Where Automation Connects.

MVI56-LTQ

ControlLogix Platform

Limitorque Valve Actuator Master Communication Module

August 5, 2021

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.

ProSoft Technology, Inc.

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

MVI56-LTQ User Manual
August 5, 2021

ProSoft Technology® Product Documentation

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

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.
North America: +1.661.716.5100
Asia Pacific: +603.7724.2080
Europe, Middle East, Africa: +33 (0) 5.3436.87.20
Latin America: +1.281.298.9109

For professional users in the European Union
If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Agency Approvals & Certifications

Please visit our website: 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:

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

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

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

ATEX Warnings and Conditions of Safe Usage:

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

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

Electrical Ratings

- Backplane Current Load: 800 mA @ 5.1 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% (without condensation)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Battery Life Advisory

The MVI46, MVI56, 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.

Contents

Guide to the MVI56-LTQ User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 11)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 37)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 55) Product Specifications (page 55) Functional Overview (page 58, page 53)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 87) Index	This section contains Support, Service and Warranty information. Index of chapters.

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 LTQ and ControlLogix devices to a power source and to the MVI56-LTQ module's application port(s)

Caution: You must be able to complete the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

1.1 System Requirements

The MVI56-LTQ 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-LTQ 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 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-LTQ module, and are all required for installation and configuration.

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

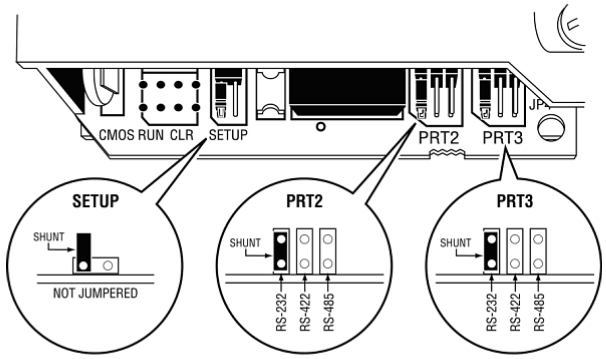
Qty.	Part Name	Part Number	Part Description
1	MVI56-LTQ Module	MVI56-LTQ	Limitorque Valve Actuator Master Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module

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

1.3 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to match the interface. There are three jumpers located at the bottom of the module.

The following illustration shows the MVI56-LTQ jumper configuration:



1. Set the PRT 2 (for application port 1) and PRT 3 (for application port 2) jumpers for RS232, RS422, or RS485 to match the wiring needed for your application. The default jumper setting for both application ports is RS-232.
2. The Setup Jumper acts as "write protection" for the module’s flash memory. In "write protected" mode, the Setup pins are not connected, and the module’s firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

1.4 Installing the Module in the Rack

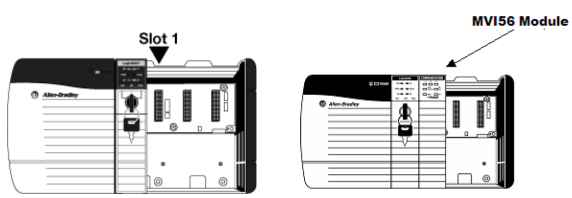
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-LTQ module. Refer to your Rockwell Automation product documentation for installation instructions.

> **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-LTQ 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.



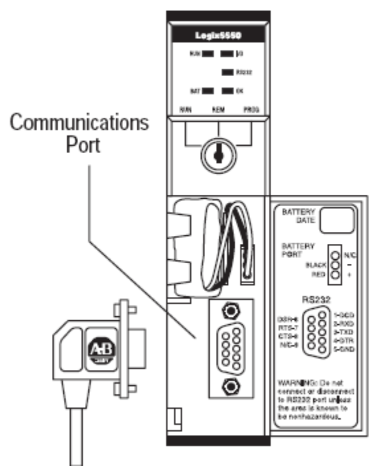
3. With a firm but steady push, snap the module into place.
4. Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5. Make a note of the slot location. You must identify the slot in which the module is installed in order for the sample program 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.

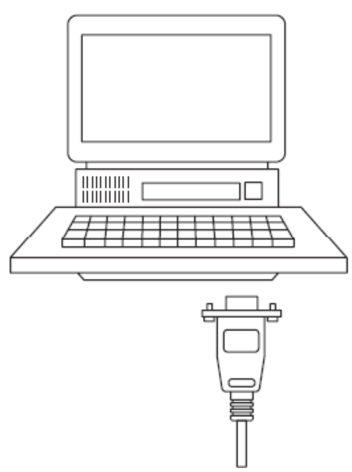
1.5 Connecting Your PC to the ControlLogix Processor

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. It is not mandatory that you use the processor's serial interface. You may access the processor through whatever 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 Opening the Sample Ladder Logic

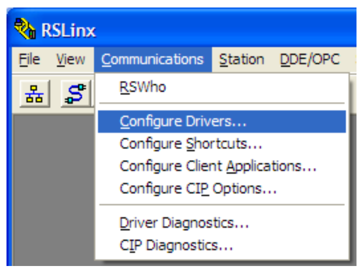
The sample program for your MVI56-LTQ module includes custom tags, data types and ladder logic for data I/O and status monitoring. For most applications, you can run the sample ladder program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

The version number appended to the file name corresponds with the firmware version number of your ControlLogix processor. The firmware version and sample program version must match.

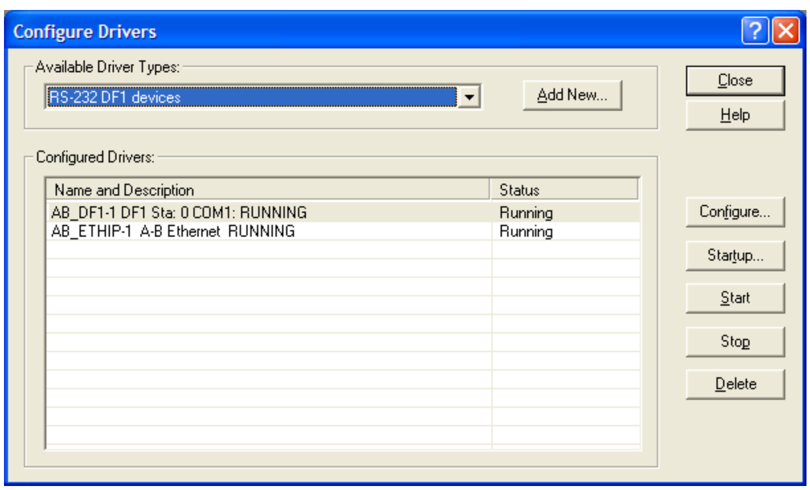
1.6.1 Configuring the RSLinx Driver for the PC COM Port

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

1. Open *RSLinx*.
2. Open the **COMMUNICATIONS** menu, and choose **CONFIGURE DRIVERS**.

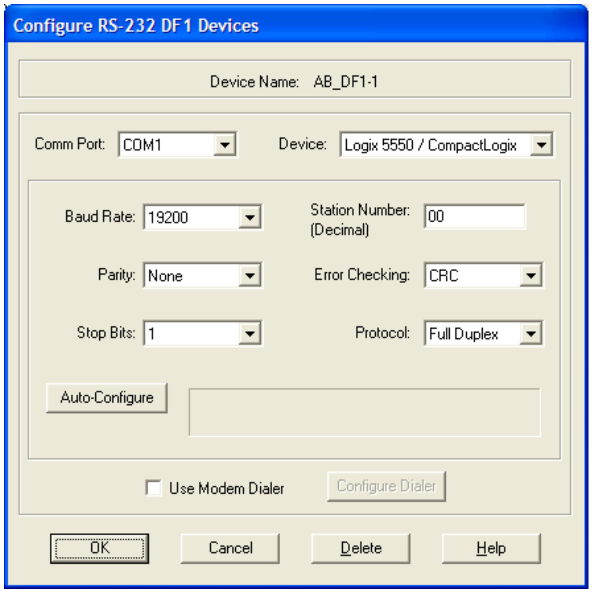


This action opens the *Configure Drivers* dialog box.



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

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



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

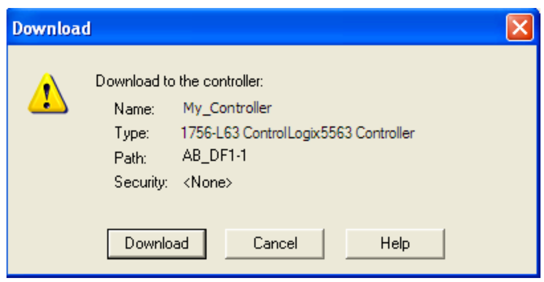
5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

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

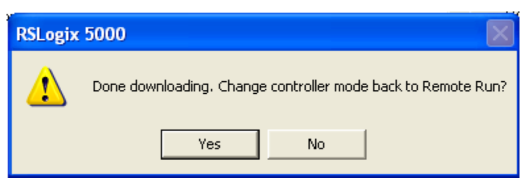
1.7 Downloading the Sample Program to the Processor

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, 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.



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.

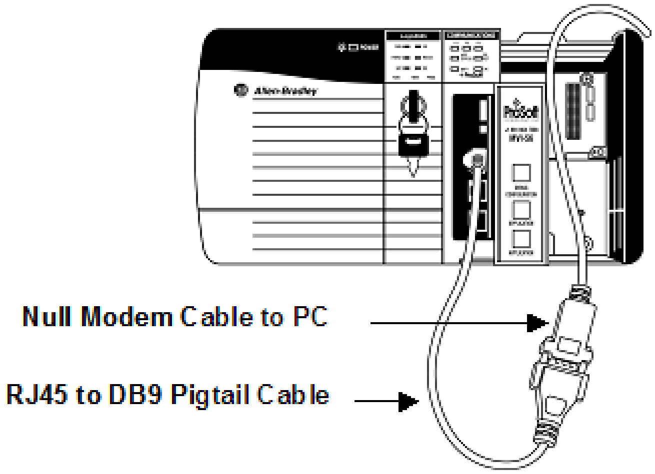


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.8 Connect your PC to the Module

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

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



2 Configuring the MVI56-LTQ Module

In This Chapter

This chapter gives you a summary of all possible configuration parameters and their allowable settings and ranges. For more detailed information on configuration and operation options, please refer to the *Reference* chapter.

2.1 Reference Documents

The following Limitorque supporting documents can assist you in deploying your network. These documents may be obtained from your local Limitorque representative or downloaded from the Flowserve Corporation website: http://www.flowserve.com/Products/Heritage-Brands/ch.Limitorque.literature

- Wiring Specifications
  http://www.flowserve.com/files/Files/Literature/ProductLiterature/FlowControl/Limitorque/440-15001.pdf
- Register mapping and communications
  http://www.flowserve.com/files/Files/Literature/Products/Flowcontrol/Limitorque/LMAIM4019.pdf
- Modbus information and wiring
  http://www.flowserve.com/files/Files/Literature/Products/Flowcontrol/Limitorque/LMAIM1329.pdf

2.2 Configuration Object (LTQCfg)

The configuration object LTQCfg contains all the configuration information required by the module. Fill in the members of this object to set the parameters required for each application. This data set will be transferred from the processor to the module each time a restart (warm or cold boot) operation is requested. 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 following table describes the structure of the object.

Name	**Data Type**	**Description**
BaudRate	INT	Baudrate for port (110 to 115.2K)
MsgRespTm	INT	Message Response Timeout
MaxSlaves	INT	Maximum number of slaves to consider
BTRMax	INT	Maximum BTR block count
BlkDelay	INT	Block transfer delay counter
LastState	INT	State for slave data on comm failure
NetPoll	INT	Network polling scheme (0=A/B, 1=A, 2=B)
PropDelay	INT	Propagation delay for messages
RTSOn	INT	RTS to TxD delay (milliseconds)
SpecPolling	INT	Special polling used (0=No, 1=T-MID, 2=Reg 6/7; other numbers of the range 1 to 65535 refer to the corresponding register).
Use_CTS	INT	Monitor CTS line (0=No, 1=Yes)
BlkFailCnt	INT	Number of failed transfers before COM shutdown
ActiveSlaves	INT[10]	Array to set active slaves (1-bit/slave)

Each parameter contained in this object is described in MVI56-LTQ Configuration Data Definition (page 23). Ensure that you fill in each parameter carefully for successful application of the module in a system.

2.3 Configuration Data

This section contains listings of the MVI56-LTQ module's configuration that is read from the ControlLogix processor when the module first initializes.

2.3.1 BaudRate

The baud rate at which the port is to operate. The available configurations are as follows:

Value	Baud Rate
1200	1200 Baud
2400	2400 Baud
4800	4800 Baud
9600	9600 Baud *
19200	19200 Baud
38400	38400 Baud
57600	57600 Baud
115200	115200 Baud

* Limitorque Field Unit Factory Default Setting

2.3.2 MsgRespTm

This register represents the message response timeout period in 1 millisecond increments. This is the time which a port configured as a Master will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending on the expected slave response times.

A value of 200 milliseconds should be the minimal setting. Values from 200 to 65535 (0xffff) are permitted.

2.3.3 MaxSlaves

This value is used by the module to optimize the number of slaves polled by the module. The value entered here can range from 1 to 150, and should always meet or exceed the last slave in the Active Slave Table.

2.3.4 BTRMax

This value is not used in the current version of the software.

2.3.5 BlkDelay

This is an empirical value used by the module to balance the amount of time the module spends block transferring and the amount spent handling port communications. The value entered is used as a loop counter in the module, where each time through the loop the count is incremented. When the count equals the Block Transfer Delay Counter a Block Transfer sequence is initiated.

Example: In Master Mode applications with the module in a remote rack, the frequency of command execution can be improved by entering a value of 75 to 150. The value must be determined empirically.

2.3.6 LastState

This value determines the state of the Limitorque read register values that are returned to the PLC upon the detection of a communication failure state (that is, comm has failed on both Port A and B).

Value	Description
0	Clear last data values (default)
1	Maintain last data values

2.3.7 NetPoll

Value	Description
0	Loop Mode (Port 1 and 2 alternating)
1	Port 1 polling only
2	Port 2 polling only

The Network Loop Mode emulates Limitorque's polling scheme that takes advantage of the actuator ability to repeat data transmissions and to operate in a looped mode. In this mode, the module will alternate communications between Port 1 and 2. Command failures on one port will be retried on the other port.

2.3.8 PropDelay

Provides a delay time between primary port polls to prevent network collisions on port changeover. Values should be no lower than the listed minimal settings. The value represents delay time in milliseconds.

Value	Number of Slaves
0	1 to 20
10	21 to 40
15	41 to 60
20	61 to 80
25	81 to 100
30	101 to 120
35	121 to 140
40	141 to 150

Note: These values are reference only. Empirical data gathered on site will enable proper adjustment of these values.

Slave #1 Channel A Fail bit (port 1) being true AND all other slave communications not in fault will be an indication of improper adjustment of this value.

2.3.9 RTSOn

This value represents the time in 1 millisecond increments for delay between asserting RTS and the actual transmission of data. Delay between the receipt of messages and transmit of new message must be greater than 10 milliseconds. When used, a value of 20 is typically inserted into this field.

Note: This value is reference only. Empirical data gathered on site will enable proper adjustment of these values.

2.3.10 SpecPolling

Enables polling of specific registers in addition to the standard polling. A value other than zero will cause an additional poll request to be sent to the slaves that are enabled. The results are placed in registers 8 and 9 in the slave response data block.

Using this feature has a performance cost as the time available for the standard polling is shared with the special polling.

Value	Description
0	Disabled
1	Register 55, TP_BEFORE_MID_T_HIGH
2	Registers 6/7, Analog Input 1 and 2
3 to 65535	Any value other than 0, 1, or 2 will tell the module to poll that corresponding register number. This been added to allow for expanded use of the SpecPolling tag. If low byte is set to 1 or 2 then the existing logic will will remain consistent for backwards compatibility.

2.3.11 Use_CTS

This parameter defines if the CTS line is to be monitored for the communication process. If the parameter is set to 0, the line will not be monitored. If the parameter is set to 1, the modem control line must be set after the RTS line is asserted for communications to occur.

2.3.12 BlkFailCnt

This parameter defines the number of successive block transfers that must occur before the communication channels are shutdown. If the value is set to 0, communications will continue on the ports even if the data transfer between the processor and module fails. If the value is set to greater-than 0, the module will suspend communications when the number of block transfer failures set is exceeded. The valid range for this parameter is 0 to 65535.

2.3.13 ActiveSlaves

These 10 words allow the user to configure the specific slaves that are active on a network. The intent of this table is to allow the user to selectively enable slave addresses and therefore not have to be concerned about activating slave addresses continuously.

All values are entered into the table in a right to left order with bit 0 representing the lower address. The slave addresses are mapped into the table as follows:

Index	Description
0	Slaves 1 to 16
1	Slaves 17 to 32
2	Slaves 33 to 48
3	Slaves 49 to 64
4	Slaves 65 to 80
5	Slaves 81 to 96
6	Slaves 97 to 112
7	Slaves 113 to 128
8	Slaves 129 to 144
9	Slaves 145 to 150

3 Ladder Logic

In This Chapter

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

The sample ladder logic 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 (LTQModuleDef)

All data related to the MVI56-LTQ 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
Cfg	LTQCfg	Module Setup information
Stat	LTQStat	Status information in each read block
BP	LTQBackplane	Data to handle backplane logic
Valve	LTQValve[150]	Valve data
DoneBits	INT[10]	Done bit data for commands
Open	INT[10]	Open Command bits
Stop	INT[10]	Stop Command bits
Close	INT[10]	Close Command bits
InitESD	INT[10]	Initiate ESD bits
TermESD	INT[10]	Terminate ESD bits
Reset	INT[10]	Reset Command bits
Contactors	LTQCont[6]	Contactors #1 to #6 engage/disengage bits
AnalogCmd	INT[10]	Command valve to set analog position bits
AnalogPos	INT[150]	Analog valve positions (0-100) to be used for set cmd

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

3.1.1 Status Object (LTQStat)

This object views the status of the module. The **LTQStat** 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
Product	INT[2]	Product code
Revision	INT[2]	Revision
OpSys	INT[2]	Operating system code
Run	INT[2]	Run number
PassCnt	INT	Program scan counter
Queue	INT	Number of entries used in command queue
BTR_Cnt	INT	Number of BTR's
BTW_Cnt	INT	Number of BTW's
BlkParse	INT	Number of BTW blocks parsed
BlkEvent	INT	Number of event commands received
BlkErr	INT	Number of block transfer errors
CurSlave	INT	Current slave number being processed
CurPort	INT	Active port used on module (0=A, 1=B)
AltPort	INT	Other port being tried after failure (0=No, 1=Yes)
SpecPoll	INT	Special Poll (0=No, 1=T_M, 2=Reg 6/7, 3 to 65535=corresponding register value)
State	INT	Communication state machine value
ComState	INT	Communication state machine for port activity
CfgErr	INT	Module Configuration Error

Refer to MVI56-LTQ Status Data Definition (page 76) for a complete listing of data stored in this object.

3.1.2 User Data Objects

These objects hold data to be transferred between the processor and the MVI56-LTQ module. The first user data object is the LTQBackplane object that contains the variables required by the ladder logic program for data transfer between the module and the processor. The following table describes the structure of this object.

Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table

Values in this structure are used by the ladder logic and initialized by the power-up function. The values in this block should change very rapidly as the transfer process is very fast.

The last set of user data is shown in the following illustration:

Name	Data Type	Description
Valve	LTQValve[150]	Valve data
DoneBits	INT[10]	Done bit data for commands
Open	INT[10]	Open Command bits
Stop	INT[10]	Stop Command bits
Close	INT[10]	Close Command bits
InitESD	INT[10]	Initiate ESD bits
TermESD	INT[10]	Terminate ESD bits
Reset	INT[10]	Reset Command bits
Contactors	LTQCont[6]	Contactors #1 to #6 engage/disengage bits
AnalogCmd	INT[10]	Command valve to set analog position bits
AnalogPos	INT[150]	Analog valve positions (0-100) to be used for set cmd

The value array stores data transferred from the module to the processor. The following table describes the structure of each element in the array.

Name	Data Type	Description
ValvePos	INT	Position of valve 0-100%
Status	INT	Status Register BIts
Fault	INT	Fault Register Bits
DOut	INT	Digital Output Bits
DIn1	INT	Digital Input 1 bits
DIn2	INT	Digital Input 2 bits
ComStatus	INT	Communication Status Code
ComCntr	INT	Communication poll counter (0-32767) for success
SPoll1	INT	Special polling register #55
SPoll2	INT	Special polling register #6 and #7
SPoll3-SPoll65535	INT	Special Polling register #3 to #65535

Information in this structure contains the status and monitor information for each valve in the system. This information should be considered by the user control logic to determine the status and control required for each valve to be utilized in a system. It is important to note that slave 1 has an array index of 0. This offset applies to all valves (slave 150 has an index of 149).

The next data set received from the module is the DoneBits array. Each slave in the system is associated with a bit in the array. Each element contains the done bit status for 16 slave units ([0]=slaves 1 to 16, [1]=slaves 17 to 32, and so on). It is important to use these bits to clear the commands issued using the next data sets. Commands issued to the module to generate messages on the Limitorque Valve Network are issued in a one-shot operation. After an event command is issued to the slave, the done bit for the slave is set. This bit is held high for one scan of the module and then is reset in the data transfer block. Ladder logic should clear the commands for each slave using the status of the done bit.

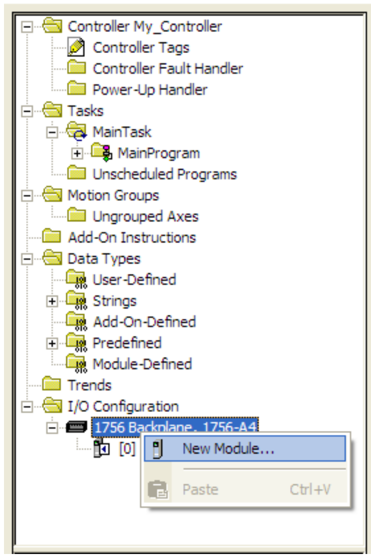
The last set of data in the module object contains the control information for the valves. This data is to be controlled by the ladder logic. An array is defined for each of the command instructions available through the module. Each bit in the array is associated with a slave. Bit 0 in word 0 corresponds to slave 1, and Bit 0 in word 1 corresponds to slave 17. An array of LTQCont objects is defined for the contactor engage/disengage commands. The following table describes the structure of this object.

Name	**Data Type**	**Description**
Engage	INT[10]	Engage Command
Disengate	INT[10]	Disengage Command

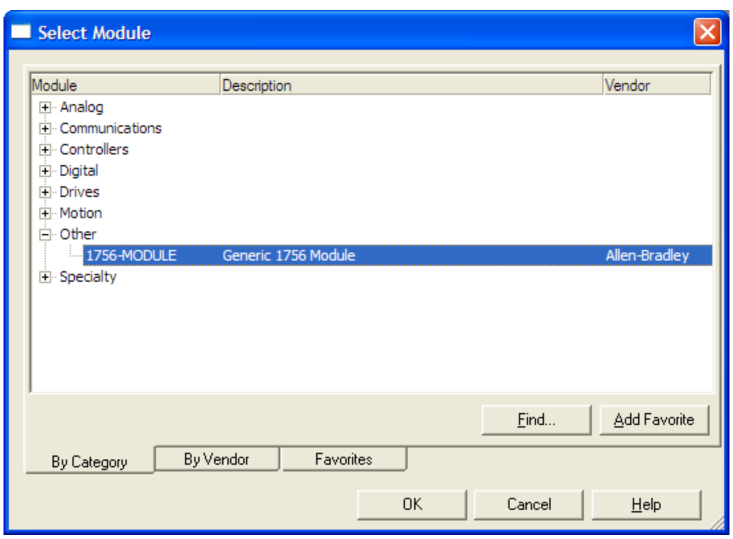
In order to execute one of the commands, set the bit for the selected slave in the array. For example, to open slave-3 valve, set bit 2 in word 0 of the Open array. For the analog control commands, set the value for the analog first (AnalogPos[ ]), then set the analog control array bit (AnalogCmd[ ]). The analog position requires two commands. The first command issued by the module sets the position in the valve. The second command instructs the valve to seek the new position.

3.2 Adding the Module to an Existing Project

1 Select the *I/O Configuration* folder in the *Controller Organization* window of RSLogix 5000, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE**.



This action opens the *Select Module* dialog box:

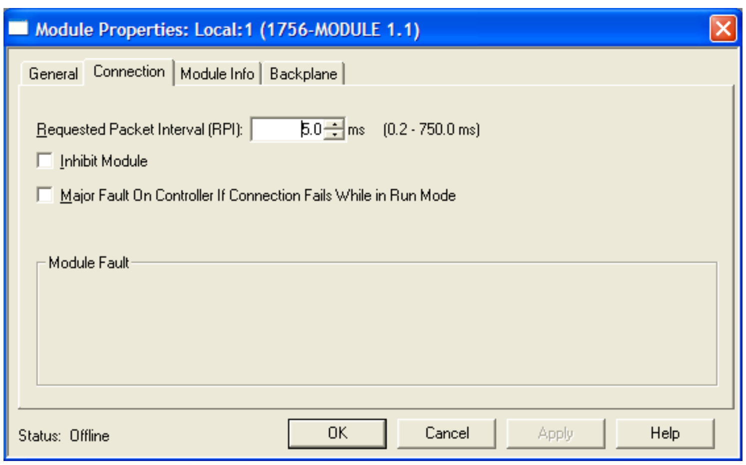


2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK**. This action opens the *New Module* dialog box.

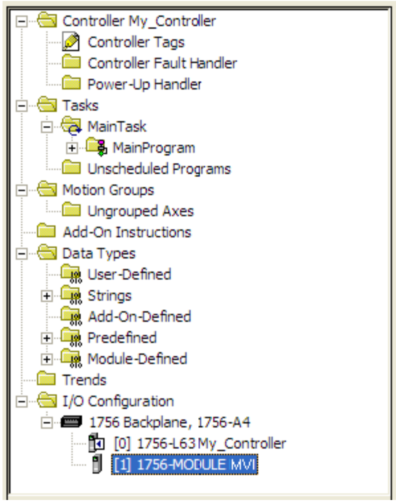
3 Enter the *Name, Description* and *Slot* options for your application. You must select the *Comm Format* as **DATA - INT** in the dialog box, otherwise the module will not communicate. Click **OK** to continue.

Parameter	**Value**
Name	Enter a module identification string. Example: LTQ_2.
Description	Enter a description for the module. Example: Limitorque Valve Actuator Master Communication Module
Comm Format	Select **DATA-INT**.
Slot	Enter the slot number in the rack where the MVI56-LTQ 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.



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



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 module 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.
- Status Data contained in the module can be viewed through the Configuration/Debug port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- Status data values can be transferred from the module to processor memory and can be monitored there manually or by customer-created logic. For details on Status Data values, see MVI56-LTQ Status Data Area.

4.1 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.
P1	Green	On	Data is being transferred between the module and the Limitorque network on Port A.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the Limitorque network on Port B.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI56-LTQ is working normally.
		On	The MVI56-LTQ module program has recognized a communication error on one of its ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

During module configuration, the OK LED will be red and the APP and BP ACT LEDs will be on. The bits in the configuration word are shown in the following table. The module configuration error word has the following definition:

Bit	Description	Value
0	Invalid baud rate specified.	0x0001
1	Message response timeout < 200.	0x0002
2	Maximum slave count > 150.	0x0004
3	Maximum slave count = 0.	0x0008
4		0x0010
5	Invalid network polling scheme value (must be 0, 1 or 2).	0x0020
7		0x0080
8		0x0100
9		0x0200
10		0x0400
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

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

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

4.1.1 Clearing a Fault Condition

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

To clear the condition, follow these steps:

1 Turn off power to the rack.
2 Remove the card from the rack.
3 Verify that all jumpers are set correctly.
4 If the module requires a Compact Flash card, verify that the card is installed correctly.
5 Re-insert the card in the rack and turn the power back on.
6 Verify correct configuration data is being transferred to the module from the ControlLogix controller.

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

4.1.2 Troubleshooting

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

Processor Errors

Problem description	Steps to take
Processor fault	Verify that the module is plugged into the slot that has been configured for the module in the I/O Configuration of 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-LTQ. Verify that all modules in the rack are correctly configured in the ladder logic.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56E 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 port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert it, and then restore power to the rack.

4.1.3 Communication Error Codes

The Error Codes returned from the module represent the outcome of the commands and responses executed by the module. This data value is returned for each slave at word offset 6. Note that in all cases, if a zero is returned, there is no current error. Valid Error Status Codes are as follows:

Note: These error codes are used for communication module diagnostics. For programming purposes, use the Slave Data Table (Slave #x Response Data Word 1 bits 7, 10, and 11) for determining slave communication status.

Code	Name	Description
0	All ok	The module is operating as desired or is currently being polled or commanded.
1	Illegal Function	An illegal function code request is being attempted.
2	Bad Data Address	The address or range of addresses, covered by a request from the master are not within allowed limits.
3	Bad Data Value	The value in the data field of the command is not allowed.
6	Module Busy	The module busy status code is returned when a write command from the master has not yet been completed when a second write command is received.
-1	CTS Line Error	The module is monitoring the CTS line and it was not asserted after the RTS line was asserted. The message was not transmitted on the port.
-2	Send Timeout	The message was not completely transmitted on the port within the 5-second message send timeout period.
-3	No Message	The command selected did not produce a message (Bad command format or illegal instruction).
-10	Long Message	The message received on the port is too long.
-11	Response Timeout	Timeout condition while waiting for response message.
253	Wrong Slave	The response message received on the port is not that in the request message.
254	Wrong Function Code	The function code in the response message does not match that in the last request message.
255	CRC Error	The CRC value returned in the message does not match the computed CRC of the packet.

4.2 Using the Configuration/Debug Port

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

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

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

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

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

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

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

4.3 Reading Status Data from the Module

The MVI56-LTQ module returns a 22-word Status Data Block that can be used to determine the module's operating status. This data is transferred to the ControlLogix processor continuously with each read block.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Complete display of the module's internal database (registers 0 to 3999)
- Version Information
- Control over the module (warm boot, cold boot, transfer configuration)
- Facility to upload and download the module's configuration file

4.3.1 Required Software

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

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

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

4.3.2 The Configuration/Debug Menu

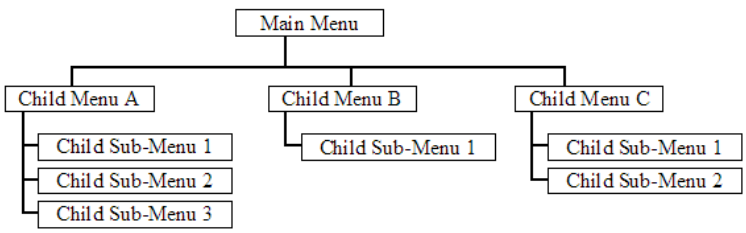
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

Navigation

All of the submenus 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 the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

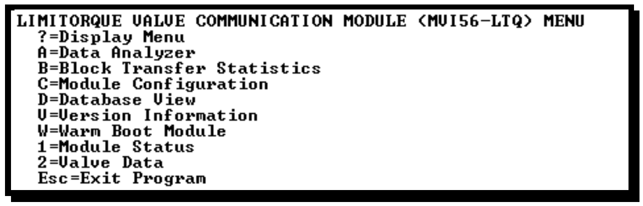
The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

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

Also, take care to distinguish the different uses for uppercase letter "eye" (**I**), lowercase letter "el" (**L**), and the number one (**1**). Likewise, uppercase letter "oh" (**O**) and the number zero (**0**) are not interchangeable. Although these characters look alike on the screen, they perform different actions on the module and may not be used interchangeably.

4.3.3 Main Menu

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



```
LIMITORQUE VALVE COMMUNICATION MODULE (MVI56-LTQ) MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  V=Version Information
  W=Warm Boot Module
  1=Module Status
  2=Valve Data
  Esc=Exit Program
```

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

Opening the Data Analyzer Menu

Press **[A]** to open the Data Analyzer Menu. Use this command to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Refer to Data Analyzer (page 47) for more information about this menu.

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

Viewing Block Transfer Statistics

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

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

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

Viewing Module Configuration

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

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

Opening the Database View Menu

Press **[D]** to open the *Database View* menu.

Use this menu command to view the current contents of the module's database. For more information about this submenu, see Database View Menu (page 52).

Viewing Version Information

Press **[G]** 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

Press **[W]** from the *Main* menu to warm boot (restart) the module.

This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to reboot.

Viewing Module Status

Press **[1]** to view information about Module Status. Use this command to view status information about the module. This screen also contains useful information for mailbox troubleshooting:

- Scan count
- Mailbox counters
- Alarm counters
- Number of acyclic read and write operations performed by the module.

You can also view the number of mailbox messages in the input and output queues, and the number of alarms in the alarm queue.

Opening the Valve Data menu

Press **[2]** to display the data values for each valve. Refer to the Reference chapter for a complete listing of this structure.

Exiting the Program

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

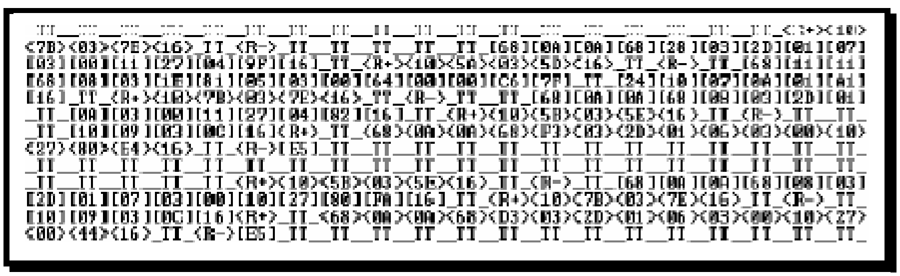
4.3.4 Data Analyzer

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

> Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.
> Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

Press **[1]** to display I/O data for the first application port in the Data Analyzer. The following illustration shows an example of the Data Analyzer output.



Analyzing Data for the second application port

Press **[2]** to display I/O data for the second application port in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

You can display timing marks for a variety of intervals in the data analyzer screen. These timing marks can help you determine communication-timing characteristics.

Key	Interval
[5]	1 milliseconds ticks
[6]	5 milliseconds ticks
[7]	10 milliseconds ticks
[8]	50 milliseconds ticks
[9]	100 milliseconds ticks
[0]	Turn off timing marks

Removing Timing Marks in the Data Analyzer

Press **[0]** to turn off timing marks in the Data Analyzer screen.

Viewing Data in Hexadecimal Format

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

Viewing Data in ASCII (Text) Format

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

Starting the Data Analyzer

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

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

The Data Analyzer displays the following special characters:

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

Stopping the Data Analyzer

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

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

Returning to the Main Menu

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

4.3.5 Data Analyzer Tips

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

```
Data Analyzer Mode Selected
```

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

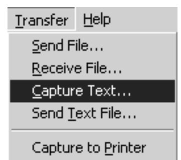
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

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

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

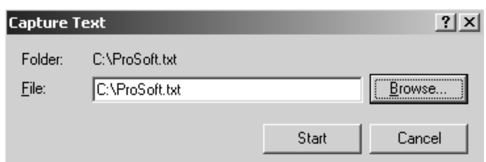
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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

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



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



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

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

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

```
[03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+><01><02>
<00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT__TT_<R+>
<01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT__TT_<R+>
<01><01><00><00><00><A0><3C><72><R->_TT__TT_[01][01][14][00][00][01][00][02]_TT_
[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]_TT__TT_<R+>
<01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+>
<01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT_
_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01]
[00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT_
_TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00][00][01]
[00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]
_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00]
[01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]
_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT_[01][02][02][00][00][B9]
[B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00]
[00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD]
[51]_TT__TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00]
[00][01][00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00]
[B7][52]_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00]
[00][00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09]
[FB][B7]_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02]
[00][00][B9][B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT__
```

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

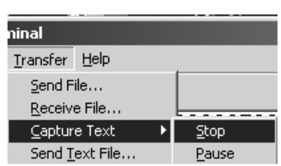
All characters shown in <> brackets are characters being sent out by the module.

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

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

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

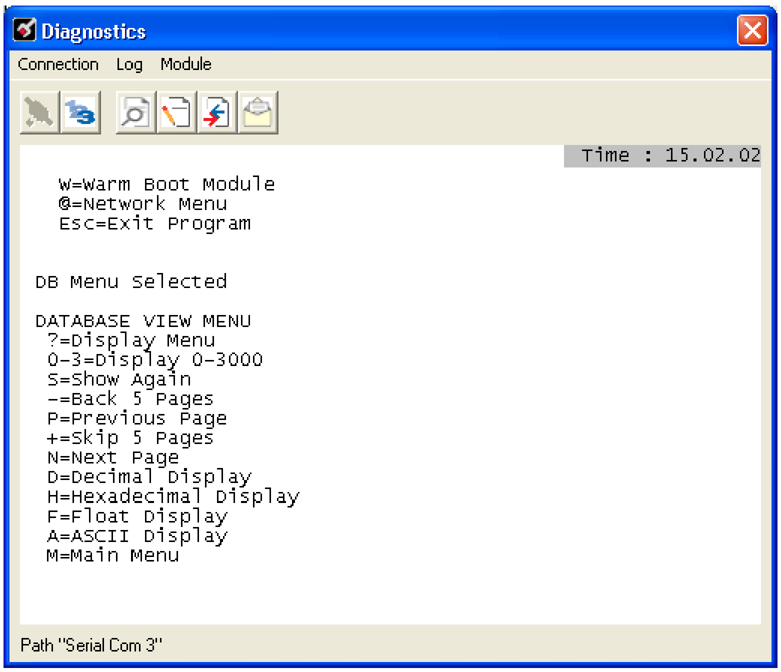
When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



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

4.3.6 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 database. Press **[?]** to view a list of commands available on this menu.



Viewing Register Pages

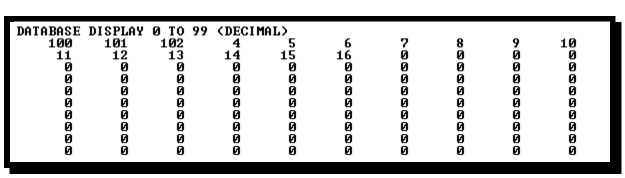
To view sets of register pages, use the keys described below:

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

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

Displaying the Current Page of Registers Again

Press **[S]** from the *Database View* menu to show the current page of registers again.



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

Moving Back Through 5 Pages of Registers

Press **[-]** from the *Database View* menu to skip five pages back in the database to see the 100 registers of data starting 500 registers before the currently displayed page.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the *Database View* menu to skip five pages ahead in the database to see the 100 registers of data starting 500 registers after the currently displayed page.

Viewing the Previous 100 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating-Point Format

Press **[F]** from the *Database View* menu 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]** from the *Database View* menu 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.

4.3.7 Valve Data Menu

Press **[2]** to display the data values for each valve. Refer to the Reference chapter for a complete listing of this structure.

Viewing Valve Registers

0 (zero) to E = Display 0 to 140

This command is used to jump to a specific set of registers in the database and displays the data. The keys perform the following functions:

Key	Function
0	Display valve 1
1	Display valve 11
2	Display valve 21
3	Display valve 31
4	Display valve 41
5	Display valve 51
6	Display valve 61
7	Display valve 71
8	Display valve 81
9	Display valve 91
A	Display valve 101
B	Display valve 111
C	Display valve 121
D	Display valve 131
E	Display valve 141

Redisplaying the Current Page

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

Viewing the Next Page of Data

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

Viewing the Previous Page of Data

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

Returning to the Main Menu

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI56 Limitorque Valve Actuator Master Communication Module allows ControlLogix compatible processors to interface easily with Limitorque Valve Actuators and other Limitorque protocol compatible devices.

The MVI56-LTQ module acts as an input/output module between the Limitorque valve network and the ControlLogix processor. The data transfer from the ControlLogix processor is asynchronous from the actions on the network. The 1500-word register space in the module transfers the valve information from the module to the processor. Ladder logic is responsible for control of the valves. Data transferred from the processor to the module instructs the module to execute commands to valves on the Limitorque network.

5.1.1 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. Sample ladder file included

5.1.2 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 VDC 3 mA @ 24 VDC
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) Full hardware handshaking control, providing radio, modem and multi-drop support	
Software configurable communication parameters	Baud rate: 110 to 115,200 baud, depending on protocol RS-232, 485 and 422 Parity: none, odd or even Data bits: 5, 6, 7, or 8 Stop bits: 1 or 2 RTS on/off delay: 0 to 65535 milliseconds
App Ports (P1, P2) (Serial modules)	RJ45 (DB-9M with supplied cable) RS-232 handshaking configurable 500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port 6-foot RS-232 configuration cable

5.1.3 Functional Specifications

- Support for the storage and transfer of up to 150 valves to the ControlLogix processor's controller tags
- Emulates Limitorque's Port A/B polling scheme using both ports on the module
- Supports the following valves
  - MX/DDC Modbus
  - UEC-3-DDC Modbus
  - DDC-100M I/O module
  - DDC-100M field unit
  - Valvcon IVO (unit in multi-drop mode only)
- Software configuration
  - Baud rate: 1200 to 115200
  - Message response timeout
  - Number of active slaves
  - Last state on communication failure for valve data
  - Network polling scheme
  - Active slave table (bit mapped)
  - Use of CTS module line option
  - Backplane failure affect on communication port availability
- Support command
  - Continuously polled
  - Read registers 40008 to 40013, optional 40055 or 40006/40007
  - Commands: open, stop, close, initiate network ESD, terminate network ESD, engage contactors 1 to 6, disengage contactors 1 to 6, position valve (0 to 100%)
- Data returned to the processor for each valve
  - Valve position
  - Status register
  - Fault register
  - Digital outputs
  - Digital inputs registers 1 and 2
  - Communication error code
  - Communication poll counter
  - Special polled registers

5.2 Functional Overview

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding module operation.

ControlLogix Processor Not in Run

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

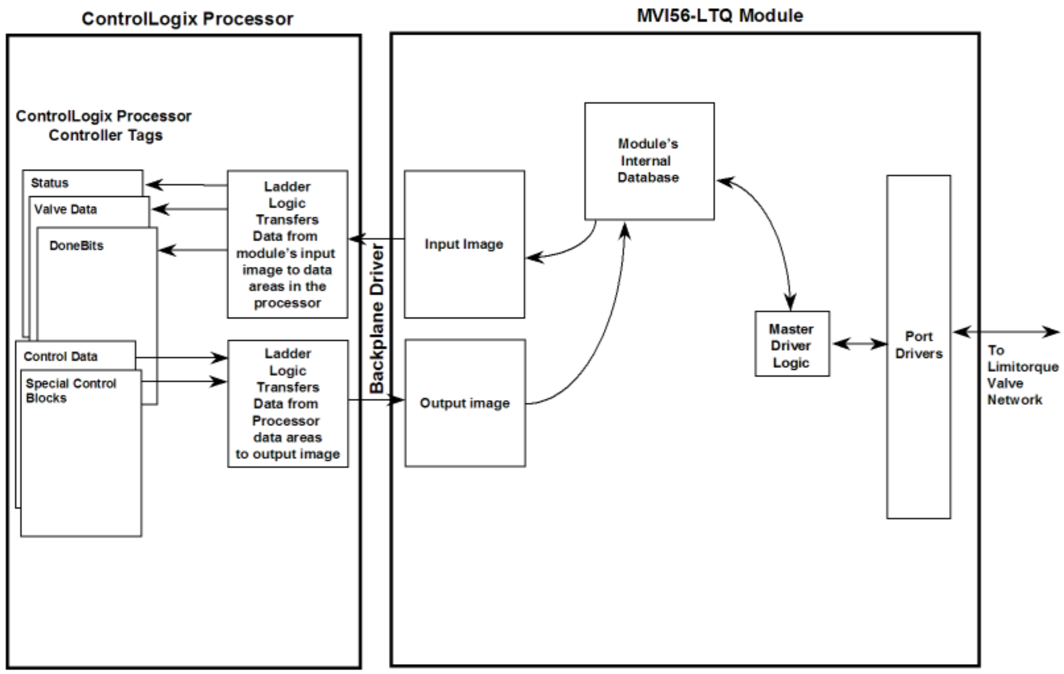
Backplane Data Transfer

The MVI56-LTQ 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 update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 1 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is 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-LTQ module and the Limitorque valve 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. The following illustration shows the layout of the database:

Module's Internal Database Structure

1500 registers for user data	Register Data	0 1499
2500 words of control data	Control Data	1500 3999

Data contained in this database is paged through the input and output images by coordination of the ControlLogix ladder logic and the MVI56-LTQ 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. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The block identification codes used by the module are listed below:

Block Number	Description
0	On read, this block contains the valve information for slaves 1 to 20. For the write operation, this block contains the digital command control data.
1	On read, this block contains the valve information for slaves 21 to 40. For the write operation, this block contains the analog command control data.
2 to 7	These blocks are only used for the read operation. They contain valve data for up to 20 slave units.
9000	This is the configuration data block transferred from the processor
9998	This block is sent from the processor to the module to instruct the module to perform the warm-boot operation.
9999	This block is sent from the processor to the module to instruct the module to perform the cold-boot operation.

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

Normal Data Transfer

Normal data transfer includes the paging of the valve data found in the module's internal database in registers 0 to 1499, the status data and the done bit data from the module to the processor. These data are transferred through read (input image) blocks 0 to 7. Data transferred from the processor to the module is through blocks 0 and 1. The structure and function of each block is discussed in the following topics.

Read Block

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

Word	Description
0	Reserved
1	BTW ID Requested (0 or 1)
2 to 201	Valve Data (20 valves each block)
202 to 211	Done Bit Data
212 to 213	Product Code
214 to 215	Revision
216 to 217	Operating System
218 to 219	Run Number
220	Pass Count
221	Queue Size
222	BTR Count
223	BTW Count
224	BT Parse Count
225	BT Event Count
226	BT Error Count
227	Current Slave
228	Current Port
229	Alt Port
230	Special Poll Request
231	State
232	Comm State
233	CfgErr
234 to 248	Spare
249	BTR ID (0 to 7)

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 representing the data for up to 20 valves. In addition to moving valve data, the block also contains status data for the module. This last set of data is transferred with each new block of data and is used for high-speed data movement.

The LTQ Module returns Done Bit Data to the ladder logic. A single bit is returned per slave address, allowing ladder logic to be used to clear the Command Enable bits. The following important points should be noted about the Command Done bits:

1 There is only one bit returned per slave address, not one bit per command per slave. The implication of this is that one Done Bit must be used to clear all possible Enable bits for one slave address.

2 The Done Bit is a positive indication that the module executed the command. It is NOT an indication of the command's success. A Done Bit is returned to the ladder logic whenever the command was completed without error or not. This allows all commands to be unlatched. To determine if there is a communication problem with a slave, verify the Channel A/B communication status bits in the valve data Status field.

3 The Done Bit data registers in the module are cleared and then updated prior to each backplane transfer sequence. This is done to ensure that the ladder logic receives the quickest possible acknowledgment of a command's execution.

The structure of the Done Bit Data in the input image from the module is as follows:

Word	Description
202	Cmd Done - Slaves 1 to 16
203	Cmd Done - Slaves 17 to 32
204	Cmd Done - Slaves 33 to 48
205	Cmd Done - Slaves 49 to 64
206	Cmd Done - Slaves 65 to 80
207	Cmd Done - Slaves 81 to 96
208	Cmd Done - Slaves 97 to 112
209	Cmd Done - Slaves 113 to 128
210	Cmd Done - Slaves 129 to 144
211	Cmd Done - Slaves 145 to 150

The following tables can be used to determine the slave associated with each bit in the array. These tables can also be used with the digital control words discussed in the following section. The following table lists the relationship between the word/bit combination and the slave address on the Limitorque valve network:

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex Value	0x0001	0x0002	0x0004	0x0008	0x0010	0x0020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x1000	0x2000	0x4000	0x8000
Word																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
2	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
3	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
4	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
5	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
6	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
7	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
8	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
9	145	146	147	148	149	150										

The following table lists the relationship between the word/bit combination and the slave index in the module:

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex Value	0x0001	0x0002	0x0004	0x0008	0x0010	0x0020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x1000	0x2000	0x4000	0x8000
Word																
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149										

The module status data starts at word 212 in the read block. This information can be used to determine the "health" of the module and should be moved to the status object in the processor logic.

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

`R0W0→R1W1→R2W0→R3W1→R4W0→R5W1→R6W0→R7W1→R0W0`

This sequence will continue until interrupted by other write block numbers sent by the controller or by operator control through the module’s Configuration/Debug port.

Write Block

These blocks of data transfer control information from the ControlLogix processor to the module. This control information issues commands to valves on the network. Two blocks are defined in the module for this operation; Block 0 and Block 1. The structure of the output image used for block 0 is shown below:

Word	Description
0	BTW ID 0
1 to 10	Open
11 to 20	Stop
21 to 30	Close
31 to 40	Initiate ESD
41 to 50	Terminate ESD
51 to 60	Reset
61 to 70	Engage Contactor #1
71 to 80	Disengage Contactor #1
81 to 90	Engage Contactor #2
91 to 100	Disengage Contactor #2
101 to 110	Engage Contactor #3
111 to 120	Disengage Contactor #3
121 to 130	Engage Contactor #4
131 to 140	Disengage Contactor #4
141 to 150	Engage Contactor #5
151 to 160	Disengage Contactor #5
161 to 170	Engage Contactor #6
171 to 180	Disengage Contactor #6
181 to 247	Spare

This data set controls the digital command set of the modules. Each 10 words for the command types represent the 150 valves in the system with the data stored as a bit-mapped image. For example, in order to perform an open command on slave 1, word 0-bit 0 is used in the open command array. In order to control the analog values, block 1 is used. The format of block 1 is shown below:

Word	Description
0	BTW ID 1
1 to 10	Analog Write Enable
11 to 160	Analog Values
161 to 247	Spare

The Analog Write Enable words are bit-mapped values with each bit representing one of the 150 valves in the system. When one of the bits is set, the associated value in the Analog Values array is passed to the selected valve on the network.

Configuration Data Transfer

When the module performs a restart operation, it requests configuration information from the ControlLogix processor. This data is transferred to the module in specially formatted write block (output image). The module will poll for the block by setting the required write block number in a read block (input image). Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required. The following topics show the format of the blocks for configuration.

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 following table shows the structure of this block.

Word	Description
0	BTW ID 9000
1	Baud Rate
2	Message Response Timeout
3	Max Number of Slaves
4	Read Data Block Count
5	Block Transfer Delay
6	Last State of Comm Fail
7	Network Polling Scheme
8	Propagation Delay
9	RTS to TxD delay
10	Special Polling
11	Active Slave Table Word 0
12	Active Slave Table Word 1
13	Active Slave Table Word 2
14	Active Slave Table Word 3
15	Active Slave Table Word 4
16	Active Slave Table Word 5
17	Active Slave Table Word 6
18	Active Slave Table Word 7
19	Active Slave Table Word 8
20	Active Slave Table Word 9
21 to 247	Spare

The read block used to request the configuration has the following structure.

Word	Description
0	Reserved
1	BTW ID 9000
2 to 248	Reserved
249	BTR ID -2 or -3

If there are any errors in the configuration, the bit associated with the error is set in the module configuration error word. The error must be corrected in order for the user configuration to be implemented without error.

Command Control Blocks

Command control blocks are special blocks used to control the module or request special data from the module. The current version of the software supports two command control blocks; warm boot and cold boot.

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. The block is commonly sent to the module any time configuration modifications are made in the controller tags data area. This causes the module to read the new configuration and restart. The following table shows this structure:

Offset	Description	Length
0	9998	1
1 to 247	Spare	247

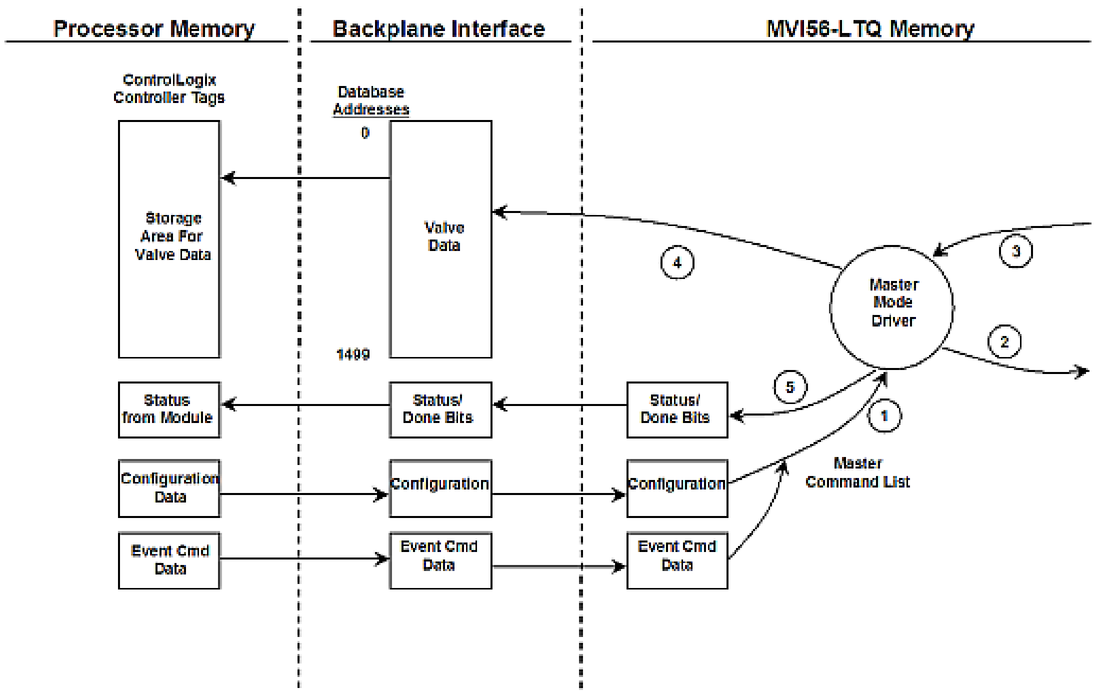
Cold Boot

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 control block.

Offset	Description	Length
0	9999	1
1 to 247	Spare	247

5.2.2 Data Flow Between MVI56-LTQ Module and ControlLogix Processor

The following topics describe the flow of data between the two pieces of hardware (ControlLinx processor and the MVI56-LTQ module) and nodes on the Limitorque valve network. In the Master mode, the MVI56-LTQ module is responsible for issuing read or write commands to slave devices on the Limitorque valve network. These commands are generated from a fixed set of commands contained in the module. The module issues the read commands continuously. Write commands are issued under ladder logic control. The following flow chart and associated table describe the flow of data into and out of the module.



1. The module obtains configuration data from the ControlLogix processor. This configuration completely defines the operation parameters for the module.
2. After configuration, the module begins transmitting read commands to the valves on the network. If an event command (a bit in one of the digital control word arrays is set) is recognized by the module, it will issue the associated write command to the valve. The done bit for the event will be set in the DoneBit array.
3. Presuming successful processing by the node specified in the command, a response message is received into the driver for processing.
4. Data received from the node on the network is passed into the module's internal database (only for read function messages).
5. Status and done bit array data is returned to the ControlLogix processor.

Refer to Module Configuration (page 21) for a complete description of the parameters required for configuration of the module.

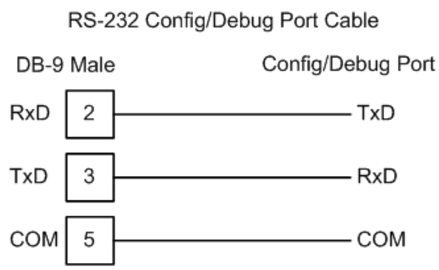
5.3 Cable Connections

The application ports on the MVI56-LTQ module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

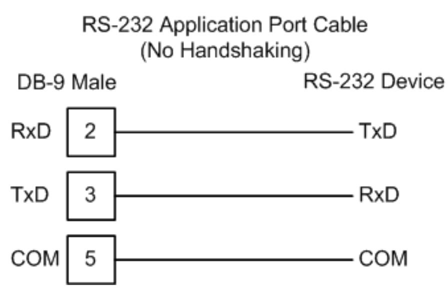
5.3.1 RS-232 Configuration/Debug Port

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



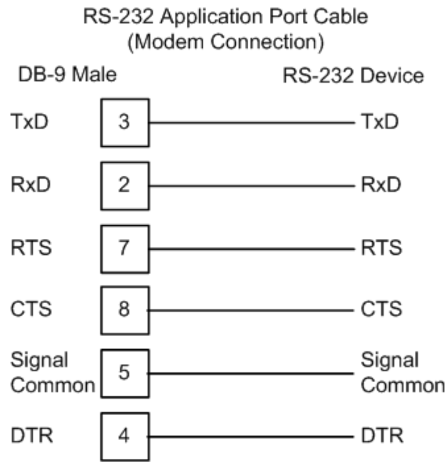
5.3.2 RS-232 Application Port(s)

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



RS-232: Modem Connection (Hardware Handshaking Required)

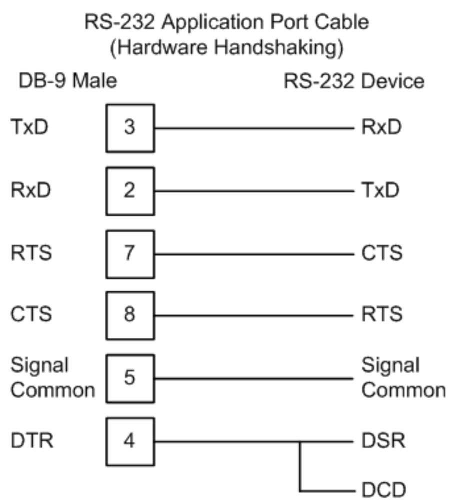
This type of connection is required between the module and a modem or other communication device.



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

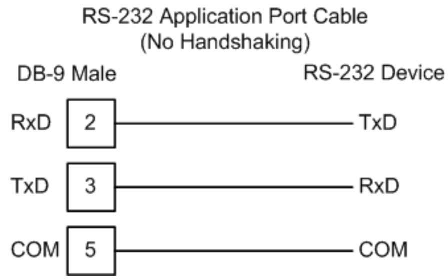
RS-232: Null Modem Connection (Hardware Handshaking)

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

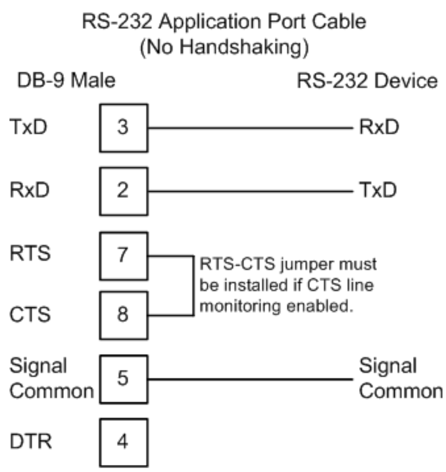


RS-232: Null Modem Connection (No Hardware Handshaking)

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



Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to N and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to Y, then a jumper is required between the RTS and the CTS lines on the port connection.



5.3.3 Network Cable Connection to Limitorque RS-232/RS-485 Converters

For the RS-485 connection, the network cable is connected to the converter via a three-pin or five-pin removable connector (depending on the converter model). This connector is located on the rear of the converter. Prepare the cable as detailed in the Limitorque Accutronix MX/DDC-100 Field Unit Installation and Operation Manual

Self-Steering Converter – P/N 61-825-1032-4

Pin	Function	Wire Color
1	DATA	White
2	DATA* (see note 1)	Blue
3	Earth ground (see note 2)	Shield

Note 1: Indicates negative side of signal.
Note 2: Must be connected to earth ground to assure surge protection.

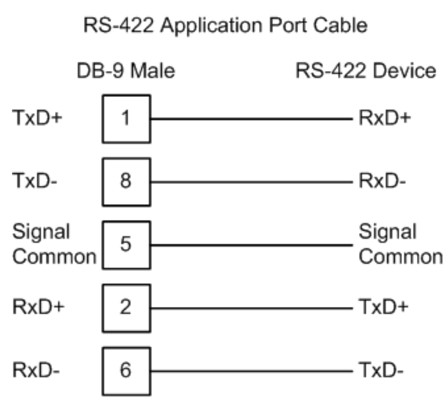
Steered Converter – P/N 61-825-0966-4

Pin	Function	Wire Color
1	DATA	White
2	DATA* (see note 1)	Blue
3	Earth ground (see note 2)	Shield
4	Not Used	Not Used
5	Not Used	Not Used

Note 1: Indicates negative side of signal.
Note 2: Must be connected to earth ground to assure surge protection.

5.3.4 RS-422

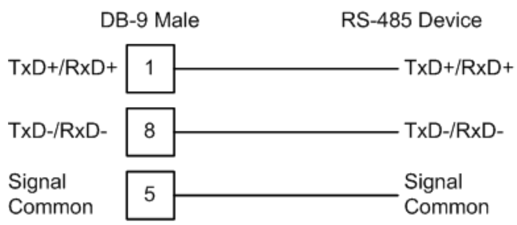
The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



5.3.5 RS-485 Application Port(s)

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

RS-485 Application Port Cable

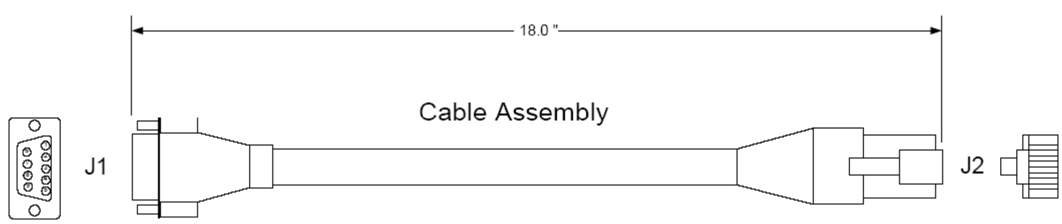


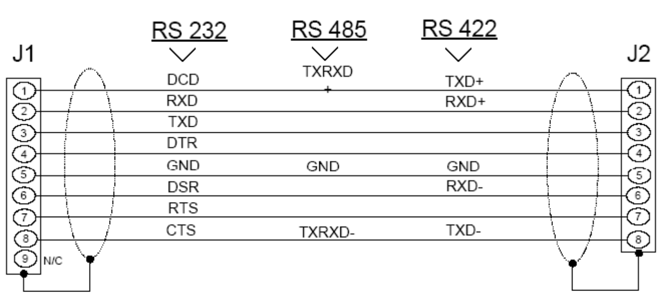
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In these cases, installing a 120-ohm terminating resistor between pins 1 and 8 on the module connector end of the RS-485 line may improve communication quality.

RS-485 and RS-422 Tip

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

5.3.6 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

(This page is intentionally left blank.)

5.4 MVI56-LTQ Database Definition

This section contains a listing of the internal database of the MVI56-LTQ module. Registers 0 to 1499 and 2340 to 2349 are transferred from the module to the processor in blocks 0 to 7. The data in registers 2000 to 2339 is that transferred in blocks 0 and 1 from the processor to the module.

Register Range	Content	Size
0 to 9	Slave #1 data	10
10 to 19	Slave #2 data	10
-	-	-
1490 to 1499	Slave # 150 data	10
2000 to 2009	Open command	10
2010 to 2019	Stop command	10
2020 to 2029	Close command	10
2030 to 2039	Initiate ESD command	10
2040 to 2049	Terminate ESD Command	10
2050 to 2059	Reset command	10
2060 to 2069	Engage Contactor 1 command	10
2070 to 2079	Disengage Contactor 1 command	10
2080 to 2089	Engage Contactor 2 command	10
2090 to 2099	Disengage Contactor 2 command	10
2100 to 2109	Engage Contactor 3 command	10
2110 to 2119	Disengage Contactor 3 command	10
2120 to 2129	Engage Contactor 4 command	10
2130 to 2139	Disengage Contactor 4 command	10
2140 to 2149	Engage Contactor 5 command	10
2150 to 2159	Disengage Contactor 5 command	10
2160 to 2169	Engage Contactor 6 command	10
2170 to 2179	Disengage Contactor 6 command	10
2180 to 2189	Enable Analog command	10
2190 to 2339	Analog values	150
2340 to 2349	Done bits	10
2500 to 2509	Last Open command	10
2510 to 2519	Last Stop command	10
2520 to 2529	Last Close command	10
2530 to 2539	Last Initiate ESD command	10
2540 to 2549	Last Terminate ESD Command	10
2550 to 2559	Last Reset command	10
2560 to 2569	Last Engage Contactor 1 command	10
2570 to 2579	Last Disengage Contactor 1 command	10
2580 to 2589	Last Engage Contactor 2 command	10
2590 to 2599	Last Disengage Contactor 2 command	10
2600 to 2609	Last Engage Contactor 3 command	10
2610 to 2619	Last Disengage Contactor 3 command	10
2620 to 2629	Last Engage Contactor 4 command	10
2630 to 2639	Last Disengage Contactor 4 command	10

Register Range	Content	Size
2640 to 2649	Last Engage Contactor 5 command	10
2650 to 2659	Last Disengage Contactor 5 command	10
2660 to 2669	Last Engage Contactor 6 command	10
2670 to 2679	Last Disengage Contactor 6 command	10
2680 to 2689	Last Enable Analog command	10
2690 to 2839	Last Analog values	150
2840 to 2849	Last Done bits	10

5.5 MVI56-LTQ Status Data Definition

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

Offset	Content	Description
212 to 213	Product	These two registers contain the product code of "LTQ5"
214 to 215	Revision	These two registers contain the product version for the current running software.
216 to 217	OpSys	These two registers contain the month and year values for the program operating system.
218 to 219	Run	These two registers contain the run number value for the currently running software.
220	PassCnt	This value is incremented each time a complete program cycle occurs in the module.
221	Queue	Number of messages present in the command queue waiting for execution.
222	BTR_Cnt	This field contains the total number of read blocks transferred from the module to the processor.
223	BTW_Cnt	This field contains the total number of write blocks transferred from the processor to the module.
224	BlkParse	This field contains the total number of blocks successfully parsed that were received from the processor.
225	BlkEvent	This field contains the total number of command event blocks received from the processor. Each command received by the module will cause this value to increment.
226	BlkErr	This field contains the total number of block errors recognized by the module.
227	CurSlave	This field contains the index of the current slave being polled by the module.
228	CurPort	This field contains the index of the current port on the module being utilized. A value of 0 indicates the first port (Port A). A value of 1 indicates the second port (Port B) is utilized.
229	AltPort	This field is set to 1 when the alternate port (not the primary port) is being used due to a communication error.
230	SpecPoll	This field contains the value of 0, 1 or 2. A value of 0 indicates a normal poll or command is being executed. A value of 1 indicates a special poll for TP_BEFORE_MID_T_HIGH (register 55) is being executed. A value of 2 indicates a special poll for registers 6 and 7 is being executed. A value of 3 to 65535 indicates the corresponding register is being polled.
231	State	This field contains the module state machine value.
232	ComState	This field contains the module communication state machine value.
233	CfgErr	This field contains the configuration error status for the module. If an error in the module's configuration is recognized, this field will be set to a value other than zero.

5.6 LTQValve Object Definition

This section describes each member of the LTQValve object of the module object.

Member	Name	MX-DDC	UEC-3-DDC
ValvePos	Analog Register	Valve Position (0 - 100%)	Valve Position (0 - 100%)
Status	Status Register		
	Bit 00	Opened	Opened
	01	Closed	Closed
	02	Stopped	Stopped
	03	Opening	Opening
	04	Closing	Closing
	05	Valve Jammed	Valve Jammed
	06	Local Mode Selected	Local Mode Selected
	07	Combined Fault *	Combined Fault *
	08	Thermal Overload Fault	Thermal Overload Fault
	09	Future Use	Fail De-Energize
	10	Channel A Fault	Channel A Fault
	11	Channel B Fault	Channel B Fault
	12	Open Torque Switch Fault	Open Torque Switch Fault
	13	Close Torque Switch Fault	Close Torque Switch Fault
	14	Manual Operation	Manual Operation
	15	Phase Error	Phase Error
Fault	Fault Register		
	Bit 00	Not Used	Open Verify Fault
	01	Not Used	Close Verify Fault
	02	Not Used	Open De-Energize Fault
	03	Not Used	Close De-Energize Fault
	04	Phases Missing	Phases Missing
	05	Phase Reversed	Phase Reversed
	06	Not Used	Manual Mid to Open
	07	Not Used	Manual Open to Mid
	08	Not Used	Manual Mid to Close
	09	Not Used	Manual Close to Mid
	10	Network ESD is ON **	Network ESD is ON**
	11	Local ESD is ON	Local ESD is ON
	12	Unit Reset since last poll	Unit Reset since last poll
	13	Local Stop Selected	Wrong Rotation
	14	Opening in Local	Opening in Local
	15	Closing in Local	Closing in Local

*** Combined Fault:**

Bit 07 of Field Unit Status Register (Word 1) indicates a fault when both bits 10 AND 11, or bit 05, or 08, or 09, or 15 indicate a fault.

**** Field unit Network ESD**

Parameter must be configured to Open, Stop, or Close.

Member	**Name**	**MX-DDC**	**UEC-3-DDC**
DOut	Digital Output		
	Bit 00	Close Contactor	Close Contactor
	01	Open Contactor	Open Contactor
	02	AS-1	User (K3)
	03	AS-2	SW-93 LED
	04	AS-3	SW-93 LED
	05	AS-4	User (K6)
	06	AR-1, Opt	N/A
	07	AR-2, Opt	N/A
	08	AR-3, Opt	Bits 08 - 15
	09	Network Relay	Field Unit Software
	10 to 15	Not Used	Version ID
Din1	Digital Inputs 1		
	Bit 00	Remote Mode Selected	Remote Mode Selected
	01	Thermal Overload Fault	Thermal Overload Fault
	02	Open Torque Switch	Open Torque Switch
	03	Open Limit Switch	Open Limit Switch
	04	Close Torque Switch	Close Torque Switch
	05	Close Limit Switch	Close Limit Switch
	06	Not Used	Aux. Open Input
	07	Not Used	Aux. Close Input
	08	User 0, Terminal-21	User 0, TB2-1
	09	User 1, Terminal-10	User 1, TB2-2
	10	User 2, Terminal-9	User 2, TB2-3
	11	User 3, Terminal-6	User 3, TB2-4
	12	User 4, Terminal-7	User 4, TB2-5
	13	User 5, Terminal-5	User 5, TB2-6
	14	User 6, Opt, Terminal-23	User 6, I/O Module Only
	15	User 7, Opt, Terminal-24	User 7, I/O Module Only
DIn2	Digital Inputs 2		
	Bit 00	Not Used	Analog In 1 Lost
	01	Not Used	Analog In 2 Lost
	02	Analog In 1 Lost	Analog In 3 Lost
	03	Analog In 2 Lost	Analog In 4 Lost
	04	Network A/B Lost	Network A/B Lost
	05	Not Used	Reserved
	06	DDC Bd. Present	Reserved
	07	I/O Opt Board Present	Reserved
	08	Not Used	Reserved
	09	Not Used	Reserved
	10	Not Used	Reserved
	11	Not Used	Reserved
	12	Phase Lost	Phase Lost
	13	Phase Reverse	Phase Reverse
	14	User 8, Opt, Terminal-25	User 8, I/O Module Only
	15	Not Used	User 9, I/O Module Only
ComStatus	Communication Status Code	See Trouble Shooting Section. Do not use this word to determine Slave communication status. Word N[ ]:1 (Status) bits 10 and 11 are preferred. This is a module diagnostic word only.	
ComCntr	Communication Counter	This is a rollover counter (0 to 32767) which increments upon completion of every successful communication transaction with a slave. This counter will increment on poll (read) commands as well as write commands.	

Member	Name	MX-DDC	UEC-3-DDC
SPoll1	Polling (Special)		
	0	Unused	
	1	TP_BEFORE_MID_T_HIGH Register 55	
	2	Analog 1 Register 6	
	3 to 65535	Corresponding register	
SPoll2	Polling (Special)		
	0	Unused	
	1	Unused	
	2	Analog 2 Register 7	

For a more complete discussion on register values for Limitorque actuators or I/O modules, please reference the Limitorque Document #435-23009, available from Limitorque.

5.7 Command Usage for Limitorque Products

This topic describes the commands supported in the MVI56-LTQ module and their association with different Limitorque interfaces.

LTQ Commands	MX-DDC	UEC-3-DDC	I/O Module
Open	Yes	Yes	Do Not Use
Stop	Yes	Yes	Do Not Use
Close	Yes	Yes	Do Not Use
Start Network ESD	Yes	Yes	Do Not Use
Stop Network ESD	Yes	Yes	Do Not Use
Reset			
Engage Relay #1	Yes (AS-1)	Do Not Use	Yes (K2)
Engage Relay #2	Yes (AS-2)	Do Not Use	Yes (K1)
Engage Relay #3	Yes (AS-3)	Yes (K3)	Yes (K3)
Engage Relay #4	Yes (AS-4)	Do Not Use	Yes (K4)
Engage Relay #5	Yes (AR-1)	Do Not Use	Yes (K5)
Engage Relay #6	Yes (AR-2)	Yes (K6)	Yes (K6)
Disengage Relay #1	Yes (AS-1)	Do Not Use	Yes (K2)
Disengage Relay #2	Yes (AS-2)	Do Not Use	Yes (K1)
Disengage Relay #3	Yes (AS-3)	Yes (K3)	Yes (K3)
Disengage Relay #4	Yes (AS-4)	Do Not Use	Yes (K4)
Disengage Relay #5	Yes (AR-1)	Do Not Use	Yes (K5)
Disengage Relay #6	Yes (AR-2)	Yes (K6)	Yes (K6)
Send Valve Position			
Enable Valve Position			

The ProSoft Technology LTQ Master Module is pre-programmed to support a subset of the Modbus protocol. The commands are hard coded into the module and have been selected to implement specific functionality. The programmed commands are documented in the following table.

Command Function	**Modbus Function Code**	**Register Address**	**Count or Write Value**	**Description**
Poll Slave	3	40008	6	Command is executed automatically to any slave in the Active Slave Table
Open Command	6	40001	256	Open Command (Interlocked with Close Command in the slave)
Stop Command	6	40001	512	Disengages Open or Close
Close Command	6	40001	768	Close Command (Interlocked with Open Command in the slave)
Start Network ESD	6	40001	1280	Initiates Network ESD function in the addressed slave
Stop Network ESD	6	40001	1536	Terminates Network ESD function in the addressed slave
Reset	6	40001	1024	
Engage Relay #1	6	40001	2304	Engages Relay #2 (I/O Module) Engages AS-1 (MX-DDC)
Engage Relay #2	6	40001	2560	Engages Relay #1 (I/O Module) Engages AS-2 (MX-DDC)
Engage Relay #3	6	40001	2816	Engages Relay #3 Engages AS-3 (MX-DDC)
Engage Relay #4	6	40001	3072	Engages Relay #4 (I/O Module) Engages AS-4 (MX-DDC)
Engage Relay #5	6	40001	3328	Engages Relay #5 (I/O Module) Engages AR-1 (MX-DDC)
Engage Relay #6	6	40001	3584	Engages Relay #6 Engages AR-2 (MX-DDC)
Disengage Relay #1	6	40001	4352	Disengages Relay #2 (I/O Module) Disengages AS-1 (MX-DDC)
Disengage Relay #2	6	40001	4608	Disengages Relay #1 (I/O Module) Disengages AS-2 (MX-DDC)
Disengage Relay #3	6	40001	4864	Disengages Relay #3 Disengages AS-3 (MX-DDC)
Disengage Relay #4	6	40001	5120	Disengages Relay #4 (I/O Module) Disengages AS-4 (MX-DDC)
Disengage Relay #5	6	40001	5376	Disengages Relay #5 (I/O Module) Disengages AR-1 (MX-DDC)
Disengage Relay #6	6	40001	5632	Disengages Relay #6 Disengages AR-2 (MX-DDC)
Send Valve Position Value	6	40002	Value from PLC	Position to move actuator 0 to 100% of Open
Enable Valve Position Value	6	40001	6656	Move-To (enable)
Poll Slave Special 1	3	40055	1	Command is executed automatically if Special Polling is set to 1
Poll Slave Special 2	3	40006	2	Command is executed automatically if Special Polling is set to 2

5.8 Polling Schemes

In Looped Mode, the LTQ provides communication redundancy to each configured slave on the network. The LTQ monitors the health of each communication path between port 1 and each configured slave and between port 2 and each configured slave. LTQ port 1 communication status between port 1 and the addressed slave is recorded in the slave Channel A status bit. LTQ port 2 communication status between port 2 and the addressed slave is recorded in the slave Channel B status bit. Both Channel A and Channel B status bits are located in the slave Status register, bits 10 and 11(Word 1, bits 10 and 11).

On a healthy network where all configured slaves are communicating; the LTQ will first poll all slaves via port 1, then poll all slaves via port 2, back to port 1, and so on. As each slave is successfully polled, the respective Channel bit is set to 0 in the slave Status register. Remember the LTQ port 1 equals Channel A and the LTQ port 2 equals Channel B.

Should a slave not be reached on a poll, the LTQ will set the corresponding Channel Fail bit to 1, switch to the other port and attempt to communicate with the same slave. Should the slave not communicate from the second port, the corresponding Channel Fail bit will be set to 1, and the LTQ will resume polling on the original port. After the LTQ has completed polling all configured slaves on the first port, the polling routine will switch to the other port and repeat the above process.

Example:

There are 5 slaves on the network and slave number 3 has been turned off. The LTQ is currently polling the slaves through the LTQ port 1 (Channel A). Slave number 1 and 2 respond to the LTQ port 1 poll. Slave number 3 does not respond to the port 1 poll causing the LTQ to set slave 3 Channel A bit to 1. The LTQ now changes to port 2 (Channel B) and polls slave number 3. Slave number 3 does not respond to the port 2 poll causing the LTQ to set the slave 3 Channel B bit to 1. Next the LTQ changes back to port 1 and attempts to poll slave number 4. This communication attempt is successful and the LTQ now polls slave number 5 through the LTQ port 1. Slave number 5 responds completing the port 1 poll.

Next the LTQ repeats the process through port 2 (Channel B). Slave 1 and 2 respond, slave 3 does not respond and the LTQ sets the slave 3 Channel B bit to 1. The LTQ changes to port 1 (Channel A) and attempts to communicate with slave 3. Slave 3 does not respond, the LTQ sets the slave 3 Channel A bit to 1, switches back to port 2 and resumes polling the remainder of the configured slaves. Once slaves 4 and 5 have been successfully polled via port 2, the LTQ then switches to port 1 and repeats the polling process. The port alternation process described above continues until slave 3 is powered on and the communication fault clears.

Commands for slave control interrupt the polling process and are issued through the current poll port. After the slave has acknowledged the command, the LTQ resumes the polling process. In the event of a communication fault between the current poll port and a commanded slave, the LTQ will issue the command through the other communication port.

Looped Network Truth Table

(Recorded in Slave Status Register Bit 10 and 11)

	Example 1		**Example 2**		**Example 3**		**Example 4**	
Slave #	Ch. A	Ch. B	Ch. A	Ch. B	Ch. A	Ch. B	Ch. A	Ch. B
1	0	0	0	0	0	1	0	1
2	0	0	0	0	0	1	0	1
3	0	0	1	1	0	1	0	1
4	0	0	0	0	1	0	0	1
5	0	0	0	0	1	0	0	1

Example 1:

The LTQ is successfully communicating to each slave and sets the bits equating to Channel A and B to 0. A value of 0 in the Channel A and B status indicate successful communication.

Example 2:

The LTQ is successfully communicating to slaves 1, 2, 4, and 5 via both ports. Slave number 3 is without power causing the slave 3 network board bypass relays to de-energize. This de-energizing of the bypass relays shorts the signal through the network board and isolates the slave from the DDC-100 network.

Example 3:

The LTQ is successfully communicating to slaves 1, 2, 3 via port 1 and 4, 5 via port 2. When a slave does not communicate within a predetermined time-out period the LTQ sets the corresponding Channel bit to a value of 1. This example indicates a wiring problem between slave 3 & 4. This problem is typically a cable breakage, short, or improperly terminated wire.

Example 4:

The LTQ is attempting to communicate to the slaves via both ports but is unable to reach any slaves via port 2. This typically indicates a broken cable connection at port 2 or at the first slave from port 2, broken or shorted cable between the LTQ and the first slave from port 2, improperly terminated wires, or loss of power to the RS-232/485 converter if attached to the LTQ port 2.

Non-Looped Network Truth Table via Port 1 Polling Only

(Recorded in Slave Status Register Bit 10, Bit 11 Channel B is always 0)

	Example 5		**Example 6**		**Example 7**	
Slave #	Ch. A	Ch. B	Ch. A	Ch. B	Ch. A	Ch. B
1	0	0	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	0	1	0
4	0	0	0	0	1	0
5	0	0	0	0	1	0

Example 5:

The LTQ is successfully communicating to each slave and sets the bits equating to Channel A to 0. A value of 0 in the Channel A status indicates successful communication.

Example 6:

The LTQ is successfully communicating to slave 1, 3, 4, and 5. Slave 2 does not respond causing the LTQ to set slave 2 Status register bit 10 to 1. In this example, slave 2 is without power causing the slave 2 network board bypass relays to de-energize. This de-energizing of the bypass relays shorts the signal through the network board and isolates the slave from the DDC-100 network.

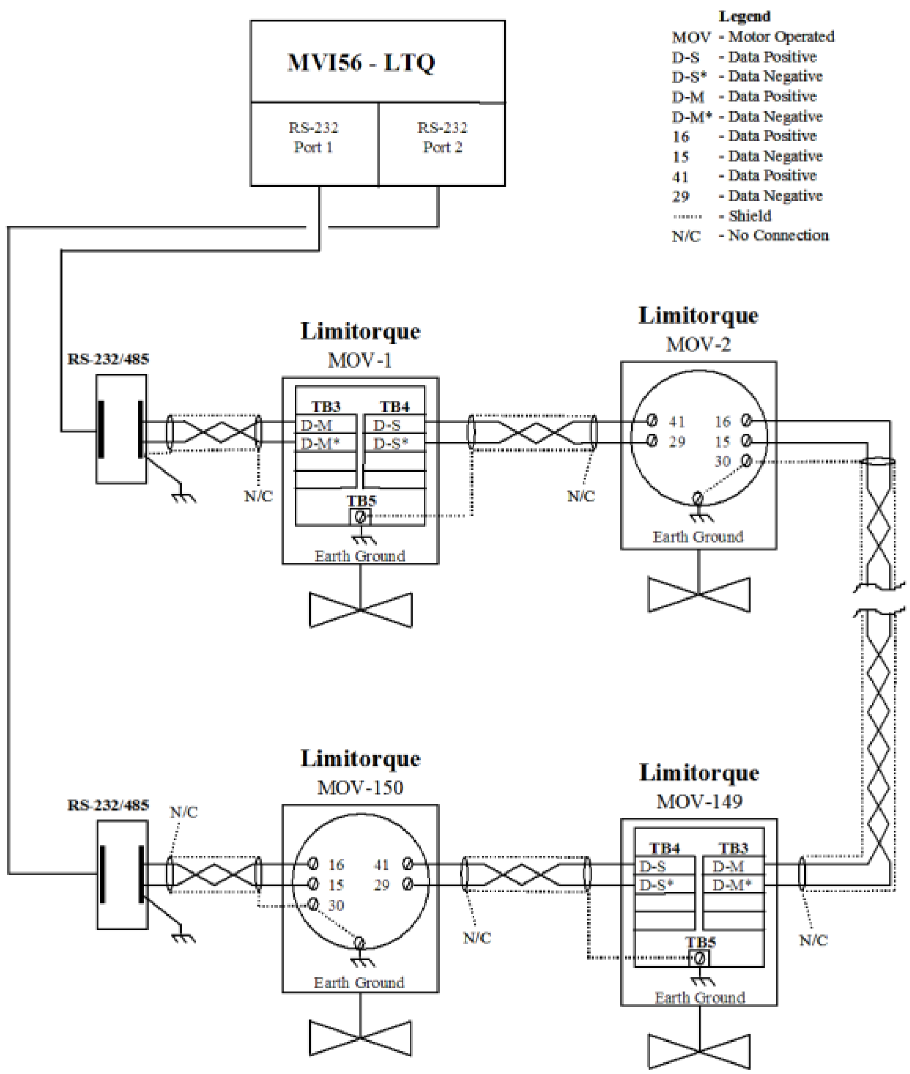
Example 7:

The LTQ is successfully communicating to slaves 1 and 2 but is not able to communicate to slaves 3, 4, and 5 causing the LTQ to set slave 3, 4, 5 Status register bit 10 to 1. This typically indicates a broken or shorted cable between slave 2 and 3, a broken cable connection at slave 2 or 3, improperly terminated wires at slave 2 or 3, or loss of power to slaves 3, 4, and 5.

5.9 Typical Network Loop

The following example shows an example of a network loop with Limitorque MX-DDC and UEC-DDC actuators.

Typical Network Loop
with
Limitorque MX-DDC and UEC-3-DDC Actuators



Notes:

1) Belden 3074F, 3105A, or 9841 shielded cable is recommended.
2) Correct polarity for field unit and MVI56 -LTQ is required for proper network operation.
3) Connections shown are typical. The number of MOVs shown may not indicate true network size.
4) Earth ground: ground rod.
5) Earth ground: ground rod or lug in actuator if actuator is grounded.

6 Support, Service & Warranty

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.*

North America (Corporate Location)	**Europe / Middle East / Africa Regional Office**
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:
https://www.prosoft-technology.com/About-Us/Contact-Us.

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