ProSoft TECHNOLOGY

Where Automation Connects.

MVI69-HART

CompactLogix Platform

HART Multi-drop Master Communication Module

June 5, 2018

USER MANUAL

Your Feedback Please

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

How to Contact Us

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

MVI69-HART User Manual
Rev 1.0.0

June 5, 2018

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

Important Installation Instructions

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

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

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

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

Class 2 Power

MVI (Multi Vendor Interface) Modules

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

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

Warnings

North America Warnings

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.

产品中有害物质的名称及含量
Name and content of hazardous substances in product

部件名称 Component Name	有害物质					
	鉛 Lead (Pb)	汞 Mercury (Hg)	鎘 Cadmium (Cd)	六价铬 Hexavalent Chromium (Cr(VI))	多溴联苯 Polybrominated Biphenyls (PBB)	多溴二苯醚 Polybrominated Diphenyl Ethers (PBDE)
印刷电路板组件 Printed Circuit Board Assemblies	X	O	O	O	O	O
金属部件 Metal Components	X	O	O	O	O	O
电池 Battery	O	O	O	O	O	O
塑料部件 Plastic Components	X	O	O	O	O	O

本表格依据SJ/T 11364的规定编制。This table is made per guidance of SJ/T 11364
X: 表示该有害物质至少在该部件的某一均质材料中的含量超出GB/T 26572规定的限量要求。
(企业可在此处，根据实际情况对上表中打"X"的技术原因进行进一步说明。)

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

Battery Life Advisory

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

Note: The battery is not user replaceable.

Markings

Electrical Ratings

- Backplane Current Load: 800 mA @ 5.1 Vdc
- Power Supply Distance Rating: 2
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Relative Humidity: 5% to 95% (with no condensation)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Label Markings

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

II 3 G

Ex nA IIC X

0°C <= Ta <= +60°C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Agency Approvals and Certifications

Agency	Applicable Standard(s)
ATEX	EN 60079-0:2006, EN 60079-15:2005
DNV	DET NORSKE VERITAS Test 2.4
CE	EMC-EN61326-1:2006; EN61000-6-4:2007
CB Safety	CA/10533/CSA, IEC 61010-1 Ed. 2, CB 243333-2056722 (2090408)
GOST-R	EN 61010

ME06

Contents

1 Start Here

In This Chapter

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

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows®:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect HART and CompactLogix or MicroLogix devices to a power source and to the MVI69-HART's application port(s)

1.1 System Requirements

The MVI69-HART module requires the following minimum hardware and software components:

- *Rockwell Automation CompactLogix* processors and *MicroLogix 1500 LRP* processors except 1769-L23E-QBFC1B, 1769-L16x, and 1769-L18x. Must have compatible power supply and one free slot in the rack, for the MVI69-HART module. The module requires 800 mA of available power.

Important: The MVI69-HART module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).

- *Rockwell Automation RSLogix 5000* (*CompactLogix*) or *RSLogix 500* (*MicroLogix*) programming software
- *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)
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

Note: MVI69/PS69 modules will not work with CompactLogix L4x processors using RSLogix 5000 v17. All other processor combinations and RSLogix versions will work correctly.

1.2 Package Contents

The following components are included with your MVI69-HART 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	MVI69-HART Module	MVI69-HART	HART Multi-drop Master Communication Module
1	Cable	Cable #15, RS-232 Null Modem	For RS-232 Connection to the CFG Port
1	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port

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

1.3 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder* (PCB) software to configure the module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology website (http://www.prosoft-technology.com). The filename contains the version of PCB. For example, **PCB_4.4.3.4.0245.exe**.

To install ProSoft Configuration Builder from the ProSoft Technology website

1. Open your web browser and navigate to www.prosoft-technology.com.
2. Search for '*PCB*' or '*ProSoft Configuration Builder*'.
3. Click on the ProSoft Configuration Builder search result link.
4. From the *Downloads* link, download the latest version of *ProSoft Configuration Builder*.
5. Choose **SAVE** or **SAVE FILE**, if prompted.
6. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
7. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

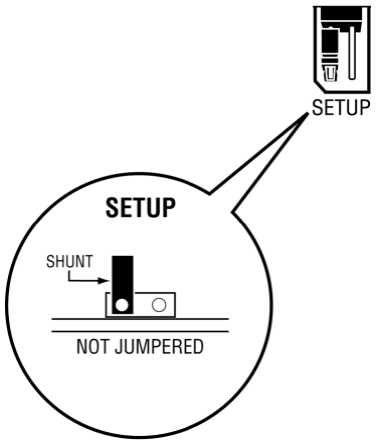
Note: To use the RSLogix under the Windows 7 OS, you must be sure to install it using the *Run as Administrator* option. To find this option, right-click the Setup.exe program icon, and then click **RUN AS ADMINISTRATOR** on the context menu. You must install using this option even if you are already logged in as an Administrator on your network or personal computer (PC). Using the *Run as Administrator* option allows the installation program to create folders and files on your PC with proper permissions and security.

If you do not use the *Run as Administrator* option, the RSLogix may appear to install correctly, but you will receive multiple file access errors whenever the RSLogix is running, especially when changing configuration screens. If this happens, you must completely uninstall the RSLogix and then re-install using the *Run as Administrator* option to eliminate the errors.

1.4 Setting Jumpers

The Setup Jumper acts as "write protection" for the module's firmware. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. The module is shipped with the Setup jumper OFF. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support (or you want to update the module firmware).

The following illustration shows the MVI69-HART jumper configuration with the Setup Jumper OFF.



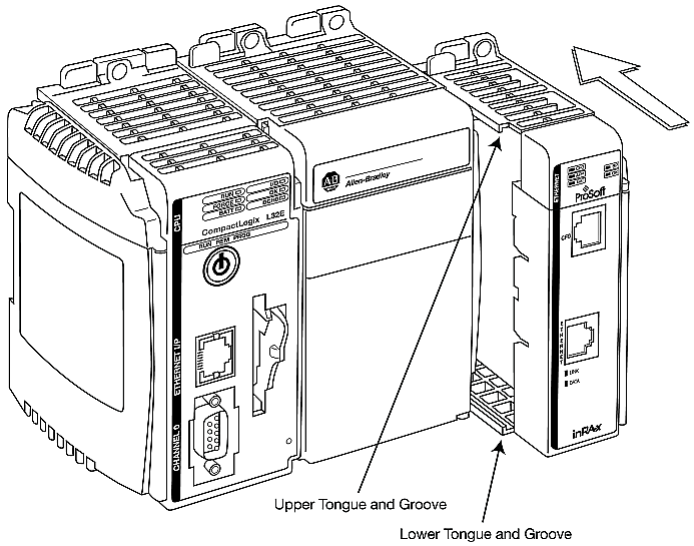
Note: If you are installing the module in a remote rack, you may prefer to leave the Setup pins jumpered. That way, you can update the module's firmware without requiring physical access to the module.

1.5 Installing the Module

Before you attempt to install the module, make sure that the bus lever of the adjacent module is in the unlocked (fully right) position.

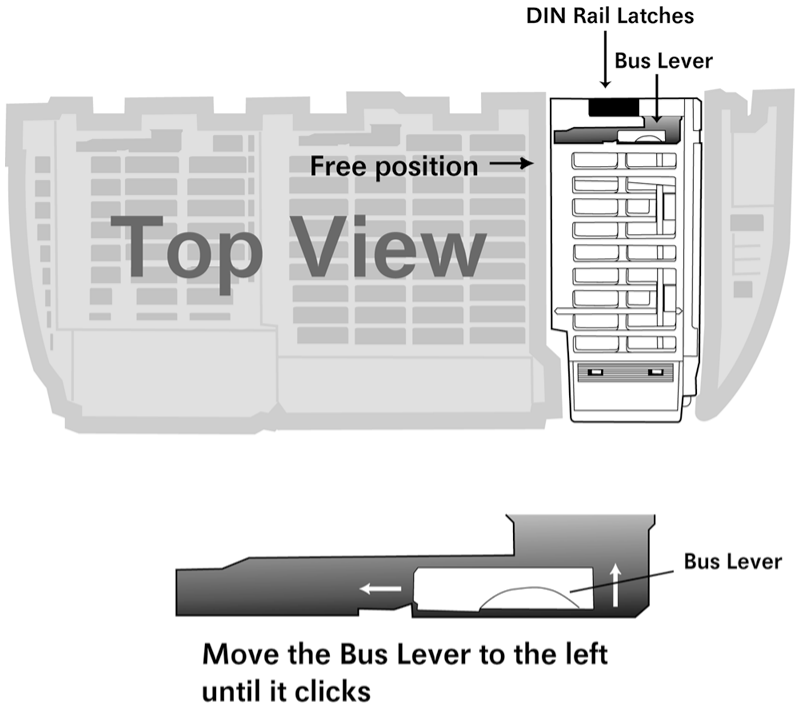
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1 Align the module using the upper and lower tongue-and-groove slots with the adjacent module and slide forward in the direction of the arrow.



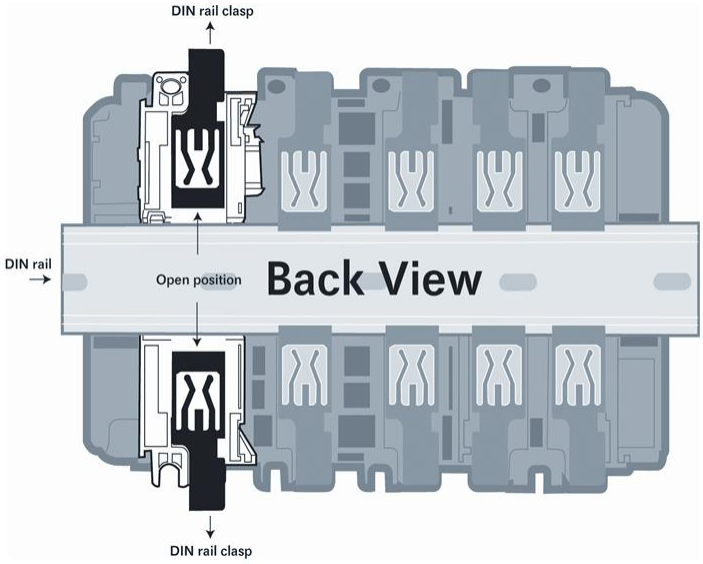
2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

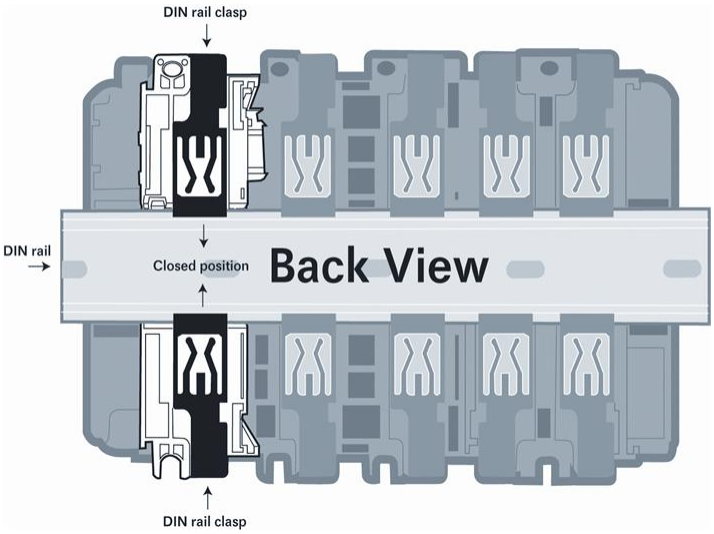
3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



4 Close all DIN-rail latches.

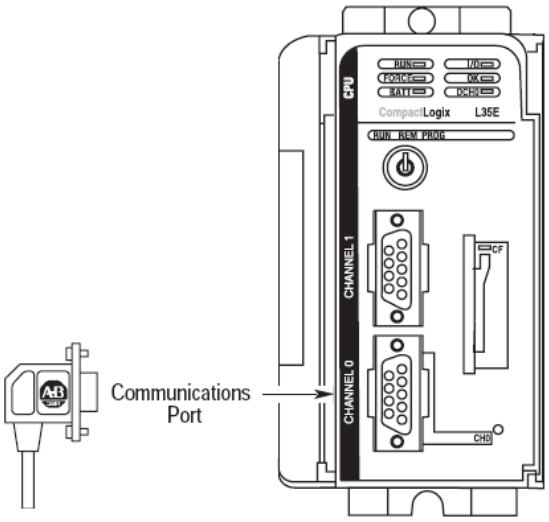
5 Press the DIN-rail mounting area of the controller against the DIN-rail. The latches will momentarily open and lock into place.



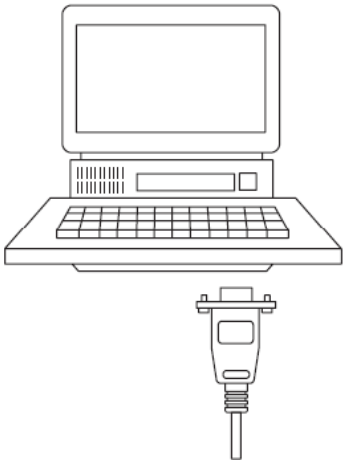


1.6 Connecting Your PC to the Processor

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



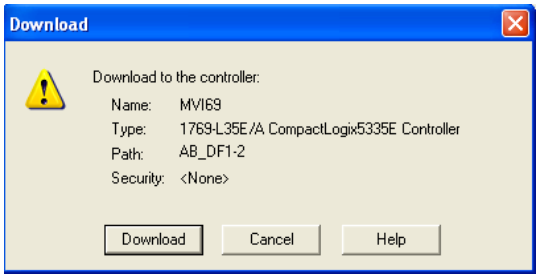
2 Connect the straight connector end of the cable to the serial port on your computer.



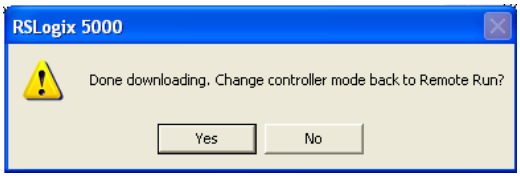
1.7 Downloading the Sample Program to the Processor

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

1. If you are not already online to the processor, open the **COMMUNICATIONS** menu, and then choose **DOWNLOAD**. RSLogix will establish communication with the processor.
2. When communication is established, RSLogix will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix will open another confirmation dialog box. Click **OK** to switch the processor from PROGRAM mode to RUN mode.

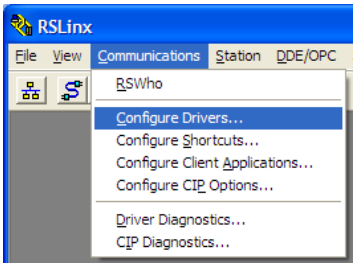


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

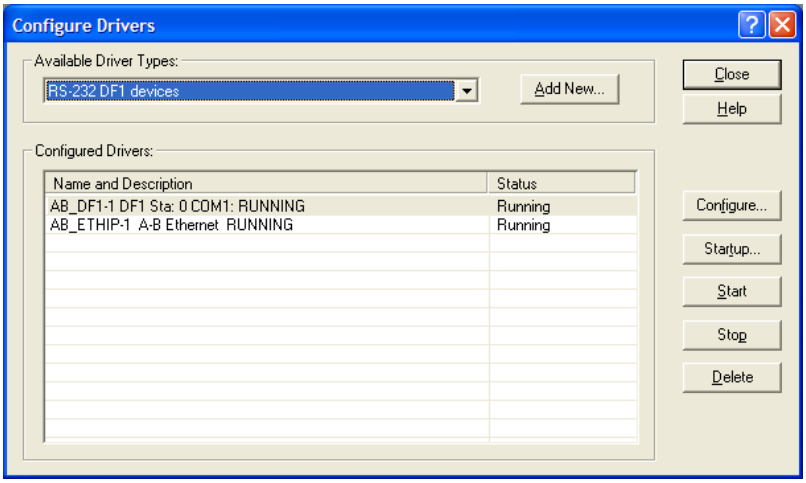
1.7.1 Configuring the RSLinx Driver for the PC COM Port

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

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

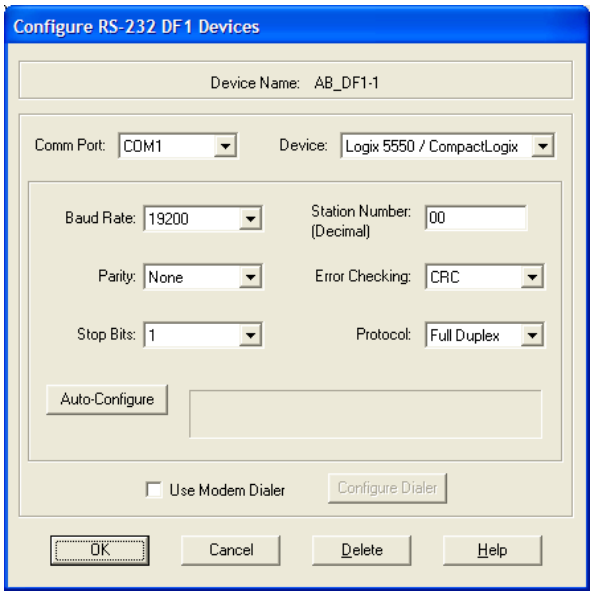


This action opens the *Configure Drivers* dialog box.



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.

1.8 Connecting Your PC to the Module

With the module securely mounted, connect your PC to the Configuration/Debug port using the RJ45-DB-9 Serial Adapter Cable and the Null Modem Cable included in the package with the MVI69-HART module.

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

2 Module Configuration

In This Chapter

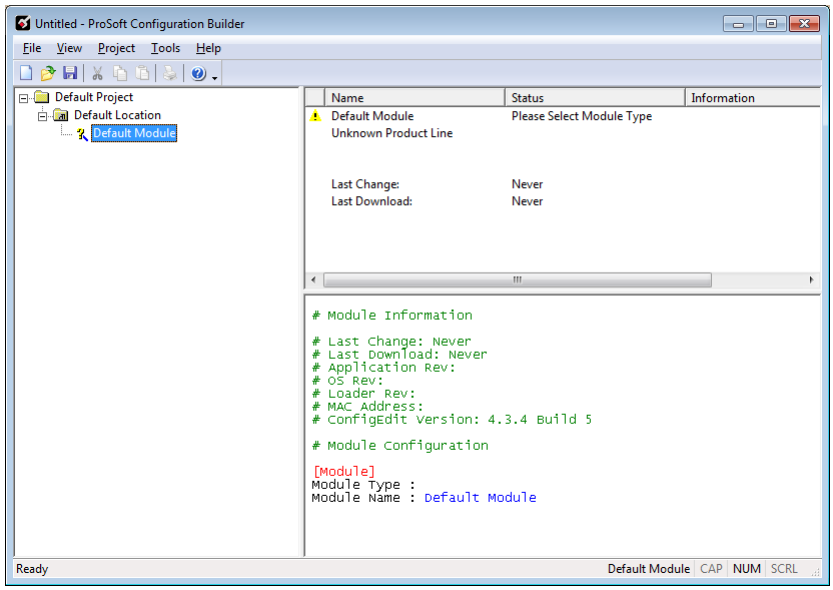
This section contains the setup procedure, data, and ladder logic for successful application of the MVI69-HART module. Each step in the setup procedure is defined in order to simplify the use of the module.

2.1 Using ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a quick and easy way to manage module configuration files customized to meet your application needs. PCB allows you to import information from previously installed (known working) configurations to new projects.

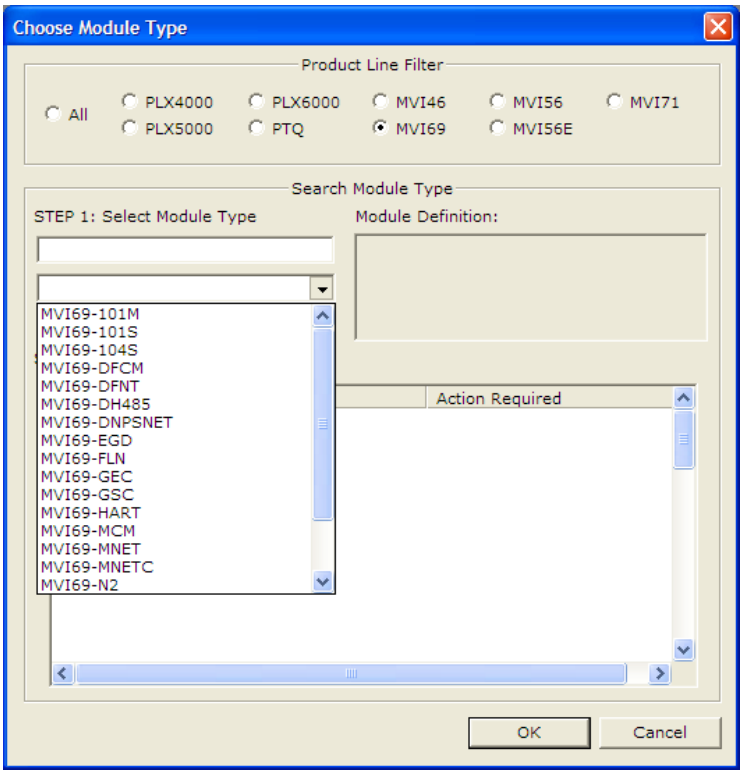
2.1.1 Setting Up the Project

To begin, start *ProSoft Configuration Builder*. If you have used other *Windows* configuration tools before, you will find the screen layout familiar. *ProSoft Configuration Builder's* window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window. When you first start *ProSoft Configuration Builder*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *ProSoft Configuration Builder* window with a new project.



1 Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, select **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



3 In the *Product Line Filter* area of the dialog box, select **MVI69**. In the *Select Module Type* dropdown list, select **MVI69-HART**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

2.1.2 Renaming PCB Objects

You can rename objects such as the *Default Project* and *Default Location* folders in the tree view. You can also rename the **MODULE** icon to customize the project.

1 Right-click the object you want to rename and then choose **RENAME.**
2 Type the new name for the object and press **Enter**.

Configuring Module Parameters

1 Click the **[+]** sign next to the module icon to expand module information.
2 Click the **[+]** sign next to any icon to view module information and configuration options.
3 Double-click any icon to open an *Edit* dialog box.
4 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
5 Click **OK** to save your changes.

Printing a Configuration File

1. In the main PCB window, right-click the **MVI69-HART MODULE** icon and then choose **VIEW CONFIGURATION**.
2. In the *View Configuration* dialog box, click the **FILE** menu and click **PRINT**.
3. In the *Print* dialog box, choose the printer to use from the drop-down list, select the printing options, and click **OK**.

2.2 MVI69-HART Configuration File

The configuration file for MVI69-HART consists of the following sections.

Section	Description
[MODULE]	General module level parameters
[HART PORT 0]	HART Port (Channel) 0 configuration parameters
[HART PORT 0 COMMANDS]	HART command list for Port (Channel) 0
[HART PORT 1]	HART Port (Channel) 1 configuration parameters
[HART PORT 1 COMMANDS]	HART command list for Port (Channel) 1

2.2.1 [MODULE]

The [Module] section contains the data that applies to the whole module and includes the backplane data transfer parameters.

Module Name

This is the given name of this particular module.

Block Transfer Size

60, **120** or **240**

This read-only parameter specifies the number of words of user data transferred in each block between the module and processor.

Read Register Start

0 to **3999**

This parameter specifies the starting register address of a block of data registers to transfer from the module to the processor. This will be used for data brought into the processor from other nodes on the network.

Write Register Count

0 to **4000**

This parameter specifies the number of registers to transfer from the processor to the module.

Write Register Start

0 to **3999**

This parameter specifies the starting register address of a module register block where data transferred from the processor will be stored. This will be used for data to be sent from the processor to other nodes on the network.

Write Register Count

Range 0 to 4000

This parameter specifies the number of registers to transfer from the processor to the module. Valid entry for this parameter is 0 to 4000.

Error/Status Offset

-1 or 0 to 3935

This parameter defines the database location where the module status data will be stored. If set to -1, data not placed in database.

Failure Flag Count

0 to **65535**

This parameter specifies the number of successive transfer errors that must occur before halting communication on the application port(s). If the parameter is set to **0**, the application port(s) will continue to operate under all conditions. If the value is set larger than **0** (**1** to **65535**), communications will cease if the specified number of failures occur.

Initialize Output Data

YES or **NO**

This parameter determines if the output data for the module is initialized with values from the processor. If the value is set to **NO** (0), the output data is initialized to 0. If the value is set to **YES** (1), the data is initialized with data from the processor. Setting this option to **YES** requires associated ladder logic to pass the data from the processor to the module.

2.2.2 [HART PORT x]

The [HART PORT *x*] sections of the configuration file set the HART channel communication parameters, define the protocol specifics and set the command list parameters. The parameters are the same for all ports. The command list for each HART channel is entered in a different section of the file.

Enabled

Y or N

This parameter enables or disables the specific HART channel. If the parameter is set to "Y", the channel will be utilized. If set to "N", the channel will not be used.

Preambles

5 to 20

This parameter sets the number of preambles to be transmitted before each message is sent from the channel. The value of 5 is normally utilized for the parameter. It can be set to a value from 5 to 20.

Primary Master

Y or N

This parameter determines if the specific HART channel will emulate a primary or secondary master. You can have only one of each type on a HART network. If you plan on using a handheld device (secondary master), you must set the parameter to 'Y'. If the parameter is set to 'Y', the channel will act as the primary master. A value of 'N' will set the channel to act as a secondary master.

Retry Count

0 to 10

This parameter sets the number of retries for a command if the command response is not received from the slave device. This parameter is normally set to a value of 3. The module will accept values of 0 to 10.

DB Address Status

-1, 0 to 3999

This parameter is utilized to set the database address in the module where the status word for the channel will be placed. If the parameter is set to -1, the word value will not be placed in the database. If a value from 0 to 3999 is set for the parameter, the status word for the channel will be placed at the specified database offset. This word is bit mapped with each bit representing a slave device. The bit will be set if slave device has a communication error.

Slave List Status

The configuration parameter "DB Address Status" defines the register address in the virtual database where the status data for each HART channel will be placed. This word has one bit for each HART device and if this bit is in "1" it means that the corresponding HART device is not answering to the poll.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HART Device	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The bit 0 of the "Slave List Status" is used for a slave connected in a point-to-point configuration. In this mode, the slave's 4 to 20 milliamp signal will also be active and can be utilized by an analog input or out module depending on the signal type.

Command Count

0 to 99

This parameter sets the number of user commands to be utilized. The first command in the list is always reserved for the auto-poll command so the user should configure this value considering one command for the auto-poll. For example, if the user configures two commands, the command count parameter should be set as 3. This parameter can be set from 0 to 99. If the parameter is set to a value other than 0, commands should be present in the [HART PORT x COMMANDS] section.

Auto-Poll Code

p2p, multidrop, or Not used

This parameter sets the auto-poll mode of the channel. If the parameter is set to "p2p", the module will automatically poll device 0 in point-to-point mode. If the parameter is set to "multidrop", the module will automatically poll multiple devices. If the parameter is set to "Not used", the auto-polling option will be disabled and only commands in the command list will be utilized for the channel. In the auto-poll mode, the module will automatically execute HART commands 0, 3, 13, 14, and 15.

Auto-Poll DB Address

0 to 3700

This parameter sets the starting address for the data obtained by the auto-poll feature. Each device on a channel requires 50 words in the database. The data area selected must not overlap any portion of the database used by other channels or the module.

Auto-Poll Swap Float

0 to 3

This parameter swaps the floating-point data values received by the auto-poll feature.

Swap Code	Description
0	None - No Change is made in the byte ordering (1234 = 1234)
1	Words - The words are swapped (1234=3412)
2	Words & Bytes - The words are swapped then the bytes in each word are swapped (1234=4321)
3	Bytes - The bytes in each word are swapped (1234=2143)

Depending on the host processor using the data, the proper swap code must be utilized to present the data in the correct format.

Max Device Count

1 to 15

This parameter sets the maximum number of slave devices to be utilized for the auto-poll feature. In point-to-point mode, the parameter should be set to a value of 1. In multi-drop mode, the parameter should be set from 1 to 15 to represent the number of slave devices attached to the channel.

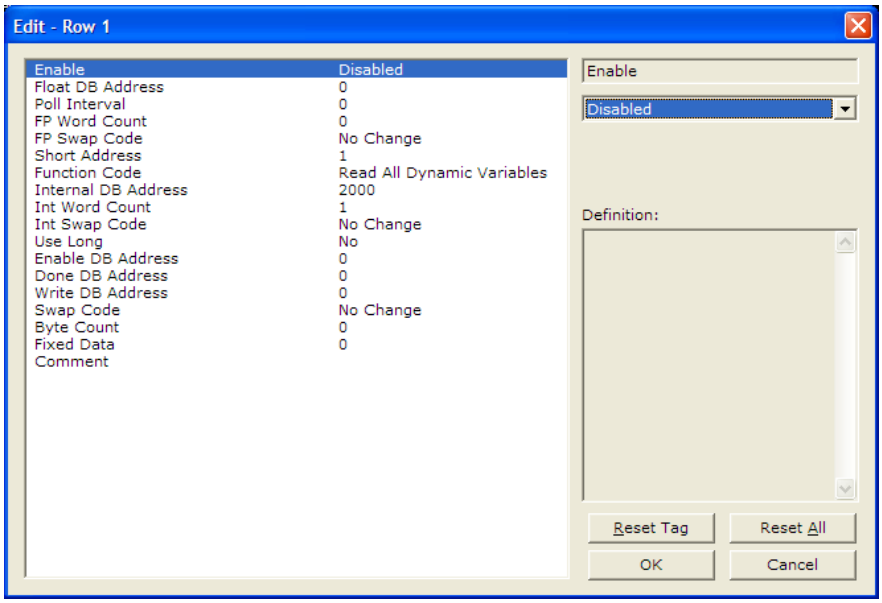
Error/Status Offset

-1 or 0 to 3935

This parameter defines the database location where the module status data will be stored. If set to -1, data not placed in database.

2.2.3 [HART PORT x COMMANDS]

The [HART PORT *x* COMMANDS] sections of the configuration file contain the user-defined HART command lists for each channel. The commands in these lists are sent to slave devices attached to the HART channels. The module supports up to 100 commands per channel.



Command List Overview

The MVI69-HART module uses a command list to interface with HART slave devices. The commands in the list specify:

- The slave device to be addressed
- The function to be performed (read or write)
- The registers in the internal database to be associated with the device data.

There is a separate command list for each HART channel, with up to 100 commands allowed per channel. The command list is processed from top (Command #0) to bottom (Command #99), then the process is repeated.

A poll interval parameter is associated with each command to specify a minimum delay time in seconds between the issuance of a command. For example, a poll interval of 10 executes the command no more frequently than every 10 seconds.

Write commands have a special feature, as they can be set to execute only if the data in the write command changes. If the register data values in the command have not changed since the command was last issued, the command will not be executed. If the data in the command has changed since the command was last issued, the command will be executed. Use of this feature can lighten the load on the HART network. In order to implement this feature, set the enable code for the command to a value of 2.

The module supports all the HART Universal Commands (page 107) and HART Common Practice Commands (page 114) commands, as well as device specific commands. A Device Specific command is supported without any translation of the data.

Commands Supported by the Module

The format of each command in the list is independent on the function being executed. All parameters in the command table must be entered. Only one parameter is optional, the Fixed Data field, which contains data to be sent to a HART device. The tables below list the functions supported by the module:

HART Universal Commands Set

Command	Definition
00	Read Unique Identifier
01	Read Primary Variable
02	Read Current And Percent Of Range
03	Read Dynamic Variables
06	Write Polling Address
07	Read Loop Configuration
08	Read Dynamic Variable Classifications
09	Read Device Variables with Status
11	Read Unique Identifier Associated With Tag
12	Read Message
13	Read Tag Descriptor Date
14	Read PV Sensor Info
15	Read Output Information
16	Read Final Assembly Number
17	Write Message
18	Write Tag Descriptor Date
19	Write Final Assembly Number

Command	Definition
20	Read Long Tag
21	Read Unique Identifier Associated With Long Tag
22	Write Long Tag

HART Common Practice Commands Set

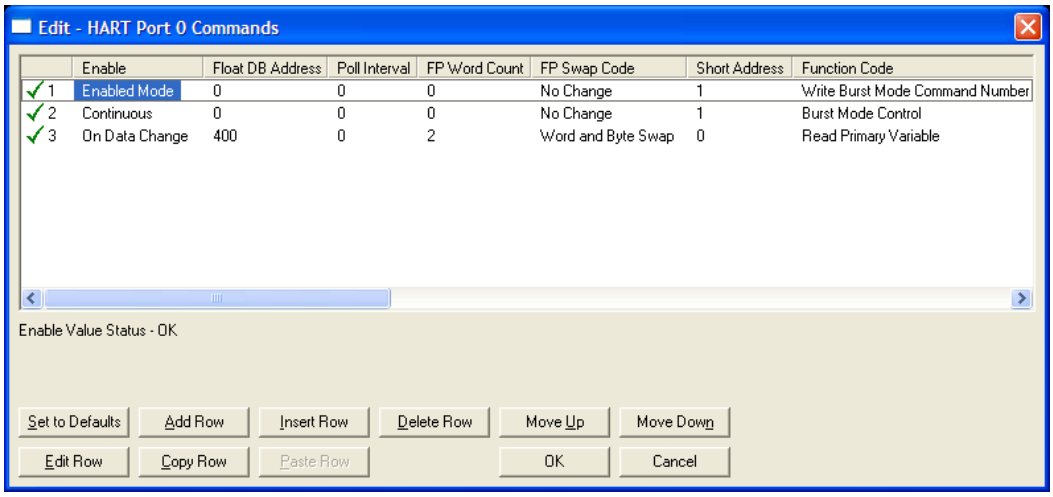
Command	Definition
33	Read Transmitter Variables
34	Write Damping Value
35	Write Range Values
36	Set Upper Range Value
37	Set Lower Range Value
38	Reset Configuration Changed Flag
39	EEPROM Control
40	Enter Exit Fixed Current Mode
41	Perform Transmitter Self Test
42	Perform Master Reset
43	Set PV Zero
44	Write PV Units
45	Trim DAC Zero
46	Trim DAC Gain
47	Write Transfer Function
48	Read Additional Transmitter Status
49	Write PV Sensor Serial Number
50	Read Dynamic Variable Assignments
51	Write Dynamic Variable Assignments
52	Set Transmitter Variable Zero
53	Write Transmitter Variable Units
54	Read Transmitter Variable Information
55	Write Transmitter Variable Damping Value
56	Write Transmitter Variable Sensor Serial Number
57	Read Unit Tag Descriptor Date
58	Write Unit Tag Descriptor Date
59	Write Number Of Response Preambles
60	Read Analog Channel and Percent of Range
61	Read Dynamic Variables and PV Analog Ch
62	Read Analog Channels
63	Read Analog Channel Information
64	Write Analog Channel Additional Damping Value
65	Write Analog Channel Range Values
66	Enter/Exit Fixed Analog Channel Mode
67	Trim Analog Channel Zero

Command	Definition
68	Trim Analog Channel Gain
69	Write Analog Channel Transfer Function
70	Read Analog Channel Endpoint Values
71	Lock Device
72	Squawk
73	Find Device
74	Read I/O System Capabilities
75	Poll Sub-Device
76	Read Lock Device State
79	Write Device Variable
80	Read Device Variable Trim Points
81	Read Device Variable Trim Guidelines
82	Write Device Variable Trim Point
83	Reset Device Variable Trim
105	Read Burst Mode Configuration
106	Flush Delayed Responses
107	Write Burst Device Variables
108	Write Burst Mode Command Number
109	Burst Mode Control
110	Read All Dynamic Variables

HART Command Entry Formats

Refer to the Reference (page 29, page 75, page 107) chapter for a complete discussion of the HART commands supported by the module, and the structure and content of the data returned for each command.

The following illustration shows a command list section of the configuration file:



Enable

0, 1, 2, 3, 4, 5

This field defines whether the command is to be executed and under what conditions.

Code	Description
0	The command is disabled and will not be executed in the normal polling sequence. This can be used to process a command from a bursting slave device.
1	Causes the command to be executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command will be executed, when the interval timer expires.
2	The command will execute only if the internal data associated with the command changes. This value is valid only when there is a specified "Write DB Address" (see below) with a non-zero byte count for write commands.
3	The HART module will send the command if either the MVI69-HART module OR the HART device is powered up. This is mainly used for configuration of HART devices on startup.
4	Places the command in enabled mode. This option is valid only if there is a specified "Enabled DB Address" (see below). If the Virtual Database word specified in "Enabled DB Address" has "-1" the command will be executed otherwise it will not.
5	Places the command in one shot enabled mode. This option is valid only if there is a specified "Enabled DB Address" (see below). If the Virtual Database word specified in "Enabled DB Address" has a value of "-1" the command will be executed otherwise it will not. When the command has been successful the Virtual Database word specified in "Enabled DB Address" will be written with "0", so the command will be executed only once.

Refer to Block 9902: Command Control (page 85) and Block 9903: Command Disable Control Block (page 85) for more information on how to use the enable code.

Float DB Address

0 to 3998

This field specifies the internal database register where the floating point values returned by the command will be placed.

Poll Interval

0 to 65535

This parameter specifies the minimum interval between executions of a continuous commands (*Enable* code of 1). The value is in seconds. Therefore, if a value of 10 is entered, the command will execute no more frequently than once every 10 seconds.

FP Word Count

-1 to 125

This parameter specifies the number of words from the floating point data returned by a HART command that will be placed on the Virtual Database. If this parameter is 0 no data will be written to the Database. If this parameter is -1 then all the floating point data will be written in the integer block of data.

Special care should be taken with this number, because is a word count and a floating point value is 2 words long. For example if you execute a HART command 3 which takes 5 floating point values from the device, you should place a word count of 10 words.

Swap Code

0, 1, 2, 3

This parameter defines the byte order of each four-byte group of data received. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard byte order for storing these data types. The following table describes the values and their associated operations:

Swap Code	Description
0	None - No Change is made in the byte ordering (1234 = 1234)
1	Words - The words are swapped (1234 = 3412)
2	Words & Bytes - The words are swapped then the bytes in each word are swapped (1234 = 4321)
3	Bytes - The bytes in each word are swapped (1234 = 2143)

Short Address

0 = Point-to-Point

1 to 15 = Multi-drop

This parameter specifies the HART slave node address on the network to which the command will be sent. Values of 0, or 1 to 15 are permitted. If the device to be addressed only accepts long address, then the parameter "Use Long" should be selected so the module can ask for the long address with the short one and then execute the command.

Function Code

0 to 255

This parameter specifies the HART function to be executed. Any HART function can be executed, even device specific ones, but only supported commands will return formatted data and classified in floating point data and integer data.

Int. DB Address

0 to 3999

This field specifies the internal database register where the integer or packed ASCII string values returned by the command will be placed.

If the Floating Point Word Count parameter is "-1", then all the data returned by the HART command will be placed in this address without any formatting.

Int Word Count

0 to 125

This parameter specifies the number of words from the integer or packed ASCII string data returned by a HART command that will be placed on the Virtual Database. If this parameter is "0", no data will be written to the Database.

Swap Code

0, 1, 2, 3

This parameter defines the byte order of each four-byte group of data received. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard byte order for storing these data types. The following table describes the values and their associated operations:

Swap Code	Description
0	None - No Change is made in the byte ordering (1234 = 1234)
1	Words - The words are swapped (1234 = 3412)
2	Words & Bytes - The words are swapped then the bytes in each word are swapped (1234 = 4321)
3	Bytes - The bytes in each word are swapped (1234 = 2143)

Use Long

0 or 1

This parameter defines if the command will be executed with short or long address. If the value is "0", then the configured command will be executed using the Short Address specified. If the value is "1" then the specified Short Address will be used only to ask for the long address and that will be used to execute the configured command.

Enable DB Address

0 to 3999

This field specifies the internal database register to be used to enable the execution of a command. This parameter is only used if "Enable" is "4" or "5". If the value of this database register is "-1", then the command will be executed, otherwise it will not. If the "Enable" value is "5", then after the successful execution of the command this value will become "0".

Done DB Address

-1 to 3999

This field specifies the internal database register to be used to signal the successful execution of a command. When a command is successfully executed a "-1" is written to this register. This "Done DB Address" can be shared with the "Enable DB Address" of another command to do a chained command execution.

Write DB Address

-1 to 3999

This field specifies the internal database register to be as a source of data for HART command which includes data. It is possible to include data with every HART command, but it depends of the command and of the device if it will accept this data.

If this value is "-1", it indicates that there will not be data with the command or that it will not come from database.

Swap Code

0, 1, 2, 3

This parameter defines the byte order of each four-byte group of data received. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard byte order for storing these data types. The following table describes the values and their associated operations:

Swap Code	Description
0	None - No Change is made in the byte ordering (1234 = 1234)
1	Words - The words are swapped (1234 = 3412)
2	Words & Bytes - The words are swapped then the bytes in each word are swapped (1234 = 4321)
3	Bytes - The bytes in each word are swapped (1234 = 2143)

Byte Count

0 to 250

This parameter specifies the number of bytes to be sent to a HART device in the command. If the command has no data then this value should be "0".

If the value of this field is different of "0" and "Write DB Address" is different of "-1" then the data for the command will be taken from the Virtual Database. If the value of this field is different of "0" and "Write DB Address" is "-1" then the data for the command will be from the "Fixed Data" field for the command.

Fixed Data

Up to 250 HEX values separated by space

This parameter is a string of HEX values to be sent with the HART command. There should be at least the number of bytes specified in the "Byte Count" parameter. The bytes should be written in hexadecimal format and separated by a space. This data will be sent if the Parameter "Write DB Address" is "-1" and "Byte Count" is greater than "0", and it will be sent in the same order that they are written.

2.3 Hart Command Examples

This section describes two examples that shows how to configure HART commands. The first example shows a read command (Function code 3 - READ DYNAMIC VARIABLES) and the second example shows a write command (Function code 34 - WRITE DAMPING VALUE).

2.3.1 Example of HART Command Function 3

The following example shows how to configure a command function 3 (READ DYNAMIC VARIABLES) to read the process variables from the HART slave device. According to the HART specification, this command will return four floating-point variables:

Word	High Byte	Low Byte
0	Current (mA)	
1		
2	Primary Variable	
3		
4	Second Variable	
5		
6	Third Variable	
7		
8	Fourth Variable	
9		

The command also returns the following integer data:

Word	High Byte	Low Byte
0	Status Word	
1	Primary Variable Units Code	Second Variable Units Code
2	Third Variable Units Code	Fourth Variable Units Code

This command returns the following number of words:

Parameter	Data Type	Direction	Number of Words	Database Address (this example)
Variable Results	Floating Point	Read from slave to MVI	10	240 (word address)
Status/Unit Codes	Integer	Read from slave to MVI	3	290 (word address)

You can configure the command parameters as described in the following table in order to correctly read the command 3 results to the module database:

Index	Parameter	Value	Observation
1	Enable	1	The command is sent continuously
2	Float DB Address	240	The floating point results will be copied to the internal database starting at word-address 240 (from 240 to 249)
3	Poll Interval	0	The command is sent without any delay
4	Word Count	10	The command returns 10 words of floating point data
5	Swap Code	3	The bytes in each returned floating point word are swapped
6	Short Address	0	This command is sent to the Hart slave device using short address 0
7	Function Code	3	The command function 3 (READ DYNAMIC VARIABLES) is used in this example
8	Int. DB Address	290	The integer data will copied to the database starting at word-address 290 (from 290 to 292)
9	Word Count	3	The command returns 3 words of integer data
10	Swap Code	3	The bytes in each returned integer data will be swapped
11	Use Long	1	This command executes using Long Address
12	Enable Address	-1	This command does not use this feature
13	DB Done Address	-1	This command does not use this feature
14	DB Write Address	-1	This parameter is ignored since no data is sent to the HART device (this is a read command)
15	DB Swap Code	3	This parameter is ignored since no data is sent to the HART device (this is a read command)
16	Byte Count	0	This parameter is ignored since no data is sent to the HART device (this is a read command)
17	Fixed Data		This command does not use this feature

These values can be entered in the configuration file as described below:

```
START
#     1        2        3     4    5       6    7       8     9   10   11      12      13       14      15    16    17
#       Float DB     Poll  Word Swap   Short Func Int. DB  Word Swap  Use  Enable DB Done DB Write DB Swap  Byte Fixed
#Enable  Address Interval Count Code Address Code Address Count Code Long Address Address  Address    Code  Count Data
      1      240        0    10    3       0    3     290     3    3    1      -1      -1       -1       3      0
END
```

In this example, verify that the command result is located inside the Read Data area, in order to have the data copied to the PLC processor (through the backplane). The following backplane configuration could be used for this example:

```
Read Register Start :    0 #Starting DB address where read by processor
Read Register Count :  300 #Number of regs for processor to read
Write Register Start:  300 #Starting DB address where write data placed
Write Register Count:  300 #Number of regs to write to module from processor
```

2.3.2 Example of HART Command Function 34

The following example shows how to configure a command function 34 (WRITE DAMPING VALUE) to write a damping values (seconds) to the HART slave device.

According to the HART specification, this command will write one floating point variables:

Word	High Byte	Low Byte
0	Floating Point Damping Value (Sec)	
1		

The HART device should also return the floating point value after it processes the request:

Word	High Byte	Low Byte
0	Floating Point Damping Value (Sec)	
1		

The HART device also returns an integer status data:

Word	High Byte	Low Byte
1	STATUS WORD	

This command returns the following number of words:

Parameter	Data Type	Direction	Number of Words	Database Address (this example)
Damping Value	Floating Point	Written from MVI to slave	2 (4 bytes)	400 (word address) or 800 (byte address)
Damping Value	Floating Point	Read from slave to MVI	2	240 (word address)
Status Word	Integer	Read from slave to MVI	1	290 (word address)

You can configure the command parameters as described in the following table:

Index	Parameter	Value	Observation
1	Enable	1	The command is sent continuously
2	Float DB Address	300	The damping value response will be copied to the internal database starting at word-address 300 (occupies addresses 300 to 301)
3	Poll Interval	0	The command is sent without any delay
4	Word Count	2	The command returns 10 words of floating point data
5	Swap Code	3	The bytes in each returned floating point word are swapped
6	Short Address	0	This command is sent to the Hart slave device using short address 0
7	Function Code	34	The command function 4 (WRITE DAMPING VALUE) is used in this example
8	Int. DB Address	320	The integer data will copied to the database word-address 320
9	Word Count	1	The command returns 1 word of integer data
10	Swap Code	3	The bytes in each returned integer data will be swapped
11	Use Long	1	This command executes using Long Address
12	Enable Address	-1	This command does not use this feature
13	DB Done Address	-1	This command does not use this feature
14	DB Write Address	800	The floating point value located at byte-address 800 (word-address 400) will be used as the damping value to be written to the HART device
15	DB Swap Code	3	The bytes in each returned integer data will be swapped
16	Byte Count	4	The command will write 4 bytes (2 words) to the HART device, since the damping value uses floating point format (2 words)
17	Fixed Data		This command does not use this feature

```
START
#      1        2        3     4    5       6    7       8     9   10   11      12      13       14      15    16    17
#        Float DB     Poll  Word Swap   Short Func Int. DB  Word Swap  Use  Enable DB Done DB Write DB Swap  Byte Fixed
# Enable  Address Interval Count Code Address Code Address Count Code Long Address Address  Address    Code Count  Data
       1      300        0     2    3       0   34     320     1    3    1      -1      -1       800       3     4
END
```

Verify that addresses 300 and 320 are located inside the Read Data area (read from the MVI database to the PLC processor). Address 400 would have to be located inside the Write Data area (written from the PLC processor to the MVI database).

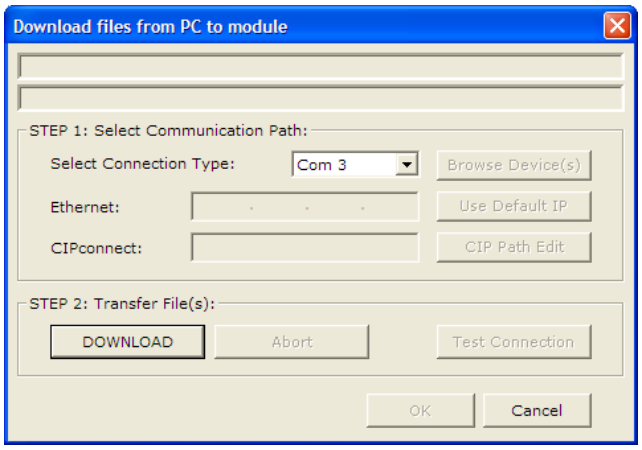
2.4 Downloading the Configuration to the Module Using Serial

For the module to use the settings you configured, you must download (copy) the updated *Project* file from your PC to the module.

Note: The first time you download the project to the module, you must use the serial COM port to download the project, including the IP address. After that, you can use the Ethernet port to communicate with the module.

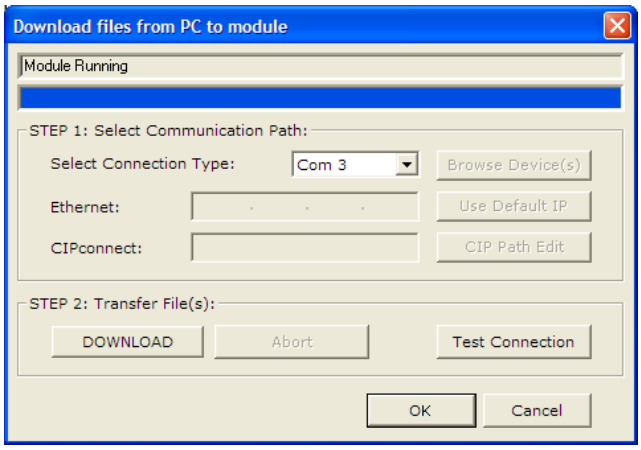
To download the project file

1 In the tree view in *RSLogix*, right-click the module icon, and choose **DOWNLOAD FROM PC TO DEVICE**. The program scans your PC for a valid com port (this may take a few seconds). When the *RSLogix* finds a valid COM port, it opens the *Download files from PC to module* dialog box.



2 Choose the COM port to use from the dropdown list, and then click **DOWNLOAD**.

The module performs a platform check to read and load its new settings. When the platform check is complete, the status bar in the *Download files* dialog box displays the message *Module Running*.



3 Ladder Logic

In This Chapter

Ladder logic is required for the MVI69-HART module to operate. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

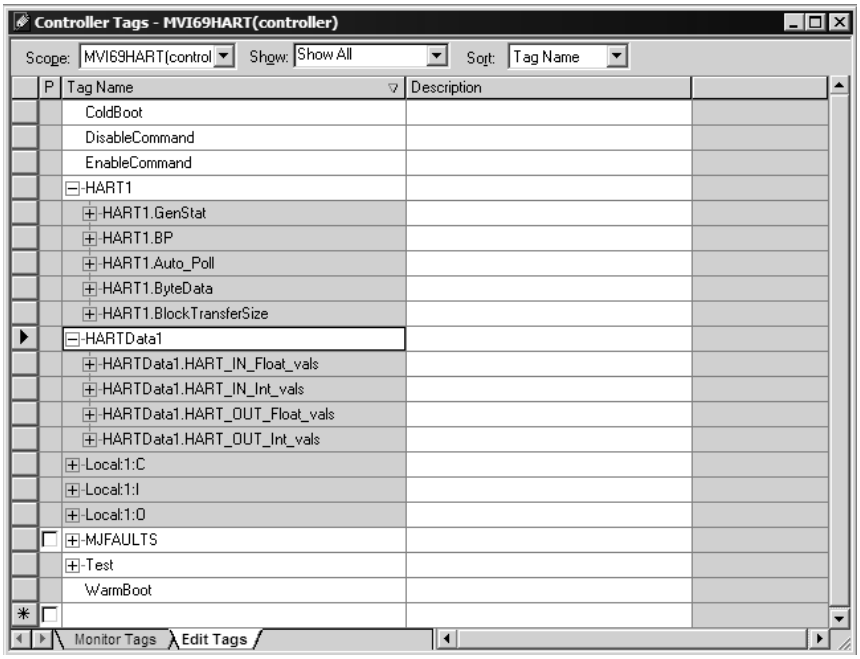
The sample ladder logic is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

3.1 Using the Sample Program

The sample program consists of user defined data types, controller tags and ladder rungs that allow communication between the CompactLogix or MicroLogix processor, the MVI69-HART module, and the HART network. You can use the sample program as-is, or you can incorporate the data types, controller tags and ladder rungs into an existing program.

3.1.1 Controller Tags

The MVI69-HART module is defined in the example as HARTData1 to hold all data related to the database (type HARTData) and HART1 to hold all status data related to the module (type HARTModuleDef). The tag name can be set to any valid tag name desired.



3.1.2 Module Status Data and Variables (HARTModuleDef)

All status and variable data related to the MVI69-HART is stored in a user-defined data type. An instance of the data type is required before the module can be used. This is accomplished 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
GenStat	HARTStat	General status information
BP	HARTBackplane	Data to handle backplane logic
Auto_Poll	HARTAutoPoll[10]	Auto-PollData
ByteData	SINT[1000]	
BlockTransferSize	INT	

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

Status Object (HARTStat)

This object stores the status data of the module. The HARTStat object (shown in the following example) 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
Scan_Cnt	INT	Program Scan Counter
Product_Name	SINT[4]	Product Code
Rev_Level	SINT[4]	Revision
Op_Sys	SINT[4]	Operating system revision
Run_Number	SINT[4]	Run number
Blk_Rd_Count	INT	Number of block read transfers
Blk_Wr_Count	INT	Number of block write transfers
Blk_Parse_Cnt	INT	Number of blocks parsed by module
Blk_Resv	INT	Reserved
Blk_Cmd	INT	Number of command enable/disable blocks
Blk_Err	INT	Number of block errors
ChStat	HARTCHStat[2]	Channel Status Data

Within the HARTStat objects are objects containing the status information for each application port (HARTCHStat). Refer to Status Data Definition (page 100) for a complete listing of the data stored in this object.

Channel Status Object (HARTCHStat)

The HARTCHStat object holds the status data related to a single HART master port. The following table describes the structure of the object.

Name	Data Type	Description
State	INT	State machine value
Comm_State	INT	Comm port state machine value
Dev_stat	INT	Device bits (OK/Err)
Dev_in_poll	INT	Device bits for poll list
Dev_long	INT	Device bits for long address
Cur_cmd	INT	Current command list index
Cmd_req	INT	Number of command requests issued
Cmd_resp	INT	Number of command responses received
Cfg_Err	INT	Configuration Error Word for channel
CurErr	INT	Current error code for channel
LastErr	INT	Last error for channel

This information is passed to the controller from the module with each normal read block image.

Configuration/Error Status Flags (Cfg_Err)

The Cfg_Err word member of the HARTCHStat reports configuration errors for the respective port. If the module is not functioning as expected, inspect the value presented in this object. If a configuration error exists, the associated bit is set. A value of zero for the bit indicates the configuration value is valid. This does not guarantee that the module is configured correctly for your application. The bits used by this member are shown in the following table.

Bit	Code	Description
0	0x0001	Enabled not set to Y or N
1	0x0002	Enable Handheld not set to Y or N
2	0x0004	Primary Master not set to Y or N
3	0x0008	Invalid Preambles (1 to 50)
4	0x0010	Invalid Short Address Retries (0 to 50)
5	0x0020	Invalid Long Address Retries (0 to 50)
6	0x0040	Invalid Retries After Error (0 to 50)
7	0x0080	Invalid Poll Time After Error (0 to 10000)
8	0x0100	Invalid DB Address Status
9	0x0200	Invalid Command Count
10	0x0400	Memory Error in allocating commands
11	0x0800	Memory Error in allocating command fixed data
12	0x1000	Memory Error in allocating TX/RX buffers
13	0x2000	HART Board not found
14	0x4000	Cannot initialize HART channel
15	0x8000	

Backplane Object (HARTBackplane)

The HARTBackplane object stores all the variables required for the data transfer operation between the module and the controller. The LastRead data member is used as the handshaking byte to indicate the arrival of new data from the module. The following table describes the structure of the 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

The other members of the object are be used in the ladder logic to assist in the data transfer operation.

3.1.3 Data Object (HARTData)

The HARTData object is defined in the example ladder logic to demonstrate how the data from a module can be stored in the processor. This object can be used to temporarily store a received data set. In this example, four data areas are set up; two for input data (data received from the module) and two for output data (data to transfer to the module). Within each of these two areas, an area for floating-point data and integer data is defined. The following table describes the structure of the object.

Name	Data Type	Description
HART_IN_Float_vals	REAL[100]	Data area to floating-point data
HART_IN_Int_vals	INT[500]	Data area to hold integer data
HART_OUT_Float_vals	REAL[100]	Data area for output floating-point data
HART_OUT_Int_vals	INT[500]	Data area for output integer data

Auto-Poll Data (HARTAutoPoll)

This data object contains the auto-poll data ready to be used by the user (multi-drop or point-to-point). First define where the auto-poll data will copied to in the MVI69-HART database using the DB AutoPoll Address parameter in the configuration file. Then copy the data from the MVI69-HART database into the processor memory (HARTAutoPoll object) using the ladder logic.

Name	Data Type	Description
Auto_Poll_CMD_Status	SINT	Auto-Polling command status bits
Last_Status_Byte	SINT	Last first status byte received from device
Last_Second_Status_Byte	SINT	Last second status byte received from device
Manufacture_ID_Code	SINT	
Device_Type_Code	SINT	
Min_Preambles	SINT	Minimum Number of Preambles
Universal_CMD_Major	SINT	Universal Command Major Rev#
Device_Rev_Level	SINT	
Software_Rev_Level	SINT	
Hardware_Rev_Level	SINT	
Device_Flags	SINT	
Device_ID	SINT[3]	
Min_Preambles_Resp	SINT	
Max_Number_Devices	SINT	
Config_Change_Count	INT	
Ext_Dev_Status	SINT	Extended Field Device Status
Primary_Var_Units	SINT	Primary Units Code
Secondary_Var_Units	SINT	Secondary Units Code
Tertiary_Var_Units	SINT	Tertiary Units Code
Quaternary_Var_Units	SINT	Quaternary Units Code
Tag_Name	SINT[8]	

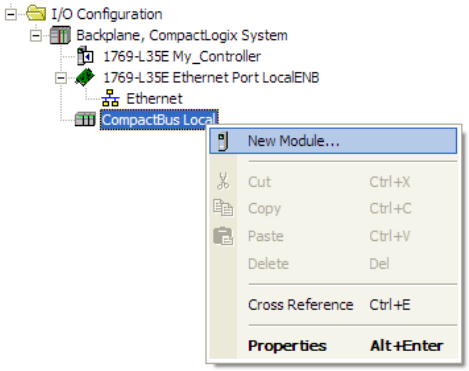
Name	Data Type	Description
Descriptor	SINT[16]	
Tag_Descriptor_Data	SINT[3]	
Transd_Serial_Num	SINT[3]	
Transd_limits	SINT	
PV_Alarms	SINT	
PV_Transfer	SINT	
PV_URLLRV_Units	SINT	
Write_Protection_Code	SINT	
Private_Label_Code	SINT	
PV_Analog_Ch_Flag	SINT	
PV_value	REAL	Primary variable value
SV_value	REAL	Secondary variable value
TV_value	REAL	Tertiary variable value
QV_value	REAL	Quaternary variable value
U_trans_limit	REAL	Upper transducer limit
L_trans_limit	REAL	Lower transducer limit
Min_span	REAL	Minimum span
PV_URV	REAL	PV upper range value
PV_LRV	REAL	PV lower range value
PV_damp	REAL	PV dampining value (in seconds)

3.2 Adding the Module to an Existing CompactLogix Project

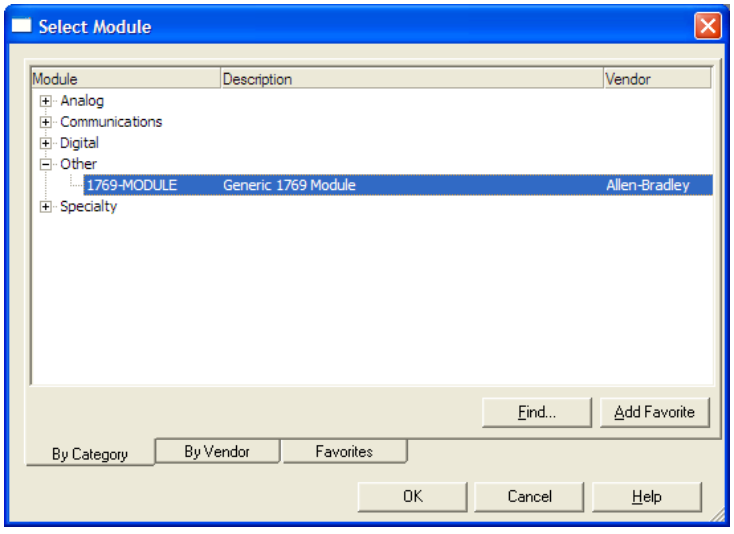
Important: The MVI69-HART module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus, to the right of the processor).

If you are installing and configuring the module with a CompactLogix controller, and you decide to use the standard sample ladder logic rather than importing the Add-On Instruction (AOI) Rung, then follow these steps. If you are using a MicroLogix controller, refer to the Adding the Module to an Existing MicroLogix Project (page 53).

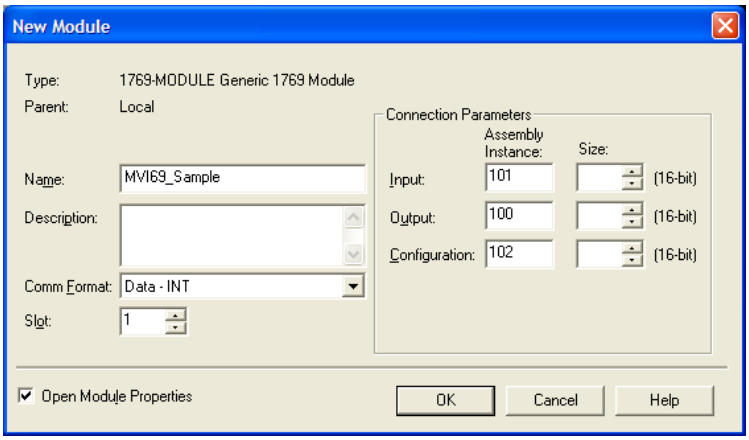
1 Add the MVI69-HART module to the project. Right-click the mouse button on the **I/O CONFIGURATION** option in the *Controller Organization* window to display a pop-up menu. Select the **NEW MODULE** option from the **I/O CONFIGURATION** menu.



This action opens the *Select Module* dialog box:



2 Select the *1769-Module* (Generic 1769 Module) from the list and click **OK**.



3 Enter the *Name, Description* and *Slot* options for your application, using the values in the illustration above. You must select the *Comm Format* as **DATA - INT** in the dialog box, otherwise the module will not communicate over the backplane of the CompactLogix rack.

4 Configure the *Connection Parameters* to match to the *Block Transfer Size* parameter in the configuration file. Use the values in the table corresponding with the block transfer size you configured.

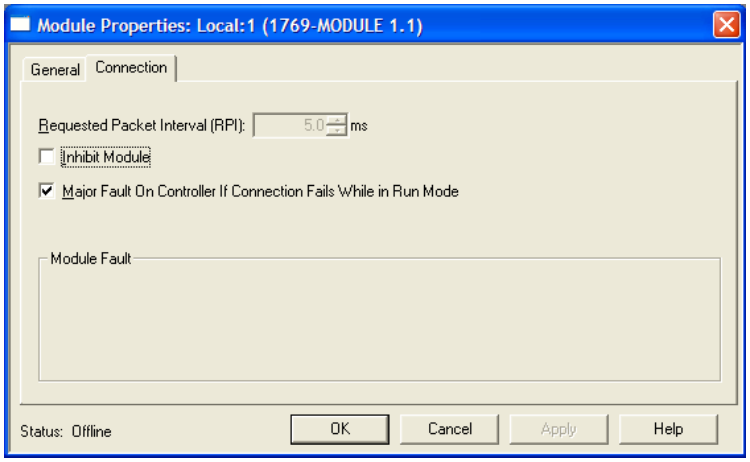
Block Transfer Size = 60	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	**62**
Output Assembly Instance	100
Output Size	**61**
Configuration Assembly Instance	102
Configuration Size	**0**

Block Transfer Size = 120	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module

Block Transfer Size = 120	
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	**122**
Output Assembly Instance	100
Output Size	**121**
Configuration Assembly Instance	102
Configuration Size	**0**

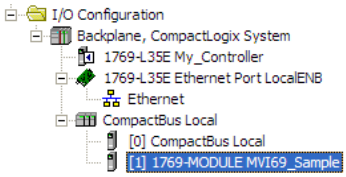
Block Transfer Size = 240	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	**242**
Output Assembly Instance	100
Output Size	**241**
Configuration Assembly Instance	102
Configuration Size	**0**

5 Click **NEXT** to continue.



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

7 **SAVE** the module. Click **OK** to dismiss the dialog box. The *Controller Organization* window now displays the module's presence. The following illustration shows the *Controller Organization* window:



8 Copy the U*ser-defined Data Types* (UDTs) from the sample program.

9 Copy the C*ontroller Tags* from the sample program.

10 Copy the *Ladder Rungs* from the sample program.

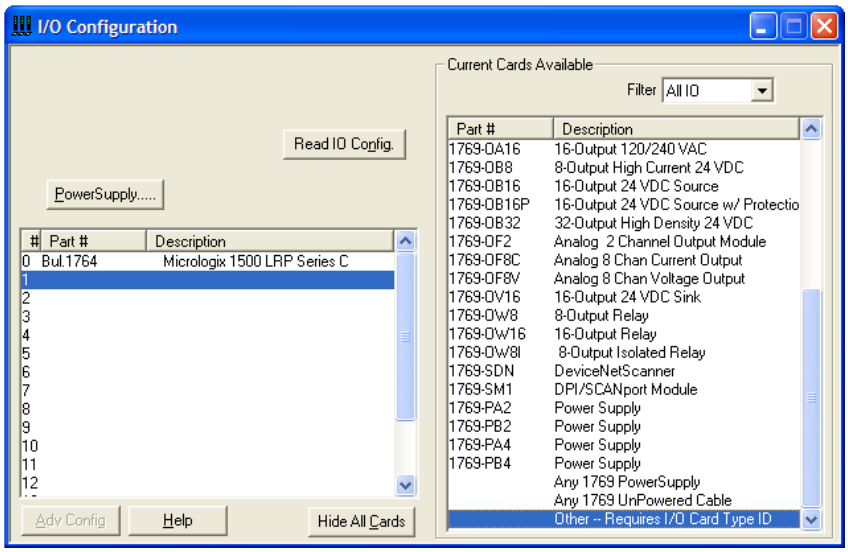
11 Save and download the new application to the controller.

3.3 Adding the Module to an Existing MicroLogix Project

If you are installing and configuring the module with a MicroLogix controller, follow these steps. If you are using a CompactLogix controller, refer to the Adding the Module to an Existing CompactLogix Project.

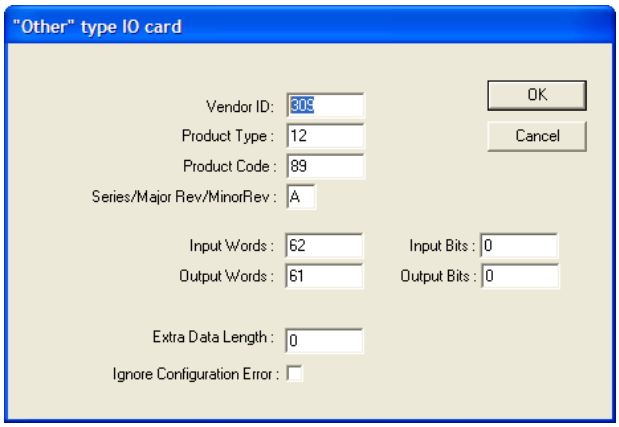
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

1. In RSLogix, open your existing application, or start a new application, depending on your requirements.
2. Double-click the **I/O CONFIGURATION** icon located in the *Controller* folder in the project tree. This action opens the *I/O Configuration* dialog box.



3. In the *I/O Configuration* dialog box, select "**OTHER - REQUIRES I/O CARD TYPE ID**" at the bottom of the list in the right pane, and then double-click to open the *"Other" type IO card* dialog box.

4 Enter the values shown in the following illustration to define the module correctly for the MicroLogix processor, and then click **OK** to save your configuration.



The *Input Words* and *Output Words* parameters will depend on the *Block Transfer Size* parameter you specify in the configuration file. Use the values from the following table.

Block Transfer Size	Input Words	Output Words
60	**62**	**61**
120	**122**	**121**
240	**242**	**241**

5 Click **OK** to continue.

6 After completing the module setup, the *I/O Configuration* dialog box will display the module's presence.

The last step is to add the ladder logic. If you are using the example ladder logic, adjust the ladder to fit your application.

Download the new application to the controller. If you encounter errors when you attempt to run the program, refer to Diagnostics and Troubleshooting (page 55) for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

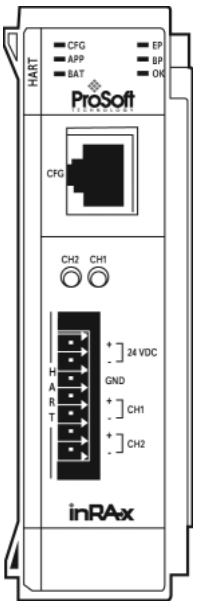
4 Diagnostics and Troubleshooting

In This Chapter

The MVI69-HART provides information on diagnostics and troubleshooting in the following forms:

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

4.1 LED Status Indicators



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

LED	Color	Status	Indication
CFG	Green	On	NA
		Off	NA
EP	Green	On	NA
		Off	NA
		Off	No data is being transferred on the Configuration/Debug port.
APP	Amber	Off	The MVI69-HART is working normally.
		On	The MVI69-HART module program has recognized a communication error.
BP	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.
CH01	Green	On	Data is being transmitted between the module and the processor.
		Off	
CH02	Green	On	Data is being transmitted between the module and the processor.
		Off	

4.1.1 Clearing a Fault Condition

Typically, if the OK LED on the front of the module remains RED for more than ten seconds, a hardware problem has been detected 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 it 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 CompactLogix or MicroLogix 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 securely plugged into the slot that has been configured for the module in the I/O Configuration in RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI69-HART. Verify that all modules in the rack are correctly configured.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI69 modules with scrolling LED display: *<Backplane Status>* condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug (or Communication) port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack, then re-insert it, and then restore power to the rack.

4.2 Reading Status Data from the Module

The MVI69-HART module returns a status data set to the CompactLogix processor in read blocks with identification codes of 0 and -1. This data is transferred to the CompactLogix processor continuously.

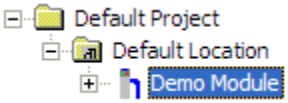
The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Version Information
- Control over the module (warm boot and cold boot)
- Facility to upload and download the module's configuration file

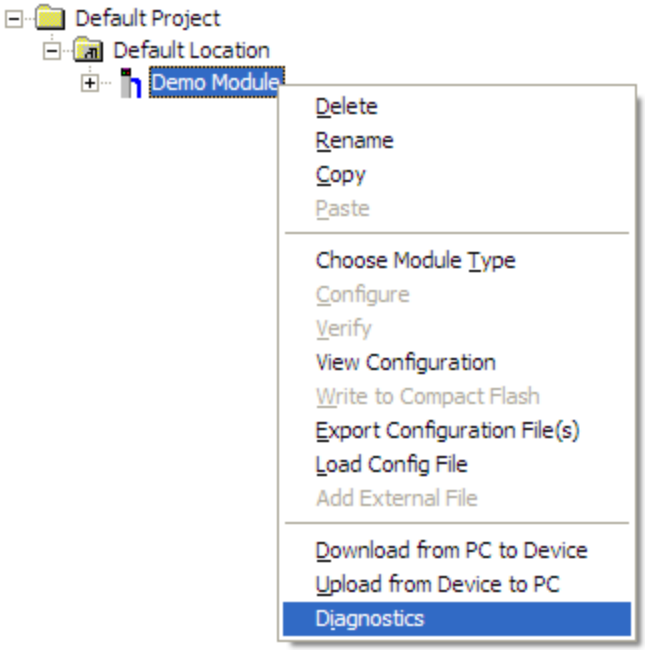
4.2.1 Using ProSoft Configuration Builder (PCB) for Diagnostics

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

1 Start *PCB*, and then right-click the module icon.

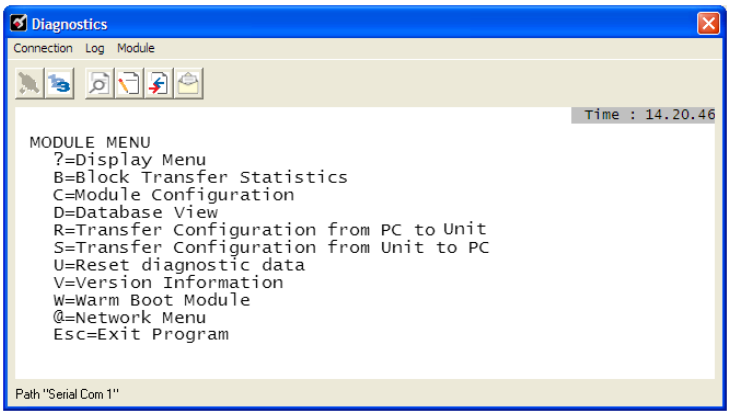


2 On the shortcut menu, choose **DIAGNOSTICS**.



This action opens the *Diagnostics* dialog box.

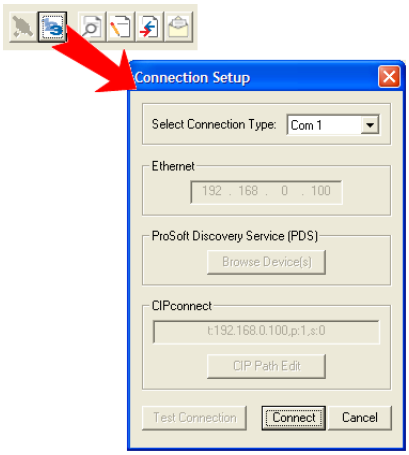
3 Press **[?]** to open the *Main* menu.



Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

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

1 Click the Setup Connection button to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.



2 For a serial connection, 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.

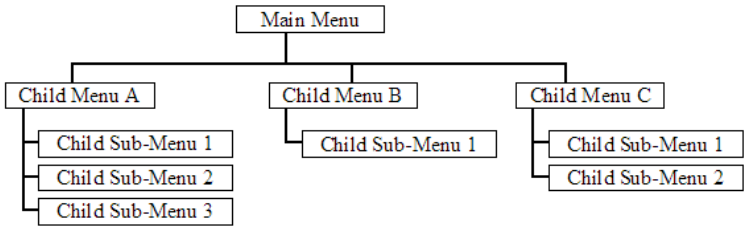
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

All of the submenus in *ProSoft Configuration Builder* for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu to the next higher menu by pressing **[M]** on your keyboard.

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



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

Keystrokes

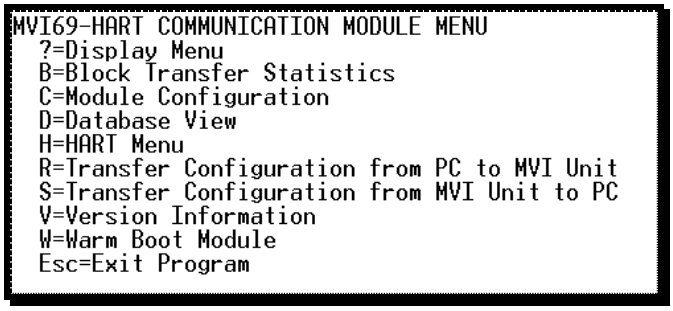
The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

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

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

4.2.2 Main Menu

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



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

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

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 number of seconds passed between the two readings.

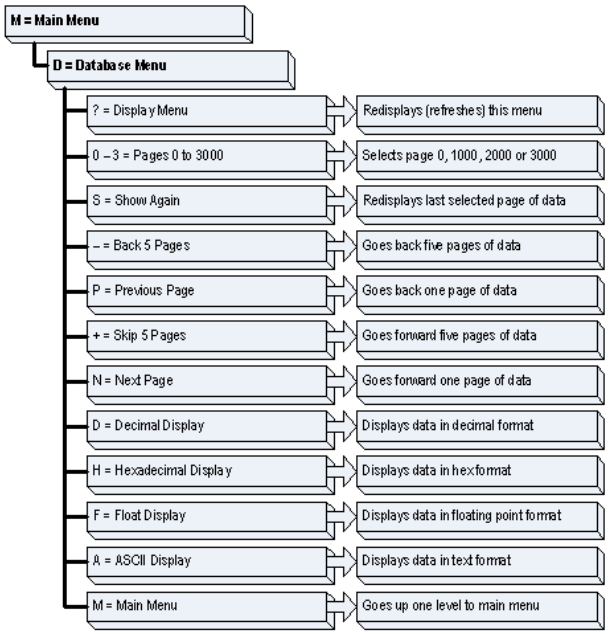
Viewing Module Configuration

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

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.



Opening the HART Master Menu

Press **[H]** to open the HART Master Menu. This menu allows you to view information about the protocol driver.

```
HART INTERFACE MENU
  ?=Display Menu
  A=HART Data Analyzer
  V=Slave Status List
  M=Main Menu
  HART Command List Errors:
      E=Port 0   F=Port 1   G=Port 2   H=Port 3
  HART Command List:
      N=Port 0   O=Port 1   P=Port 2   Q=Port 3
  Port Status and Configuration:
      1=Port 0   2=Port 1   3=Port 2   4=Port 3
```

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module to the PC. For more information on receiving and sending configuration files, refer to Downloading the PCB File to the Module.

Sending the Configuration File

Press **[S]** to upload (send) a configuration file from the module to your PC. For more information on receiving and sending configuration files, refer to Downloading the PCB File to the Module.

Viewing Version Information

Press **[V]** to view version information for the module. Use this command to view the current firmware version of the software (Software Revision Level) for the module, as well as other important values.

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 program execution frequncy.

Warm Booting the Module

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

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

Press **[W]** from the *Main* menu to warm boot (restart) the module. This command causes the program to exit and reload, refreshing configuration parameters that must be set on program initialization.

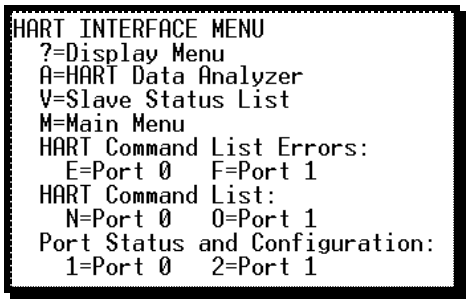
Exiting the Program

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

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

Press **[ESC]** to restart the module and force all drivers to be loaded. The module uses its flash memory configuration to configure the module.

4.2.3 HART Master Menu



Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

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 68) 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 the Slave Status List

Press **[V]** to view the slave status values associated with the ports. The slave status values are defined as follows:

- ERR = Device in Error
- OK = Device OK
- [Blank] = Device Not Polled.

Opening the Command Error List Menu

Press **[E]** (port 1) or **[F]** (port 2) to open the Command Error List. This list consists of multiple pages of command list error/status data. Press **[?]** to view a list of commands available on this menu.

Opening the Command List Menu

Press **[E]** (port 1) or **[F]** (port 2) to open the Command List menu. Use this command to view the configured command list for the module.

Viewing the Master Command List Help

Press **[H]** to view a help screen with explanations of each item that appears on the command list.

```
HART MASTER COMMAND LIST HELP

 EN     = ENABLE
 F.REG  = FLOATING POINT DATA ADDRESS IN DATABASE
 P.INT  = POLL INTERVAL IN SECONDS
 F.CNT  = COUNT OF FLOATING POINT DATA REQUIRED
 S      = FLOATING POINT DATA SWAP CODE
 DEV    = HART DEVICE ID (SHORT ADDRESS)
 FUN    = HART FUNCTION
 I.REG  = INTEGER DATA ADDRESS IN DATABASE
 I.CNT  = COUNT OF INTEGER DATA REQUIRED
 S      = INTEGER DATA SWAP CODE
 LNG    = LONG ADDRESS POLLING TO THE HART NETWORK
 E.REG  = ENABLE REGISTER IN DATABASE FOR THIS COMMAND
 D.REG  = DONE REGISTER IN DATABASE FOR THIS COMMAND
 W.REG  = WRITE DATA ADDRESS IN DATABASE
 S      = WRITE DATA SWAP CODE
 BC     = WRITE DATA BYTECOUNT
 STAT.  = LAST HART STATUS FOR THIS COMMAND
```

Viewing Port Status and Configuration

Press **[1]** (port 1) or **[2]** (port 2) to view status and configuration for ports 0 through 3 respectively.

HART Error Descriptions

Error Type	Description
Gap Errors	Increments when a delay of more than 20 milliseconds occurs between characters in a HART message
Overflow	Increments when a received HART message is longer than the internal buffer can hold
SOM Errors	**Start of Message** error - Increments whenever the module does not see at least 3 preambles (FF characters or all bits set ON) at the beginning of a HART message
Retry Count	Increments every time a HART command fails and is retried
Check Byte	Increments when a Checksum error is detected in the received HART packet.
Overrun/Parity/Frame	Increments every time one of three errors occur: **Overrun** - The HART driver wasn't able to read the current data byte before a new one arrived, causing the current character to be lost. **Parity -** HART communications uses EVEN parity. A byte is received with parity error. **Frame** - There is a zero bit where the stop bit should be. The message is not formatted correctly. These errors are typically caused by electrical wiring problems or electrical interference on the network loop.
Response Timeout	Increments when the response to a HART Command is not received within the configured timeout period.

Returning to the Main Menu

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

4.2.4 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.

```
DB Menu Selected
DATABASE VIEW MENU
 ?=Display Menu
 0-3=Display data page 0-3000
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 D=Decimal Display
 H=Hexadecimal Display
 F=Float Display
 A=ASCII Display
 M=Main Menu
```

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.

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

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

Moving Back Through 5 Pages of Registers

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

Moving Forward (Skipping) Through 5 Pages of Registers

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

Viewing the Previous Page of Registers

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

Viewing the Next Page of Registers

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

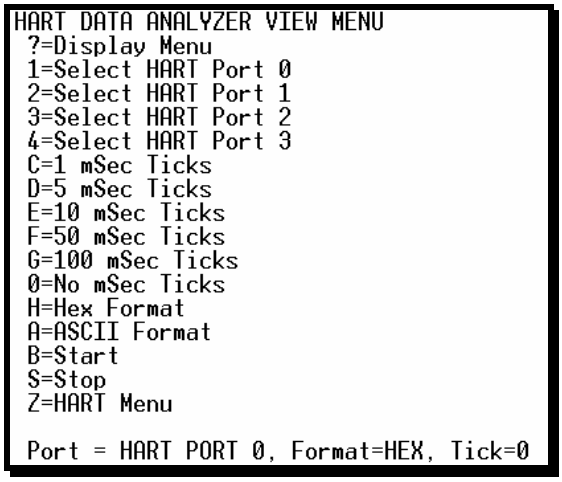
Press **[A]** from the *Database View* menu to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.2.5 Data Analyzer

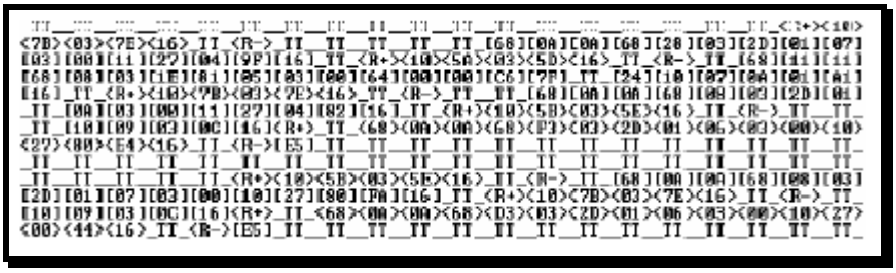
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.



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

Analyzing Data for Port 1

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



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

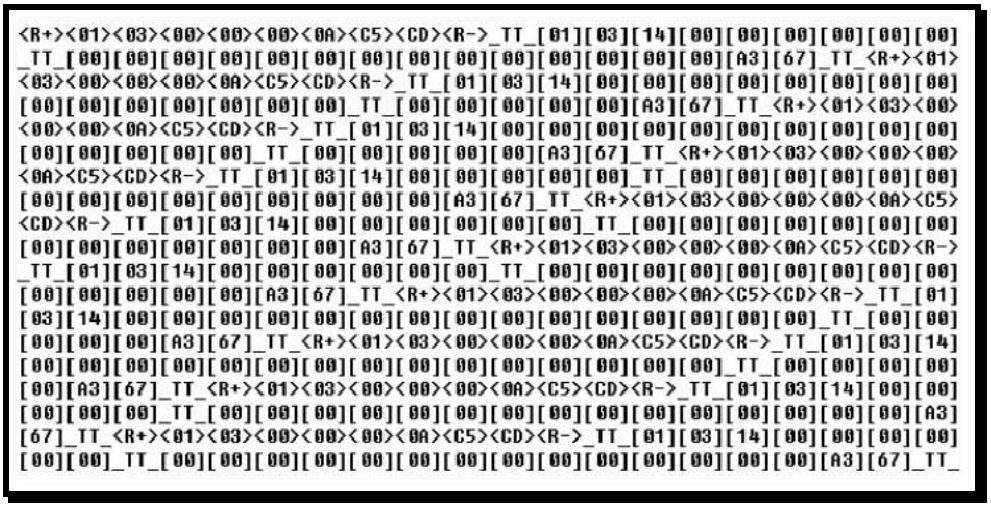
Press **[H]** from the *Database View* menu to display the data on the current page in hexadecimal format.

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.



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.

4.2.6 Data Analyzer Tips

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

```
Data Analyzer Mode Selected
```

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

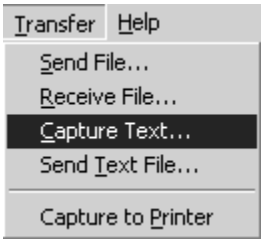
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

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

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

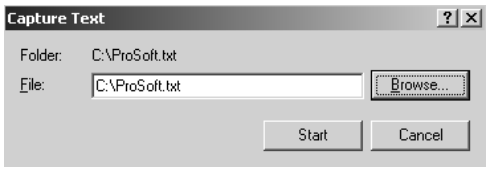
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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

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



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



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

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

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

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

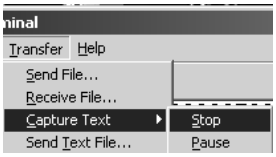
All characters shown in <> brackets are characters being sent out by the module.

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.

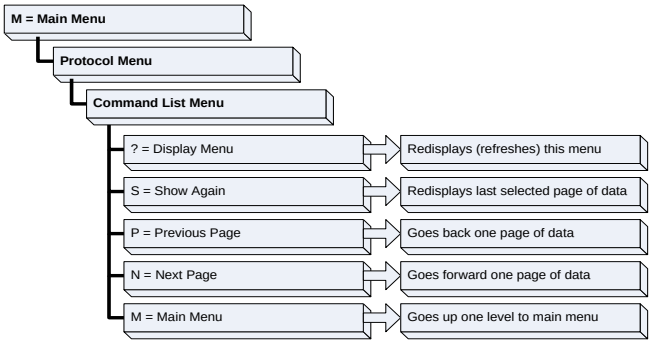
When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



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.

4.2.7 Master Command Error List Menu

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



Redisplaying the Current Page

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

Moving Back Through 5 Pages of Commands

Press **[-]** to display data for last 5 page commands.

Viewing the Previous Page of Commands

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

Moving Forward (Skipping) Through 5 Pages of Commands

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

Viewing the Next Page of Commands

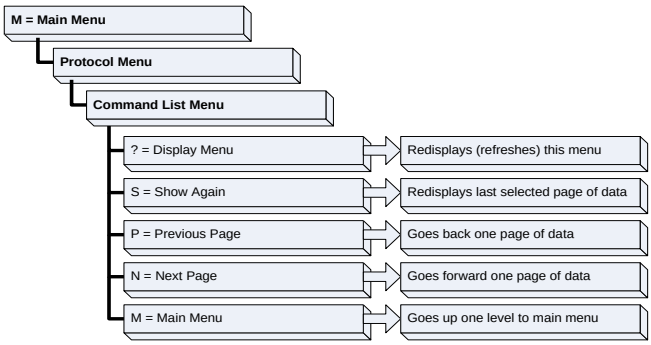
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.2.8 Master Command List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 50 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 50 Commands

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

Viewing the Next Page of Commands

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

Returning to the Main Menu

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI69 HART Multi-drop Master Communication Module is an CompactLogix or MicroLogix backplane-compatible module that allows CompactLogix or MicroLogix processors to interface easily with HART compatible devices. Devices commonly supporting the protocol include pressure, temperature, flow transmitters, as well as other similar instruments commonly found in the process (and other) industry.

The MVI69-HART Master Communication Module allows Rockwell Automation CompactLogix or MicroLogix compatible processors to easily communicate with HART slave devices. The MVI69-HART module interfaces up to 15 devices on each HART channel with the CompactLogix or MicroLogix processor. HART channels on the module support master protocol commands to interface with slave devices on their own networks. Each port is individually configurable. Data is exchanged between the HART network and the CompactLogix or MicroLogix processor backplane using the internal database contained in the module and direct control by the processor's ladder logic and pre-defined data objects (4000 registers maximum).

The MVI69-HART module is the perfect solution for industrial applications in chemical and refining operations, to gas and liquid distribution systems, and remote offshore monitoring stations are addressing virtually all aspects of control, data acquisition, and maintenance.

HART® is a registered trademark of the HART Communication Foundation.

5.1.1 General Specifications

- Single-slot, 1769 backplane-compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module.
- Ladder Logic is used for data transfer between module memory and processor controller tags. A sample ladder file with AOI is included.
- Configuration data obtained from configuration text file downloaded to module. A sample configuration file is included.
- Supports MicroLogix 1500 LRP Controllers.
- Supports CompactLogix Programmable Automation Controllers (CPLX PAC) that have a 1769 backplane bus and that can provide at least 800 mA of 5 VDC on the backplane. In some systems, one or more additional power supplies may be needed to provide adequate backplane power.

5.1.2 Hardware Specifications

Specification	Description
Dimensions	Standard 1769 single-slot module
Current Load	800 mA max @ 5 VDC Power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)
Operating Temp.	0°C to 60°C (32°F to 140°F)
Storage Temp.	-40°C to 85°C (-40°F to 185°F)
Relative Humidity	5% to 95% (with no condensation)
LED Indicators	Power and Module Status Application Status HART Channel Status Serial Port Activity Serial Activity and Error Status
Debug/Configuration Port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only No hardware handshaking
Application Port	
App Port	HART terminal connector

5.1.3 Functional Specifications

The MVI69-HART module supports the HART Multi-drop implementation of the protocol. Following are some general specifications for the module:

- Built accordance to the HART Bell 202 Frequency Shift Keying (FSK) standard to superimpose digital signals at a low level on top of the 4 to 20 mA
  - Two independent HART master ports that are completely user-configurable
  - Supports up to 15 devices per port
  - Point-to-point (slave address 0) or multi-drop (slave address 1 to 15) modes supported
- Supports 99 universal and common practice commands per port to control and monitor devices with integer, IEEE754 floating-point and packed ASCII character string data blocks
- Burst mode can be used for faster update of data from a single slave
- Supports an auto polling feature that will automatically collect data from each HART instrument on the channel and store the data in the module database
- Communication ports can be configured as a secondary Master (that is, handheld configuration device)

Protocol Supported: HART protocol uses the Bell 202 standard frequency shift-keying (FSK) digital signal to communicate at 1200 baud, superimposed at a low level on the 4 to 20 mA analog measurement signal. The MVI69-HART module supports version 5 of the HART protocol.

Supported Function Codes: HART Universal Commands Set supported are 00 to 03, 06 to 09, and 11 to 22. HART Common Practice Commands Set supported are 33 to 83 and 105 to 110.

HART Network Communications: Supports two Master channels. Each channel on the module is configured independently to emulate a HART Master. Burst mode can be used for faster update of data from a slave device.

Command polling is also user-configurable, including disabled, continuous, on change of data (write only), and dynamically user or automatic enabled.

Status: Error codes returned by the HART protocol available on an individual command basis. In addition, a slave status list is maintained per active channel.

5.2 Functional Overview

5.2.1 General Concepts

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

Module Power Up

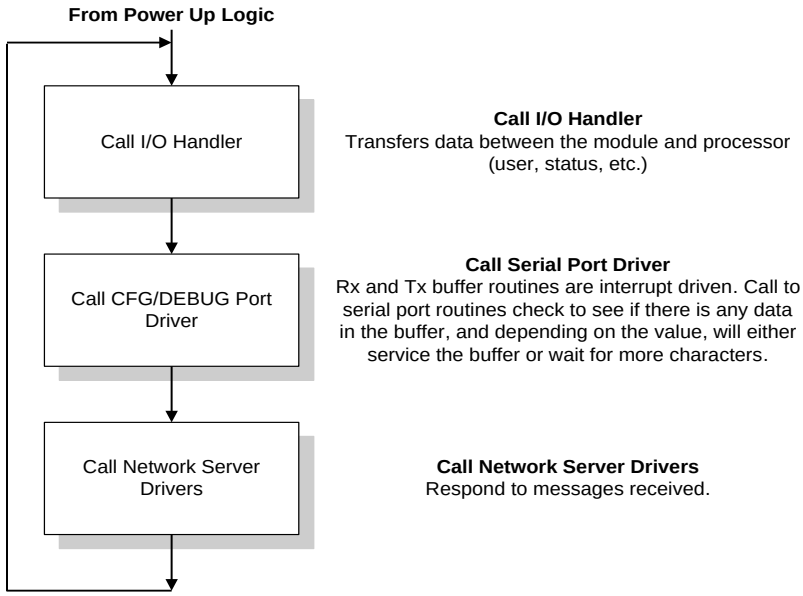
On power up the module begins performing the following logical functions:

- Initialize hardware components
  - Initialize CompactLogix or MicroLogix backplane driver
  - Test and Clear all RAM
  - Initialize the serial communication ports
- Read module configuration from the Compact Flash
- Initialize Module Register space
- Set up the communication interface for the debug/configuration port

When this initialization procedure is complete, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.



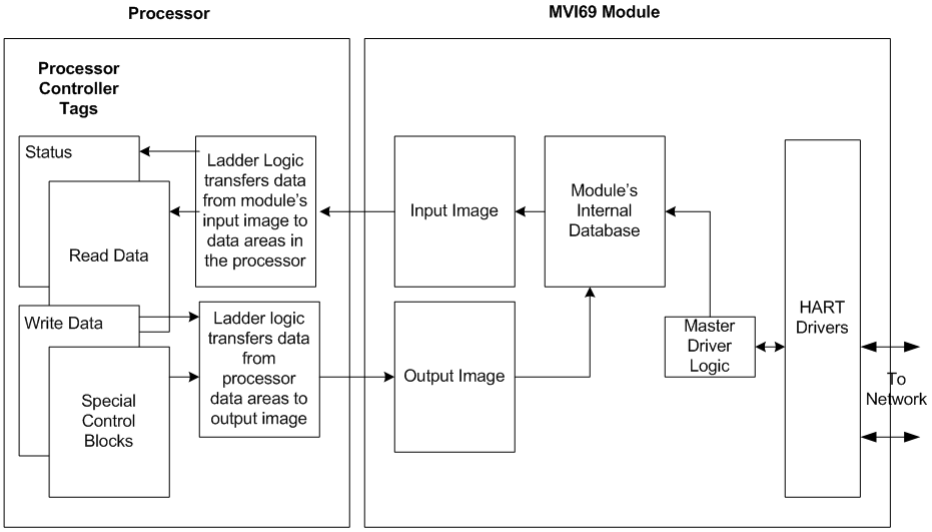
5.2.2 Backplane Data Transfer

The MVI69-HART module communicates directly over the CompactLogix or MicroLogix backplane. Data is paged between the module and the CompactLogix 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.

Data received by the master drivers is placed in the module's input image. This data is processed by the ladder logic in the CompactLogix processor. The input image for the module varies depending on the block size specified in the configuration file. 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 may be set to 61, 121, or 241 words depending on the block transfer size parameter set in the configuration file.

The following illustration shows the data transfer method used to move data between the CompactLogix processor, the MVI69-HART module and the HART 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 CompactLogix 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.

As blocks are transferred between the module and the processor, each block contains block identification codes that define the content or function of the block of data transferred.

Block Range	Descriptions
-1	Null block (Status Data Only)
0	Null block (Status Data Only)
1 to 20	Read or write data
9902	Command Enable Control Block
9903	Command Disable Control Block
9950	Command List Error data
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Blocks -1 and 0 transfer status data from the module to the processor and they contain no data when transferred from the processor to the module. Blocks 1 to 20 are utilized to transfer data stored or to be stored in the module's database. These data blocks send data from module to the processor (monitored data received from the devices on the HART network) and to send data from the processor to the module (control data to send to the end devices). Block identification codes 9900 to 9999 are used for special control blocks to control the module.

5.2.3 Normal Data Transfer

Normal data transfer includes the transferring of data received by, or to be transmitted to, the master drivers and the status data. These data are transferred through read (input image) and write (output image) blocks. 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 CompactLogix processor. The structure of the input image used to transfer this data is shown below:

Offset	Description	Length
0	Read Block ID	1
1	Write Block ID	1
2 to n	Read Data	n

n=60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Read Block ID is an index value used to determine the location of where the data will be placed in the CompactLogix processor controller tag array of module read data. The number of data words per transfer depends on the configured Block Transfer Size parameter in the configuration file (possible values are 60, 120, or 240).

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

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

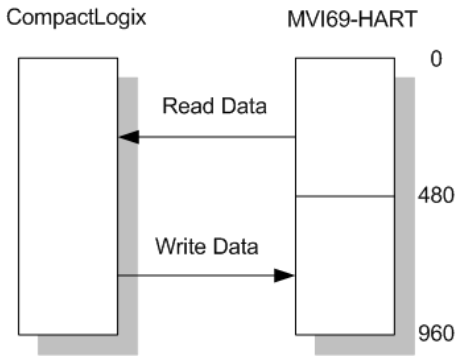
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the HART network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

If the backplane parameters are configured as follows:

```
Read Register Start:      0
Read Register Count:    480
Write Register Start:   480
Write Register Count:   480
```

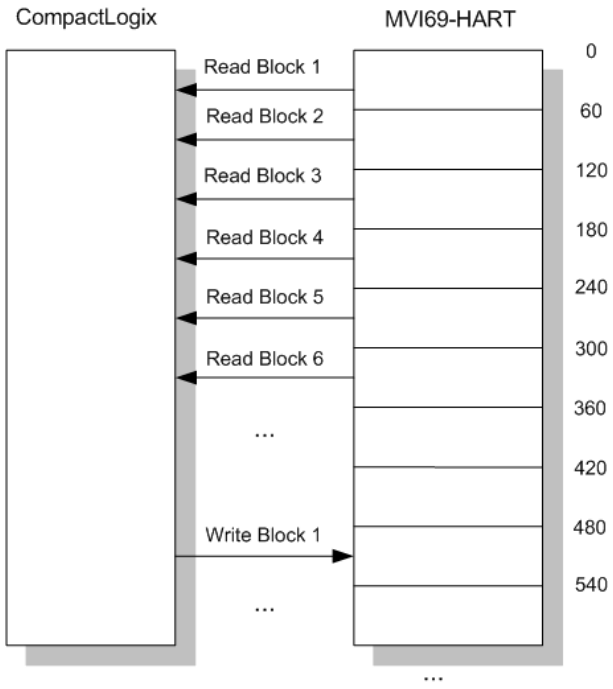
The backplane communication would be configured as follows:



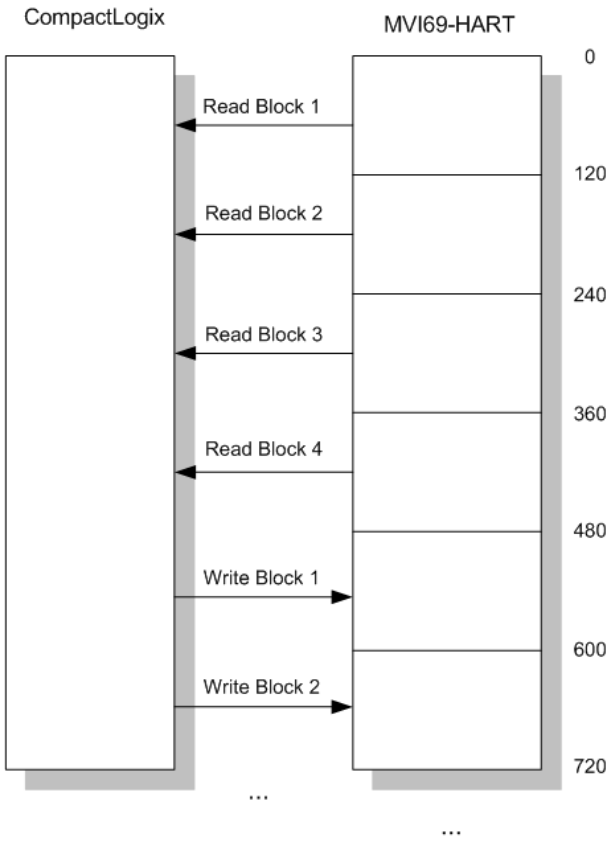
Database address 0 to 479 will be continuously transferred from the module to the processor. Database address 480 to 959 will continuously be transferred from the processor to the module.

The Block Transfer Size parameter basically configures how the Read Data and Write Data areas are broken down into data blocks (60, 120, or 240).

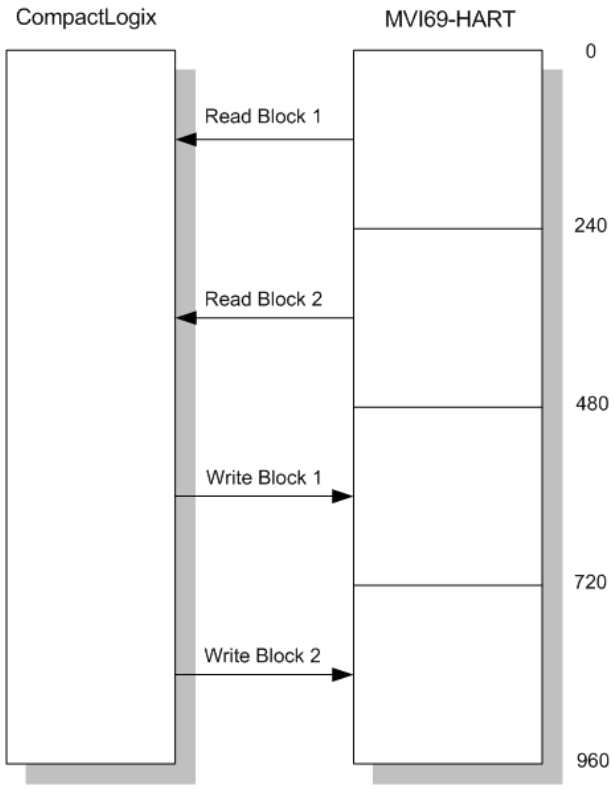
If Block Transfer Size = 60



If Block Transfer Size = 120



If Block Transfer Size = 240



Write Block

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

Offset	Description	Length
0	Write Block ID	1
1 to n	Write Data	n

n=60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed.

5.2.4 Command Control Blocks

Block identification codes greater than 9900 are utilized to perform special functions in the module. Each control block recognized and used by the module is defined in the following topics.

Block 9902: Command Control

The block 9902 identification code is used by the processor to enable a set of commands that have their enable code set to 4 or 5 (one shot). The value referenced by the address associated with the command, Enable DB Address, will be set to -1 to enable the command for a one-shot (enable code 5) or continuous (enable code 4) execution. After the command is executed, the module resets the register to 0 to disable the command for the one-shot mode. The format for this command block is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9902 identifying the enable command to the module.
1	Port Number	This is the HART port number (0 to 3) for the command. Each port has its own command list.
2	Command count	This field contains the number of commands to enable in the command list. Valid values for this field are 1 to 60.
3 to 62	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to enable. The value referenced by the Enable DB address associated with each command will be set to -1.
63	Spare	Not Used

There is no response to this block by the module.

Block 9903: Command Disable Control Block

The block 9903 identification code is used by the processor to disable a set of commands that have an enable code set to 4 or 5 (one shot). The value referenced by the address associated with the command, Enable DB Address, will be set to 0 to disable the command. The format for this command block is shown in the following table.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9903 identifying the enable command to the module.
1	Port Number	This is the HART port number (0 to 3) for the command. Each port has its own command list.
2	Command count	This field contains the number of commands to disable in the command list. Valid values for this field are 1 to 60.
3 to 62	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to disable. The value referenced by the Enable DB address associated with each command will be set to 0.
63	Spare	Not Used

There is no response to this block by the module.

Block 9950: Read Command Error List

Block 9950 identification code requests the Command List Error/Status Table for a single port from the module for the 99 user configurable commands. The following table describes the format of this block.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the module.
1	Port Number	This field contains the port number to report in the response block. Valid values are 0 to 3 for ports 0 to 3 on the module.
2	Number of Commands to report	This field contains the number of commands to report in the response message. The value has a range of 1 to 50. This number must not be greater that the "command count" value configured in the configuration file.
3	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands - 1.
4 to 63	Spare	Not Used

Response to a block 9950 request: The module will respond to a valid request with a block containing the requested error information. The format for the block is shown in the following example:

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the PLC.
1	Block ID	This is the next block requested by the module.
2	Port Number	This field contains the port number (0 to 3) corresponding to the data contained in the block.
3	Number of Commands reported	This field contains the number of commands contained in the block that must be processed by the PLC. This field will have a value of 1 to 50.
4	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 1 to MaxCommands-1.
5 to 54	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the request block. The number of valid command errors in the block is set in word 3 of the block. Refer to the command error list to interpret the error codes reported.
55 to 63	Spare	Not Used

Block 9998: Warm Boot

This block is sent from the CompactLogix or MicroLogix processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. The following table describes the format of the control block.

Offset	Description	Length
0	9998	1
1 to n	Spare	n

n=60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

Block 9999: Cold Boot

This block is sent from the CompactLogix 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 n	Spare	n

n=60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

5.2.5 HART Channels

The MVI69-HART module supports the HART protocol as a Master on up to two channels. Each channel is individually configurable.

The HART protocol uses the Bell 202 standard frequency shift-keying (FSK) signal to communicate at 1200 baud, superimposed at a low level on the 4 to 20 mA analog measurement signal. Having an average value of zero, and FSK signal causes no interference with the analog value. The HART devices are powered from this 4 to 20 mA analog loop.

Both HART protocol channels in the module generate the Bell 202 FSK signal to communicate in multi-drop mode with up to 15 HART devices and provide up to 250 mA supply for analog loop.

Auto-Poll Modes

Each HART channel can be set to operate in three different modes:

- Point-to-Point
- Multi-drop
- User Mode

Using the configuration file, choose the auto-poll mode through the Auto-Poll Code parameter (P, M, or N). In the first two modes, the module will automatically collect data from each HART instrument on the channel (auto-poll) and store the data in the module's database.

In User Mode, the module will only execute the commands in the user command list and will not automatically acquire data. Refer to the Auto-Polling section for more information.

Mode	Use Auto-Poll	Use Command List
N (None)	N	Y
M (Multi-drop)	Y	Y
P (Point-to-Point)	Y	Y

When configured for Point-to-Point communication, the channel will automatically poll data from the connected slave address 0.

When configured for Multi-drop, the channel will automatically poll data from the connected slaves in the network (address 1 to 15). Refer to the Auto-Polling section for more information about auto-poll mode.

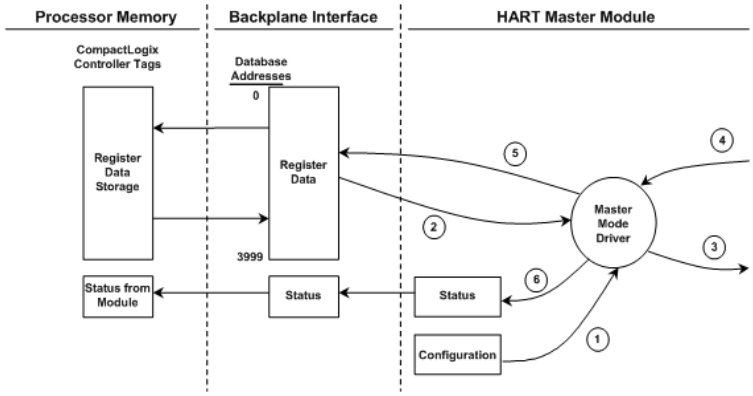
User-configured commands can be issued on each channel to the HART devices. Up to 99 commands can be defined for each port. Data read from the devices is placed in the virtual database of the module, which is passed between the module and the processor.

Any write requests or device-specific commands for the HART slave devices are sourced with data from the virtual database, or from a configured constant data block. Within the commands, it can be specified whether or not to use the HART device's short or long address. If the long address is selected, the device is polled first with short address to ask for the long one. Then, the device is polled with the long address. The module does this processing automatically.

In a HART network, it is possible to have two masters. The module fully supports the existence of a second master, but it can reduce the throughput on the HART network. This facility is enabled or disabled in the module's configuration. If the ability to have a second master on the network is disabled, then the maximum communication throughput is achieved.

5.2.6 Master Driver Mode

The master driver supported on each application port of the module emulates a HART master device. Configuration of each port is independent and should be connected to different HART networks. Up to 15 devices can be connected to each channel. Control of the slave devices by the master port is using a user-constructed command list auto-polling, if enabled. Data for these commands is sourced from the module's database. Data read from the slave devices is placed in the module's database. The following diagram shows the operation of the master driver:



1 The master driver is configured as specified by the HART69.CFG file
2 The master will construct control commands using the data in the database
3 The master will send these commands and read requests out on the network
4 Response messages generated by controlled devices on the network are received by the master driver
5 Read data received by the master is passed to the module's database and passed to the processor
6 Additionally, status data for the module is passed to the processor

HART Command List

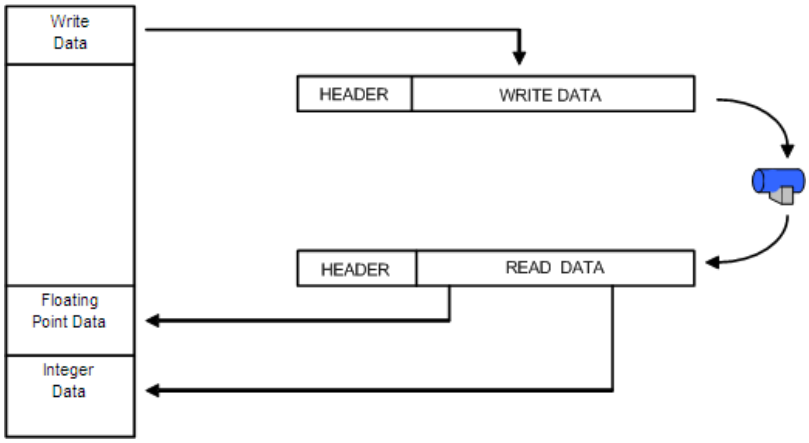
The HART Command List specifies the commands to be executed to the HART devices connected to a channel. A HART command can be seen as an outgoing message to the HART devices that provides Write Data for a specific command or a response message that carries process data (Read Data) back to the module. The MVI69 module supports three kinds of data blocks in the Universal and Common Practice commands. These data blocks are:

- Integers
- IEEE 754 Floating Point Numbers (32 bits)
- Packed ASCII character strings

The Packed ASCII character strings are unpacked and placed with the integers data block.

For all commands, it is possible to select where the Write Data comes from; it can be in the module's internal database or it can be configured as a fixed data block in the command.

For response messages from HART devices, it is possible to configure where the Floating Point Data and Integer Data will be placed in the module's internal database, but this is only possible for the Universal and Common Practice commands. In the case of Device Specific commands, all the Read Data is placed in the Integer data section. See HART Command Support for a listing of supported HART commands.



Burst Mode

If a slave on a network will be placed in burst mode, its data can be placed in the module's database. For the command to be burst by slave device, enter a command in the user command list with the appropriate HART command number. Set the type field to 0 to disable the command. The parameters in the command will be used to store the data received from the bursting slave. There can only be one slave bursting on the network at any one time. This mode can be used for faster update of data from a slave.

Auto-Polling

This feature is enabled by setting the AUTO-POLL CODE (in the configuration file) to a value of P (point-to-point). If the value N is entered for the parameter, the auto-poll feature is disabled. When the feature is disabled, the channel will only execute the commands enabled in the user command list. When the auto-polling mode is enabled, the module will automatically acquire data from the HART instruments attached to a channel without the use of user commands. If user commands are present and enabled when the feature is enabled, they will also be executed independent of auto-polling.

With the auto-poll feature enabled, the module automatically generates the following HART commands and stores the data in the module's database at the user-specified location:

CMD	Description
0	Read Unique Identifier
3	Read Current and Four Dynamic Variables
13	Read Tag, Descriptor and Date
14	Read PV Sensor Information
15	Read Output Information

If the unit is set for point-to-point mode, the module will automatically gather the information for the device with the polling address (short address) of zero and place the data into the database. Each device requires a 50-word database area with the format shown in the following table.

DB Byte Offset	Type	Description	Byte Cnt	Data Source	Use of Data
0	byte	Auto-polling command status bits	1	App	Status
1	byte	Last first status byte received from device	1	Resp	Status
2	byte	Last second status byte received from device	1	Resp	Status
3	byte	Manufacture ID Code	1	CMD 0	LongAddress
4	byte	Device Type Code	1	CMD 0	LongAddress
5	byte	Minimum number of preambles	1	CMD 0	Msgconstruction
6	byte	Universal Command Major Rev #	1	CMD 0	Msgchoice
7	byte	Device Revision Level	1	CMD 0	Info
8	byte	Software Revision Level	1	CMD 0	Info
9	byte	Hardware Revision Level/Physical Signaling Code	1	CMD 0	Info
10	byte	Device Flags	1	CMD 0	Info
11 to 13	byte	Device ID	3	CMD 0	Long Address
14	byte	Minimum number of preambles to be sent with the response message from the slave to the master.	1	CMD 0	
15	byte	Maximum number of device variables	1	CMD 0	Info
16 to 17	word	Configuration Change Counter	2	CMD 0	Info
18	byte	Extended Field Device Status	1	CMD 0	Info
19	byte	Primary variable units code	1	CMD 3	Cfg
20	byte	Secondary variable units code	1	CMD 3	Cfg
21	byte	Tertiary variable units code	1	CMD 3	Cfg
22	byte	Quaternary variable units code	1	CMD 3	Cfg
23 to 30	byte	Tag name	8	CMD 13	Info
31 to 46	byte	Descriptor	16	CMD 13	Info

DB Byte Offset	Type	Description	Byte Cnt	Data Source	Use of Data
47 to 49	byte	Tag/Descriptor data	3	CMD 13	Info
50 to 52	byte	Transducer serial number	3	CMD 14	Info
53	byte	Transducer limits and min span units code	1	CMD 14	Info
54	byte	PV alarm selection code	1	CMD 15	Info
55	byte	PV transfer function code	1	CMD 15	Info
56	byte	PV upper and lower range value units code	1	CMD 15	Info
57	byte	Write protection code	1	CMD 15	Status
58	byte	Private label distributor code	1	CMD 15	Info
59	byte	PV analogchannel flag	1	CMD 15	Info
		TOTAL BYTE COUNT	60		
		TOTAL WORD COUNT	30		

DB Byte Offset	Type	Description	Byte Cnt	Data Source	Use of Data
60 to 63	float	Primary variable value	4	CMD 3	Status
64 to 67	float	Secondary variable value	4	CMD 3	Status
68 to 71	float	Tertiary variable value	4	CMD 3	Status
72 to 75	float	Quaternary variable value	4	CMD 3	Status
76 to 79	float	Upper transducer limit	4	CMD 14	Cfg
80 to 83	float	Lower transducer limit	4	CMD 14	Cfg
84 to 87	float	Minimum span	4	CMD 14	Cfg
88 to 91	float	PV upper range value	4	CMD 15	Cfg
92 to 95	float	PV lower range value	4	CMD 15	Cfg
96 to 99	float	PV damping value (in seconds)	4	CMD 15	Cfg
		TOTAL FLOAT BYTE COUNT	40		
		TOTAL FLOAT WORD COUNT	20		

DB Regs/Device	50
Max DB Regs/channel for 15 Devices	750
Max DB Regs for HART Card	3000

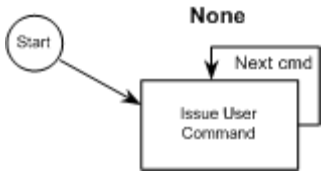
The following table defines the auto-polling command status bits:

Bit #	Description
0	Long Address Set (command 0 successful)
1	Command 13 successful (configuration)
2	Command 14 successful (configuration)
3	Command 15 successful (configuration)
4	Command 3 successful (data polling)
5	Reserved
6	Reserved
7	Reserved

The following topics describe the modes of module operation.

Auto-Poll Disabled Mode

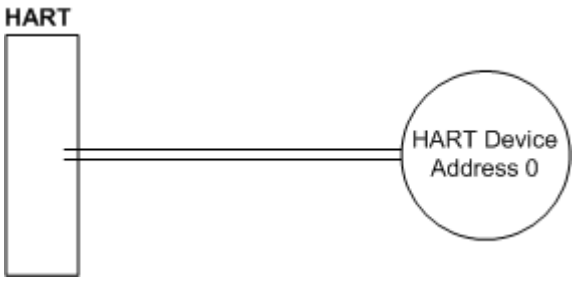
If the auto-polling feature is disabled (Auto-Poll Code = N), the module functions as shown in the following diagram:



Only the user commands are executed and all data is placed in and sourced from the module's internal database. The user is responsible for constructing all commands to control and monitor the instruments attached to the channel.

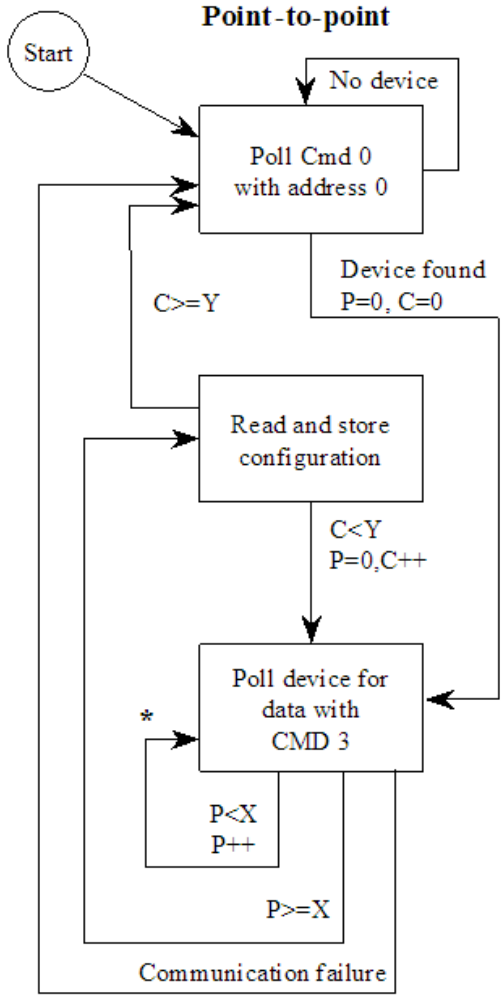
Point-to-Point Mode

Important: If the HART device address is 0 you must configure the channel for Point-to-Point mode.



In point-to-point mode, the module only polls for a single instrument with a polling address of zero. When the instrument is found by the channel, it continuously polls for the data using command 3. Occasionally, it will poll for the configuration information for the device. This is accomplished with HART commands 13, 14 and 15. Less frequently, the channel will perform a HART command 0 request to see if any of the data for the instrument has changed. If communications is lost with the device, the module will try to establish communications with the device using command 0. If user commands are present and enabled, they will be executed after each data poll.

When the point-to-point mode of auto-polling is enabled (Auto-Poll Code = P), the following diagram applies to the channel operation:

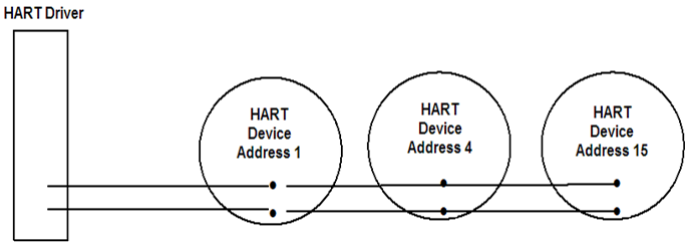


Where
P, C = loop iterators
X = Poll Counter (internal fixed value

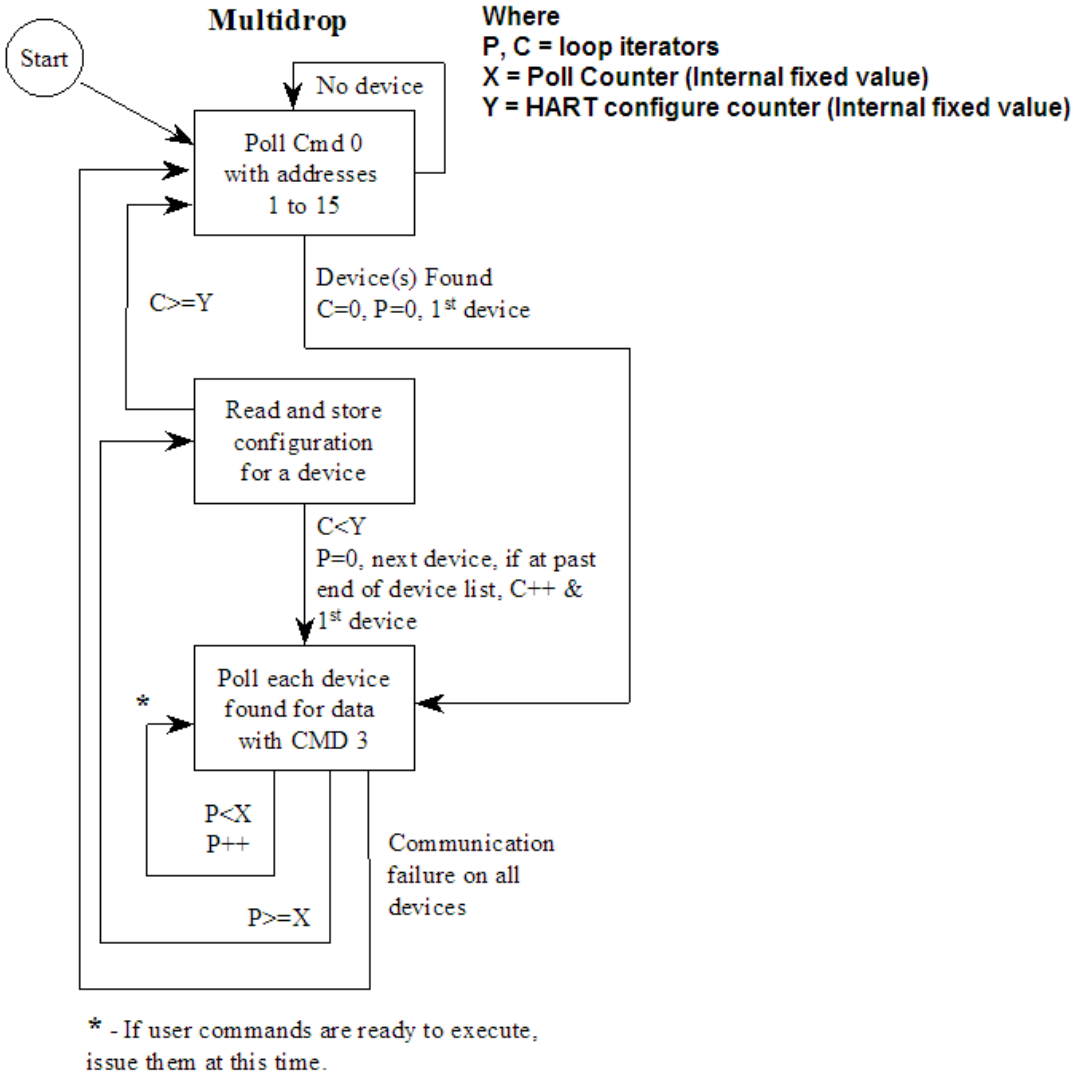
Multi-drop Mode

Important: If the HART device address is between 1 and 15, you must configure the channel for multi-drop mode.

If the unit is set for multi-drop mode, the module will poll each unit attached to the channel starting with polling address 1. The parameter MAX DEVICE COUNT in the configuration determines the maximum slave address number to be polled in multi-drop mode. For example, if the MAX DEVICE COUNT parameter is set to 3, the channel will poll for polling addresses 1, 2, and 3. It is important when assigning the device polling addresses for instruments on a channel to start with 1 and successfully increase the value by one until the last instrument is assigned an address. If you set the MAX DEVICE COUNT parameter to 2, and assign the two instruments addresses 1 and 15, the polling address 15 will never be used (only addresses 1 and 2). Therefore, the second instrument will never be polled and will be in communication failure. Up to 15 instruments can be assigned to a single channel (polling address 1 to 15).



If the module is configured for multi-drop auto-polling (Auto-Poll Code = M), the following diagram applies:



In multi-drop mode, the channel will poll for instruments 1 to the value set in the MAX DEVICE COUNT parameter. For each device found, it will continuously poll for data using command 3. After a certain number of iterations, the HART Driver will ready the configuration data for the next device (if there is one). Each time configuration information is acquired, it will be for a different slave device if more than 1 slave is recognized by the channel. Less frequently, the channel will poll for devices not found in the original search of the network using HART command 0. If communications is lost with all devices, the module will try to establish communications with the devices using command 0. If user commands are present and enabled, they will be executed after the data polling of the devices.

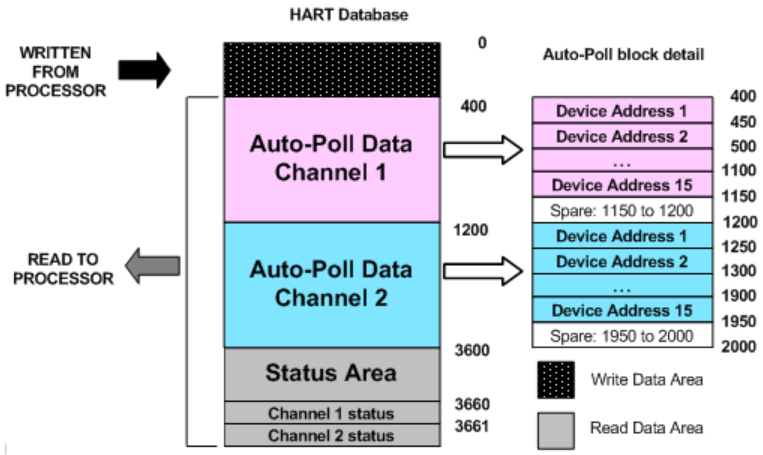
Multi-drop Mode Example:

The intent is to show when using Multi-drop mode how the Auto-Poll DB Address and Max Device Count parameters should be used.

If the configuration file sets the following parameters:

Parameter	Value	Config File Section
DB Address Status	3660	[HART PORT 0]
Auto-Poll DB Address	400	
Max Device Count	15	
DB Address Status	3661	[HART PORT 1]
Auto-Poll DB Address	1200	
Max Device Count	15	

The HART database has the following structure.

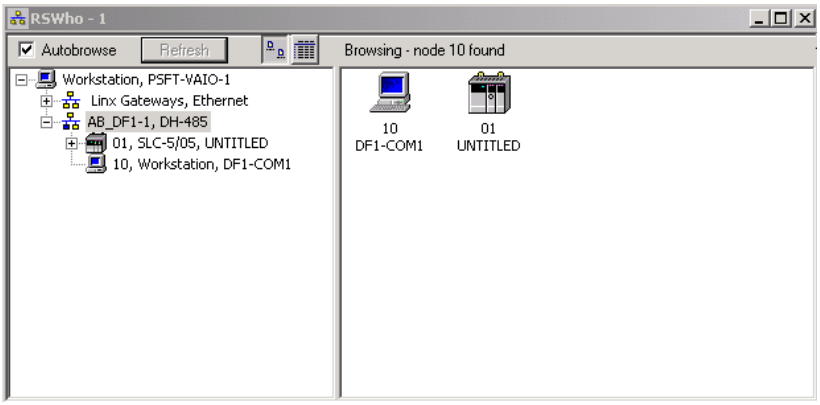


The configuration file for this example shows that when the MAX Device Address parameter is configured as 15, the module will reserve 750 words for each channel (15 devices x 50 words each) even though not all devices may be present on the network.

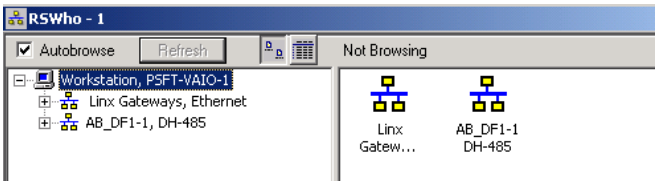
5.3 Disabling the RSLinx Driver for the Com Port on the PC

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

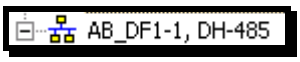
1. Open *RSLinx* and go to **COMMUNICATIONS** > **RSWHO**.
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network.



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

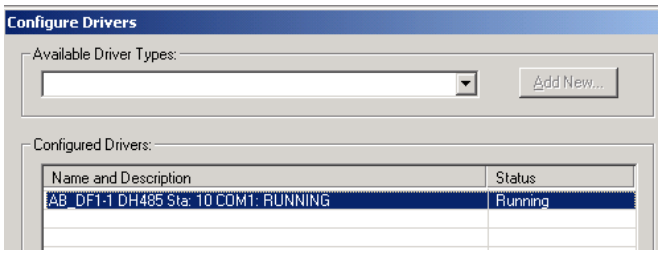


Branches are displayed or hidden by clicking on the  or the  icons.



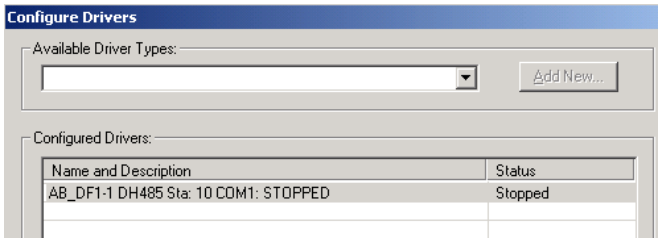
4. When you have verified that the driver is not being browsed, go to **COMMUNICATIONS** > **CONFIGURE DRIVERS**.

You may see something like this:



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

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

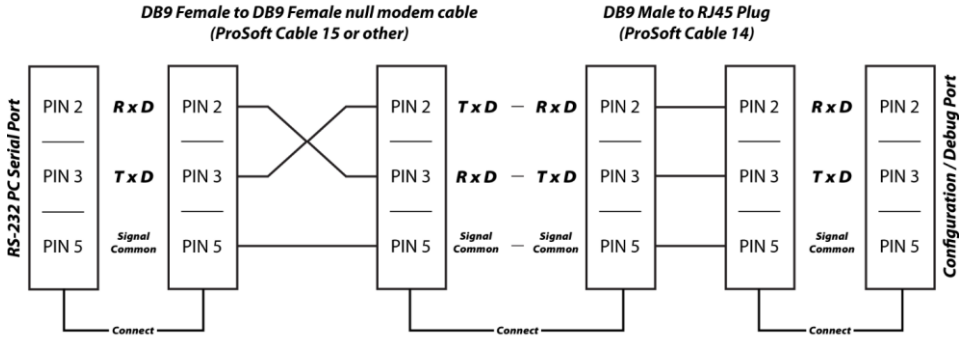


6 You may now use the com port to connect to the debug port of the module.

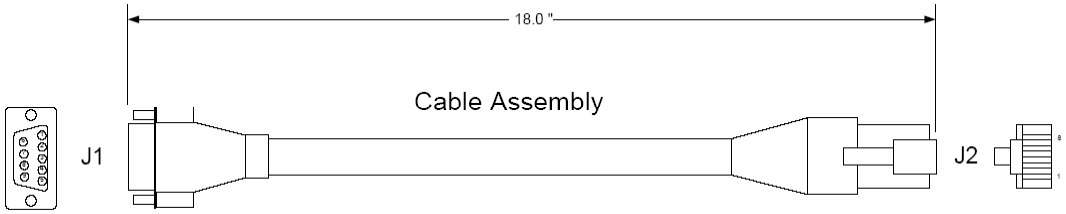
Note: You may need to shut down and restart your PC before it will allow you to stop the driver (usually only on *Windows NT* machines). If you have followed all of the above steps, and it will not stop the driver, then make sure you do not have *RSLogix* open. If *RSLogix* is not open, and you still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

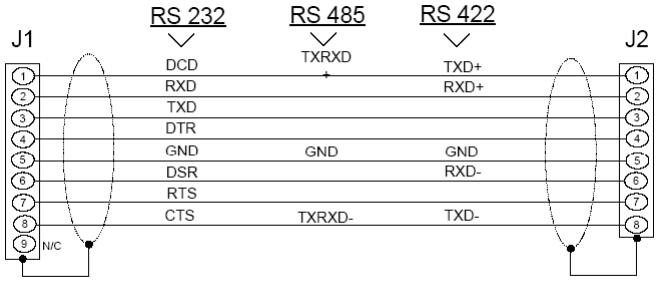
5.4 RS-232 Configuration/Debug Port

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



5.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.6 Status Data Definition

This section contains a listing of the data contained in the MVI69-HART status data object, configuration error word and module error codes.

ERROR/STATUS DATA BLOCK

This block is generated with a Read Block ID 0 or -1.

Offset	Content	Description
0	Reserved	Not used
1	Write Block ID	Block identification code for block requested by the module.
2	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
3 to 4	Product Code	These two registers contain the product code of "HRT5 "
5 to 6	Product Version	These two registers contain the product version for the currently running software.
7 to 8	Operating System	These two registers contain the month and year values for the program operating system.
9 to 10	Run Number	These two registers contain the run number value for the currently running software.
11	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
12	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
13	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
14	Reserved	Not used
15	Command Block Count	This field contains the total number of command blocks received from the processor for enable and disable requests.
16	Error Block Count	This field contains the total number of block errors recognized by the module.
17	Channel 1 State	Channel 1 state machine value (used for debugging) 0 Polling 1 Config 2 Data Poll 3 User Poll 4 Wait Unique ID

Offset	Content	Description
18	Comm State	Channel 1 communication state machine value (used for debugging) 0 Startup 1 Idle 2 Transmit Pending 3 Transmit 4 Receive 5 Post Transmit 6 Idle, Waiting
19	Device Status (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave is in error. If the bit is clear (0), the slave is not in error.
20	Device Poll List (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave is in the poll list. If the bit is clear (0), the slave is not in the poll list.
21	Device With Long Address (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave uses the long address. If the bit is clear (0), the slave does not use the long address.
22	Current Command	This field contains the index of the current command to execute.
23	Command Request Count	This field contains the total number of request messages issued on the port.
24	Command Response Count	This field contains the total number of response messages received from devices on the network.
25	Configuration Error Word	Configuration error word (see table)
26	Current Error Code	Current error code for port
27	Last Error Code	Last error code reported for port
28	Channel 2 State	Channel 2 state machine value (used for debugging) 0 Polling 1 Config 2 Data Poll 3 User Poll 4 Wait Unique ID
29	Comm State	Channel 2 communication state machine value (used for debugging) 0 Startup 1 Idle 2 Transmit Pending 3 Transmit 4 Receive 5 Post Transmit 6 Idle, Waiting

Offset	Content	Description
30	Device Status (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave is in error. If the bit is clear (0), the slave is not in error.
31	Device Poll List (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave is in the poll list. If the bit is clear (0), the slave is not in the poll list.
32	Device With Long Address (bit mapped)	Each bit in this word corresponds to a slave address on the network starting at bit 1 for slave address 1. Bit 0 is not used. If the bit is set, the slave uses the long address. If the bit is clear (0), the slave does not use the long address.
33	Current Command	This field contains the index of the current command to execute.
34	Command Request Count	This field contains the total number of request messages issued on the port.
35	Command Response Count	This field contains the total number of response messages received from devices on the network.
36	Configuration Error Word	Configuration error word (see table).
37	Current Error Code	Current error code for port.
38	Last Error Code	Last error code reported for port.
61 to 248	Reserved	Not used.
249	Read Block ID	Block identification code of 0 or -1 to indicate a status data block.

The following table defines the contents of the configuration error word. Each bit in the word corresponds to an error condition recognized when the module is configured. There is a separate word for each application port. This data is reported in the status data area previously defined.

Bit	Code	Description
0	0x0001	Enabled not set to Y or N
1	0x0002	Enable Handheld not set to Y or N
2	0x0004	Primary Master not set to Y or N
3	0x0008	Invalid Preambles (1 to 50)
4	0x0010	Invalid Short Address Retries (0 to 50)
5	0x0020	Invalid Long Address Retries (0 to 50)
6	0x0040	Invalid Retries After Error (0 to 50)
7	0x0080	Invalid Poll Time After Error (0 to 10000)
8	0x0100	Invalid DB Address Status
9	0x0200	Invalid Command Count
10	0x0400	Memory Error in allocating commands
11	0x0800	Memory Error in allocating command fixed data
12	0x1000	Memory Error in allocating TX/RX buffers

Bit	Code	Description
13	0x2000	HART Board not found
14	0x4000	Cannot initialize HART channel
15	0x8000	

Each command in the command list for each HART channel has a word value for a status/error code. This error data list can be read using the Configuration/Debug Port and can be placed in the module's internal database using the Error/Status Offset parameter for each port. Accessing the Debug capabilities of the module is accomplished by connecting a PC to the Debug port, then running a terminal emulation program. This status/error code is the first word of the Integer Data Block returned from every HART command executed. This word has information about the execution of the command by the HART device.

Refer to the following section for a listing of the Error Codes to interpret the status/error codes present in the integer data area. The following illustration shows the location of each error word in the data block:

Word Offset	Description
0	Command 0 Error Value
…	
…	
99	Command 99 Error Value

The module error codes are listed in this section. Error codes returned from the HART device are placed at the first word of the integer data block in the Virtual Database. The error codes are formatted in the word as follows:

If the most-significant bit of the word contains "1", then the most significant byte has a communication error code. The least-significant byte of the word will contain "0".

If the most-significant bit of the word contains "0", the most significant byte contains a command error code. In this case, the least-significant byte contains device malfunction information.

If this word value is "-1" (or 0xFFFF), it means that the command timed out.

Use the error codes returned for each command in the list to determine the success or failure of the command. If the command fails, use the error code to determine the cause of failure.

5.6.1 Protocol Error Codes

These are error codes that are part of the HART protocol. The standard HART error codes are shown in the following tables:

Error Code Word

First Byte								Second Byte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Bit 7 of First Byte = 1 **COMMUNICATION ERROR**	
First Byte	
Bit 6	Parity Error
Bit 5	Overrun Error
Bit 4	Framing Error
Bit 3	Checksum Error
Bit 2	Reserved
Bit 1	Rx Buffer Overflow
Bit 0	Undefined
Second Byte	
Bit 0 To Bit 7	All 0

Bit 7 of First Byte = 0 **Command Error**	
First Byte	
Bits 6 To 0 (not Bit-mapped):	
0	No Error
1	Undefined
2	Invalid Selection
3	Passed Parameter Too Large
4	Passed Parameter Too Small
5	Too Few Data Bytes Received
6	Transmitter-specific Command Error
7	In Write-protect Mode
8 to 15	Command Specific Errors (see Below)
16	Access Restricted
32	Device Is Busy
64	Command Not Implemented
Second Byte	
Bit 7	Device Malfunction
Bit 6	Configuration Changed
Bit 5	Cold Start

Bit 7 of First Byte = 0 **Command Error**	
Bit 4	Unused
Bit 3	Output Current Fixed
Bit 2	Analog Output Saturated
Bit 1	Variable (not Primary) Out Of Limits
Bit 0	Primary Variable Out of Limits

Command Specific Errors	
8	Update Failed Update In Progress Set to Nearest Possible Value
9	Applied Process Too High Lower Range Value Too High Not in Fixed Current Mode
10	Applied Process Too Low Lower Range Value Too Low Multi-drop Not Supported
11	In Multi-drop Mode Invalid Transmitter Variable Code Upper Range Value Too High
12	Invalid Unit Code Upper Range Value Too Low
13	Both Range Values Out of Limits
14	Pushed Upper Range Value Over Limit Span Too Small

5.7 HART Universal Commands

5.7.1 COMMAND 00 - Read Unique Identifier

Description

This command gets the long address of the HART device plus other manufacturer information like Manufacturer ID, Device Type Code, Software Revision, Hardware Revision, and so on.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Constant "254"	Manufacturer Identification Code
2	Manufacturer Device Type Code	Number of Preambles
3	Universal Command Revision	Transmitter Specific Command Revision
4	Software Revision	Hardware Revision
5	Device Function Flags	Device ID Number 1
6	Device ID Number 2	Device ID Number 3

5.7.2 COMMAND 01 - Read Primary Variable

Description

This command gets the device Primary Variable and the Primary Variable Units

Write Parameters

NONE

Floating Point Data Returned

Word	High Byte	Low Byte
0	Primary Variable Value	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	0

5.7.3 COMMAND 02 - Read Current And Percent Of Range

Description

This command gets the current of the loop that is forced by the HART device and the Percent of Range of the Current.

Write Parameters

NONE

Floating Point Data Returned

Word	High Byte	Low Byte
0	Current (mA)	
1		
2	Percent of Range	
3		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.7.4 COMMAND 03 - Read Dynamic Variables

Description

This command gets the current and four (predefined) dynamic Variables.

Write Parameters

NONE

Floating Point Data Returned

Word	High Byte	Low Byte
0	Current (mA)	
1		
2	Primary Variable	
3		
4	Second Variable	
5		
6	Third Variable	
7		
8	Fourth Variable	
9		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	Second Variable Units Code
2	Third Variable Units Code	Fourth Variable Units Code

5.7.5 COMMAND 06 - Write Polling Address

Description

This command sets the polling address of a HART device. Extreme care should be taken when you use this command because you can loose the communication with the device.

Write Parameters

Word	High Byte	Low Byte
0	Polling Address	Polling Address

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Polling Address	0

5.7.6 COMMAND 12 - Read Message

Description

This command reads an ASCII message contained in the HART Device and written by the Write Message command 17.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Message ASCII Character 0	Message ASCII Character 1
2	Message ASCII Character 2	Message ASCII Character 3
.	.	
.	.	
.	.	
14	Message ASCII Character 28	Message ASCII Character 29
15	Message ASCII Character 30	Message ASCII Character 31

5.7.7 COMMAND 13 - Read Tag, Descriptor and Date

Description

This command reads an ASCII Tag which identifies the device, an ASCII descriptor of the device and the last Date it has been configured.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	TAG ASCII Character 0	TAG ASCII Character 1
2	TAG ASCII Character 2	TAG ASCII Character 3
3	TAG ASCII Character 4	TAG ASCII Character 5
4	TAG ASCII Character 6	TAG ASCII Character 7
5	Descriptor ASCII Character 0	Message ASCII Character 1
.	.	
.	.	
.	.	
12	Descriptor ASCII Character 14	Message ASCII Character 15
13	Date	Date
14	Date	0

5.7.8 COMMAND 15 - Read Output Information

Description

This command gets information about the Primary Variable Output Information.

Write Parameters

NONE

Floating Point Data Returned

Word	High Byte	Low Byte
0	Upper Range Value	
1		
2	Lower Range Value	
3		
4	Damping Value (Sec)	
5		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Alarm Select Code	Transfer Function Code
2	Primary Variable Range Units Code	Write-Protect Code
2	Private-Label Distributor Code	0

5.7.9 COMMAND 16 - Read Final Assembly Number

Description

This command reads the final assembly number of the HART device.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Final Assembly Number 0	Final Assembly Number 1
2	Final Assembly Number 2	0

5.7.10 COMMAND 17 - Write Message

Description

This command writes an ASCII message contained in the HART Device and that can be read with command 12.

Write Parameters

Word	High Byte	Low Byte
0	Packed ASCII Message Byte 0	Packed ASCII Message Byte 1
1	Packed ASCII Message Byte 2	Packed ASCII Message Byte 3
.	.	
.	.	
.	.	
11	Packed ASCII Message Byte 22	Packed ASCII Message Byte 23

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.7.11 COMMAND 18 - Write Tag, Descriptor and Date

Description

This command writes an ASCII Tag which identifies the device, an ASCII descriptor of the device and the last Date it has been configured.

Write Parameters

Word	High Byte	Low Byte
0	Packed ASCII TAG Byte 0	Packed ASCII TAG Byte 1
1	Packed ASCII TAG Byte 2	Packed ASCII TAG Byte 3
2	Packed ASCII TAG Byte 4	Packed ASCII TAG Byte 5
3	Packed ASCII Descriptor Byte 0	Packed ASCII Descriptor Byte 1
.	.	
.	.	
.	.	
8	Packed ASCII Descriptor Byte 10	Packed ASCII Descriptor Byte 11
9	Date	Date
10	Date	0

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	TAG ASCII Character 0	TAG ASCII Character 1
2	TAG ASCII Character 2	TAG ASCII Character 3
3	TAG ASCII Character 4	TAG ASCII Character 5
4	TAG ASCII Character 6	TAG ASCII Character 7
5	Descriptor ASCII Character 0	Message ASCII Character 1
.	.	
.	.	
.	.	
12	Descriptor ASCII Character 14	Message ASCII Character 15
13	Date	Date
14	Date	0

5.7.12 COMMAND 19 - Write Final Assembly Number

Description

This command writes the final assembly number of the HART device.

Write Parameters

Word	High Byte	Low Byte
0	Final Assembly Number 0	Final Assembly Number 1
1	Final Assembly Number 2	Final Assembly Number 2

Floating Point Data Returned:

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Final Assembly Number 0	Final Assembly Number 1
2	Final Assembly Number 2	0

5.8 HART Common Practice Commands

5.8.1 COMMAND 33 - Read Transmitter Variables

Description

This command gets four user selected dynamic Variables.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code For Slot 0	Transmitter Variable Code For Slot 1
1	Transmitter Variable Code For Slot 2	Transmitter Variable Code For Slot 3

Floating Point Data Returned

Word	High Byte	Low Byte
0	Variable for Slot 0	
1		
2	Variable for Slot 0	
3		
4	Variable for Slot 2	
5		
6	Variable for Slot 3	
7		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code For Slot 0	Units Code for Slot 0 Variable
2	Transmitter Variable Code For Slot 1	Units Code for Slot 1 Variable
3	Transmitter Variable Code For Slot 2	Units Code for Slot 2 Variable
4	Transmitter Variable Code For Slot 3	Units Code for Slot 3 Variable

5.8.2 COMMAND 34 - Write Damping Value

Description

This command writes the damping value of a HART device.

Write Parameters

Word	High Byte	Low Byte
0	Floating Point Damping Value (Sec)	
1		

Floating Point Data Returned

Word	High Byte	Low Byte
0	Floating Point Damping Value (Sec)	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.3 COMMAND 35 - Write Range Values

Description

This command writes the Upper and Lower range of the Primary Variable

Write Parameters

Word	High Byte	Low Byte
0	Range Units Code	Floating Point Upper Range Value (Byte 0)
1	Floating Point Upper Range Value (Byte 1)	Floating Point Upper Range Value (Byte 2)
2	Floating Point Upper Range Value (Byte 3)	Floating Point Lower Range Value (Byte 0)
3	Floating Point Lower Range Value (Byte 1)	Floating Point Lower Range Value (Byte 2)
4	Floating Point Lower Range Value (Byte 3)	Floating Point Lower Range Value (Byte 3)

Floating Point Data Returned

Word	High Byte	Low Byte
0	Upper Range Value	
1		
2	Lower Range Value	
3		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Range Units Code	0

5.8.4 COMMAND 36 - Set Upper Range Value

Description

This command is similar in effect to push the SPAN button of the HART device but doing it remotely through the HART network.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.5 COMMAND 37 - Set Lower Range Value

Description

This command is similar in effect to push the ZERO button of the HART device but doing it remotely through the HART network.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.6 COMMAND 38 - Reset Configuration Changed Flag

Description

This command resets the status bit that indicates that configuration has been changed.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.7 COMMAND 39 - EEPROM Control

Description

This command operates over the EEPROM changing its settings.

Write Parameters

Word	High Byte	Low Byte
0	EEPROM Control Code	EEPROM Control Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	EEPROM Control Code	0

5.8.8 COMMAND 40 - Enter Exit Fixed Current Mode

Description

This command writes the damping value of a HART device.

Write Parameters

Word	High Byte	Low Byte
0	Floating Point Current (mA)	
1		

Floating Point Data Returned

Word	High Byte	Low Byte
0	Floating Point Current (mA)	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.9 COMMAND 41 - Perform Transmitter Self Test

Description

This command starts the HART device Self Test to find if there is any problem with the device hardware.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.10 COMMAND 42 - Perform Master Reset

Description

This command performs a master reset in the HART device.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.11 COMMAND 43 - Set PV Zero

Description

This command forces the Primary Value to Zero.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.12 COMMAND 44 - Write PV Units

Description

This command changes the setting of the Primary Variable units.

Write Parameters

Word	High Byte	Low Byte
0	Primary Variable Units Code	Primary Variable Units Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	0

5.8.13 COMMAND 45 - Trim DAC Zero

Description

This command calibrates the device Digital to analog converter zero. This is done by writing the same current value that is measured in the loop.

Write Parameters

Word	High Byte	Low Byte
0	Floating Point measured current (mA)	
1		

Floating Point Data Returned

Word	High Byte	Low Byte
0	Floating Point Current (mA)	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.14 COMMAND 46 - Trim DAC Gain

Description

This command calibrates the device Digital to analog converter gain. This is done by writing the same current value that is measured in the loop.

Write Parameters

Word	High Byte	Low Byte
0	Floating Point Measured Current (mA)	
1		

Floating Point Data Returned

Word	High Byte	Low Byte
0	Floating Point Current (mA)	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	

5.8.15 COMMAND 47 - Write Transfer Function

Description

This command changes the setting of the Transfer Function of the HART device.

Write Parameters

Word	High Byte	Low Byte
0	Transfer Function Code	Transfer Function Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transfer Function Code	0

5.8.16 COMMAND 48 - Read Additional Transmitter Status

Description

This command gets extended information about the status of the Transmitter. This information is specific for each transmitter.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Additional Status (Byte 0)	Additional Status (Byte 1)
2	Additional Status (Byte 2)	Additional Status (Byte 3)
.	.	
.	.	
.	.	

Word	High Byte	Low Byte
13	Additional Status (Byte 24)	0

5.8.17 COMMAND 49 - Write PV Sensor Serial Number

Description

This command changes the sensor serial number of the Primary Variable.

Write Parameters

Word	High Byte	Low Byte
0	Sensor Serial Number (Byte 0)	Sensor Serial Number (Byte 1)
1	Sensor Serial Number (Byte 2)	Sensor Serial Number (Byte 2)

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Sensor Serial Number (Byte 0)	Sensor Serial Number (Byte 1)
2	Sensor Serial Number (Byte 2)	0

5.8.18 COMMAND 50 - Read Dynamic Variable Assignments

Description

This command gets the actual assignment of the Dynamic Variables returned with command 3.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code For Primary Variable	Transmitter Variable Code For Second Variable
2	Transmitter Variable Code For Third Variable	Transmitter Variable Code For Fourth Variable

5.8.19 COMMAND 51 - Write Dynamic Variable Assignments

Description

This command sets the assignment of the Dynamic Variables returned with command 3.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code For Primary Variable	Transmitter Variable Code For Second Variable
1	Transmitter Variable Code For Third Variable	Transmitter Variable Code For Fourth Variable

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code For Primary Variable	Transmitter Variable Code For Second Variable
2	Transmitter Variable Code For Third Variable	Transmitter Variable Code For Fourth Variable

5.8.20 COMMAND 52 - Set Transmitter Variable Zero

Description

This command forces a selected transmitter variable to zero.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code	Transmitter Variable Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	0

5.8.21 COMMAND 53 - Write Transmitter Variable Units

Description

This command changes a selected transmitter variable unit.

Write Parameters:

Word	High Byte	Low Byte
0	Transmitter Variable Code	Transmitter Variable Units Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	Transmitter Variable Units Code

5.8.22 COMMAND 54 - Read Transmitter Variable Information

Description

This command gets information about any selected transmitter variable sensor.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code	Transmitter Variable Code

Floating Point Data Returned

Word	High Byte	Low Byte
0	Transmitter Variable Upper Limit	
1		
2	Transmitter Variable Lower Limit	
3		
4	Transmitter Variable Damping Value (Sec)	
5		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code	Sensor Serial Number 0
1	Sensor Serial Number 1	Sensor Serial Number 2
2	Unit Code for Limits	0

5.8.23 COMMAND 55 - Write Transmitter Variable Damping Value

Description

This command writes the Damping Value of a user selected transmitter variable.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code	Transmitter Variable Damping Value (Byte 0)
1	Transmitter Variable Damping Value (Byte 1)	Transmitter Variable Damping Value (Byte 2)
2	Transmitter Variable Damping Value (Byte 3)	Transmitter Variable Damping Value (Byte 3)

Floating Point Data Returned

Word	High Byte	Low Byte
0	Transmitter Variable Damping Value	
1		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code	0

5.8.24 COMMAND 56 - Write Transmitter Variable Sensor Serial Number

Description

This command writes the Serial Number of a user selected transmitter variable.

Write Parameters

Word	High Byte	Low Byte
0	Transmitter Variable Code	Transmitter Variable Sensor Serial Number (Byte 0)
1	Transmitter Variable Sensor Serial Number (Byte 1)	Transmitter Variable Sensor Serial Number (Byte 2)

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Transmitter Variable Code	Transmitter Variable Sensor Serial Number (Byte 0)
2	Transmitter Variable Sensor Serial Number (Byte 1)	Transmitter Variable Sensor Serial Number (Byte 2)

5.8.25 COMMAND 57 - Read Unit Tag Descriptor Date

Description

This command reads an ASCII Tag which identifies the device, an ASCII descriptor of the device and the last Date it has been configured.

Write Parameters

NONE

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	TAG ASCII Character 0	TAG ASCII Character 1
2	TAG ASCII Character 2	TAG ASCII Character 3
3	TAG ASCII Character 4	TAG ASCII Character 5
4	TAG ASCII Character 6	TAG ASCII Character 7
5	Descriptor ASCII Character 0	Message ASCII Character 1
.	.	
.	.	
.	.	
12	Descriptor ASCII Character 14	Message ASCII Character 15
13	Date	Date
14	Date	0

5.8.26 COMMAND 58 - Write Unit Tag Descriptor Date

Description

This command writes an ASCII Tag which identifies the device, an ASCII descriptor of the device and the last Date it has been configured.

Write Parameters:

Word	High Byte	Low Byte
0	Packed ASCII TAG Byte 0	Packed ASCII TAG Byte 1
1	Packed ASCII TAG Byte 2	Packed ASCII TAG Byte 3
2	Packed ASCII TAG Byte 4	Packed ASCII TAG Byte 5
3	Packed ASCII Descriptor Byte 0	Packed ASCII Descriptor Byte 1
.	.	
.	.	
.	.	
8	Packed ASCII Descriptor Byte 10	Packed ASCII Descriptor Byte 11
9	Date	Date
10	Date	0

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	TAG ASCII Character 0	TAG ASCII Character 1
2	TAG ASCII Character 2	TAG ASCII Character 3
3	TAG ASCII Character 4	TAG ASCII Character 5
4	TAG ASCII Character 6	TAG ASCII Character 7
5	Descriptor ASCII Character 0	Message ASCII Character 1
.	.	
.	.	
.	.	
12	Descriptor ASCII Character 14	Message ASCII Character 15
13	Date	Date
14	Date	0

5.8.27 COMMAND 59 - Write Number Of Response Preambles

Description

This command sets the number of preambles that the HART slave will use in every command response.

Write Parameters

Word	High Byte	Low Byte
0	Number of Response Preambles	Number of Response Preambles

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Number of Response Preambles	0

5.8.28 COMMAND 108 - Write Burst Mode Command Number

Description

This command sets the command number that the HART device will use in Burst Mode.

Write Parameters

Word	High Byte	Low Byte
0	Burst Mode Command Number	Burst Mode Command Number

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Burst Mode Command Number	0

5.8.29 COMMAND 109 - Burst Mode Control

Description

This command sets the HART device Burst Mode.

Write Parameters

Word	High Byte	Low Byte
0	Burst Mode Control Code	Burst Mode Control Code

Floating Point Data Returned

NONE

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Burst Mode Control Code	0

1 = ON

0 = Off

5.8.30 COMMAND 110 - Read All Dynamic Variables

Description

This command gets all dynamic Variables.

Write Parameters

NONE

Floating Point Data Returned

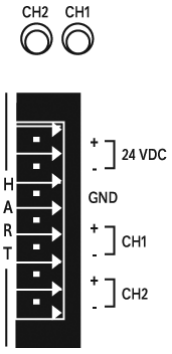
Word	High Byte	Low Byte
0	Primary Variable	
1		
2	Second Variable	
3		
4	Third Variable	
5		
6	Fourth Variable	
7		

Integer Data Returned

Word	High Byte	Low Byte
0	STATUS WORD	
1	Primary Variable Units Code	Second Variable Units Code
2	Third Variable Units Code	Fourth Variable Units Code

5.9 Application Port Connection

The MVI69-HART module has a single terminal connector to attach the module to the HART networks. The following diagrams display the configuration of the terminal connector:



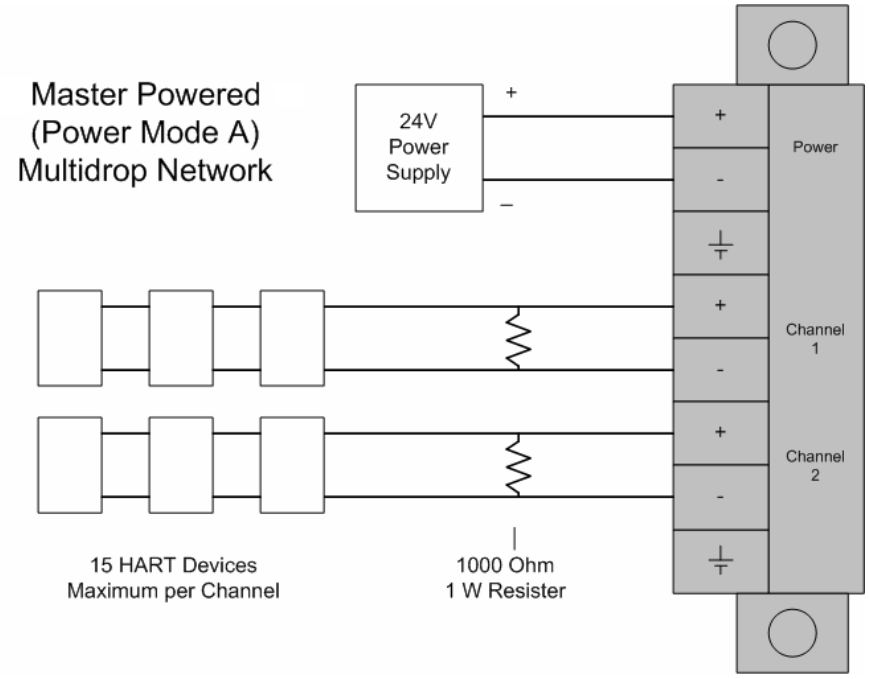
5.10 Example Field Terminations

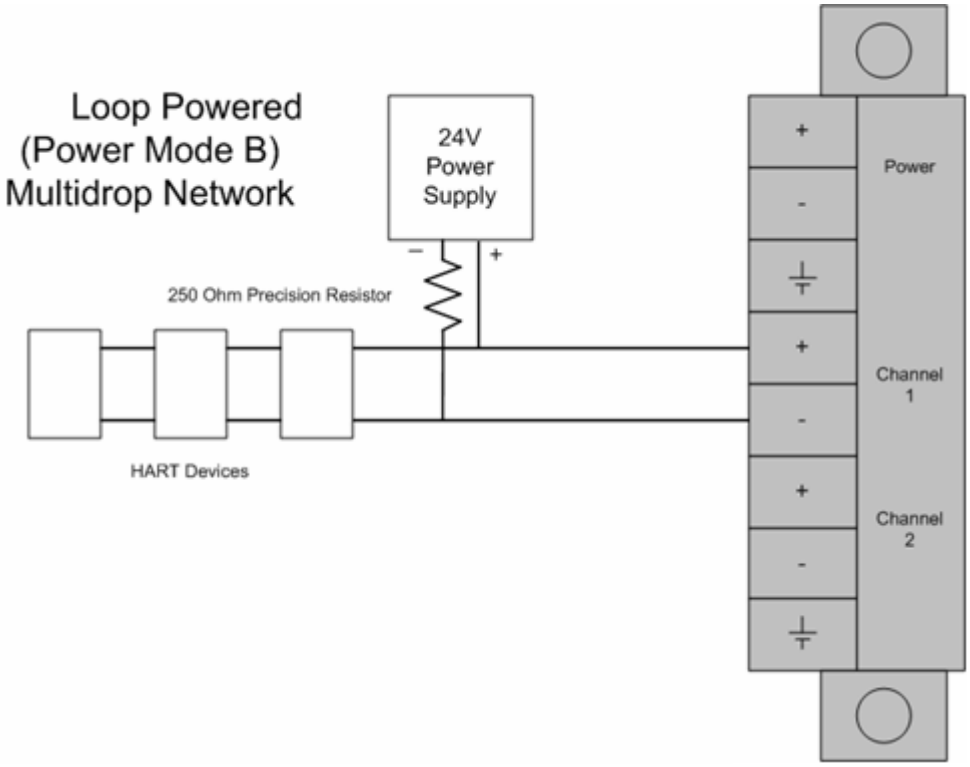
		Up to 15 instruments per channel	One instrument per channel	
		Multi-drop Mode (address 1 to 15)	**Point to Point Mode (address 0 to 15)**	**Point to Point Mode with Analog (address 0 only)**
Power Mode A 24VDC connected to HART power terminals 1 and 2 on terminal block	Master Powered instruments on all 4 channels	OK Install 250Ω 1KΩ	OK Install 1K resistor	OK
Power Mode B HART power terminals 1 and 2 on terminal block left open	Loop Powered instruments on 1 or more channels	OK	OK	OK - differential or single ended analog signal
	Self Powered instruments on 1 or more channels	Not possible	OK	OK -differential or single ended analog signal

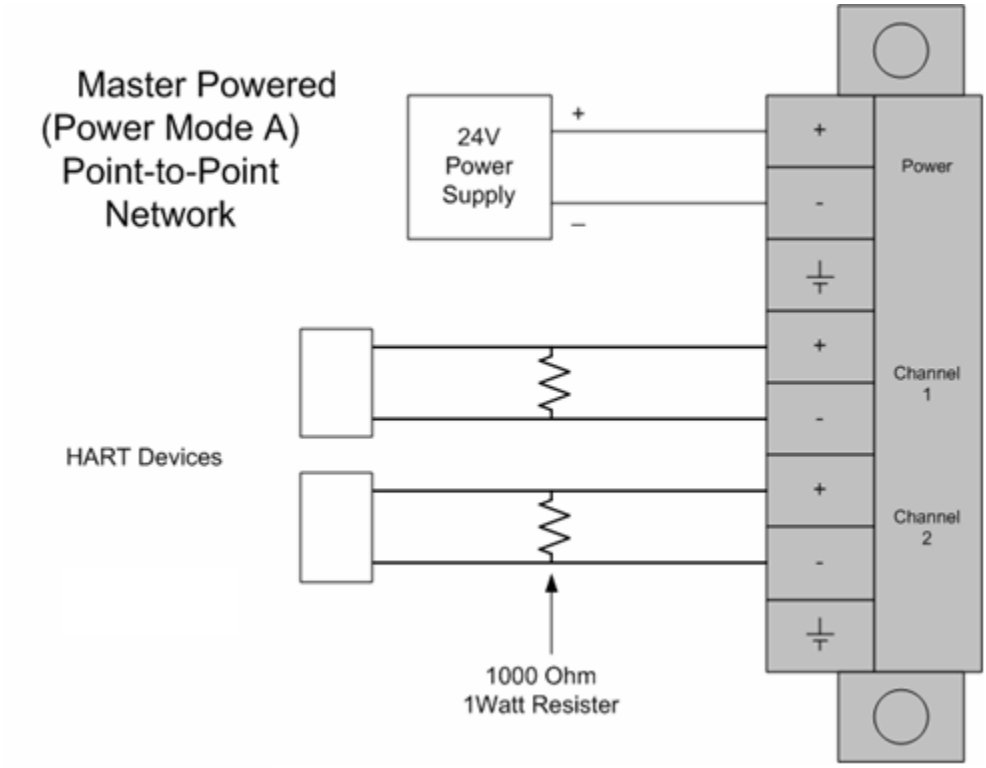
5.10.1 Notes

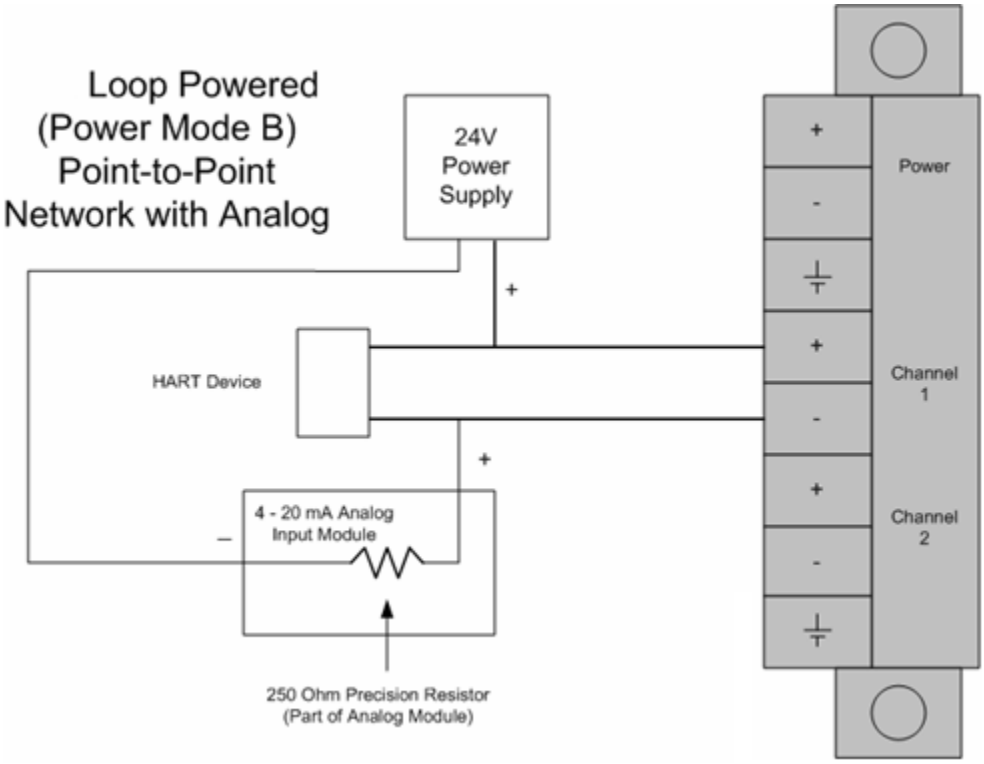
- Both channels on the unit must be either Master Powered (Power Mode A) or must be Loop or Self powered (Power Mode B). You cannot mix Power Mode A and Power Mode B on the same unit. You can mix Loop powered and Self powered instruments on the same unit. However, you cannot mix Loop powered and Self powered instruments on the same channel.
- For Master Powered, Multi-drop Mode and Point to Point Mode, a 1K, 1W resistor must be connected across the + and - input terminals of each HART channel. Resistors are supplied with each product.
- When configuring loop powered instruments in a Power Mode B type multi-drop network, use only one power supply per HART channel if you wish to maintain channel to channel isolation. If channel to channel isolation is not required, the same power supply may be used for both channels. However, a separate series resistor is required for each channel. An appropriately sized resistor must be connected in series with the power supply for each channel. The value and power rating of the resistor is dependant upon the number of instruments in the loop and the length and gauge of the pair of wires connecting the instruments to the unit. 250 Ohm, 3 Watt is a good, first choice value.
- If field instrument wiring is shielded, terminate shields at the ground terminals of the HART connector. Do not terminate the shields at the instrument.

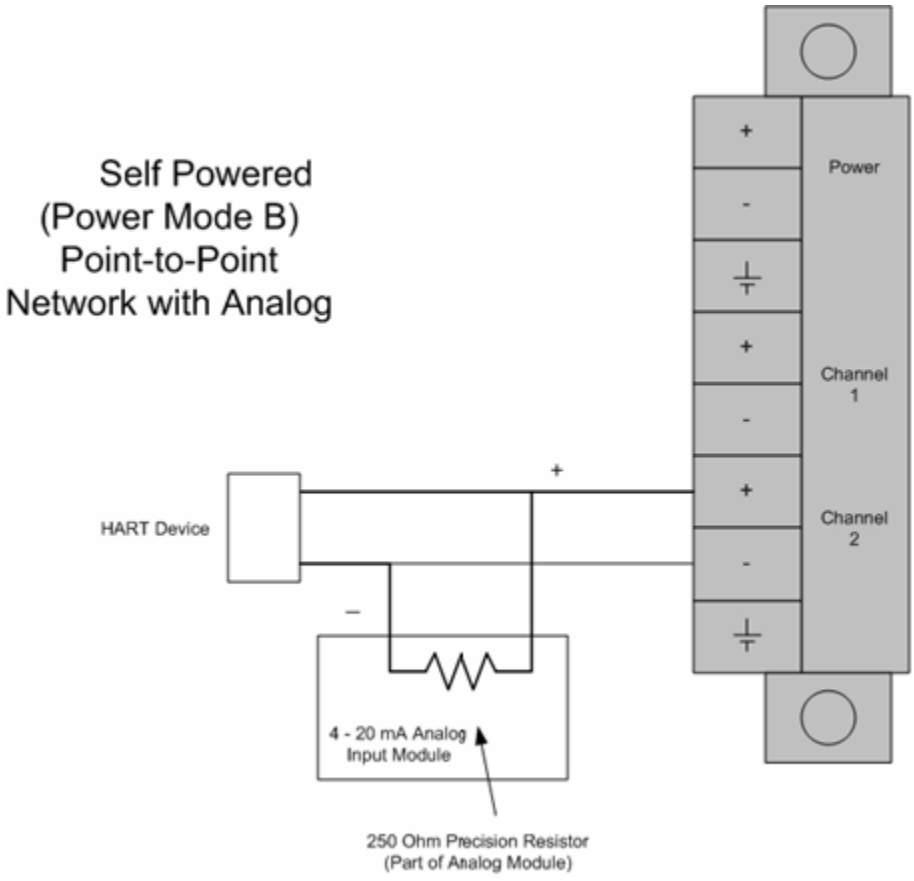
- For multi-drop applications, the 1KΩ resistor should be normally used when one transmitter is connected to the port. If you increase the number of devices connected to a single port, you may need to reduce the size of the resistor.

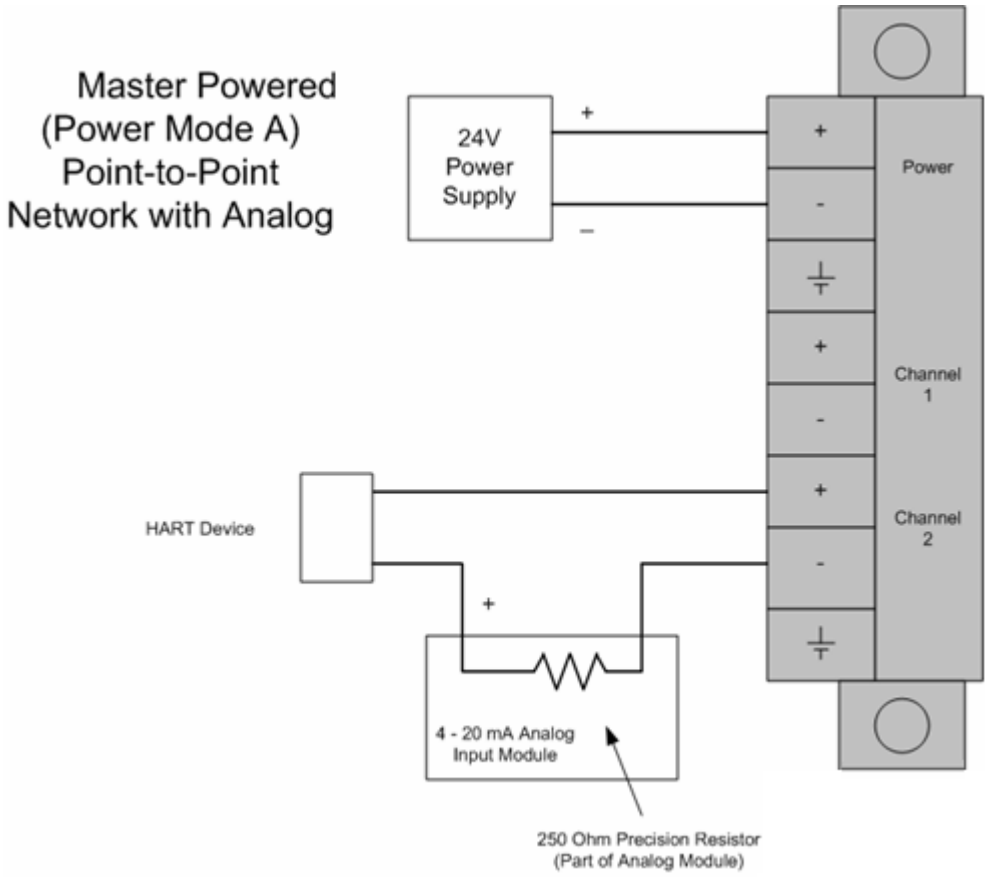





Master Powered
(Power Mode A)
Point-to-Point
Network
24V
Power
Supply
+
–
+
-
Power
+
-
Channel
1
+
-
Channel
2
HART Devices
1000 Ohm
1Watt Resister


Loop Powered
(Power Mode B)
Point-to-Point
Network with Analog
24V
Power
Supply
+
Power
-
+
Channel
1
-
+
Channel
2
-
HART Device
+
4 - 20 mA Analog
Input Module
–
250 Ohm Precision Resistor
(Part of Analog Module)


Self Powered
(Power Mode B)
Point-to-Point
Network with Analog
+
-
Power
+
-
Channel
1
+
-
Channel
2
HART Device
+
–
4 - 20 mA Analog
Input Module
250 Ohm Precision Resistor
(Part of Analog Module)


Master Powered
(Power Mode A)
Point-to-Point
Network with Analog
24V
Power
Supply
+
–
+
-
Power
Channel
1
Channel
2
HART Device
+
4 - 20 mA Analog
Input Module
250 Ohm Precision Resistor
(Part of Analog Module)

6 Support, Service & Warranty

In This Chapter

6.1 Contacting Technical Support

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

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

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

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

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

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

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

6.2 Warranty Information

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

Documentation is subject to change without notice.

Index