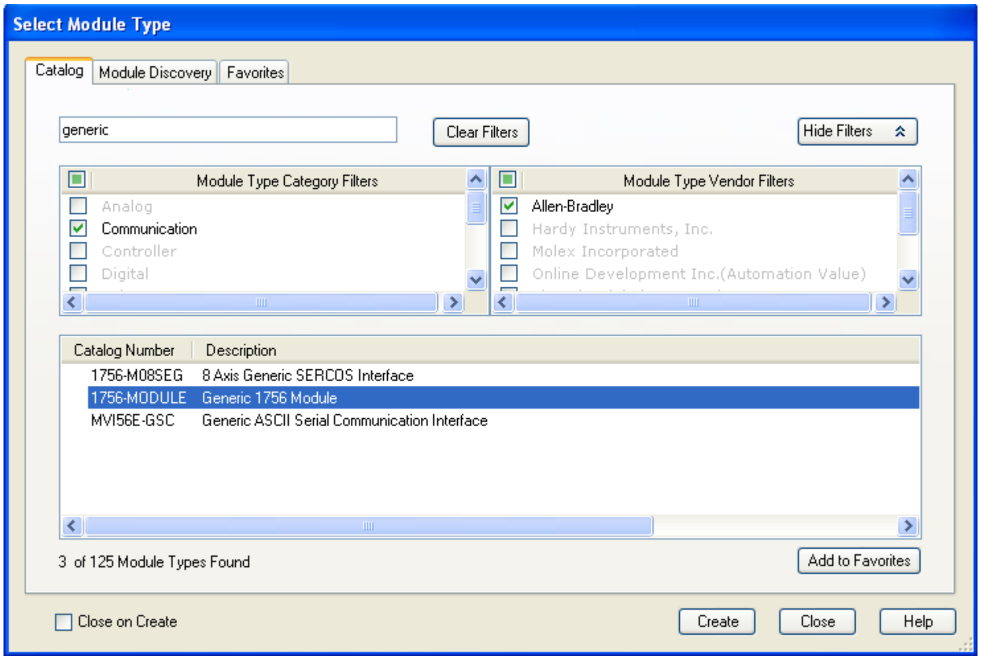
Event command control bits apply only to commands that have a value of two for their **Enable** parameter in the master command list.

3.2 Adding the Module to an Existing Project

1 In the *Controller Organizer* window, right-click **I/O CONFIGURATION** or the controller backplane and then choose **NEW MODULE...**



This action opens the *Select Module* dialog box:



generic

☑	Module Type Category Filters
☐	Analog
☑	Communication
☐	Controller
☐	Digital

☑	Module Type Vendor Filters
☑	Allen-Bradley
☐	Hardy Instruments, Inc.
☐	Molex Incorporated
☐	Online Development Inc.(Automation Value)

Catalog Number	Description
1756-M08SEG	8 Axis Generic SERCOS Interface
1756-MODULE	Generic 1756 Module
MVI56E-GSC	Generic ASCII Serial Communication Interface

3 of 125 Module Types Found

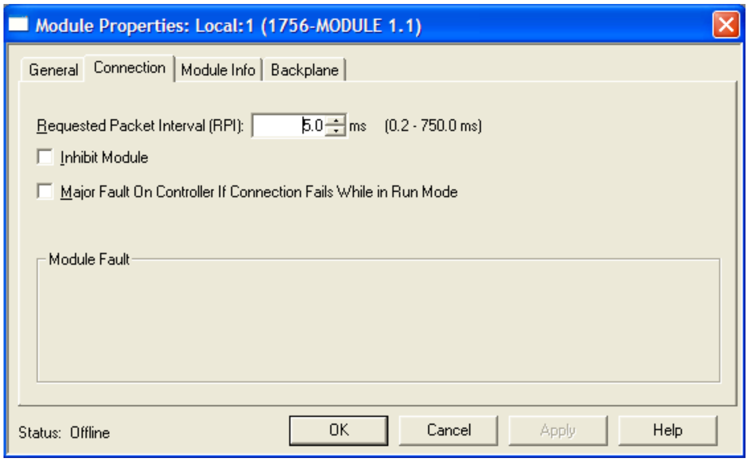
☐ Close on Create

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

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

Parameter	Value
Name	Enter a module identification string. Example: **MBP_2**
Description	Enter a description for the module. Example: **MODBUS PLUS COMMUNICATION MODULE**
Comm Format	Select **DATA-INT**.
Slot	Enter the slot number in the rack where the MVI56-MBP module is located.
Input Assembly Instance	**1**
Input Size	**250**
Output Assembly Instance	**2**
Output Size	**248**
Configuration Assembly Instance	**4**
Configuration Size	**0**

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



Module Properties: Local:1 (1756-MODULE 1.1)

Requested Packet Interval (RPI): 5.0 ms (0.2 - 750.0 ms)

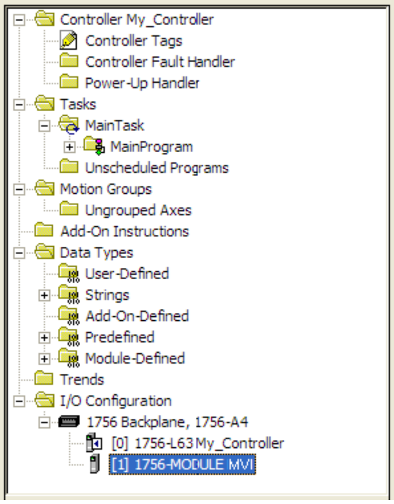
□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

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



Controller My_Controller
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled Programs
Motion Groups
Ungrouped Axes
Add-On Instructions
Data Types
User-Defined
Strings
Add-On-Defined
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L63 My_Controller
[1] 1756-MODULE MVI

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

4 Diagnostics and Troubleshooting

In This Chapter

The MVI56-MBP provides information on diagnostics and troubleshooting in the following forms:

- LED status indicators on the front of the module provide general information on the module's status.
- You can view status data contained in the module through the Configuration/Debug port or the Ethernet port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- You can transfer status data values from the module to processor memory and can monitor them in the processor manually or by customer-created logic. For details on Status Data values, see Error Status Table.

4.1 Reading Status Data from the Module

The MVI56-MBP module returns a 365-word Status Data block that can be used to determine the module’s operating status. This data is located in the module’s database at registers 4000 to 4365. This data is transferred to the ControlLogix processor approximately every second. Nodes on the Modbus Plus network can read this data area through the issuance of read commands to the module

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 *Prosoft Configuration Builder (PCB)*. The module does not respond to mouse movements or clicks. The command executes as soon as you press the command letter — you do not need to press **[Enter]**. When you type a command letter, a new screen will be displayed in the *Prosoft Configuration Builder (PCB)* application.

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:

Parameter	**Setting**
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:

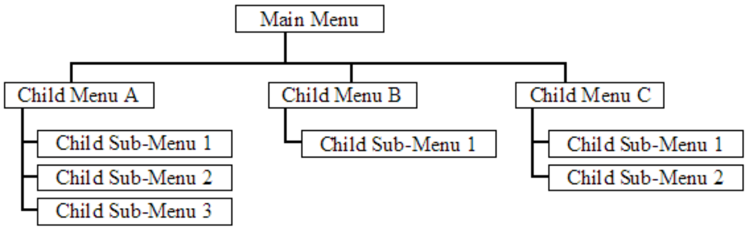
1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2 Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC (page 68).
3 Verify that your communication 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 submenus in *ProSoft Configuration Builder* for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu to the next higher menu by pressing **[M]** on your keyboard.

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



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

Keystrokes

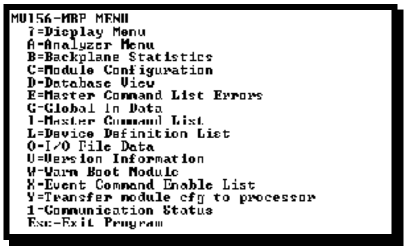
The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

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

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

4.1.2 Main Menu

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



```
MVI56-MBP MENU
  ?=Display Menu
  A=Analyzer Menu
  B=Backplane Statistics
  C=Module Configuration
  D=Database View
  E=Master Command List Errors
  G=Global In Data
  I=Master Command List
  L=Device Definition List
  O=I/O File Data
  V=Version Information
  W=Warm Boot Module
  X=Event Command Enable List
  Y=Transfer module cfg to processor
  1=Communication Status
  Esc=Exit Program
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Use these commands only if you fully understand their potential effects, or if you are specifically directed to do so by ProSoft Technology Technical Support staff.

Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Opening the Data Analyzer Menu

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

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please 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 Block Transfer Statistics

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

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

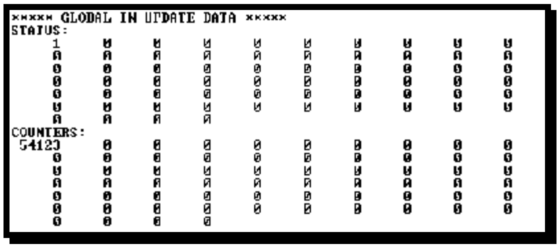
Tip: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

Press **[C]** to view the *Module Configuration* screen. Use this command to display the current module configuration and statistics.

Viewing Global In Data

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



```
***** GLOBAL IN UPDATE DATA *****
STATUS:
      1     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0
COUNTERS:
  54123     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0
```

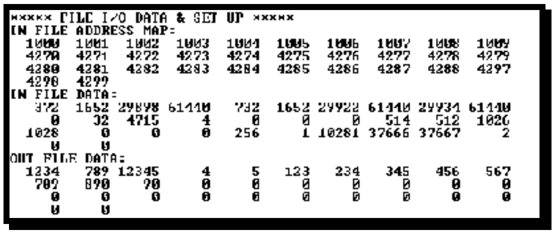
The **Status** data area displays the current status of each node on the network. The **Counters** section displays the number of Global Input data messages received from the device.

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 view the module's global input and output data.



```
***** FILE I/O DATA & SET UP *****
IN FILE ADDRESS MAP:
  1000  1001  1002  1003  1004  1005  1006  1007  1008  1009
  4270  4271  4272  4273  4274  4275  4276  4277  4278  4279
  4280  4281  4282  4283  4284  4285  4286  4287  4288  4297
  4298  4299
IN FILE DATA:
   372  1652 29898 61440   732  1652 29922 61440 29934 61440
     0    32  4715     4     0     0     0   514   512  1026
  1028     0     0     0   256     1 10281 37666 37667     2
     0     0
OUT FILE DATA:
  1234   789 12345     4     5   123   234   345   456   567
   789   890    90     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0
```

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 ControlLogix 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

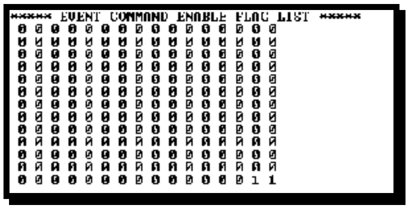
Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Use these commands only if you fully understand their potential effects, or 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 cause 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 ControlLogix 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

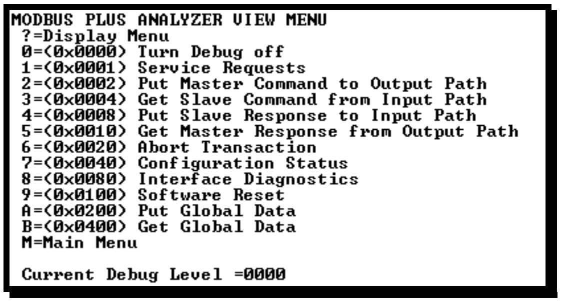
Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Use these commands only if you fully understand their potential effects, or 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 uses its flash memory configuration to configure the module.

4.1.3 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.



```
MODBUS PLUS ANALYZER VIEW MENU
 ?=Display Menu
 0=(0x0000) Turn Debug off
 1=(0x0001) Service Requests
 2=(0x0002) Put Master Command to Output Path
 3=(0x0004) Get Slave Command from Input Path
 4=(0x0008) Put Slave Response to Input Path
 5=(0x0010) Get Master Response from Output Path
 6=(0x0020) Abort Transaction
 7=(0x0040) Configuration Status
 8=(0x0080) Interface Diagnostics
 9=(0x0100) Software Reset
 A=(0x0200) Put Global Data
 B=(0x0400) Get Global Data
 M=Main Menu

 Current Debug Level =0000
```

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 MVI56-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.

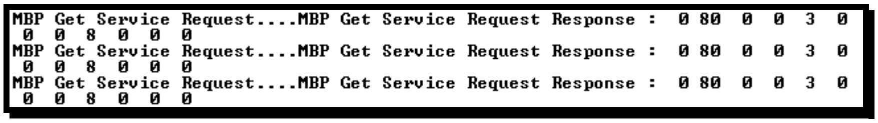
The **Current Debug Level** parameter displayed at the bottom of the display shows the current debug level being monitored by the analyzer. A value of 0000 indicates that the analyzer is not monitoring any messages.

(0x0000) Turn Debug off

Press **[0]** (zero) to turn the analyzer off. This action freezes the analyzer screen, showing the most recent data capture. If your terminal emulation program buffers previously received data, you can scroll through the acquired data to view older data.

(0x0001) Service Requests

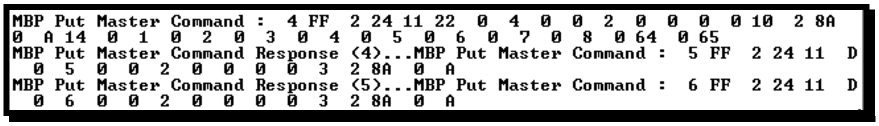
Press **[1]** to view all service request operations. The screen will scroll very rapidly as it displays these messages. The following illustration shows an example.



```
MBP Get Service Request....MBP Get Service Request Response :  0 80  0  0  3  0
 0  0  8  0  0  0
MBP Get Service Request....MBP Get Service Request Response :  0 80  0  0  3  0
 0  0  8  0  0  0
MBP Get Service Request....MBP Get Service Request Response :  0 80  0  0  3  0
 0  0  8  0  0  0
```

(0x0002) Put Master Command to Output Path

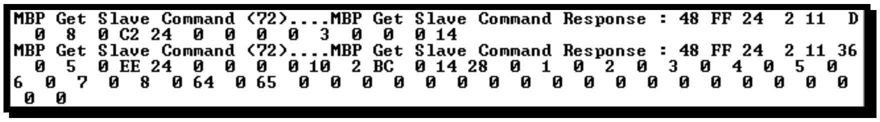
Press **[2]** to display master command requests sent from the module to the network. The following illustration shows an example.



```
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

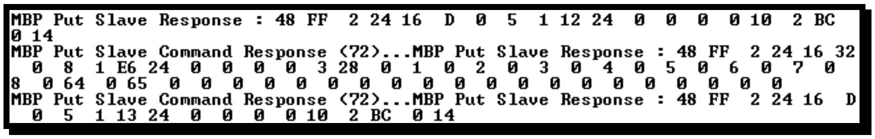
Press **[3]** to display slave request messages received for the module. The following illustration shows an example.



```
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  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

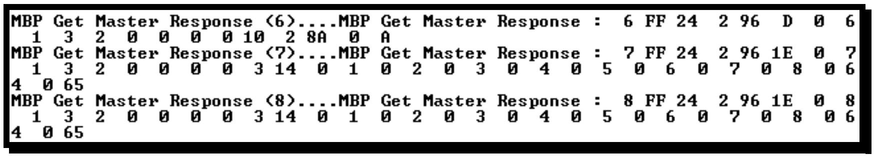
Press **[4]** to display slave response messages sent to requests made by other nodes on the network to the module. The following illustration shows an example.



```
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
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. The following illustration shows an example.



```
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

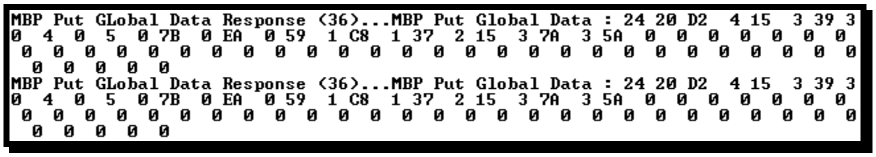
Press **[8]** to display any interface diagnostic messages processed by the module.

(0x0100) Software Reset

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

(0x0200) Put Global Data

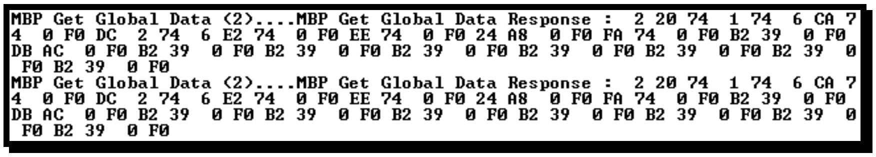
Press **[A]** to display any global output messages processed by the module. The following illustration shows an example of the output.



```
MBP Put GLobal Data Response (36)...MBP Put Global Data : 24 20 D2  4 15  3 39 3
0  4  0  5  0 7B  0 EA  0 59  1 C8  1 37  2 15  3 7A  3 5A  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0
MBP Put GLobal Data Response (36)...MBP Put Global Data : 24 20 D2  4 15  3 39 3
0  4  0  5  0 7B  0 EA  0 59  1 C8  1 37  2 15  3 7A  3 5A  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0
```

(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.



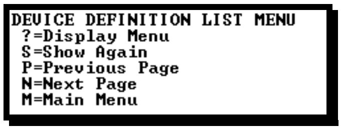
```
MBP Get Global Data (2)....MBP Get Global Data Response :  2 20 74  1 74  6 CA 7
4  0 F0 DC  2 74  6 E2 74  0 F0 EE 74  0 F0 24 A8  0 F0 FA 74  0 F0 B2 39  0 F0
DB AC  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0
 F0 B2 39  0 F0
MBP Get Global Data (2)....MBP Get Global Data Response :  2 20 74  1 74  6 CA 7
4  0 F0 DC  2 74  6 E2 74  0 F0 EE 74  0 F0 24 A8  0 F0 FA 74  0 F0 B2 39  0 F0
DB AC  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0 F0 B2 39  0
 F0 B2 39  0 F0
```

Returning to the Main Menu

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

4.1.4 Device Definition List Menu

This list consists of multiple pages of device definition data. Press **[?]** to view a list of commands available on this menu.



```
DEVICE DEFINITION LIST MENU
 ?=Display Menu
 S=Show Again
 P=Previous Page
 N=Next Page
 M=Main Menu
```

Redisplaying the Current Page

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

Viewing the Previous Page of Commands

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

Viewing the Next Page of Commands

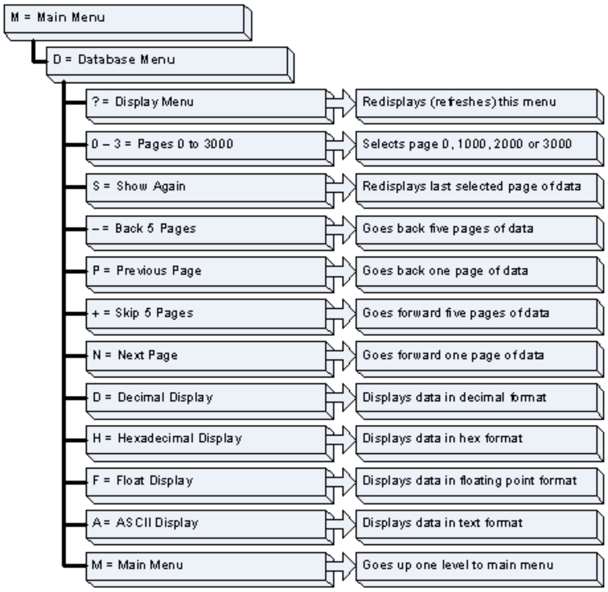
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.1.5 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.


M = Main Menu
D = Database Menu
? = Display Menu
Redisplays (refreshes) this menu
0 – 3 = Pages 0 to 3000
Selects page 0, 1000, 2000 or 3000
S = Show Again
Redisplays last selected page of data
– = Back 5 Pages
Goes back five pages of data
P = Previous Page
Goes back one page of data
+ = Skip 5 Pages
Goes forward five pages of data
N = Next Page
Goes forward one page of data
D = Decimal Display
Displays data in decimal format
H = Hexadecimal Display
Displays data in hex format
F = Float Display
Displays data in floating point format
A= ASCII Display
Displays data in text format
M = Main Menu
Goes up one level to main menu

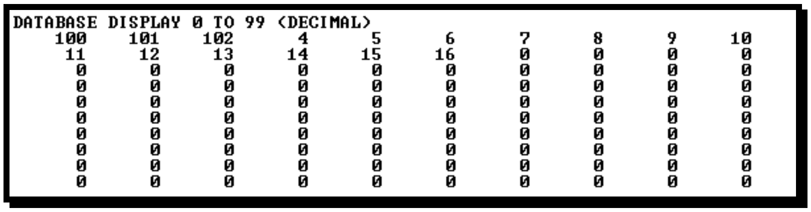
Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again



```
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

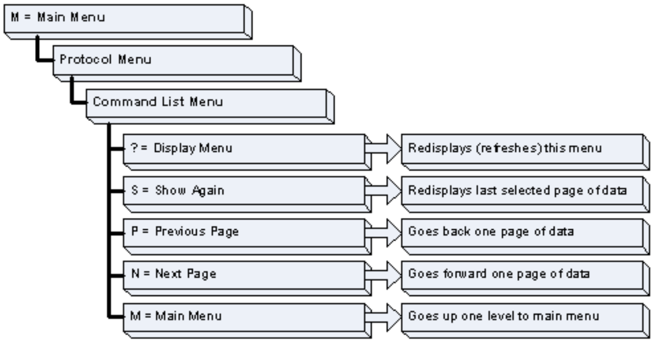
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.1.6 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.


M = Main Menu
Protocol Menu
Command List Menu
? = Display Menu
Redisplays (refreshes) this menu
S = Show Again
Redisplays last selected page of data
P = Previous Page
Goes back one page of data
N = Next Page
Goes forward one page of data
M = Main Menu
Goes up one level to main 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.

4.2 LED Status Indicators

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

LED	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
APP STATUS	Amber	Flashing	Module is functioning normally
		On	Communication error or module malfunction
		Off	Communication error or module malfunction
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.
BATT	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.
OK	Green	Off	The card is not receiving any power and is not plugged securely into the rack.
		On	The module is operating normally.
MBP ACT	Amber	6 flashes per second	The MVI56-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.

4.3 Clearing a Fault Condition

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

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

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

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

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56 modules with scrolling LED display: <*Backplane Status*> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug (or Communication) 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, then re-insert it, and then restore power to the rack.

5 Reference

In This Chapter

5.1 Product Specifications

The MVI56 Modbus Plus Communication Module allows Rockwell Automation ControlLogix 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).

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 MVI56-MBP module acts as an input/output module between the Modbus Plus network and the ControlLogix backplane. The data transfer from the ControlLogix 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 ControlLogix backplanes.

5.1.1 Functional Specifications

- Communication parameters (Config port only)
  - Baud: 57,600 (fixed)
  - Parity: none (fixed)
  - Stop: 1 (fixed)
- Modbus Plus ports
  - Memory usage is user definable
  - 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 200 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 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 200 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 returned to ladder logic on a per-command basis

5.1.2 General Specifications

- Single Slot - 1756 backplane compatible
- Local or remote rack
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module.
- Ladder Logic is used for data transfer between module and processor.
- Configuration data obtained through user-defined ladder (included).

5.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 V DC 3mA @ 24V DC
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Shock	30g Operational 50g non-operational Vibration: 5 g from 10 to 150 Hz
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Module Status Backplane Transfer Status Application Status Serial Activity
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Application ports (PRT1 & PRT2)	
Modbus Plus ports	Dual DB9 Application ports for redundant operation
Shipped with Unit	6-foot RS-232 configuration cable

5.2 Functional Overview

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI56-MBP module.

Module Power Up

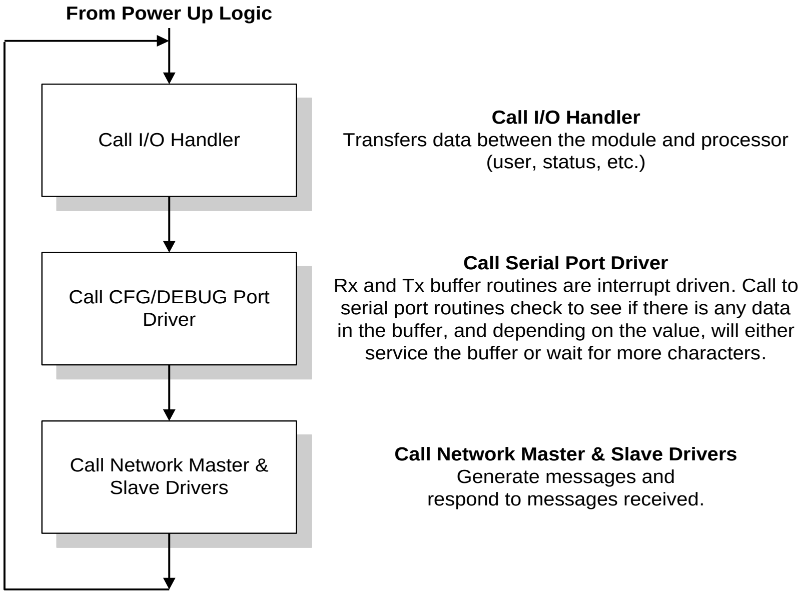
On power up the module begins performing the following logical functions:

1. Initialize hardware components
   - Initialize backplane driver
   - Test and Clear all RAM
   - Reset Modbus Plus Chipset
2. Wait for Module Configuration from 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 has received the Module Configuration Block from the processor, the Modbus Plus chipset will be enabled (presuming valid configuration values were received), and will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

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


From Power Up Logic
Call I/O Handler
Call CFG/DEBUG Port Driver
Call Network Master & Slave Drivers
Call I/O Handler
Transfers data between the module and processor (user, status, etc.)
Call Serial Port Driver
Rx and Tx buffer routines are interrupt driven. Call to serial port routines check to see if there is any data in the buffer, and depending on the value, will either service the buffer or wait for more characters.
Call Network Master & Slave Drivers
Generate messages and respond to messages received.

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 will resume communications on the network.

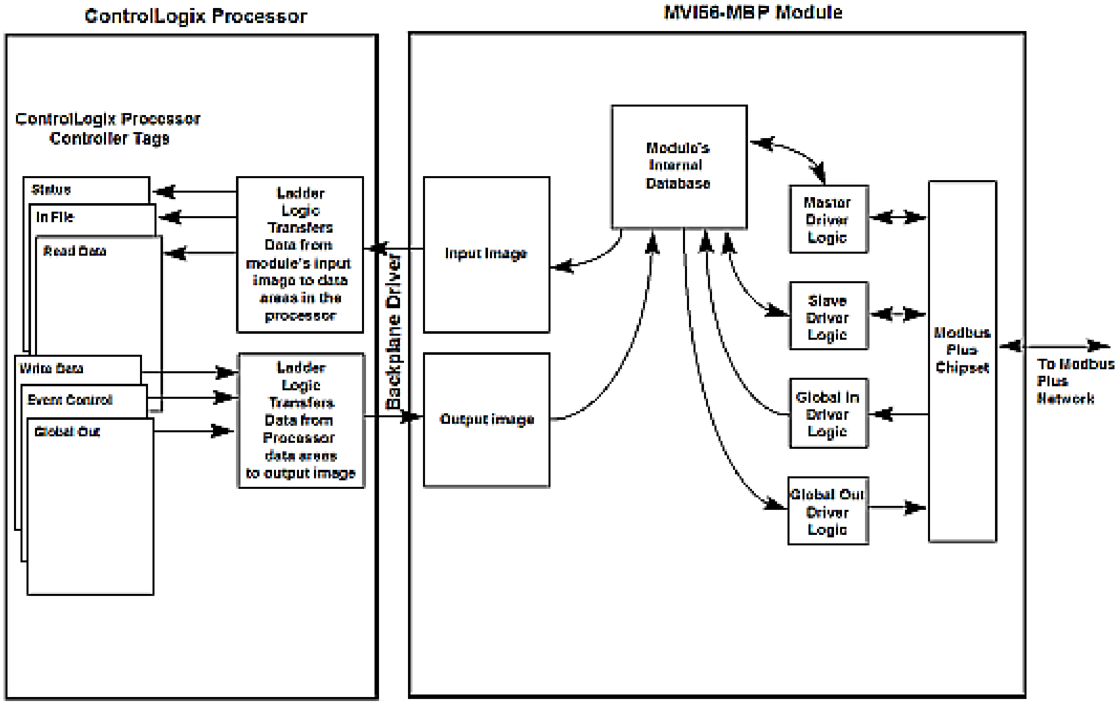
Backplane Data Transfer

The MVI56-MBP module communicates directly over the ControlLogix backplane. Data is paged between the module and the ControlLogix processor across the backplane using the module's input and output images. The frequency of update of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 2.5 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is set to 250 words. This large data area permits fast throughput of data between the module and the processor.

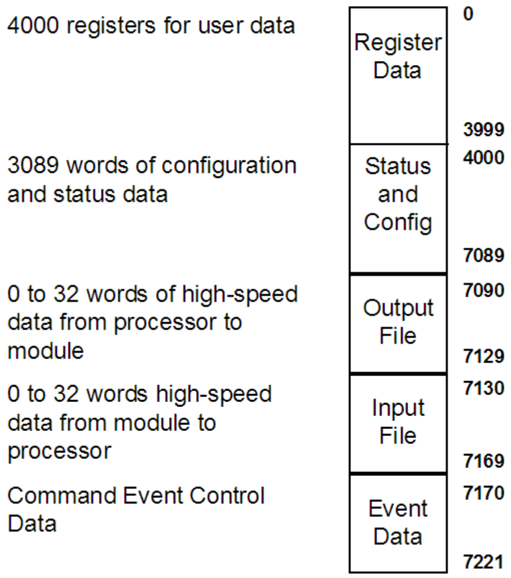
The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module is set to 248 words. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-MBP module and the Modbus Plus network.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the ControlLogix processor to interface the input and output image data with data defined in the Controller Tags. 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 (47222 Modbus). The following illustration shows the layout of the database:

Module's Internal Database Structure


4000 registers for user data
Register Data
0
3999
3089 words of configuration and status data
Status and Config
4000
7089
0 to 32 words of high-speed data from processor to module
Output File
7090
7129
0 to 32 words high-speed data from module to processor
Input File
7130
7169
Command Event Control Data
Event Data
7170
7221

Data contained in this database is paged through the input and output images by coordination of the ControlLogix ladder logic and the MVI56-MBP module's program. Up to 248 words of data can be transferred from the module to the processor at a time. Up to 247 words of data can be transferred from the processor to the module. Each image has a defined structure depending on the data content and the function of the data transfer as defined in the following sections.

Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 3999, a portion of the status data, input file data and global output data. These data are transferred through read (input image) and write (output image) blocks. Refer to **Module Configuration** for a description of the data objects used with the blocks and the ladder logic required. The structure and function of each block is discussed in the following sections.

Read Block

These blocks of data are used to transfer information from the module to the ControlLogix processor. The following table describes the structure of the input image.

Read Block from Module to Processor

Word Offset	Description	Length
0	Read Block ID	1
1	Write Block ID	1
2 to 201	Read Data	200
202 to 217	Status Data	16
218 to 249	Global Input Data	32

The Read Block ID is an index value used to determine the location of where the data will be placed in the ControlLogix processor controller tag array of module read data. Each transfer can move up to 200 words (block offsets 2 to 201) of data. In addition to moving user data, the block also contains a short list of status data for the module and the global input data (Input File Data). This last set of data is transferred with each new block of data and is used for high-speed data movement.

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

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

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

Write Block

These blocks of data are used to transfer information from the ControlLogix processor to the module. The structure of the output image used to transfer this data is shown below:

Write Block from Processor to Module

Word Offset	Description	Length
0	Write Block ID	1
1 to 200	Write Data	200
201 to 213	Command Event Enable Bits	13
214 to 215	Spare	2
216 to 247	Global Output Data	32

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed. Each transfer can move up to 200 words (block offsets 1 to 200) of data. In addition to the user data, the block also contains the command event enable bits and the global output data. This last set of data is transferred with each new block of data and is used for high-speed data movement.

Configuration Data Transfer

When the module performs a restart operation, it will request configuration information from the ControlLogix processor. This data is transferred to the module in specially formatted write blocks (output image). The module will poll for each block by setting the required write block number in a read block (input image). The format of the blocks for configuration is given in the following sections.

Module Configuration Data

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

Configuration Block from Processor to Module

Word Offset	Description	Length
0	9000	1
1 to 5	Backplane Parameters	5
6 to 15	Modbus Plus Configuration	10
16 to 47	Input file Map	32
48 to 247	Spare	200

Device Definition Data

Four blocks are used (Write Block IDs of 9100 to 9103) to transfer the device definition data from the processor to the module. The module will sequentially poll for each block from the processor. Ladder logic must handle each and every one of the data transfers. The structure of each block is shown in the following table.

Configuration Block from Processor to Module

Word Offset	Description	Length
0	9100 to 9103	1
1 to 8	Device Definition	8
9 to 16	Device Definition	8
17 to 24	Device Definition	8
25 to 32	Device Definition	8
33 to 40	Device Definition	8
41 to 48	Device Definition	8
49 to 86	Device Definition	8
57 to 64	Device Definition	8
65 to 72	Device Definition	8
73 to 80	Device Definition	8
81 to 88	Device Definition	8
89 to 96	Device Definition	8

Word Offset	Description	Length
97 to 104	Device Definition	8
105 to 112	Device Definition	8
121 to 128	Device Definition	8

Master Command List Data

Eight blocks are used (Write Block IDs 6000 to 6007) to transfer the master command list data from the processor to the module. The module will sequentially poll for each block from the processor. Ladder logic must handle each and every one of the data transfers. The structure of each block is shown in the following table.

Configuration Block from Processor to Module

Word Offset	Description	Length
0	6000 to 6007	1
1 to 8	Command Definition	8
9 to 16	Command Definition	8
17 to 24	Command Definition	8
25 to 32	Command Definition	8
33 to 40	Command Definition	8
41 to 48	Command Definition	8
49 to 86	Command Definition	8
57 to 64	Command Definition	8
65 to 72	Command Definition	8
73 to 80	Command Definition	8
81 to 88	Command Definition	8
89 to 96	Command Definition	8
97 to 104	Command Definition	8
105 to 112	Command Definition	8
121 to 128	Command Definition	8
129 to 136	Command Definition	8
137 to 144	Command Definition	8
145 to 152	Command Definition	8
153 to 160	Command Definition	8
161 to 168	Command Definition	8
169 to 176	Command Definition	8
177 to 184	Command Definition	8
185 to 192	Command Definition	8
193 to 200	Command Definition	8

Status Blocks

Status blocks are used to send status data from the module to the processor. This data can be used to determine the "health" of the module and the Modbus Plus network. These data are transferred from the module to the processor approximately every second. Two separate blocks are required to send all the status data contained in the module. The first block of data contains the global input status and general module status data. The structure of this block is shown in the following table.

Status Block from Module to Processor

Word Offset	Description	Length
0	9500	1
1	Write Block ID	1
2 to 71	Global Input Status Data	70
72 to 101	Module Status Data	30
102 to 171	Global Input Counter Data	70
172 to 249	Spare	78

The second block contains the status value for each master command list entry. This is a list of 200 status values each one corresponding to a master command list entry. The structure of this block is shown below.

Status Block from Module to Processor

Word Offset	Description	Length
0	9501	1
1	Write Block ID	1
2 to 201	Command Status Data	200
201 to 249	Spare	48

Special Function Blocks

Special function blocks are special blocks used to request specific tasks from the module. The current version of the software supports the following special function blocks:

- User Command block
- Command Execution block
- Command Enable/Disable/Event blocks
- Warm Boot block
- Cold Boot block

User Command Block

This block is sent from the ControlLogix processor to the module to execute up to six commands generated from the ladder logic. 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 process is shown in the following table.

Block Request from Processor to Module

Word 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
62 to 247	Spare	186

Word 1 of the block is used to set 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.

Command Execution Block

This command block is used to insert commands in the command list into the module's command queue. The command queue is executed at a high priority. Commands that have their enable code set to zero can be executed by the module using this feature. There is no response block issued by the module for this block to the ControlLogix processor. The format of the block sent by the processor to the module has the following format:

Block Request from Processor to Module

Word Offset	Description	Length
0	9003	1
1	Number of commands to add	1
2 to 61	List of command indexes	60
62 to 247	Spare	186

Word 1 of the block is used to set 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.

Command Enable/Disable Blocks

Block codes 9010, 9011 and 9012 are used to 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.

Block 9010 is used to disable 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.

Block Request from Processor to Module

Word Offset	Description	Length
0	9010	1
1	Number of commands to disable	1
2 to 61	List of command indexes	60
62 to 247	Spare	186

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

Block Request from Processor to Module

Word Offset	Description	Length
0	9011	1
1	Number of commands to enable	1
2 to 61	List of command indexes	60
62 to 247	Spare	186

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

Block Request from Processor to Module

Word Offset	Description	Length
0	9012	1
1	Number of commands to enable	1
2 to 61	List of command indexes	60
62 to 247	Spare	186

Write Configuration

This block is sent from the ControlLogix processor, and causes 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 module will respond with blocks containing the module configuration data. Ladder logic must be written to handle the receipt of these blocks. The blocks transferred from the module are as follows.

Block Response from Module to Processor

Word Offset	Description	Length
0	-9000	1
1	-9000	1
2 to 6	Backplane Parameters	5
7 to 16	Modbus Plus Configuration	10
17 to 48	Input File Map	32
49 to 247	Spare	200

Block Response from Module to Processor (Device Definition Data)

Word Offset	Description	Length
0	-9100 to -9103	1
1	-9100 to -9103	1
2 to 9	Device Definition	8
10 to 17	Device Definition	8
18 to 25	Device Definition	8
26 to 33	Device Definition	8
34 to 41	Device Definition	8
42 to 49	Device Definition	8
50 to 57	Device Definition	8
66 to 73	Device Definition	8
74 to 81	Device Definition	8
82 to 89	Device Definition	8
90 to 94	Device Definition	8
98 to 105	Device Definition	8

Word Offset	Description	Length
106 to 113	Device Definition	8
114 to 121	Device Definition	8
122 to 129	Device Definition	8

Block Response from Module to Processor (Master Command List Data)

Wrod Offset	Description	Length
0	-6000 to -6007	1
1	-6000 to -6007	1
2 to 9	Command Definition	8
10 to 17	Command Definition	8
18 to 25	Command Definition	8
26 to 33	Command Definition	8
34 to 41	Command Definition	8
42 to 49	Command Definition	8
50 to 57	Command Definition	8
66 to 73	Command Definition	8
74 to 81	Command Definition	8
82 to 89	Command Definition	8
90 to 94	Command Definition	8
98 to 105	Command Definition	8
106 to 113	Command Definition	8
114 to 121	Command Definition	8
122 to 129	Command Definition	8
130 to 137	Command Definition	8
138 to 145	Command Definition	8
146 to 153	Command Definition	8
154 to 161	Command Definition	8
162 to 169	Command Definition	8
170 to 177	Command Definition	8
178 to 185	Command Definition	8
186 to 193	Command Definition	8
194 to 201	Command Definition	8

Each of these blocks must be handled by the ladder logic for proper module operation.

Warm Boot

This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made in the controller tags data area. This will cause the module to read the new configuration information and to restart. The following table describes the format of the Warm Boot block.

Block Request from Processor to Module

Word Offset	Description	Length
0	9998	1
1 to 249	Spare	249

Cold Boot

This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The following table describes the format of the Cold Boot block.

Block Request from Module to Processor

Word Offset	Description	Length
0	9999	1
1 to 249	Spare	249

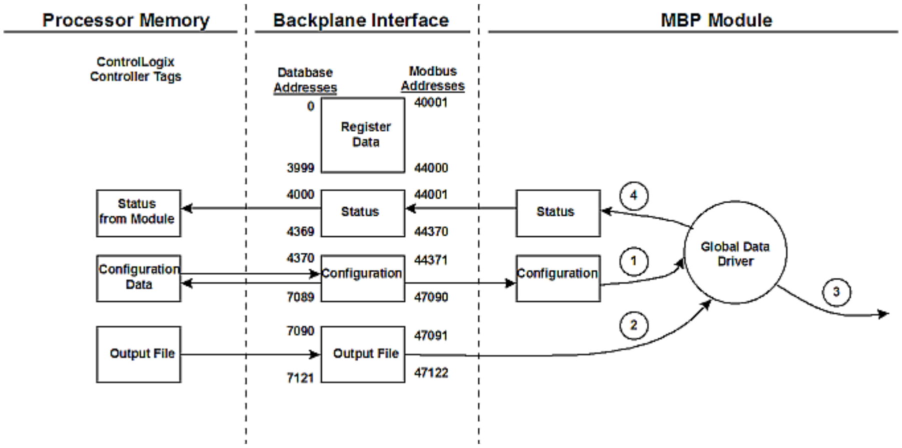
5.2.2 Data Flow between MVI56-MBP Module and ControlLogix Processor

The following topics describe the flow of data between the two pieces of hardware (ControlLogix processor and MVI56-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 will be operating in conjunction with either the Master or the Slave driver.

Global Data Out Mode

When the MVI56-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. 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 ControlLogix 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 ControlLogix processor.

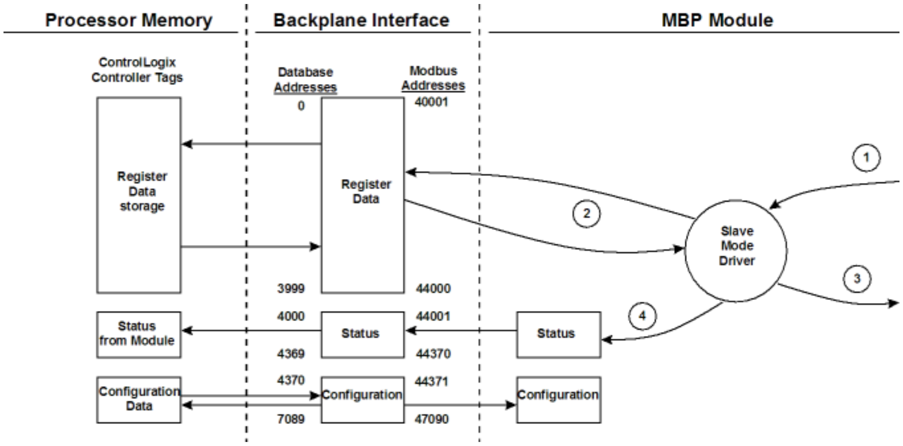
Slave Driver Mode

The Slave Driver Mode allows the MVI56-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 describe 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 MVI56-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 diagram details an example configuration for a MSTR Type 2 command.

```
          +---------+
enable -¦         ¦- active
          ¦  40050  ¦
          +---------¦
 abort -¦         ¦- error
          ¦  40060  ¦
          +---------¦
          ¦  MSTR   ¦- success
          ¦  00020  ¦
          +---------+
```

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 MVI56-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 MVI56-MBP module

The MSTR 2 instruction shown above reads 20 words from the MVI56-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 diagram details an example configuration for a MSTR Type 1 command.

```
          +---------+
enable -¦         ¦- active
          ¦  40001  ¦
          +---------¦
 abort -¦         ¦- error
          ¦  40010  ¦
          +---------¦
          ¦  MSTR   ¦- success
          ¦  00002  ¦
          +---------+
```

Contents of registers in the control block

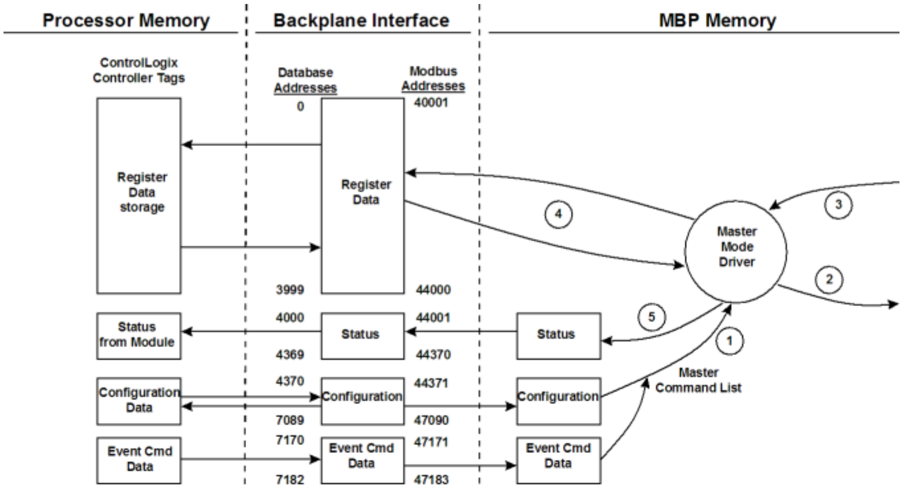
40001 = 1 Write instruction
40002 = 0 Error code
40003 = 30 Length of the write
40004 = 100 register to Write to in module
40005 = 6 MVI56-MBP Node address to write to
40006 = 1 Slave Input Path for routing
40007 = 0 Routing Address 3
40008 = 0 Routing Address 4
40009 = 0 Routing Address 5
40010 = Source of the data in the Modicon PLC to send to the MVI56-MBP module

This instruction will write 30 words (from 40010 to 40039) in the PLC to the MVI56-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 MVI56-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 ControlLogix processor. 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 ControlLogix 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 ControlLogix processor for each command in the Master Command List.

In order for the MVI56-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 ControlLogix processor. The following sections describe these configuration requirements:

Device Definition File

An entry must be made in the Device Definition File for each node on the network which 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 - MBP

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, then, the module will look to the Event Command Control words received from the ControlLogix processor to determine when to execute. The following describes the structure of the Control words:

Database Address	Bit Offset in Controller Tag 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 details 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

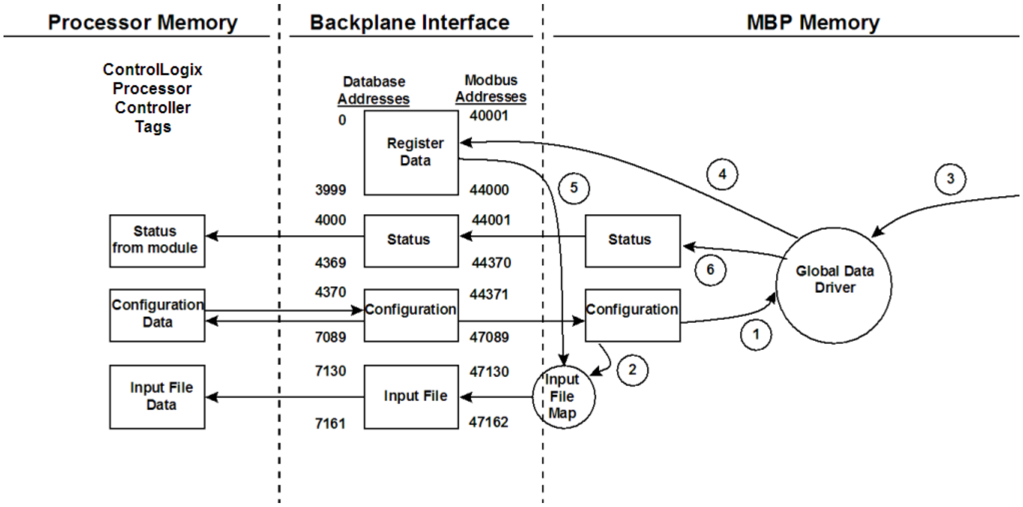
Execution status can be monitored in the same fashion as other commands in the Master Command List, via the Master Command Status block.

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).

Global Data In Mode

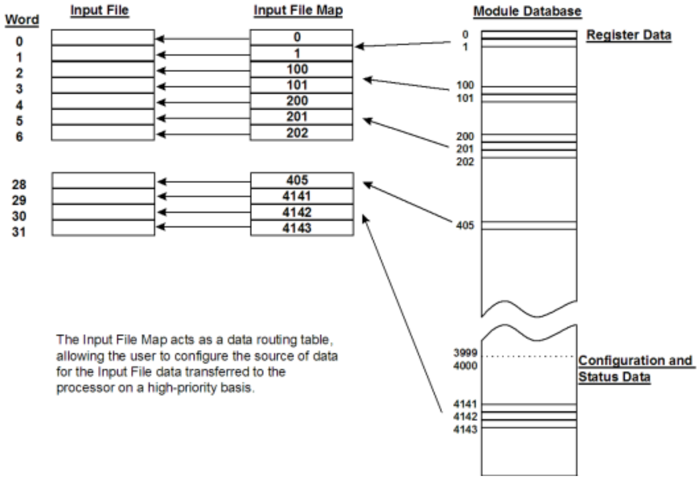
When the Global Data In mode is operational, the MVI56-MBP module is receiving Global Input data from up to 64 other nodes on the Modbus Plus network. Each node is capable of transferring up to 32 words, and therefore, the MVI56-MBP module is capable of accepting up to 2048 words in this manner. The amount of data and from which slaves to collect it from are all user defined through the Device Definition File. The following flow chart and associated table describe 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, and so on. 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.
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 ControlLogix 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 ControlLogix 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:



The Input File Map acts as a data routing table, allowing the user to configure the source of data for the Input File data transferred to the processor on a high-priority basis.

Each time the module enters a new read block into the input image it uses this Input File Map to select data out of the module's database and places it in the input image. Any data in the module's database can be assigned to the high-speed Input File data using this map.

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.

Power Up

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

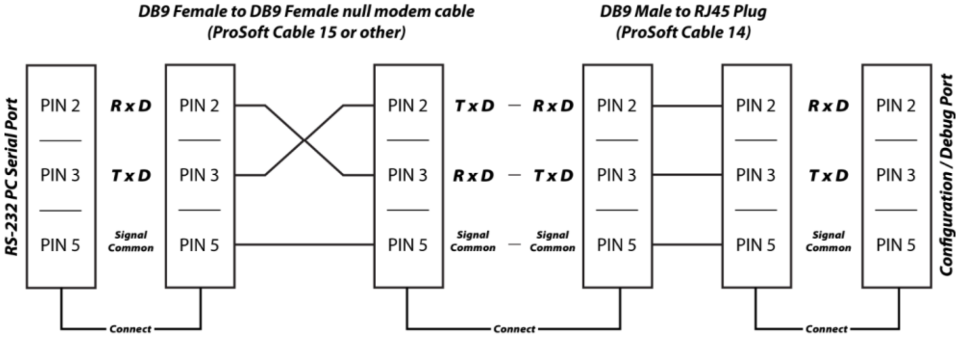
Changing Parameters During Operation

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 will 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 ControlLogix processor by executing a write command with a value of 9997 to register 4370 in the module's internal database. To cause 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.

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 pinout for communications on this port is shown in the following diagram.

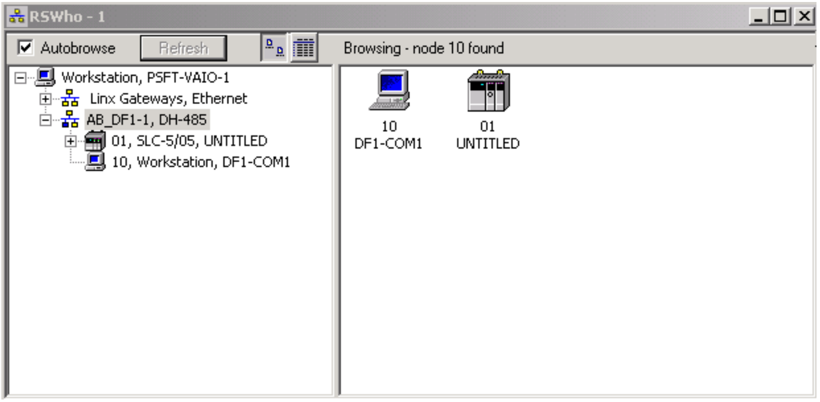


5.3.1 Disabling the RSLinx Driver for the Com Port on the PC

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.

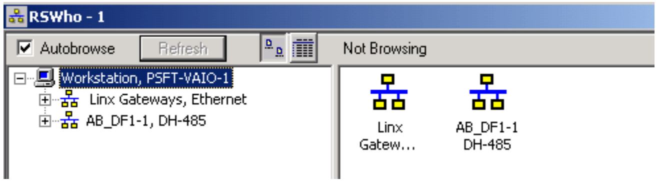
1 Open *RSLinx* and go to **COMMUNICATIONS** > **RSWHO**.

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

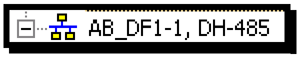


Browsing - node 10 found

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:

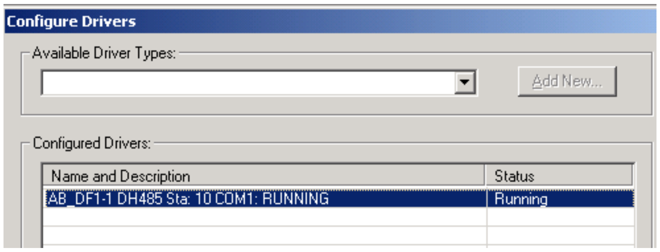


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



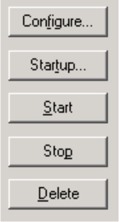
4 When you have verified that the driver is not being browsed, go to **COMMUNICATIONS** > **CONFIGURE DRIVERS**.

You may see something like this:

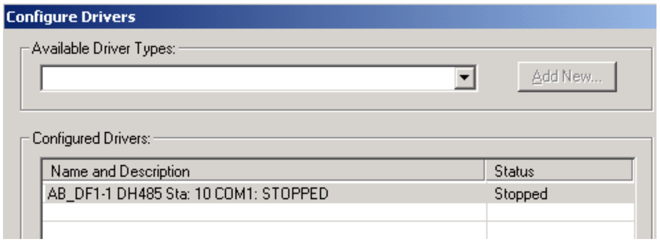


Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: RUNNING	Running

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** button on the side of the window:



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



Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: STOPPED	Stopped

6 You may now use the 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

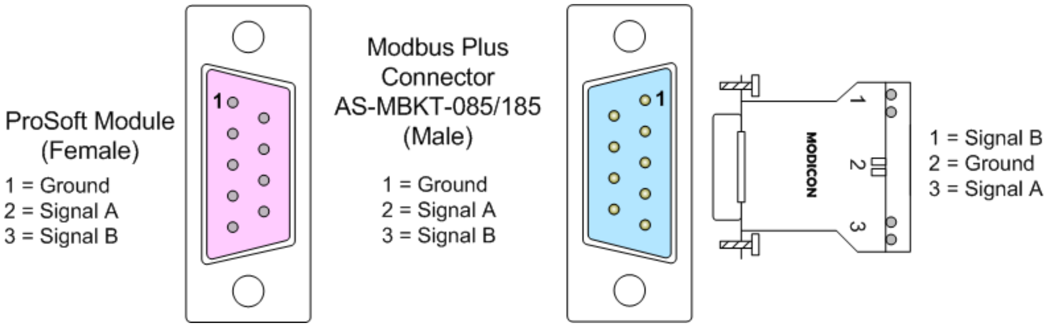
The MVI56-MBP module has one or two physical Modbus Plus connectors (depending on hardware configuration) with a DB-9 Female plug located on the front of the module.

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

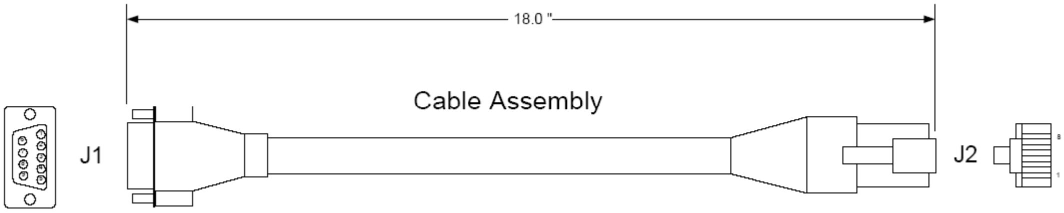
Modicon Part Number	Description
AS-MBKT-085	Inline Connector
AS-MBKT-185	Terminating Connector

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*.

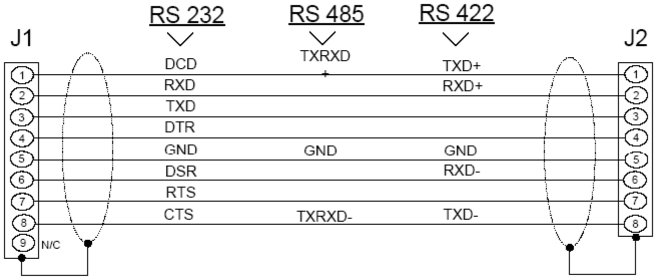
If the Modicon connectors are not available during installation, the following pin out applies to the DB-9 Modbus Plus port connections:



5.5 DB9 to RJ45 Adaptor (Cable 14)



Cable Assembly



Wiring Diagram

5.6 Database Definition

This section contains a listing of the internal database of the MVI56-MBP module. This information can be used to interface other devices to the data contained in the module.

5.6.1 Module Memory Map

Overview

Access	Module Address Range		Modbus Address Range		Description	Block Size
	Low	High	Low	High		
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 73). 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

5.6.2 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.

Array assignments for InUpdate member of MBPStat object

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
61	4061	Global In Update Stat - Device #62
62	4062	Global In Update Stat - Device #63
63	4063	Global In Update Stat - Device #64

Element	DB Address	Description
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	

Array assignments for MstrCmd member of MBPStat object

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	

Array assignments for Misc member of MBPStat object

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

Element	DB Address	Description
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
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
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)

Code	Description
0x10	Global data not configured (length set to zero)
0x20	Not assigned
0x40	Invalid Global Data Length (Length > 32 words)
0x80	Not assigned

Array assignments for InCntr member of MBPStat object

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
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

5.6.3 Configuration Data

This section contains listings of the MVI56-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.
4375	44376	Input File Update Timing	This parameter is not used by the MVI56-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.

DB Address	Modbus Address	Parameter	Description
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.
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 MVI56-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

DB Address	Modbus Address	Parameter	Description
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
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
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

DB Address	Modbus Address	Parameter	Description
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 MVI56-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	
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.

DB Address	Modbus Address	Parameter	Description
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

5.6.4 Global Output Data

Global Output Data transmitted from the MVI56-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	Parameter	Description
7090	47091	Global Out 0	Global output data for module word # 0
7091	47092	Global Out 1	Global output data for module word # 1
7092	47093	Global Out 2	Global output data for module word # 2
7093	47094	Global Out 3	Global output data for module word # 3
7094	47095	Global Out 4	Global output data for module word # 4
-	-		
7121	47122	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

5.6.5 Input File Data

The MVI56-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	Parameter	Description
7130	47131	In File Word 0	Data from Input File Map index 0 for word 0
7131	47132	In File Word 1	Data from Input File Map index 1 for word 1
7132	47133	In File Word 2	Data from Input File Map index 2 for word 2
7133	47134	In File Word 3	Data from Input File Map index 3 for word 3
-	-		
7161	47162	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

5.6.6 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
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

DB Address	Modbus Address	Parameter	Description
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

6 Support, Service & Warranty

In This Chapter

6.1 Contacting Technical Support

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

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

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

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

> **Note:** *For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues. Detailed contact information for all our worldwide locations is available on the following page.*

Asia Pacific	**Europe / Middle East / Africa**
Regional Office Phone: +603.7724.2080 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com **North Asia (China, Hong Kong)** Phone: +86.21.5187.7337 china@prosoft-technology.com Languages spoken: Chinese, English REGIONAL TECH SUPPORT support.ap@prosoft-technology.com **Southwest Asia (India, Pakistan)** Phone: +91.98.1063.7873 india@prosoft-technology.com Languages spoken: English, Hindi, Urdu **Australasia (Australia, New Zealand)** Phone: +603.7724.2080 pacific@prosoft-technology.com Language spoken: English **Southeast Asia (Singapore, Indonesia, Philippines)** Phone: +603.7724.2080 seasia@prosoft-technology.com Languages spoken: English, Bahasa, Tamil **Northeast & Southeast Asia** **(Japan, Taiwan, Thailand, Vietnam, Malaysia)** Phone: +603.7724.2080 neasia@prosoft-technology.com Languages spoken: English, Chinese, Japanese **Korea** Phone: +603.7724.2080 korea@prosoft-technology.com Languages spoken: English, Korean	Regional Office Phone: +33.(0)5.34.36.87.20 europe@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com **Middle East & Africa** Phone: +971.4.214.6911 mea@prosoft-technology.com Languages spoken: Hindi, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com **North Western Europe (UK, IE, IS, DK, NO, SE)** Phone: +44.(0)7415.864.902 nweurope@prosoft-technology.com Language spoken: English **Central & Eastern Europe, Finland** Phone: +48.22.250.2546 centraleurope@prosoft-technology.com Languages spoken: Polish, English, Russia & CIS Phone: +7.499.704.53.46 russia@prosoft-technology.com Languages spoken: Russian, English **Austria, Germany, Switzerland** Phone: +33.(0)5.34.36.87.20 germany@prosoft-technology.com Language spoken: English, German **BeNeLux, France, North Africa** Phone: +33(0)5.34.36.87.27 france@prosoft-technology.com Languages spoken: French, English **Mediterranean Countries** Phone: +39.342.8651.595 italy@prosoft-technology.com Languages spoken: Italian, English, Spanish

Latin America	**North America**
Regional Office Phone: +52.222.264.1814 support.la@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com **Brazil** Phone: +55.11.5084.5178 brasil@prosoft-technology.com Languages spoken: Portuguese, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com **Mexico** Phone: +52.222.264.1814 mexico@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com **Andean Countries, Central America & Caribbean** Phone: +507.6427.48.38 andean@prosoft-technology.com Languages spoken: Spanish, English **Southern Cone (Argentina, Bolivia, Chile, Paraguay & Uruguay)** Phone: +54.911.4565.8119 scone@prosoft-technology.com Languages spoken: Spanish, English	Regional Office Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com

6.2 Warranty Information

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

Documentation is subject to change without notice.

Index