



Where Automation Connects.



inRax[®]

MVI69-DH485

CompactLogix or MicroLogix Platform

DH485 Communication Module

11/3/2008

USER MANUAL

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation CompactLogix or MicroLogix hardware, the MVI69-DH485 Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

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

Battery Life Advisory

All modules in the MVI series use a rechargeable Lithium Vanadium Pentoxide battery to backup the 512K SRAM memory, 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 it becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and configuration data, the real-time clock, and the 512K SRAM memory for approximately 21 days.

Before you remove a module from its power source, ensure that the battery within the module is fully charged. A fully charged battery will hold the BIOS settings (after being removed from its power source) for a limited number of days. When the battery is fully discharged, the module will revert to the default BIOS settings.

Note: The battery is not user replaceable.

Your Feedback Please

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

ProSoft Technology

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

Copyright © ProSoft Technology, Inc. 2000 - 2008. All Rights Reserved.

MVI69-DH485 User Manual
11/3/2008

ProSoft Technology®, ProLinX®, inRAX®, ProTalk® and RadioLinX® are Registered Trademarks of ProSoft Technology, Inc.

ProSoft® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD and are available at no charge from our web site: <http://www.prosoft-technology.com>
Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33.5.34.36.87.20

Latin America: +1.281.298.9109

North America: +1.661.716.5100

Contents

Please Read This Notice	2
Battery Life Advisory	2
Your Feedback Please.....	2
ProSoft® Product Documentation.....	3
Guide to the MVI69-DH485 User Manual	7
1 Start Here	9
1.1 System Requirements	9
1.2 Package Contents	10
1.3 Install ProSoft Configuration Builder Software	11
1.4 Setting Jumpers	12
1.5 Install the Module in the Rack	13
1.6 Connect your PC to the Processor.....	16
1.7 Download the Sample Program to the Processor	17
1.8 Connect your PC to the Module	20
2 Configuring the MVI69-DH485 Module	21
2.1 ProSoft Configuration Builder.....	21
2.2 [Module].....	26
2.3 [DH485 Port x].....	28
2.4 [DH485 Port x Commands]	30
2.5 [DH485 Port x Maps]	33
2.6 File Override Mapping	34
2.7 Timer, Counter, and Control Data Types	35
2.8 Download the Project to the Module	38
3 Ladder Logic	39
3.1 Module Data Object (DH485ModuleDef)	40
3.2 Adding the Module to an Existing CompactLogix Project	45
3.3 Adding the Module to an Existing MicroLogix Project.....	49
4 Diagnostics and Troubleshooting	51
4.1 Reading Status Data from the Module	51
4.2 LED Status Indicators.....	61
4.3 Clearing a Fault Condition.....	62
4.4 Troubleshooting.....	62
5 Reference	63
5.1 Product Specifications.....	63

5.2	Functional Overview	65
5.3	DH485 Communication Ports	81
5.4	Status Data Definition	86
6	Support, Service & Warranty	91
6.1	How to Contact Us: Technical Support.....	91
6.2	Return Material Authorization (RMA) Policies and Conditions	92
6.3	LIMITED WARRANTY	94
Index		99

Guide to the MVI69-DH485 User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 61) Diagnostics and Troubleshooting (page 51)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 63) Functional Overview (page 65) Product Specifications (page 63)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 91)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

❖ System Requirements	9
❖ Package Contents	10
❖ Install ProSoft Configuration Builder Software.....	11
❖ Setting Jumpers	12
❖ Install the Module in the Rack	13
❖ Connect your PC to the Processor	16
❖ Download the Sample Program to the Processor.....	17
❖ Connect your PC to the Module	20

Installing the MVI69-DH485 module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-DH485 Module and the application in which they will be used.



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

1.1 System Requirements

The MVI69-DH485 module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix processor, with compatible power supply and one free slot in the rack, for the MVI69-DH485 module. The module requires 800mA of available power.

Important: The MVI69-DH485 module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).

Important: For 1769-L23x processors, please make note of the following limitations.

- 1769-L23-QBFC1B = 800mA at 5Vdc (1 MVI69-DH485 will use all 800mA of available power. No other modules can be used with an MVI69 module connected to this processor).
- 1769-L23E-QB1B = 1000mA at 5Vdc (1 MVI69-DH485 will use 800mA of available power. One other module can be used on this rack provided it consumes less than 200mA at 5Vdc).
- 1769-L23E-QBFC1B = 450mA at 5Vdc (no MVI69 module can be used with this processor)
- 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)
- CD-ROM drive
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

1.2 Package Contents

The following components are included with your MVI69-DH485 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-DH485 Module	MVI69-DH485	DH485 Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI69-DH485 module.

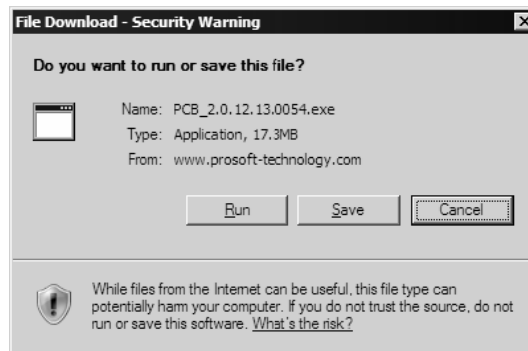
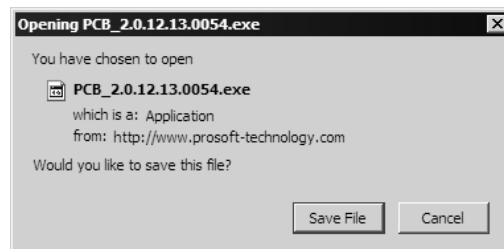
If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the MVI69-DH485 module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

To install ProSoft Configuration Builder from the ProSoft Web Site

- 1 Open your web browser and navigate to <http://www.prosoft-technology.com/pcb>
- 2 Click the **Download Here** link to download the latest version of ProSoft Configuration Builder.
- 3 Choose "Save" or "Save File" when prompted. The following illustrations show the file download prompt for two of the most common web browsers.



- 4 Make a note of the location where you saved the file, for example "Desktop", or "My Documents", so you can start the installation program.
- 5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install ProSoft Configuration Builder from the ProSoft Solutions CD-ROM, included in the package with your MVI69-DH485 module.

To install ProSoft Configuration Builder from the CD-ROM

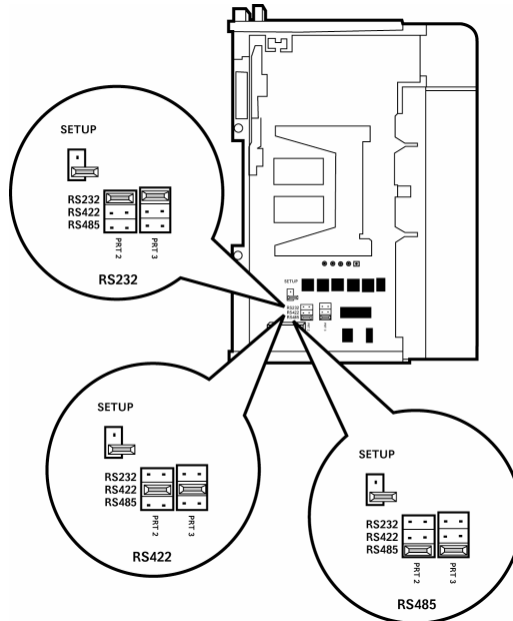
- 1 Insert the ProSoft Solutions CD-ROM into the CD drive of your PC. Wait for the startup screen to appear.
- 2 On the startup screen, click *Product Documentation*. This action opens an explorer window.

- 3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.
- 4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

Note: Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the Utilities folder on the CD-ROM to a convenient location on your hard drive.

1.4 Setting Jumpers

Changing from RS-232 to RS-485 requires that you change the jumper on the module to the appropriate settings. To do so, use a pair of needle-nose pliers to move the jumper shunt to the appropriate pins as shown in the following diagram:



Note: The RS-422 jumper setting shown in the diagram is not used with this module configuration.

The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

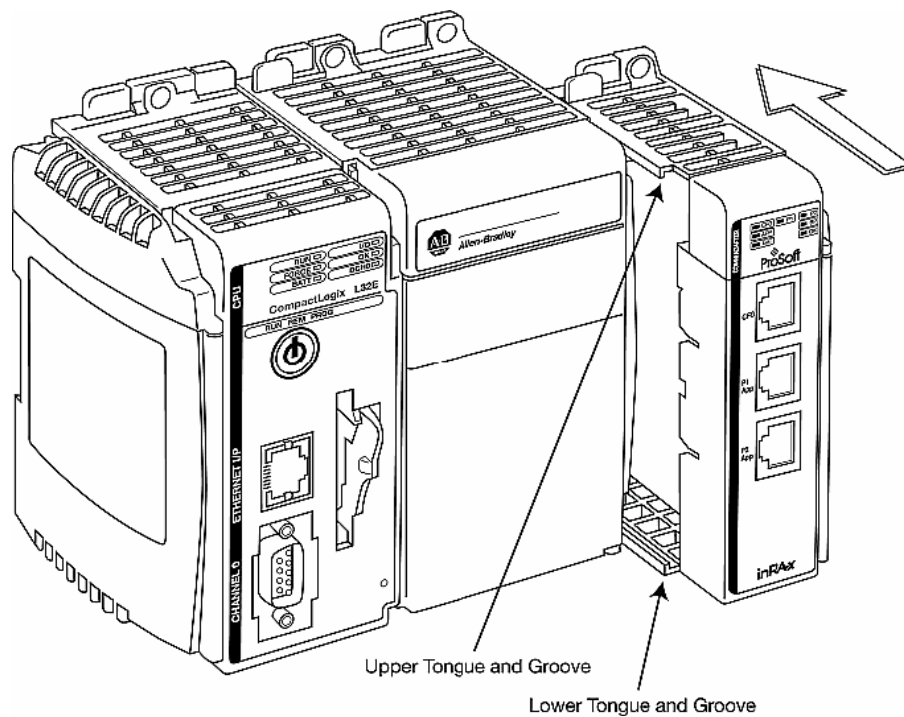
1.5 Install the Module in the Rack

This section describes how to install the module into a CompactLogix or MicroLogix rack

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

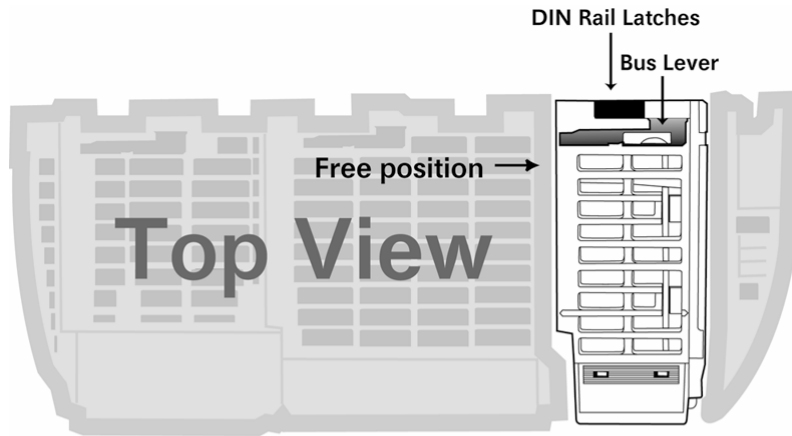
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

- 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.

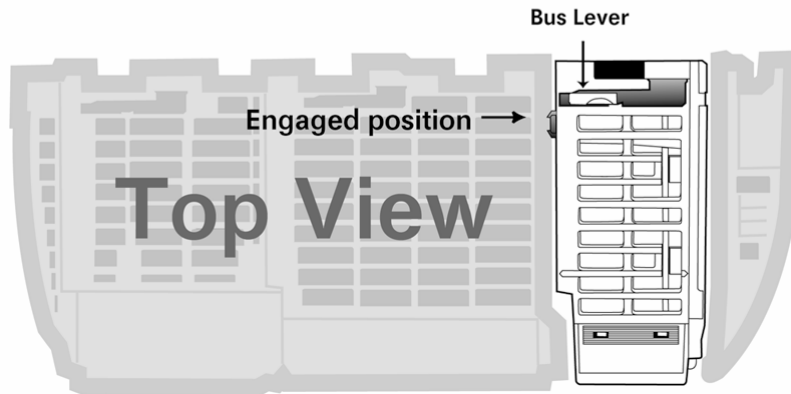


- 2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

- 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.

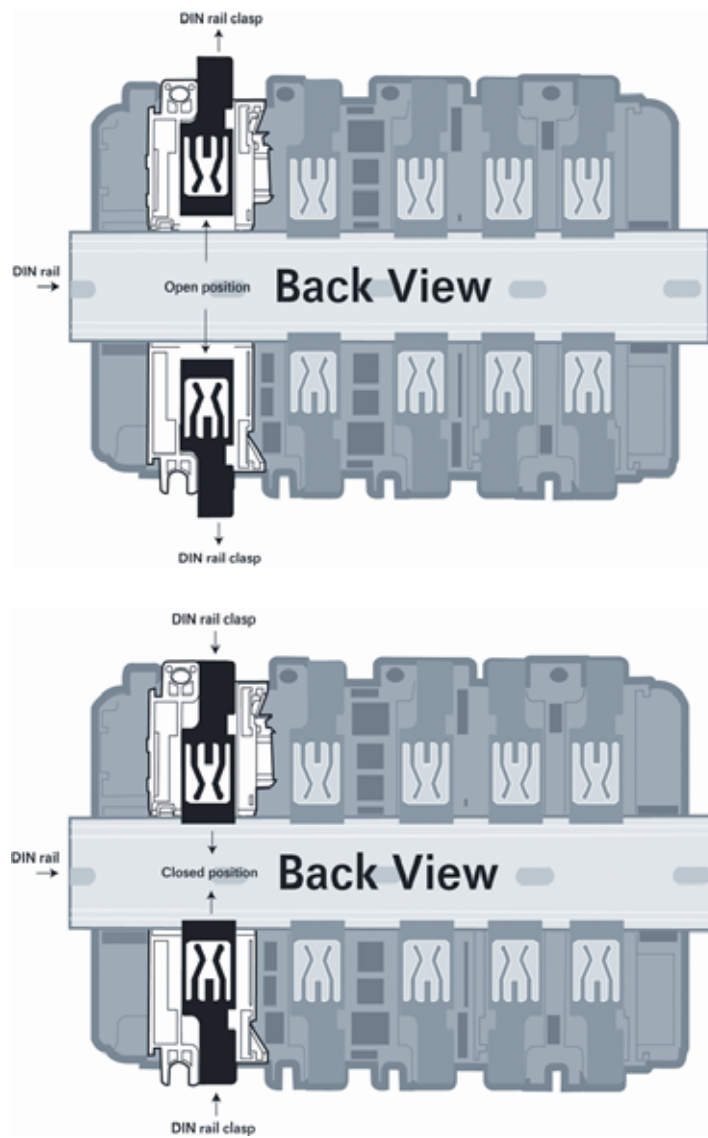


Move the Bus Lever to the left
until it clicks



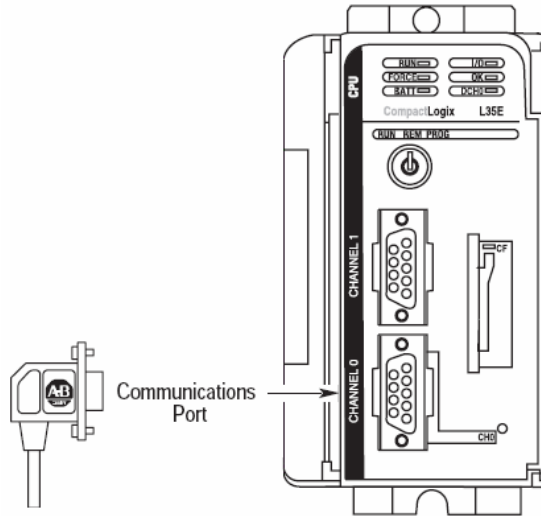
- 4 Close all DIN rail latches.

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

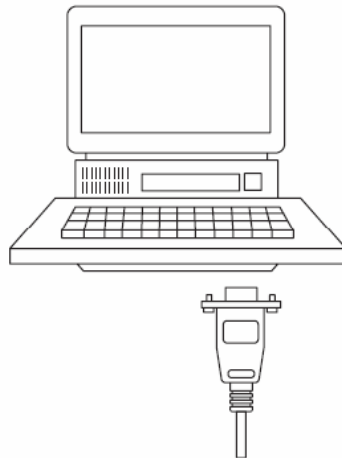


1.6 Connect your PC to the Processor

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



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

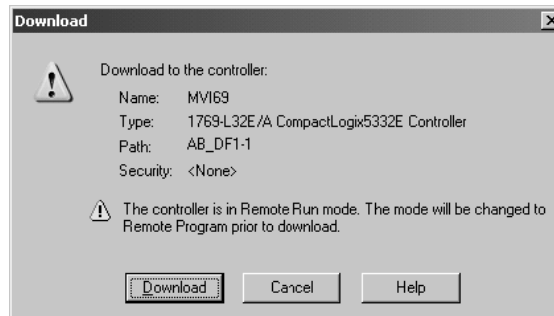


1.7 Download the Sample Program to the Processor

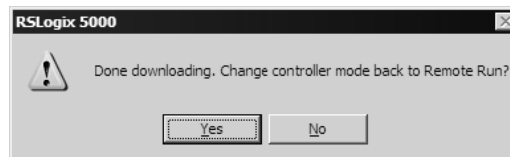
Important: For most applications, the sample program will work without modification.

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

- 1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.
- 2 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.

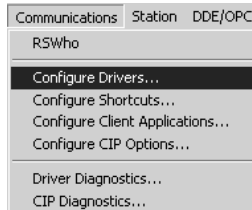


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

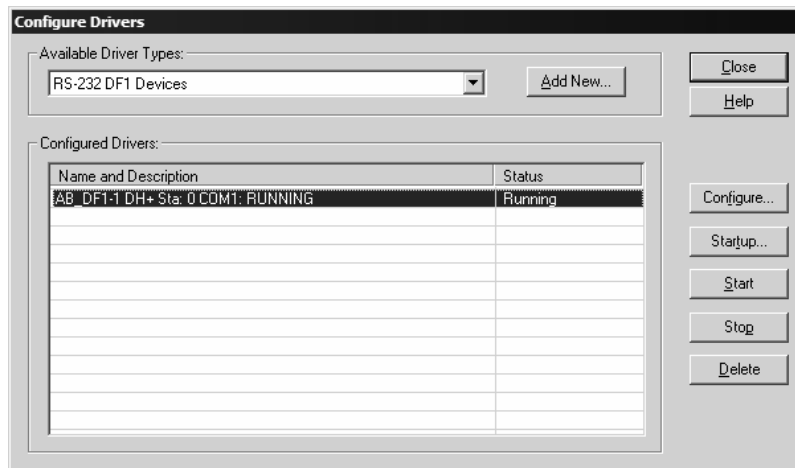
1.7.1 Configuring RSLinx

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

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



This action opens the Configure Drivers dialog box.



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

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



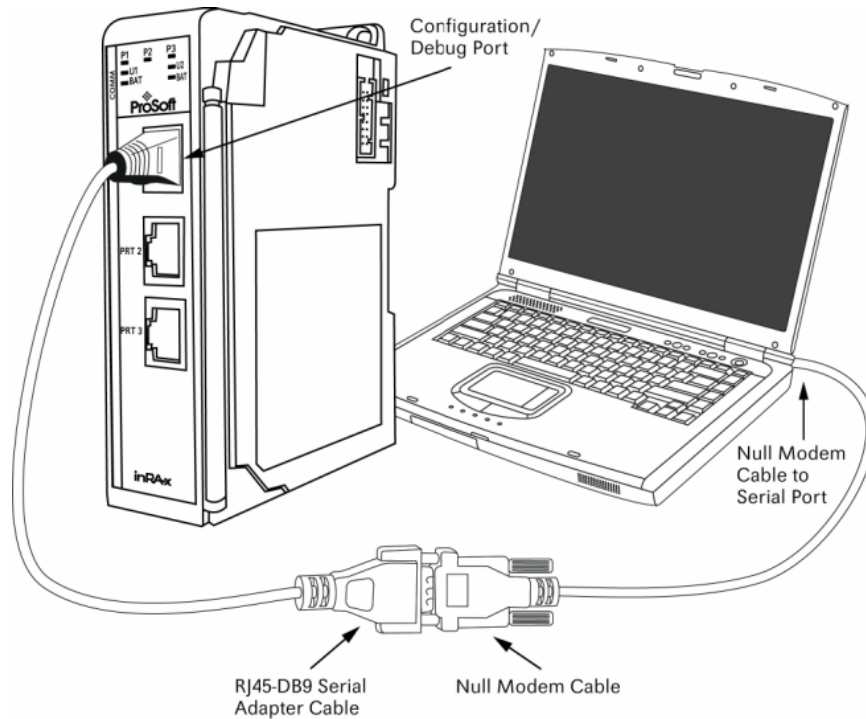
- Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.
- When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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 Connect your PC to the Module

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

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



2 Configuring the MVI69-DH485 Module

In This Chapter

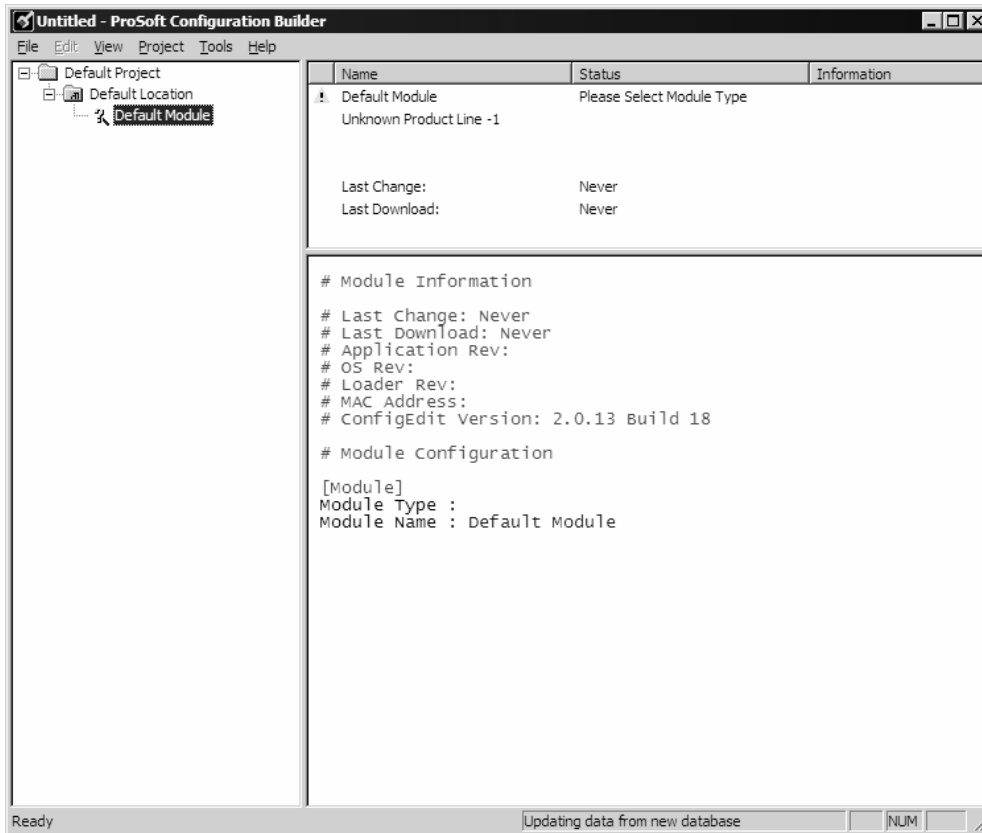
❖ ProSoft Configuration Builder	21
❖ [Module].....	26
❖ [DH485 Port x].....	28
❖ [DH485 Port x Commands].....	30
❖ [DH485 Port x Maps].....	33
❖ File Override Mapping	34
❖ Timer, Counter, and Control Data Types.....	35
❖ Download the Project to the Module.....	38

2.1 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 is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

2.1.1 Set 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.



Your first task is to add the MVI69-DH485 module to the 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.

- On the shortcut menu, choose "Choose Module Type". This action opens the Choose Module Type dialog box.



- In the Product Line Filter area of the dialog box, select MVI69. In the Select Module Type dropdown list, select MVI69-DH485, and then click OK to save your settings and return to the ProSoft Configuration Builder window.

Adding a Project

To add a project to an existing project file:

- Select the Default Project icon.
- Choose Project from the Project menu, then choose Add Project. A new project folder appears.

Adding a Module

To add a module to your project:

- Double-click the Default Module icon to open the Choose Module Type dialog box.
- On the Choose Module Type dialog box, select the module type.

Or

- Open the Project menu and choose Location.
- On the Location menu, choose Add Module.

To add a module to a different location:

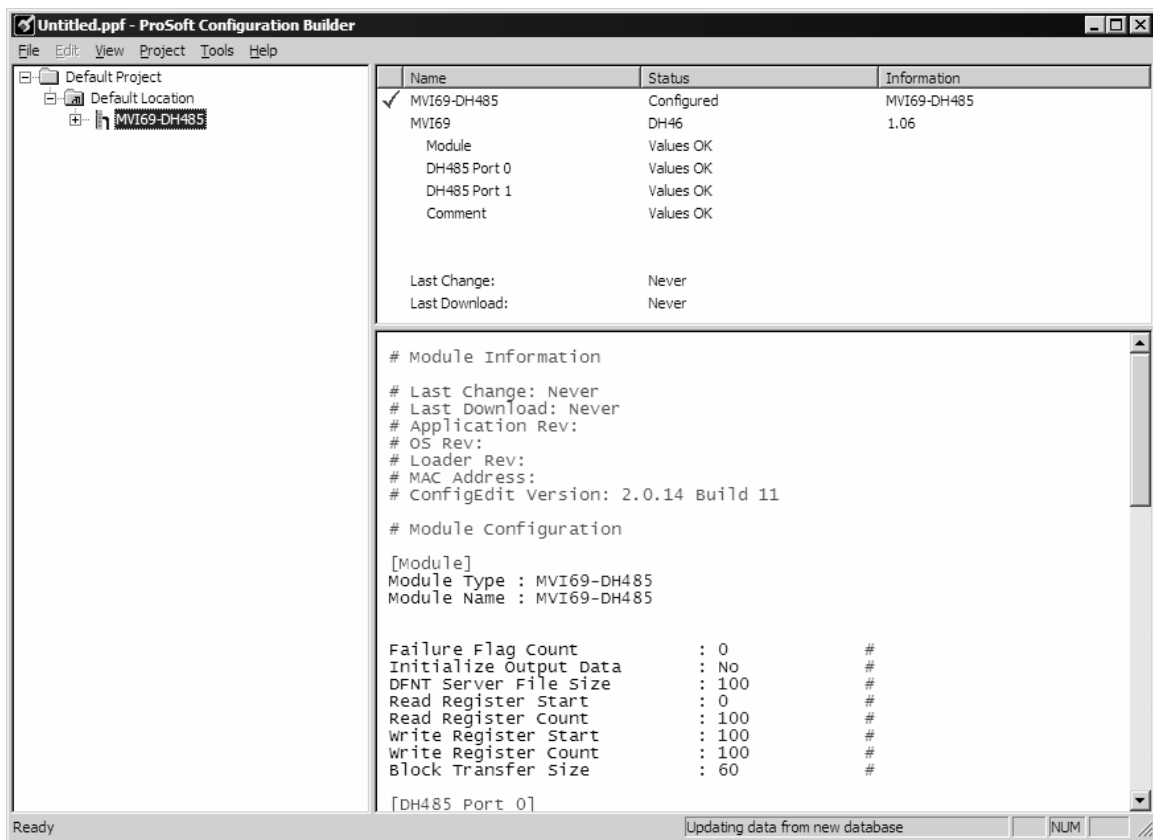
- 1 Right-click the Location folder and choose Add Module. A new module icon appears.

Or

- 1 Select the Location icon.
- 2 From the Project menu, select Location, then select Add Module.

2.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the MVI69-DH485 module to the project.





At this time, you may wish to rename the "Default Project" and "Default Location" folders in the tree view.

To rename an object:

- 1 Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose Rename.
- 2 Type the name to assign to the object.
- 3 Click away from the object to save the new name.



Module Entries

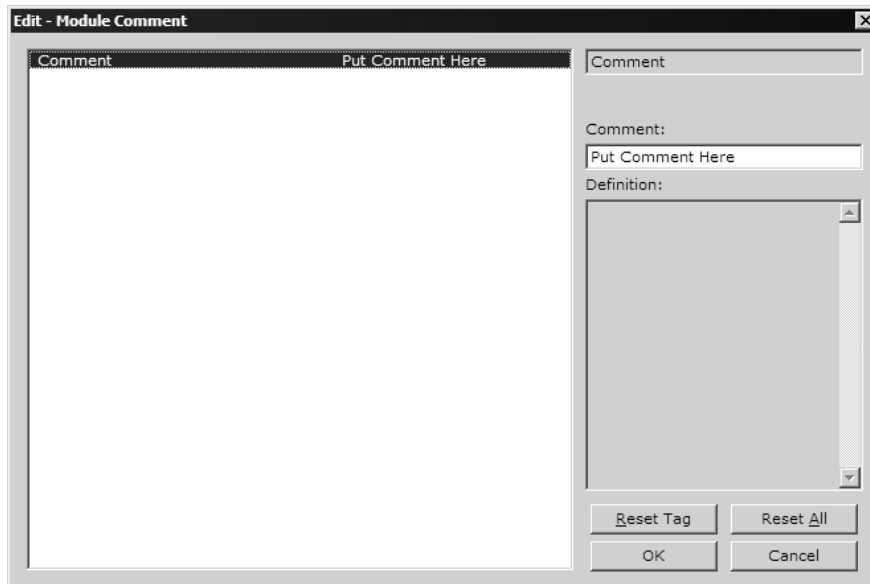
To configure module parameters

- 1 Click on the plus sign next to the icon  Comment to expand module information.
- 2 Double-click the  Module Comment icon to open the Edit dialog box.
- 3 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
- 4 Click OK to save your changes.

Comment Entries

To add comments to your configuration file:

- 1 Click the plus sign to the left of the  Comment icon to expand the Module Comments.
- 2 Double-click the  Module Comment icon. The Edit - Module Comment dialog appears.



- 3 Enter your comment and click OK to save your changes.

Printing a Configuration File

To print a configuration file:

- 1 Select the Module icon, and then click the right mouse button to open a shortcut menu.
- 2 On the shortcut menu, choose View Configuration. This action opens the View Configuration window.
- 3 On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.
- 4 On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

2.2 [Module]

This section provides the module with a unique name, identifies the method of failure for the communications for the module if the PLC is not in run, and describes how to initialize the module upon startup.

The following example shows a sample [Module] section:

```
# This section of the file describes the database setup and module level
# parameters.

[Module]
Module Name : DH485 Communication Module Test

Error/Status Pointer      : 3900

#Required for all except MVI46 and PROTOC
Read Register Start      : 600 #Starting DB address where read data stored
Read Register Count      : 600 #Number of regs to read from module by processor
Write Register Start     : 0   #Starting DB address where write data accessed
Write Register Count     : 600 #Number of regs to write to module from processor

#This parameter is required for MVI69 applications
Block Transfer Size      : 60  #Number of data words for msgs 60, 120 or 240
Failure Flag Count       : 0   #Determines if BP failure will cause protocol to
                             #be disabled (0=Ignore, >0 = failure count to
                             #disable)
Initialize Output Data   : Yes #Read output values from controller (Yes or No)
```

Modify each of the parameters based on the needs of your application.

2.2.1 Module Name

0 to 80 characters

This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.

2.2.2 Error / Status Pointer

-1 to 3999

Starting register location in virtual database for the error/status table. If a value of -1 is entered, the error/status data will not be placed in the database. All other valid values determine the starting location of the data. This data area includes the module version information.

2.2.3 Read Register Start

Range 0 to 3999

This parameter specifies the starting register in the module where data will be transferred from the module to the processor. Valid range for this parameter is 0 to 3999.

2.2.4 Read Register Count

Range 0 to 3999

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 3999.

2.2.5 Write Register Start

0 to 3999

This parameter specifies the starting register in the module where the data will be transferred from the processor to the module.

2.2.6 Write Register Count

Range 0 to 3999

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 3999

2.2.7 Block Transfer Size

60, 120 or 240

This read-only parameter specifies the number of words in each block transferred between the module and processor. Valid values for this parameter are 60, 120 and 240.

2.2.8 Initialize Output Data

Yes or No

This parameter determines if the output data for the module should be initialized with values from the processor. If the value is set to No (0), the output data will be initialized to 0. If the value is set to Yes (1), the data will be initialized with data from the processor. Use of this option requires associated ladder logic to pass the data from the processor to the module.

2.2.9 Failure Flag Count

0 through 65535

This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to 0, the communication ports 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.

2.3 [DH485 Port x]

General configuration information for the specified DH-485 port on the module

```
[DH485 Port 0]
Baud Rate           : 19200 #Baud rate of 1200, 2400, 9600 or 19200
Node Address        :      1 #My node address of 0 to 31
Maximum Node Address :     31 #Maximum node address in network (1-31)
Token Hold Factor   :     10 #Value from 0 to 31 for this node to hold token
Response Timeout    :     10 #Response timeout in 100mSec increments (0-50)
Status DB Offset    :    3800 #DB loc for port status data (-1=ignore)
Command Error DB Offset:  3100 #DB loc for cmd list error data (-1=ignore)
CIF Read DB Offset  :    4000 #DB byte loc <- CIF read data (-1=ignore)
CIF Read Count      :     100 #Numb of bytes to read from CIF data area
                    :          #(0-244)
CIF Write DB Offset :    4200 #DB byte loc -> CIF write data (-1=ignore)
First File          :       7 #First file number to emulate in DB
File Size           :     200 #Number of words in each file emulated
File Offset         :       0 #DB start address for file emulation
Min Command Delay   :    2000 #Min number of milliseconds between commands
```

2.3.1 Baud Rate

Range 1200 to 19200

This is the baud rate to use for the DH485 network. Select one of the listed baud rates.

2.3.2 Node Address

Range 0 to 31

This is the node address to be utilized by the DH-485 driver for this port on the network. Enter a value not already used on the network in the range of 0 to 31. If a value of 255 is utilized or set by the module, the port is disabled. Note: All nodes on the network should be set to the lowest set of values in the range of 1 to 4).

2.3.3 Maximum Node Address

Range 0 to 31

Enter the maximum address that the initiator searches for before wrapping to zero. The default is 31. This parameter should be set to the maximum node address set in the DH-485 network.

2.3.4 Token Hold Factor

Range 0 to 31

Enter the number of transmissions (plus retries) that a node holding a token can send onto the data link each time that it receives the token. Enter a value between 0 and 31. The default is 1.

2.3.5 Response Timeout

Range 1 to 50

This parameter sets the number of 100 millisecond time intervals to wait for a response to a request from the module. If the module does not receive the response within the time period specified, a timeout condition will be set for the command.

2.3.6 Status DB Offset

Range -1 to 3980

This parameter sets the location of the status data for the port in the module's internal database. If the parameter is set to -1, the data is not placed in the database. If a valid value is entered, the module's status data will be placed in the database starting at the location indicated.

2.3.7 Command Error DB Offset

Range -1 to 3900

This parameter sets the location of the command error list data for the port in the module's internal database. If the parameter is set to -1, the data is not placed in the database. If a valid value is entered, the module's error list data will be placed in the database starting at the location indicated.

2.3.8 CIF Read DB Offset

Range -1 to 7500 (Only even values)

This parameter sets the starting byte location in the module's database where the CIF file Read will be placed. This data is passed from CIF memory area to the set location in the module's database. If this parameter is set to -1, no CIF read data will be utilized. When the CIF Read Area is disabled (CIF Read DB Offset = -1) or CIF Read Count = 0, the Debug menu shows this parameter as 65535.

2.3.9 CIF Read Count

Range 0 to 242

This parameter sets the number of bytes to transfer from the CIF file to the database. The CIF write count will be calculated as (244 - CIF Read Count)

2.3.10 CIF Write DB Offset

Range -1 to 7500 (Only even values)

This parameter sets the starting byte location in the module's database where the CIF file Write data will be read from. This data is passed to the CIF memory area from the set location in the module's database. If this parameter is set to -1, no CIF write data will be utilized. When the CIF Write Area is disabled (CIF Write DB Offset = -1) or CIF Write Count = 0, the Debug menu shows this parameter as 65535.

2.3.11 First File

Range 0 to 255

This parameter sets the file number for the first file to be emulated by the module.

2.3.12 File Size

Range 1 to 1000

This parameter sets the word size of all the files emulated in the module

2.3.13 File Offset

Range 1 to 999

This parameter sets the word offset into the module's database where the file emulation will start.

2.3.14 Min Command Delay

Range 0 to 10000

This parameter sets the minimum number of milliseconds to wait before issuing each command. This parameter is utilized to keep the network from being flooded with requests from the module.

2.4 [DH485 Port x Commands]

The [DH485 Port x Commands] section for each port defines the commands to be issued by the module to other devices on the network. These commands can be used for data collection and/or control.

2.4.1 Command List Overview

In order to interface the MVI69-DH485 module to act as a master device, you must construct a command list for each port. The commands in the list specify the node to be addressed, the function to be performed (read or write), the data area in the device to interface with and the registers in the internal database to be associated with the device data. The command list supports up to 100 commands. The command list is processed from top (command #0) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in seconds between the issuance of a command. If the user specifies a value of 10 for the parameter, the command will be executed 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 network. In order to implement this feature; set the enable code for the command to a value of 2.

2.4.2 Command Entry Format

Each command entered in the command list section has the same format. The following is an example section for Port 0:

```
[DH485 Port 0 Commands]
#
# Enable Types : 0=Disable, 1=Poll at interval, 2=Conditional Poll
# Swap Codes : 0=None, 1=Swap Words, 2=Swap bytes and words, 3=Swap bytes
# Function Codes: 0=Read, 1=Write
# FileType Codes: 0=status, 1=bit, 2=timer, 3=counter, 4=control, 5=int,
#               6=float, 10=CIF
#
#Enbl      DB  Elem  Swap  Poll  Node  Func  File  File  Element
#Type      Address count Code  Int   #   Code Type  #   Number
START
  1    1000    10    0    0    6    0    5   11    0
  1    1000    10    0    0    6    1    5   10    0
  0    1000   100    0    0    5    0    5    7    0 //read port 0 status data
  0    1000   100    0    0    5    1    5   16    0 //read port 1 status data
  0    1500    10    0    0    6    1    5   10    0 //write status data
  0    1500    10    0    0    6    0    5   11    0 //read status data
  1     250     1    0    0    2    1    6    8    5 //write status data
  1     250     1    0    0    2    0    6    8    0 //write status data
  0    2260    10    0    0    6    1   10    9   10 //write status data
  0    2260    10    0    0    6    0   10    9    0 //write status data
END
```

The first part of each record in the section relates to the module interface and the last part relates to the node to be interfaced with. The following topics describe each field required for a user command

2.4.3 Enable Type Code

This field defines if the command is enabled and when it should be executed. The following codes are recognized by the application:

- 0 = Command is disabled
- 1 = Command is executed at the polling interval specified
- 2 = The write command is only executed when data changes

2.4.4 Database Start Address

This field defines the starting address in the module's internal database to associate with the command. This field can have a value from 0 to 3999. The address supplied is a word address in the database.

2.4.5 Element Count

This field defines the number of elements to be used with the command. If the command is interfacing with CIF data, this parameter represents a byte count. For data file access, the data size utilized is dependent on the file type used.

2.4.6 Swap Code

This field changes the order of the bytes and/or words used when sending or receiving the data. The following codes are utilized:

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)

2.4.7 Poll Interval

This field sets the time interval between successive execution of the command. This parameter is specified in seconds. If the field is set to 10, the command will not be executed more frequently than every 10 seconds.

2.4.8 Node Number

This field defines the node address of the DH-485 node to send the command request. This field should be set to a value from 0 to 31.

2.4.9 Function Code

This field defines the function to be executed by the command. The module uses the following codes:

- 0 = Read
- 1 = Write

2.4.10 File Type

This field defines the file type to be interfaced with in the other DH-485 node. The program utilizes the following codes for this field:

- 0 = Status File (2 bytes per element)
- 1 = Bit File (2 bytes per element)
- 2 = Timer File (6 bytes per element)
- 3 = Counter File (6 bytes per element)
- 4 = Control File (6 bytes per element)
- 5 = Integer File (2 bytes per element)
- 6 = Floating-point File (4 bytes per element)
- 10 = CIF File (1 byte per element)

2.4.11 File Number

This field defines the file number to access. This field is ignored for CIF file access and should be set to 0. For Bit, Timer, Counter, Control, Integer, and Float data types, a maximum value of 255 is valid.

2.4.12 Element Number

This field defines the first element in the file specified to be associated with the command. For a CIF file, this parameter is given as the byte location. For Bit, Timer, Counter, Control, Integer, and Float data types, the maximum value is 255. For CIF data types the maximum value is 510.

Maximum Element Counts for Read/Write Commands

The maximum number of elements requested from a remote node is determined by the file type and function code utilized in the command. The following table lists the maximum element count for each file type:

File Type	Bytes/Elements	Read Result	Write Result
Status	2	83	83
Bit	2	118	115
Timer	6	39	38
Counter	6	39	38
Control	6	39	38
Integer	2	118	115
Float	4	59	57
CIF	1	236	234

2.5 [DH485 Port x Maps]

Each file map entered in the configuration file has the same format. The following is an example section for Port 0:

```
# This section contains a list of file override values to overlay the database.
# The list will be searched by the driver for each data request made of the
# driver on the specified port. If the file:element value in the request
# matches an entry in the list, the database offset assigned to the entry
# will be used in conjunction with the other data in the entry to return
# or set the data.

[DH485 Port 0 Maps]
# DB_Address   File_Number   Element   Length
START
#           0           110       0       100
#           100          112       0       100
END
```

These file maps are searched first when a node on the network makes a request. If the requested data area is found in the map list, the database area associated with the command will be used. If the requested data area is not found in the map list, the fixed mapping data configuration will be used.

The following topics describe each field required for an override map.

2.5.1 Database Start Address

This field defines the starting address in the module's database for the file emulation. This parameter can be assigned a value of 0 to 3999.

2.5.2 File Number

This field defines the file number to be emulated at the database location specified.

2.5.3 Element

This field specifies the first element in the file to be emulated. This element number corresponds to the database start address set for the record.

2.5.4 Word Count (Length)

This field defines the number of word registers to be emulated in the file.

2.6 File Override Mapping

The [DH485 Port x Maps] section for each port defines file override mappings for the module. Use of these maps provides flexibility defining the file emulation supported in the slave driver of the module. Up to 50 maps can be defined in the module to override the fixed file-mapping feature of the module.

2.6.1 File Override Entry Format

Each file map entered in the configuration file has the same format. The following is an example section for Port 0:

```
[DH485 Port 0 Maps]          Override file maps for first DH485 port
# DB_Address  File_Number  Element  Length
START
      0         10         0         100
     100        12         0         100
END
```

The following table describes the definition of each field required for an override map:

Field	Definition
Database Start Address	This field defines the starting address in the module's database for the file emulation. This parameter can be assigned a value of 0 to 3999.
File Number	This field defines the file number to be emulated at the database location specified.
Element	This field specifies the first element in the file to be emulated. This element number corresponds to the database start address set for the record.
Word Count (Length)	This field defines the number of word registers to be emulated in the file.

These file maps are searched first when a node on the network makes a request. If the requested data area is found in the map list, the database area associated with the command will be utilized. If the requested data area is not found in the map list, the fixed mapping data configuration will be used.

2.7 Timer, Counter, and Control Data Types

2.7.1 Timer Data Type

The Timer data type uses its 3 words (6 bytes) as described below (for example, T4:0):

Word 0:

- Bit 9: timebase selection
- Bit 13: T4:0.DN
- Bit 14: T4:0.TT
- Bit 15: T4:0.EN

Word 1:

T4:0.PRE

Word 2:

T4:0.ACC

2.7.2 Counter Data Type

The Counter data type uses its 3 words (6 bytes) as described below (for example, C5:0)

Word 0:

- Bit 10:UA
- Bit 11:UN
- Bit 12:OV
- Bit 13:DN
- Bit 14:CD
- Bit 15:CV

Word 1: C5:0.PRE

Word 2: C5:0.ACC

2.7.3 Control Data Type

The Control Data Type uses its three words (6 bytes) as shown in the following example. For example, R6:0.

Word 0:

- Bit 8: FD
- Bit 9: IN
- Bit 10: UL
- Bit 11: ER
- Bit 12: EM
- Bit 13: DN
- Bit 14: EU
- Bit 15: EN

Word 1: R6:0.LEN

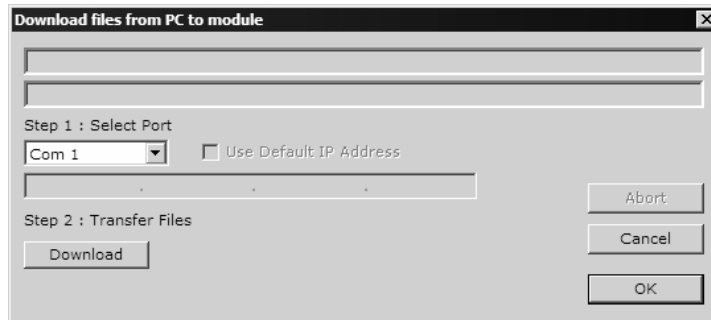
Word 2: R6:0.POS

2.8 Download the Project to the Module

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

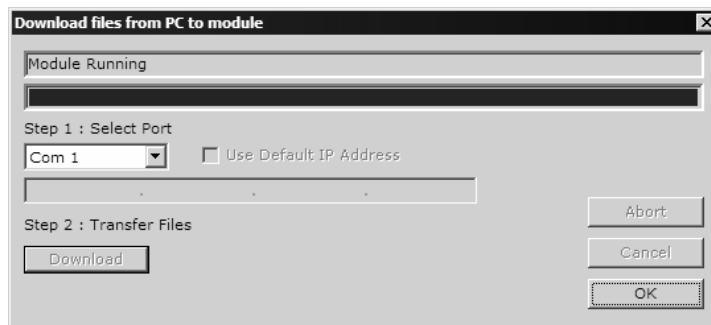
To Download the Project File

- 1 In the tree view in ProSoft Configuration Builder, click once to select the MVI69-DH485 module.
- 2 Open the **Project** menu, and then choose **Module / Download**. The program will scan your PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the following dialog box will open.



- 3 Choose the com port to use from the dropdown list, and then click the Download button.

The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in ProSoft Configuration Builder will be updated with the message *"Module Running"*.



3 Ladder Logic

In This Chapter

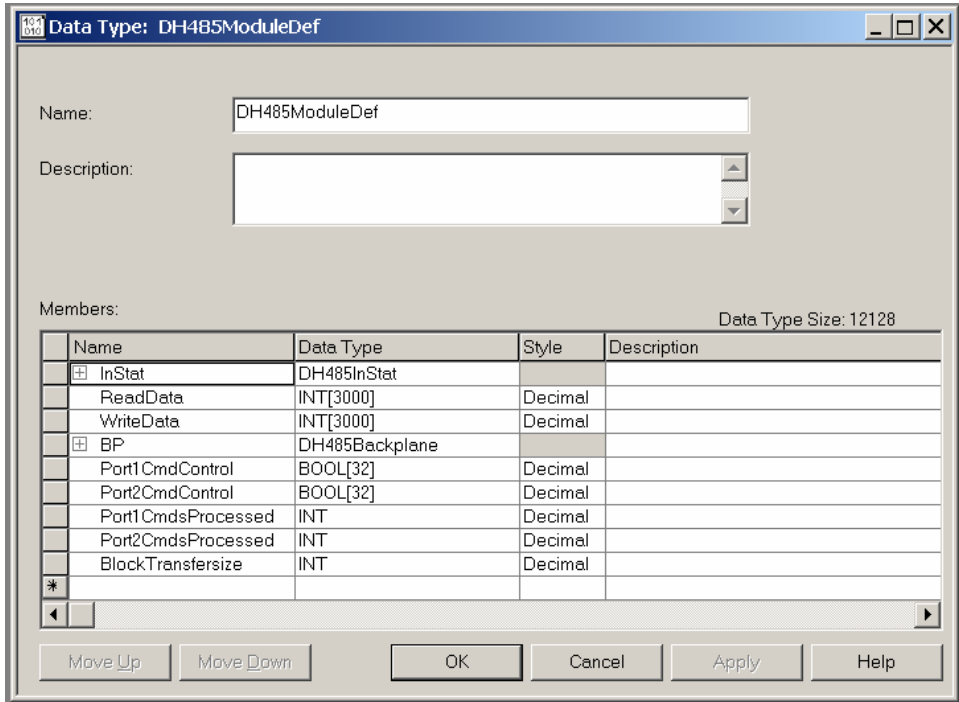
- ❖ Module Data Object (DH485ModuleDef) 40
- ❖ Adding the Module to an Existing CompactLogix Project 45
- ❖ Adding the Module to an Existing MicroLogix Project..... 49

Ladder logic is required for application of the MVI69-DH485 module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

3.1 Module Data Object (DH485ModuleDef)

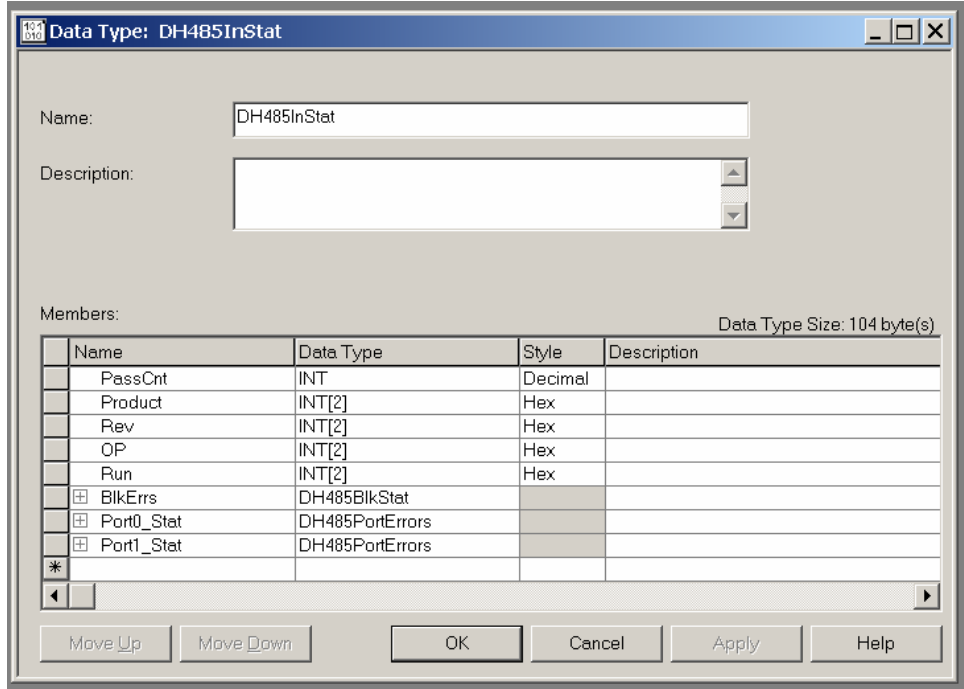
All status and variable data related to the MVI69-DH485 is stored in a user defined data type. An instance of the data type is required before the module can be used. This is done by declaring a variable of the data type in the Controller Tags Edit Tags dialog box. The object has the following structure.



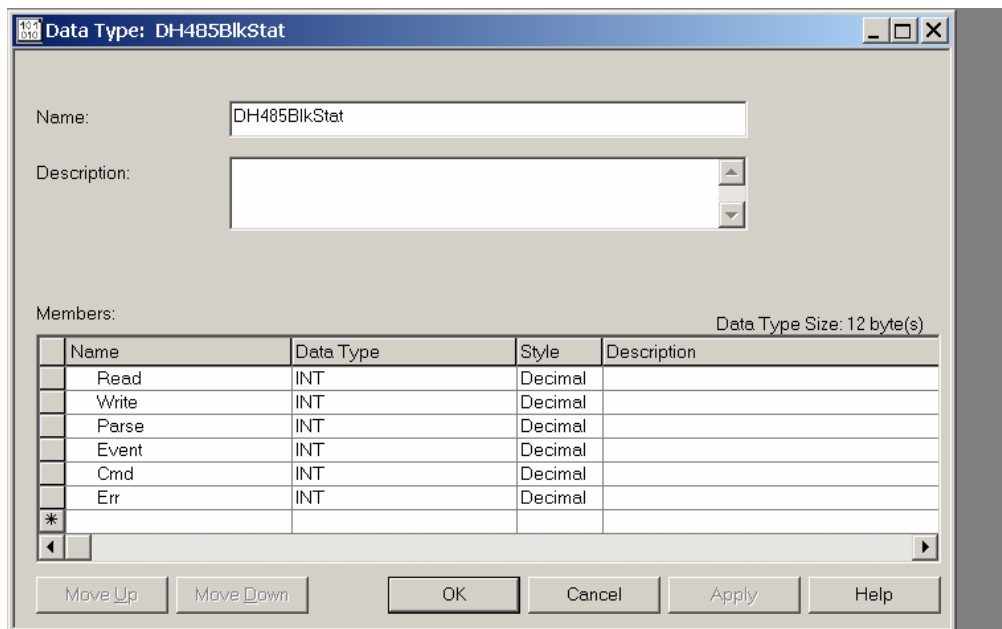
This object contains objects that define variables to be used with the module and status data related to the module. Each of these object types is discussed in the following topics of the document.

3.1.1 Status Object (DH485InStat)

This object stores the status data of the module. The DH485InStat object shown below is updated each time a read block is received by the processor. The status data for each port is passed in blocks with identification codes of 0 and -1. Use this data to monitor the state of the module at a "real-time rate".



The structure of the DH485BlkStat is shown in the following diagram and stores the status information for the backplane data transfer operation:



The structure of the DH485PortErrors is displayed in the following diagram:

Name:

Description:

Members: Data Type Size: 36 byte(s)

Name	Data Type	Style	Description
Active_Nodes	DINT	Binary	
Online_Status	INT	Decimal	
Node_address	INT	Decimal	
Cur_Cmd	INT	Decimal	
Requests	INT	Decimal	
Responses	INT	Decimal	
Cmd_Err	INT	Decimal	
Cfg_Err_Word	INT	Hex	
Reserved_18_19	INT	Decimal	
Rec_Packets	INT	Decimal	
Trans_Packets	INT	Decimal	
Retry_Cnt	SINT	Decimal	
Retry_Fail	SINT	Decimal	
NAK_rx	SINT	Decimal	
NAK_tx	SINT	Decimal	
Bad_Packets	SINT	Decimal	
Bad_Cntl	SINT	Decimal	
CRC_err	SINT	Decimal	
Parity_err	SINT	Decimal	
Frame_err	SINT	Decimal	
Overrun_err	SINT	Decimal	
Unexp_err	SINT	Decimal	
Bad_LSAP_rx	SINT	Decimal	
*			

Move Up Move Down OK Cancel Apply Help

Refer to Status Data Definition (page 86) for a complete list of the data stored in this object. This data is passed in blocks 0 and -1 for each port on the module.

3.1.2 User Data Objects

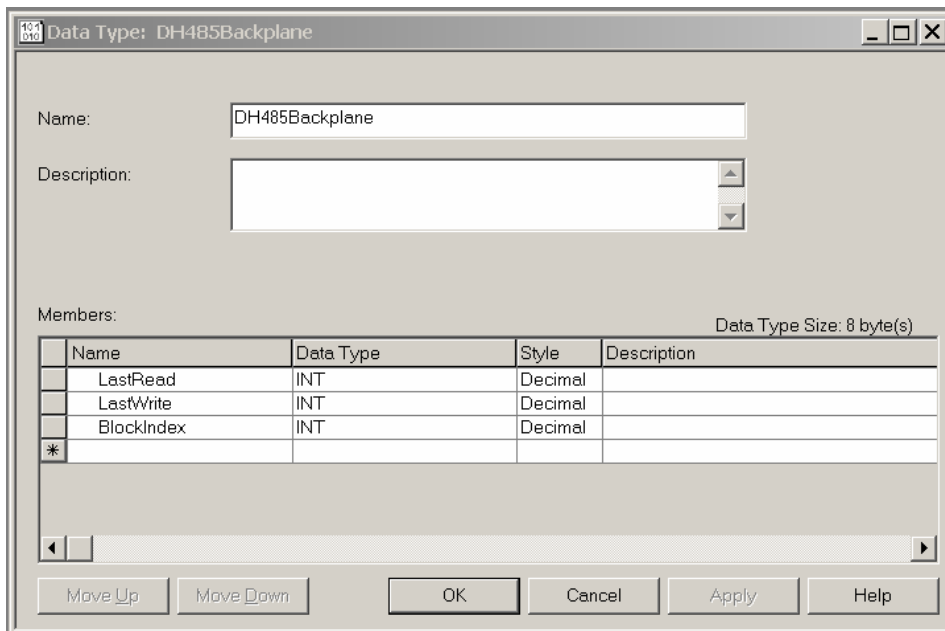
These objects hold data to be transferred between the processor and the MVI69-DH485 module and to aid in the generation of the ladder logic. The user data is the read and write data transferred between the processor and the module as "pages" of data up to 200 words long.

▶	⊕ DH485_1.ReadData	{ ... }	{ ... }	Decimal	INT[3000]
	⊕ DH485_1.WriteData	{ ... }	{ ... }	Decimal	INT[3000]

The read data (**ReadData**) is an array set to match the value entered in the **Read Register Count** parameter of the DH485.CFG file. For ease of use, this array should be dimensioned as an even increment of 200 words. This data is paged up to 200 words at a time from the module to the processor. The ReadData task is responsible for placing the data received into the proper position in the read data array. Use this data for status and control in the ladder logic of the processor.

The write data (**WriteData**) is an array set to match the value entered in the **Write Register Count** parameter of the DH485.CFG file. For ease of use, this array should be dimensioned as even increments of 200 words. This data is paged up to 200 words at a time from the processor to the module. The WriteData task is responsible for placing the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.

The DH485Backplane object is utilized by the ladder logic to store the variables used in the backplane transfer operation. The structure of the object is shown in the following diagram:



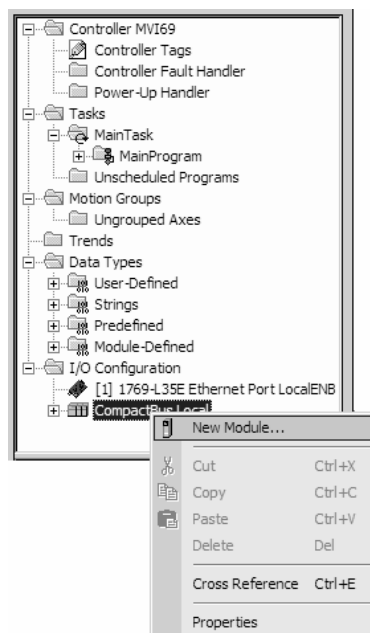
Other data objects can be added to the DH485ModuleDef object as required for specific applications. It is recommended to only add objects or members to the DH485ModuleDef object that relate to the module.

3.2 Adding the Module to an Existing CompactLogix Project

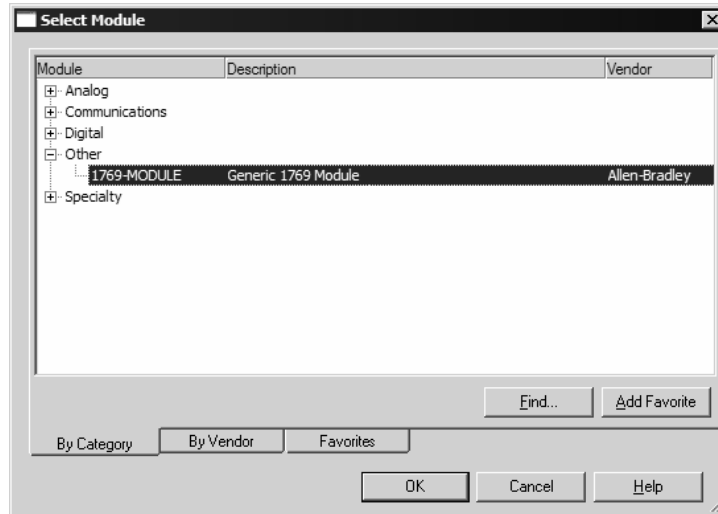
Important: The MVI69-DH485 module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

If you are installing and configuring the module with a CompactLogix processor, follow these steps. If you are using a MicroLogix processor, refer to the next section.

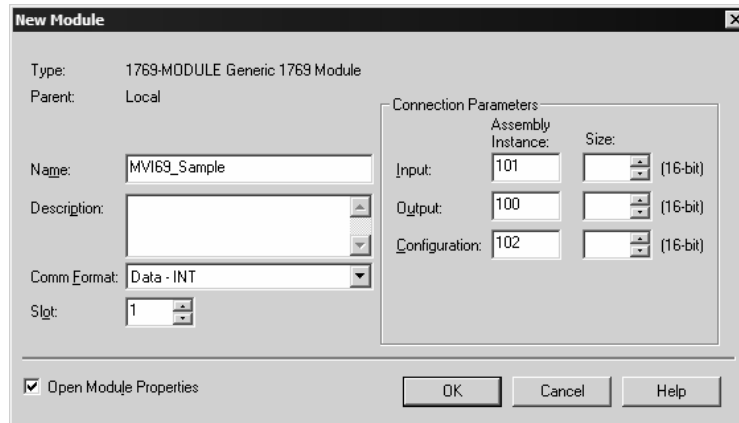
- 1 Add the MVI69-DH485 module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



This action opens the following 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

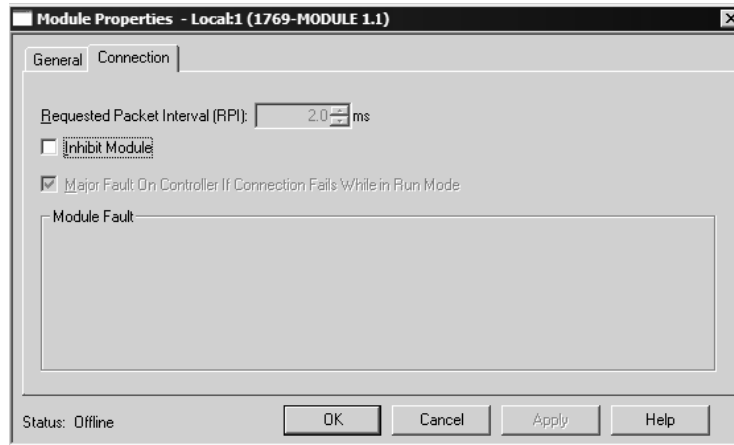
Field	Recommended Value
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT

Block Transfer Size = 60	
Field	Recommended Value
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

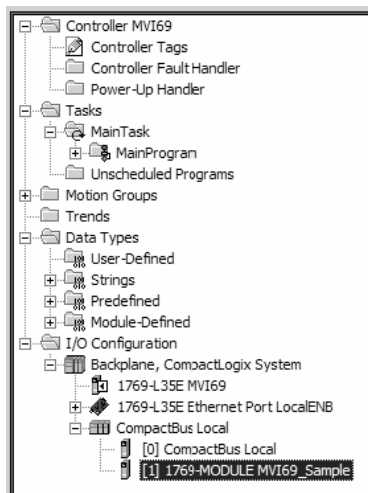
Block Transfer Size = 120	
Field	Recommended Value
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	122
Output Assembly Instance	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 Request Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 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:



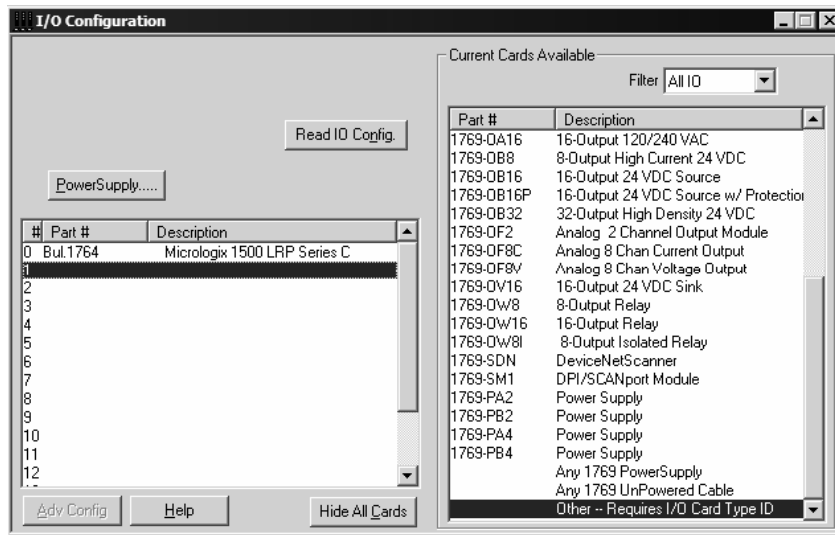
- 8 Copy the Controller Tags from the sample program.
- 9 Copy the User Defined Data Types from the sample program.
- 10 Copy the Ladder Rungs from the sample program.
- 11 Save and Download the new application to the controller and place the processor in run mode.

3.3 Adding the Module to an Existing MicroLogix Project

If you are installing and configuring the module with a MicroLogix processor, follow these steps. If you are using a CompactLogix processor, refer to the previous section.

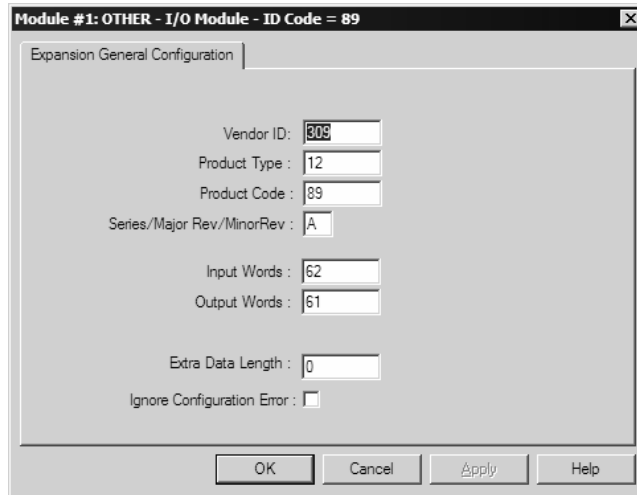
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

- 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 On 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 Module 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 parameter 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 **Next** 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. Refer to the example Ladder Logic section in this manual.

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 51) 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

- ❖ Reading Status Data from the Module 51
- ❖ LED Status Indicators..... 61
- ❖ Clearing a Fault Condition 62
- ❖ Troubleshooting..... 62

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

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

4.1 Reading Status Data from the Module

The MVI69-DH485 module returns a 29-word Status Data block that can be used to determine the module's operating status. This data is located in the module's database at registers 6670 to 6698 and at the location specified in the configuration. This data is transferred to the CompactLogix processor continuously.

4.1.1 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

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

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

4.1.2 The Configuration/Debug Menu

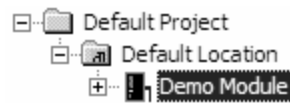
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

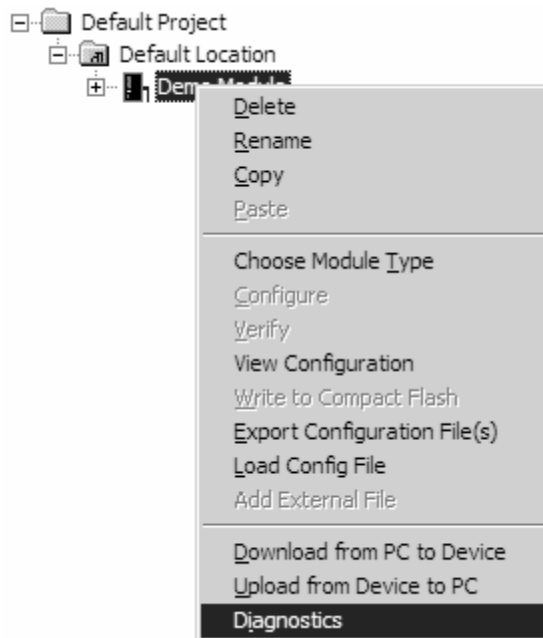
Using the Diagnostic Window in ProSoft Configuration Builder

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

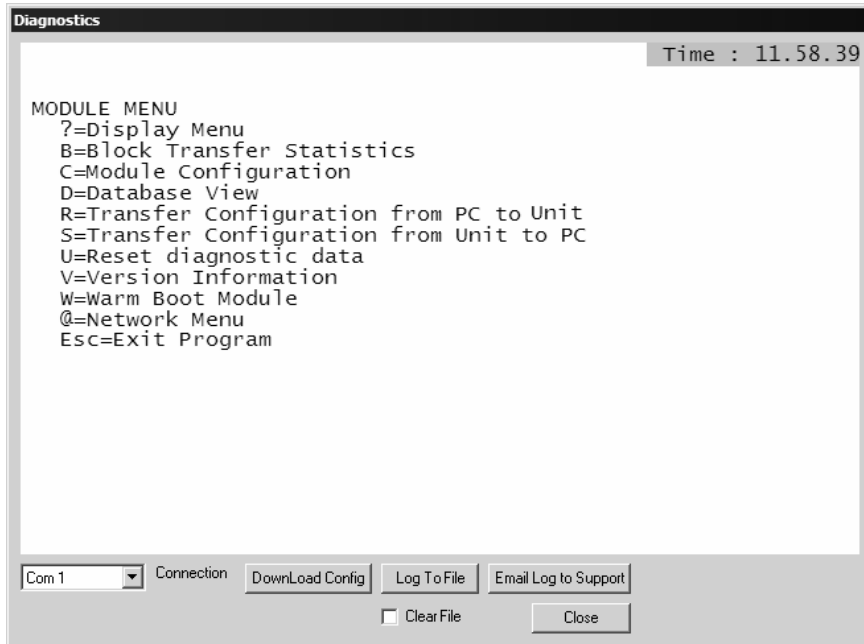
- 1 Start PCB program with the application file to be tested. Right click over the module icon.



- 2 On the shortcut menu, choose Diagnostics.



- 3 This action opens the Diagnostics dialog box. Press "?" to display 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:

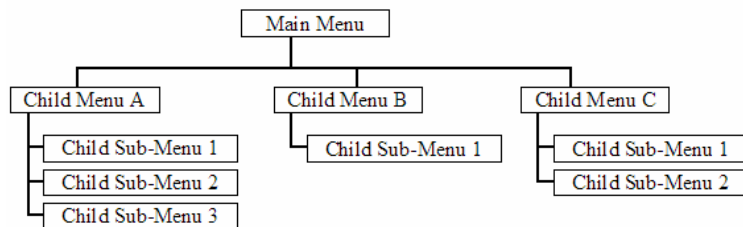
- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

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



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

Keystrokes

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

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

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

4.1.3 Main Menu

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

```
DH485 APPLICATION MENU
?=Display Menu
B=Block Transfer Statistics
C=Module Configuration
D=Database View
E=DH485 Port 0 Menu
F=DH485 Port 1 Menu
R=Transfer Configuration from PC to Module
S=Transfer Configuration from Module to PC
V=Version Information
W=Warm Boot Module
Esc=Exit Program
```

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

Viewing Block Transfer Statistics

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

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

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

Viewing Module Configuration

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

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

Opening the Database Menu

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

Opening the DH-485 Port x Menu

Press **[E]** (port 1) or **[F]** (port 2) to view the configuration and status data related to the DH-485 driver for each of the application ports.

Transferring the Configuration File from PC to MVI69 module

Press **[R]** to send (upload) the configuration file from your PC to the module and store the file on the module's Compact Flash Disk.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, the module will restart the program and load the new configuration information. Review the new configuration using menu commands **[6]** and **[0]** to verify that the module is configured correctly.

Transferring the Configuration File from MVI69 module to PC

Press **[S]** to receive (download) the configuration file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, you can open and edit the file to change the module's configuration.

Warm Booting the Module

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

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

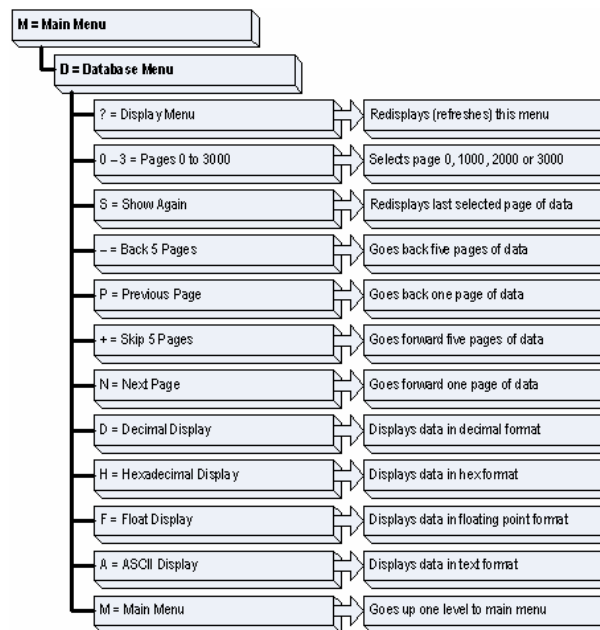
Exiting the Program

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

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

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



Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again

DATABASE DISPLAY 0 TO 99 <DECIMAL>										
100	101	102	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

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

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

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

Returning to the Main Menu

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

4.1.5 DH485 Port x Menu

Use these commands to view the configuration and status data related to the DH-485 driver for each of the application ports. After selecting one of the commands, press the '?' key to display the following menu:



Select one of the options from the menu to perform one of the operations available on the menu. Each command is discussed in the following topics:

Viewing Port Configuration

Press **[C]** to view configuration information for the application port.

Use this command to display detailed configuration information for the port.

Opening the Command Error List Menu

Press **[E]** 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 **[L]** to open the Command List menu. Use this command to view the configured command list for the module.

Opening the Override File Map List Menu

Press **[O]** to view the Override File Map List. Use this command to view the user defined file map configuration data.

Viewing Port Status Data

Press **[S]** to view detailed status information for the application port.

The Cfg Error: parameter displayed will have a value of 0x0000 if there are no configuration errors for the driver. This is a bit mapped value (displayed in hexadecimal format) with each bit representing a configuration error. The following table lists the bits used by the module:

Bit	Code	Description
0	0x0001	Invalid baud rate
1	0x0002	Invalid node address
2	0x0004	Invalid maximum node address
3	0x0008	Invalid token hold factor
4	0x0010	Invalid response timeout
5	0x0020	Invalid status or command error DB offset
6	0x0040	Invalid CIF read count or DB offset
7	0x0080	Invalid CIF write DB offset
8	0x0100	Invalid file size
9	0x0200	Invalid file offset
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

Viewing Version Information

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

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

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

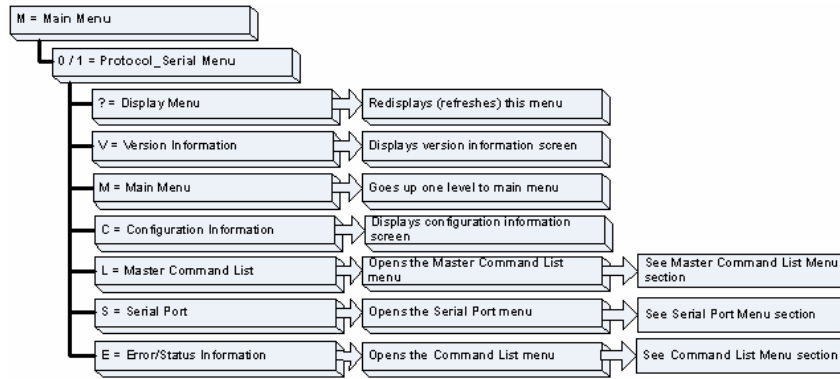
Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Returning to the Main Menu

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

4.1.6 Protocol Serial DH485 Menu

Press **[0]** or **[1]** to view protocol serial information for ports 1 and 2, respectively. Use this command to view a variety of error and status screens for the port. Press **[?]** to view a list of commands available on this 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.

Viewing Version Information

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

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

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

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

Returning to the Main Menu

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

Viewing Configuration Information

Press **[C]** to view configuration information for the selected port, protocol, driver or device.

Opening the Command List Menu

Press **[L]** to open the Command List menu. Use this command to view the configured command list for the module.

Opening the Serial Port Menu

Press **[S]** to open the Serial Port menu. Use this command to view and change additional serial port driver settings.

Viewing Error and Status Data

Press **[E]** to display the error/status data for the module.

4.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the DH-485 network on Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the DH-485 network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	On	The MVI69-DH485 module program has recognized a communication error on one of its ports.
		Off	The MVI69-DH485 is functioning normally.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

During module configuration, the OK LED will be red and the APP and BP ACT LEDs will be on.

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

4.3 Clearing a Fault Condition

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

To clear the condition, follow these steps:

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

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 Support.

4.4 Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic. Module has a power supply distance rating of 2 on Compact Logix. The module must be within 2 slots of the power supply on Compact Logix, or that the MicroLogix backplane can supply the 800ma required for the module.

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: <ul style="list-style-type: none"> ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

5 Reference

In This Chapter

❖ Product Specifications	63
❖ Functional Overview	65
❖ DH485 Communication Ports	81
❖ Status Data Definition	86

5.1 Product Specifications

The MVI69 DH-485 Communication Module allows CompactLogix I/O compatible processors to interface easily with other DH-485 protocol compatible devices. Compatible devices include not only Rockwell Automation PLCs (which support the DH-485 protocol) but also a wide assortment of end devices.

The MVI69-DH485 module actively issues DH-485 commands to other nodes on the DH-485 network. One hundred user-defined commands are supported by the module on each port. The CompactLogix processor can be programmed to control the activity on the port by actively enabling or disabling each command directly from the ladder logic.

The module also accepts DH-485 commands from an attached DH-485 master unit (for example, CompactLogix processor ladder logic MSG instruction). The module permits a remote master to interact with all data contained in the module. This data can be derived from other DH-485 slave devices on the network through a master port or from the CompactLogix processor.

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 and processor. Sample ladder file included.
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included.
- Supports all CompactLogix processors: L20/L23/L30/L31/L32/L35, L43 and L45 (L43 and L45 supported with RSLogix 5000 v16.03 or later)
- Also supports MicroLogix 1500 LRP

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 to 60°C (32 to 140°F)
Storage Temp.	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Battery and Module Status Application Status Serial Port Activity CFG Port Activity
CFG Port (CFG)	RJ45 (DB-9F with supplied cable) RS-232 only No hardware handshaking
App Ports (P1,P2) (Serial modules)	RS-232, RS-485 or RS-422 (jumper selectable) RJ45 (DB-9F with supplied cable) RS-232 handshaking configurable 500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port 6-foot RS-232 configuration Cable

5.1.3 Functional Specifications

The MVI69-DH485 module acts as an input/output module between the DH-485 network and the CompactLogix processor. The data transfer from the CompactLogix processor is asynchronous from the actions on the DH-485 network. A 4000-word register space in the module exchanges data between the processor and the DH-485 network.

- The module has a Common Interface File (CIF) with 244 words that can be accessed by remote DH-485 nodes
- Read and write commands can be sent associated to the following file types: status, bit, timer, counter, control, integer, float, and CIF
- User-definable module memory usage
- Two ports emulate the DH-485 devices (peer-to-peer) with distinct node addresses
- All configuration information contained in a single, user-defined text file
- CompactLogix processor interface
- All module-related data is contained in an internal database
- Ability to add or remove nodes without disruption of the network
- Baud rates: 9600 or 19200 baud

5.2 Functional Overview

This section provides an overview of how the MVI69-DH485 module transfers data using the DH485 protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI69-DH485 module.

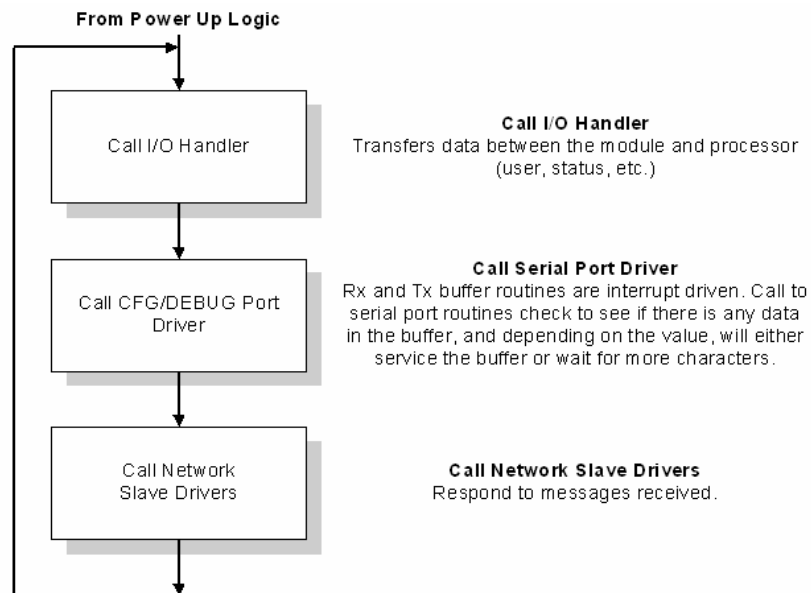
Module Power Up

- 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
- Module configuration
- Initialize Module Register space
- Enable Slave Driver on selected ports
- Enable Master Driver on selected ports

Once 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.



Backplane Data Transfer

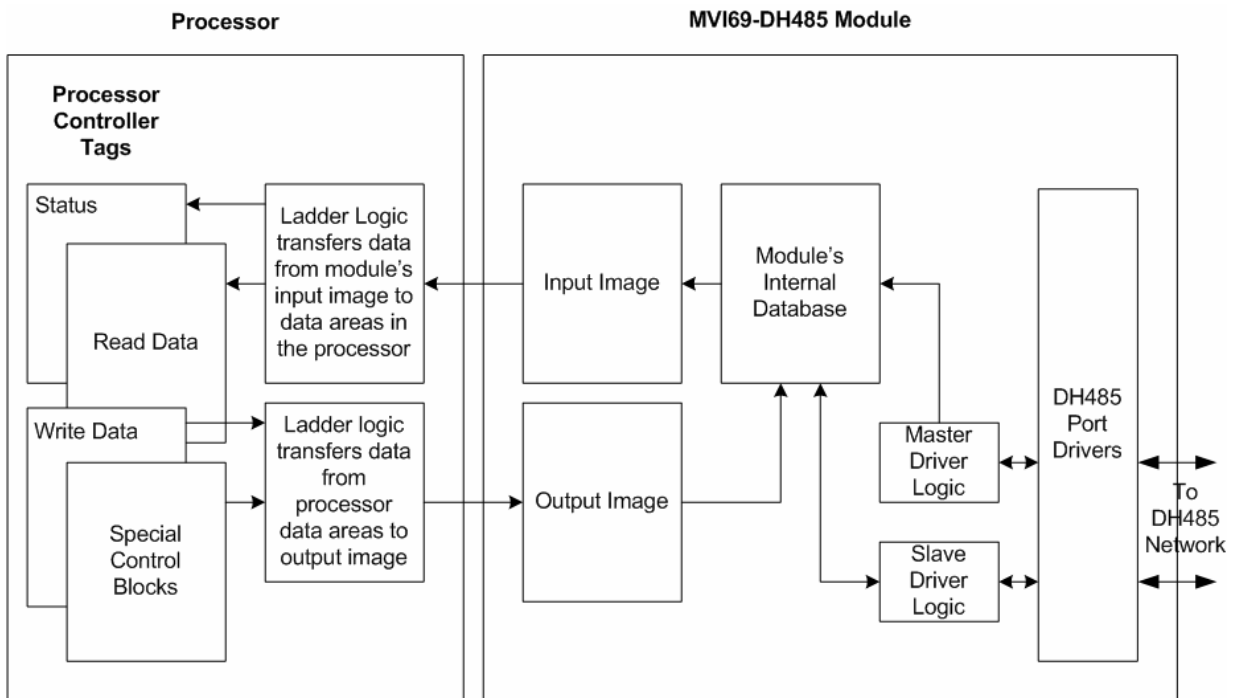
The MVI69-DH485 module communicates directly over the CompactLogix or MicroLogix backplane. Data is paged between the module and the CompactLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

The data is paged between the processor and the module using input and output image blocks. You can configure the size of the blocks using the Block Transfer Size parameter in the configuration file. You can configure blocks of 60, 120, or 240 words of data depending on the number of words allowed for your own application.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module may be set to 62, 122, or 242 words depending on the block transfer size parameter set in the configuration file.

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-DH485 module and the DH-485 network.

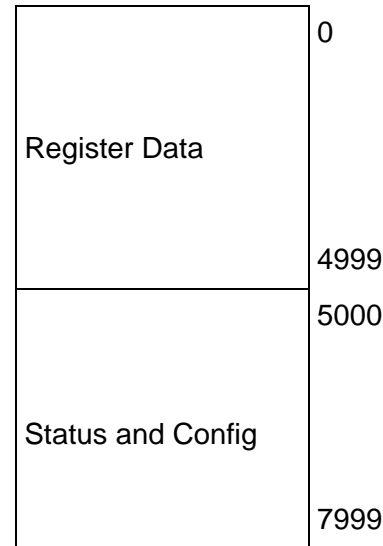


All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the CompactLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Module's Internal Database Structure

5000 registers for user data

3000 words of configuration and status data



Data contained in this database is paged through the input and output images by coordination of the CompactLogix ladder logic and the MVI69-DH485 module's program. Up to 242 words of data can be transferred from the module to the processor at a time. Up to 241 words of data can be transferred from the processor to the module. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The block identification codes used by the module are listed below:

Block Range	Descriptions
0 or -1	Status Block
1 to 20	Read or write data
1000 to 1019	Read data initialization blocks
3000	Port 0 command disable block
3001	Port 0 command enable block
3002	Port 0 command conditional block
3100	Port 1 command disable block
3101	Port 1 command enable block
3102	Port 1 command conditional block
9998	Warm-boot control block
9999	Cold-boot control block

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

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999 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+1)	Read Data	n

where $n = 60, 120, \text{ or } 240$ 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 three read and two write blocks are used with the application, the sequence will be as follows:

R1W1 → R2W2 → R3W1 → R1W2 → R2W1 → R3W2 → R1W1 →

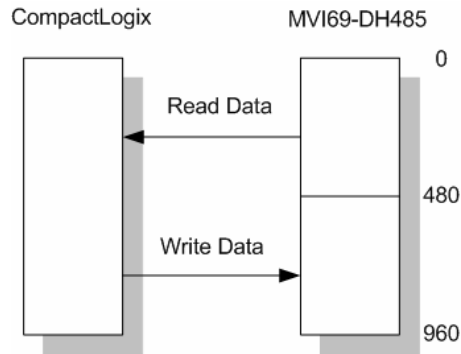
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the DH-485 network or operator control through the module's Configuration/Debug port.

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
```

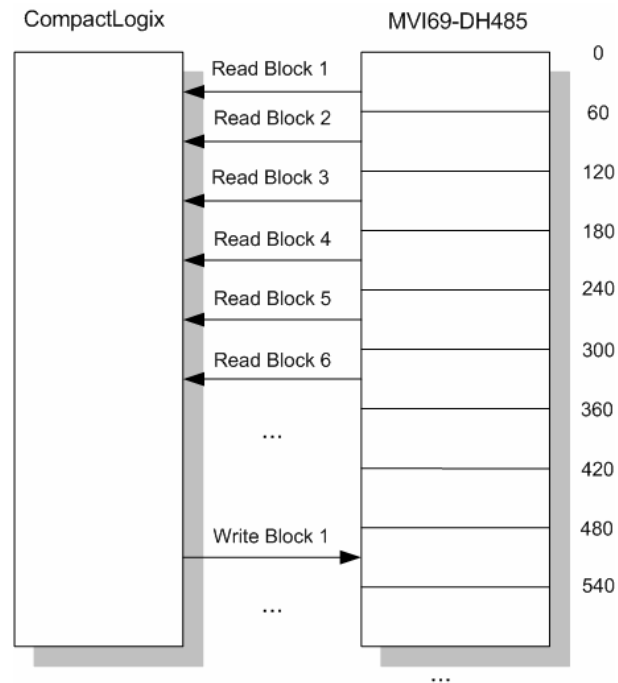
The backplane communication would be configured as follows:



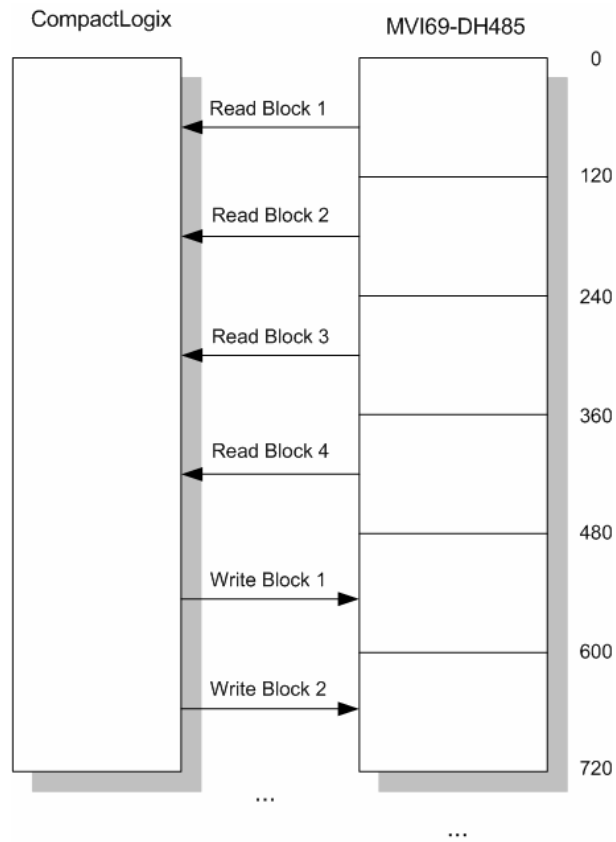
Database address 0 to 479 will be continuously transferred from the module to the processor. Database address 480 to 959 will continuously be transferred from the processor to the module.

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).

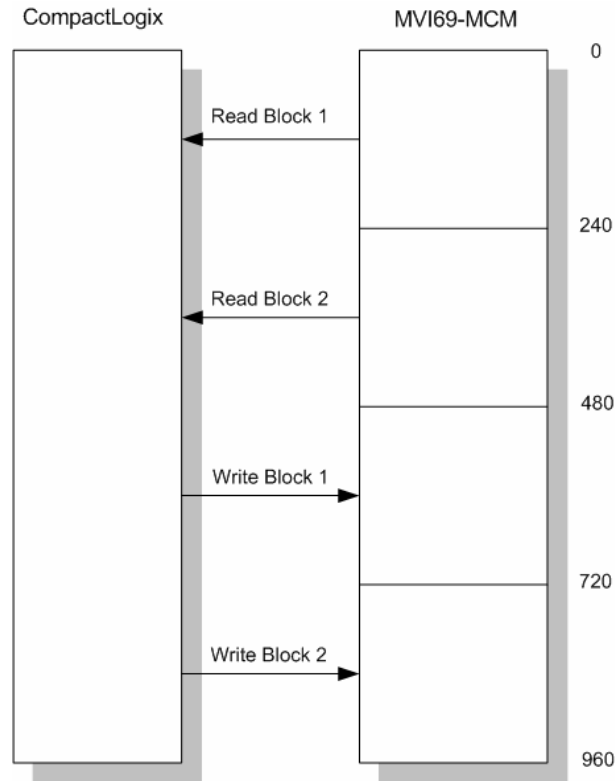
If Block Transfer Size = 60:



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

where $n = 60, 120, \text{ or } 240$ 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.

5.2.3 Command Control Blocks

Command control blocks are special blocks used to control the module or request special data from the module. The current version of the software supports five command control blocks: initialize the read data, change command type control, warm boot and cold boot.

Initialize Output Data

Important: In order to use this feature, you must change the "Initialize Output Data" parameter to "Yes" in the configuration file.

Blocks 1000 to 1019 initialize the read data when the module starts. This is the data passed from the module's internal database to the processor during normal operation. This optional function can be used to set the read data area to the values last read from the module before the first transfer of data from the module to the processor. The module will request this data from the processor using a block with the following format:

Offset	Description	Length
0	Reserved	1
1	1000 to 1019	1
2 to 248	Spare	247
249	1000 to 1019	1

The block number utilized represents the 200-word offset of data requested. Therefore, when the module request block 1000, the processor should return the first 200 words of read data. For block 1001, the processor should return the second 200 words of read data. The format of the response block built by the module is as follows:

Offset	Description	Length
0	1000 to 1019	1
1 to 200	Data to place in output area	200
201 to 247	Spare	47

The number of blocks requested by the module from the processor is dependent on the number of read registers configured for the module. If the module is configured for 600 registers, 3 blocks (1000 to 1002) will be requested. This data will be placed in the module's internal database starting at the read register configured for the module.

Use of this feature requires ladder logic to process the blocks requested. The example ladder logic displays a rung that will handle this optional feature.

Change Command Type Control Block

Blocks 3000, 3001, 3002, 3100, 3101 and 3102 alter the command type for entries in the user command list. When commands are processed during the startup of the module, the command type is set to the value specified in the configuration file. These control blocks alter the configured command type.

Each block contains a command index list. The values entered in this list correspond to the indices of the commands configured by the user. The first command in the user command list has an index of 0 and the second has an index of 1. Therefore, to alter the command type of commands 3 and 5, set the number of command indexes (word 1) to 2 and enter the values 3 and 5 in words 2 and 3. Each block can handle up to 60 commands in the list.

Block 3000 and 3100 requests change the commands listed in the block to the disable type (type = 0). Block 3000 is used for Port 0 and 3100 is used for Port 1 commands. The format for the request block from the ladder logic is as follows:

Offset	Description	Length
0	3000 or 3100	1
1	Number of command indexes	1
2 to 62	Command index list	60
63 to 247	Spare	186

The response block from the module to the processor has the following format:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of commands processed	1
3 to 248	Spare	246
249	3000 or 3100	1

Block 3001 and 3101 requests change the commands listed in the block to the enable type (type = 1). Block 3001 is used for Port 0 and 3101 is used for Port 1 commands. The format for the request block from the ladder logic is as follows:

Offset	Description	Length
0	3001 or 3101	1
1	Number of command indexes	1
2 to 62	Command index list	60
63 to 247	Spare	186

The response block from the module to the processor has the following format:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of commands processed	1
3 to 248	Spare	246
249	3001 or 3101	1

Block 3002 and 3102 requests change the commands listed in the block to the conditional type (type = 2). Block 3002 is used for Port 0 and 3102 is used for Port 1 commands. This type code should only be utilized for write commands. The format for the request block from the ladder logic is as follows:

Offset	Description	Length
0	3002 or 3102	1

Offset	Description	Length
1	Number of command indexes	1
2 to 62	Command index list	60
63 to 247	Spare	186

The response block from the module to the processor has the following format:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of commands processed	1
3 to 248	Spare	246
249	3002 or 3102	1

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 structure of the control block is shown below:

Offset	Description	Length
0	9998	1
1 to n	Spare	247

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

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 structure of the control block is shown below:

Offset	Description	Length
0	9999	1
1 to n	Spare	247

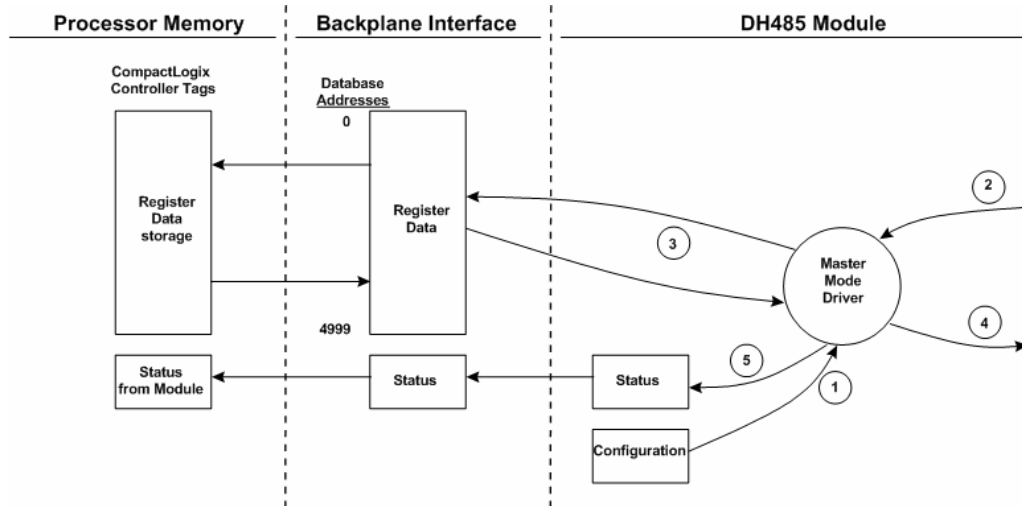
n=60, 120, or 240 depending on what is entered in the Block Transfer Size parameter (refer to the configuration file).

5.2.4 Data Flow between MVI69-DH485 Module and CompactLogix Processor

The following topics describe the flow of data between the two pieces of hardware (CompactLogix processor and MVI69-DH485 module) and other nodes on the DH-485 network under the module's different operating modes. Each port on the module is configured to emulate a DH-485 master device or a DH-485 slave device. The operation of each port is dependent on this configuration. The sections below discuss the operation of each mode.

Slave Driver

The Slave Driver Mode allows the MVI69-DH485 module to respond to data read and write commands issued by a master on the DH-485 network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The DH-485 slave port driver receives the configuration information from the configuration file on the compact flash disk. This information configures the serial port and define the slave node characteristics. The module simulates N-files and CIF to permit remote access of the database.
2	A Host device, such as the Rockwell Automation SLC or an HMI application issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the command, the data is immediately transferred to or from the internal database in the module or the CIF data area. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

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

The slave driver supports the following DH-485 command set:

TYPE	ACCESS	Description
CIF	Read	485CIF, Peer-to-Peer, Read MSG requests
CIF	Write	485CIF, Peer-to-Peer, Write MSG requests
Data Table	Read	500CPU, Peer-to-Peer, Read MSG requests
Data Table	Write	500CPU, Peer-to-Peer, Write MSG requests

The Data table commands require the use of files. These files are emulated in the module. The user configuration of the module defines how these files are emulated in the module. Two file mappings are available. The first sets the first file number and file size for each file. Using this mapping, the files are set as contiguous files overlaying the database. For example, if the user sets the file size to 200 and sets the first file number as 7, the files will overlay the database as shown in the following diagram:

		Database Register
N7:0	----->	0
N8:0	----->	200
N9:0	----->	400
N10:0	----->	600
N11:0	----->	800
N12:0	----->	1000
N13:0	----->	1200
N14:0	----->	1400
N15:0	----->	1600
N16:0	----->	1800
N17:0	----->	2000
N18:0	----->	2200
N19:0	----->	2400
N20:0	----->	2600
N21:0	----->	2800
N22:0	----->	3000
N23:0	----->	3200
N24:0	----->	3400
N25:0	----->	3600
N26:0	----->	3800

In addition to the fixed file emulation, the module also supports user defined mapping of files. With this mapping, the user defines the database offset, file number, the starting element number and the length. This data is entered in the [DH485 Port x Maps] section of the configuration file for each port (x=0 for Port 0 and 1 for Port 1). The following is an example for Port 0:

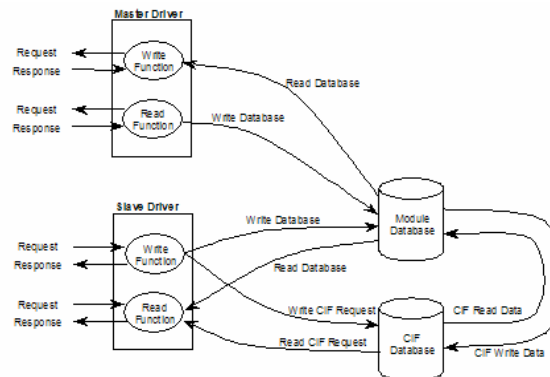
```
[DH485 Port 0 Maps]
# DB_Address   File_Number   Element   Length
START
           0           10           0         100
          100          12          100        100
END
```

With the configuration displayed above, requests for file 10 elements 0 to 99 will be associated with the module's internal data registers 0 to 99 and file 12 elements 100 to 199 will be associated with the data base registers 100 to 199. The module supports up to 50 of these data mappings. When a request is made of the slave driver, the mapping defined in this override section will first be searched first. If the data area in the request is found in the list, that data mapping will be utilized. If the data area in the request is not found in the list, the fixed data area mapping will be used.

5.2.5 CIF Data

The module supports the common interface file (CIF) of the DH-485 protocol. A separate data area can be defined for each application port. This data area is divided into read and write data and is mapped into the module's internal database. The module's application constantly transfers the read CIF data from the CIF database to the module database and the write CIF data from the module to the CIF database. The parameters used to define the CIF database are contained in the configuration file. This optional feature of the module should only be utilized if required by a remote master on the network. Otherwise, the data file functions (500CPU message) instructions should be used to access and control the module's data. The following diagram shows the relationship of the DH-485 drivers, the CIF database and the module's database:

DH-485 network



Important: The user can configure the size of the CIF Read Area using the CIF Read Count Parameter. The size of the CIF Write area is calculated using the following formula:

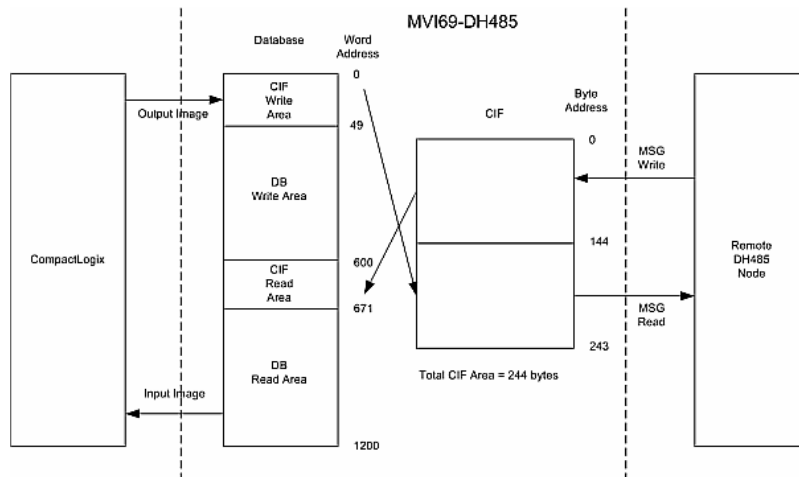
$$\text{CIF Write Count} = 244 - \text{CIF Read Count}.$$

In order to clarify the direction of data flow, the following shows an example:

Parameter	Value	Format
Read Register Start	600	Word
Read Register Count	600	Word
Write Register Start	0	Word
Write Register Count	600	Word

Parameter	Value	Format
CIF Read DB Offset	1200	Byte
CIF Read Count	144	Byte
CIF Write DB Offset	0	Byte

This configuration will imply that the data will be transferred as shown in the following diagram:

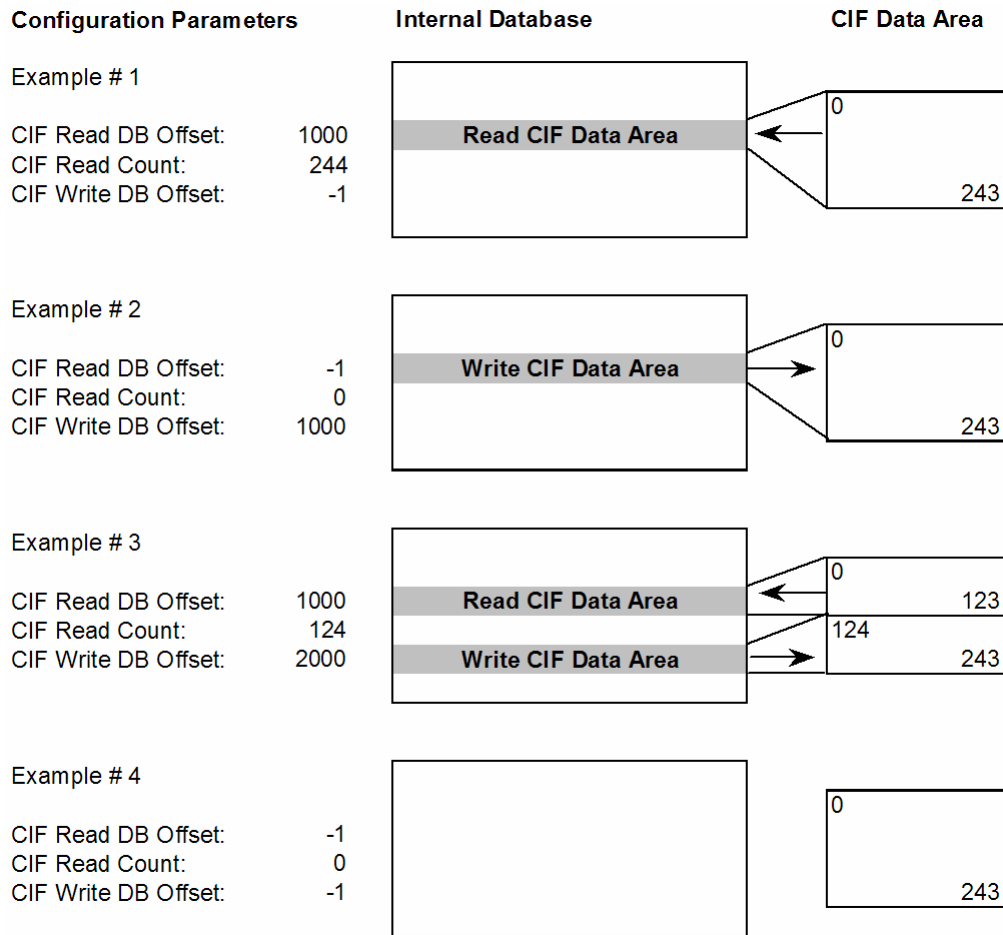


As shown in the diagram, the CIF area is split into the Read and Write areas. The CIF Read area always starts at an offset in the CIF file, except when the CIF Read Count is equal to 0. In this case, the CIF Read Area will not exist.

The diagram shows how the CIF area should interact with the backplane read and write area in order to achieve the correct flow of data.

If a Write MSG from a remote SLC processor is sent to byte addresses 0 and 1 in the module's CIF area, the data would be copied to word address 600 in the database and would then be read to the CompactLogix processor.

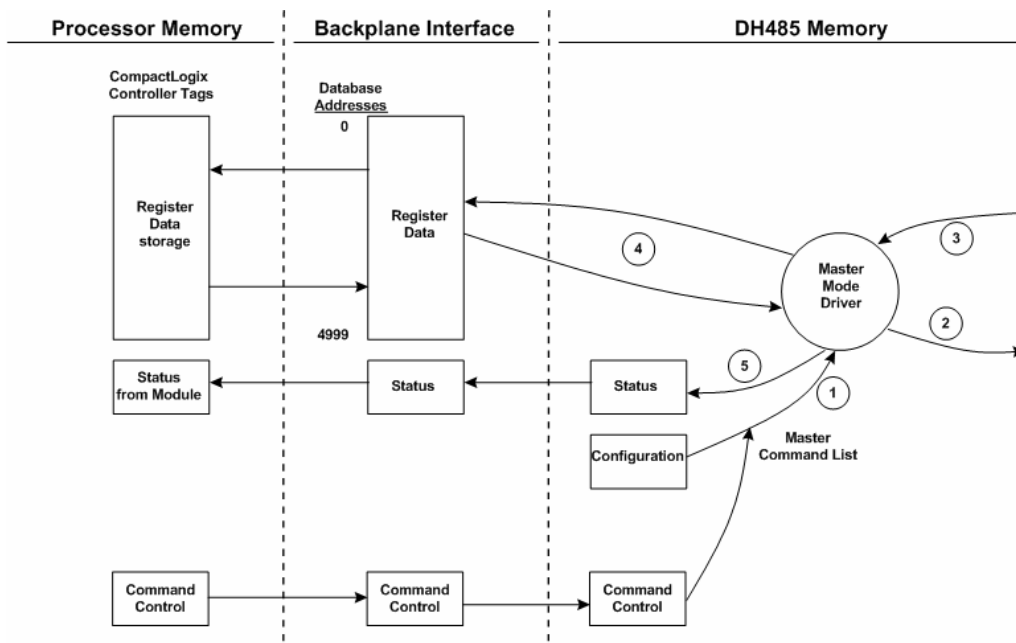
The following illustration shows four different uses and configurations of the CIF data area:



Example 1 only uses CIF read data and utilizes the maximum read data area. Example 2 only uses the CIF write data and utilizes the maximum write data area. Example 3 uses both CIF read and write data. 124 bytes of read data are used and 120 bytes of write data are used. Example 4 does not use the CIF data in the application.

Master Driver Mode

In the Master mode, the MVI69-DH485 module is responsible for issuing read or write commands to slave devices on the DH-485 network. These commands are user-configured in the module via the Master Command List received from the CompactLogix processor or issued directly from the CompactLogix processor (event command control). Command status is returned to the processor for each individual command in the command list status block. The location of this status block in the module's internal database is user defined. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Master driver obtains configuration data from the configuration file on the compact flash disk in the module. This information is used by the Master driver to determine the type of commands to be issued to the other nodes on the DH-485 network.
2	After configuration, the Master driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from the module's internal database to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the Master driver for processing.
4	Data received from the node on the network is passed into the module's internal database, assuming a read command.
5	Status is returned to the CompactLogix processor for each command in the Master Command List.

Refer to the **Module Set Up** section for a complete description of the parameters required to define the virtual DH-485 master port. Refer to the DH-485 **Driver** documentation for a complete discussion of the structure and content of each command. Care must be taken in constructing each command in the list for predictable operation of the module. If two commands write to the same internal database address of the module, the results will not be as desired. All commands containing invalid data are ignored by the module.

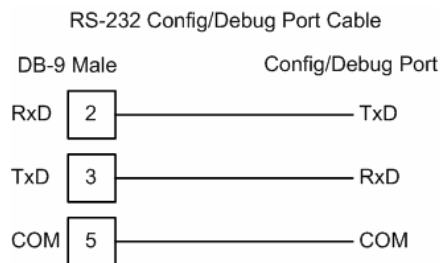
5.3 DH485 Communication Ports

The application ports on the MVI69-DH485 module support RS-232 and RS-485 interfaces. Please ensure that the jumpers on the module are set correctly for the type of interface you are using.

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

5.3.1 RS-232 Configuration/Debug Port

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

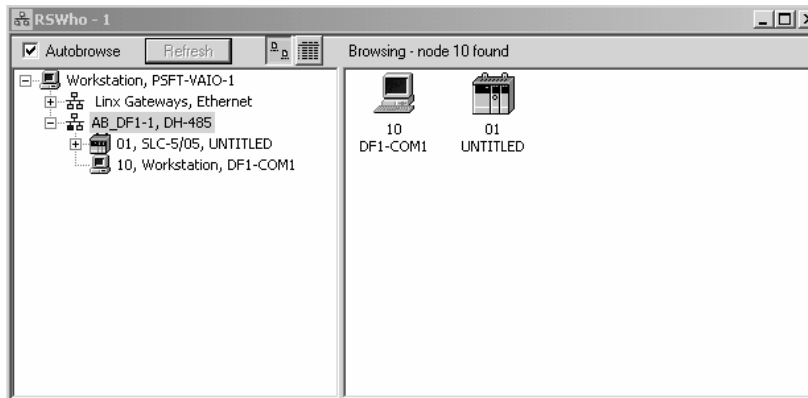


Disabling the RSLinx Driver for the Com Port on the PC

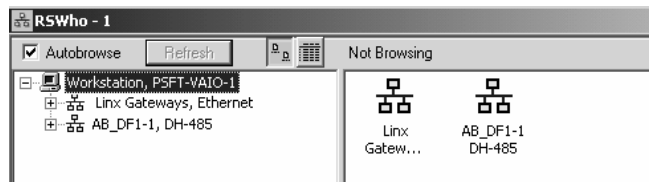
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable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



- 1 Open RSLinx and go to Communications>RSWho

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



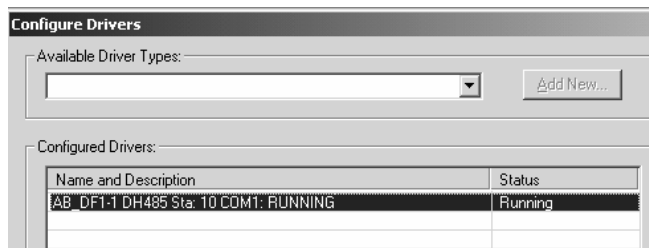
- 3 Notice how the DF1 driver is opened, and the driver is looking for a processor on node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your RSWho screen should look like this:



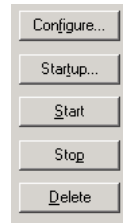
Branches are displayed or hidden by clicking on the  or the  icons.



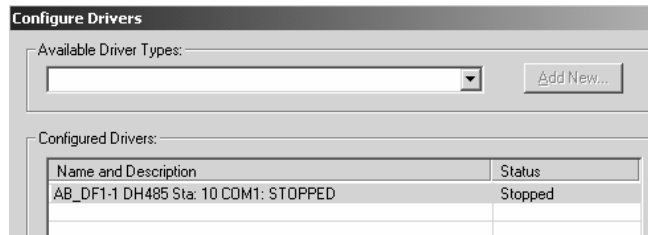
- 4 When you have verified that the driver is not being browsed, go to **Communications>Configure Drivers**
You may see something like this:



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



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



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

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

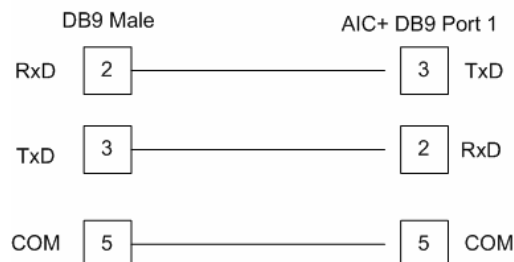
5.3.2 Cable Connections

ProSoft provides a single RJ45 to male DB-9 adapter to permit simpler interfacing to other devices for each port. The module's DH485 ports can be configured to operate in RS-232 or RS-485 mode. The interface to be associated with each port is set with jumpers on the module. The following sections describe each interface.

RS-232

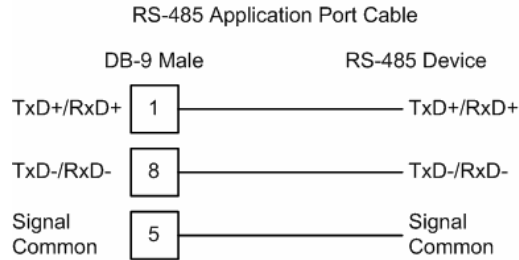
When the RS-232 interface is selected, you must use an AIC+ to connect the port to a DH485. The cable required for this connection is shown in the following illustration:

DH485 Port RS-232 Cable

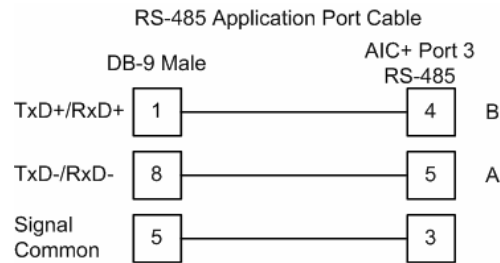


RS-485

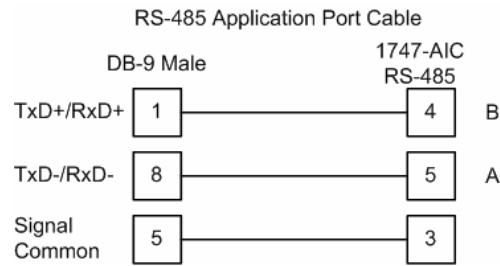
The RS-485 uses a single two or three wire cable. The use of the ground is optional and dependent on the RS-485 network. The cable required for this interface is shown in the following diagram:



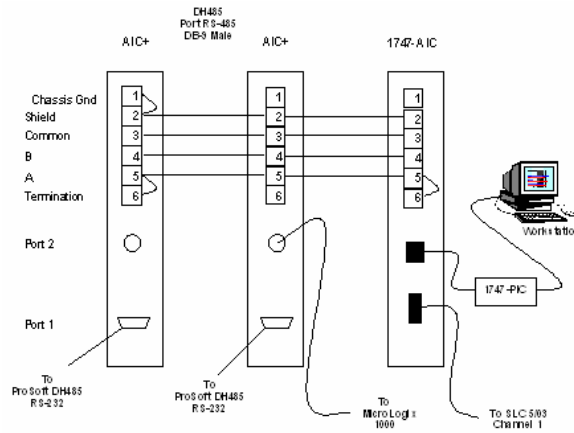
When connecting to port 3 of an AIC+ module, the following is the correct wiring:



When connecting to a 1747-AIC module, the following is the correct wiring:



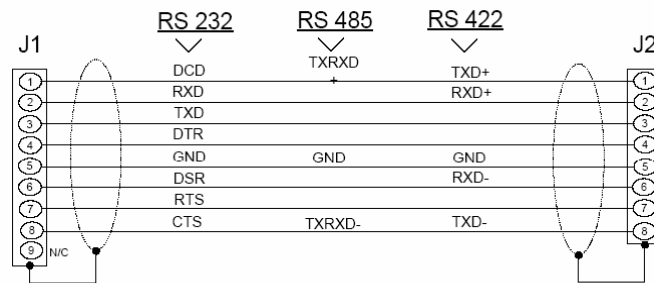
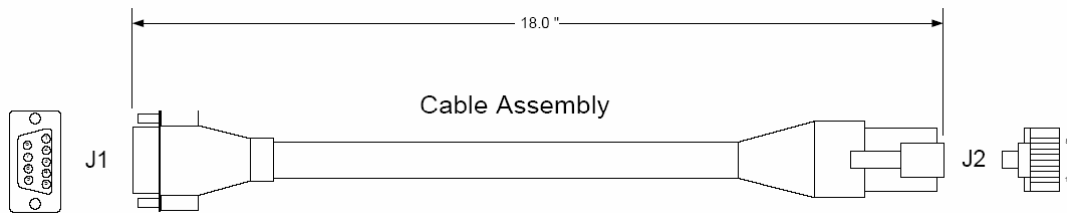
The following illustration shows an example DH-485 network.



This network displays the two different methods to configure the module for a DH-485 network. Please note there is no place on the module's RS-485 to land the shield, and when used in the configuration shown, it must be wired externally. Verify that the RS interface jumper on the module is set to the correct position: RS-232 or RS-485.

Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

5.3.3 DB9 to RJ45 Adaptor (Cable 14)



Wiring Diagram

5.4 Status Data Definition

This section contains a listing status data presented to the processor by the module. The following table lists the data returned with each normal read block:

Word#	Byte#	Description
202		Scan Counter
203 to 204		Product Name (ASCII)
205 to 206		Revision (ASCII)
207 to 208		Operating System Revision (ASCII)
209 to 210		Production Run Number (ASCII)
211		Read Block Count
212		Write Block Count
213		Parse Block Count
214		Reserved
215		Reserved
216		Number of Block Errors

The user can also copy this data block to the module using the Error/Status Pointer parameter in the configuration file.

The following table lists the status data passed from the module to the CompactLogix or MicroLogix processor in blocks 0 and -1.

Note: Word offset can change to any place in the database (0-3950).

Port 0

Word#	Byte#	Description
2 to 3	0 to 3	Active node bits for stations 0 to 31
4	4 to 5	Online status (0=Offline, 1=Online)
5	6 to 7	Node address of the unit/port emulated
6	8 to 9	Current command index being issued
7	10 to 11	Total number of request messages
8	12 to 13	Total number of response messages received
9	14 to 15	Total number of command list errors
10	16 to 17	Configuration error word. Each bit represent a configuration error condition.
11	18 to 19	Reserved for future use.
12	20 to 21	Total number of packets received
13	22 to 23	Total number of packets transmitted
14	24	Total retry count
	25	Retry failure counter
15	26	Total number of NAK's because of no memory for reception
	27	Total number of NAK's because of no memory for transmission
16	28	Total number of bad packets

Word#	Byte#	Description
	29	Total number of bad control packets
17	30	Total number of packets received with a bad CRC value
	31	Total number of parity errors
18	32	Total number of framing errors
	33	Total number of overrun errors
19	34	Total number of unexpected bytes received
	35	Total number of bad LSAP's received
20 to 51		Reserved for future use.

Port 1

Word#	Byte#	Description
52 to 53	0 to 3	Active node bits for stations 0 to 31
54	4 to 5	Online status (0=Offline, 1=Online)
55	6 to 7	Node address of the unit/port emulated
56	8 to 9	Current command index being issued
57	10 to 11	Total number of request messages
58	12 to 13	Total number of response messages received
59	14 to 15	Total number of command list errors
60	16 to 17	Configuration error word. Each bit represent a configuration error condition.
61	18 to 19	Reserved for future use.
62	20 to 21	Total number of packets received
63	22 to 23	Total number of packets transmitted
64	24	Total retry count
	25	Retry failure counter
65	26	Total number of NAK's because of no memory for reception
	27	Total number of NAK's because of no memory for transmission
66	28	Total number of bad packets
	29	Total number of bad control packets
67	30	Total number of packets received with a bad CRC value
	31	Total number of parity errors
68	32	Total number of framing errors
	33	Total number of overrun errors
69	34	Total number of unexpected bytes received
	35	Total number of bad LSAP's received

Common

Word#	Byte#	Description
102 to 201		Reserved for future use.
202		Scan Counter
203 to 204		Product Name (ASCII)
205 to 206		Revision (ASCII)

Word#	Byte#	Description
207 to 208		Operating System Revision (ASCII)
209 to 210		Production Run Number (ASCII)
211		Read Block Count
212		Write Block Count
213		Parse Block Count
214		Reserved
215		Reserved
216		Number of Block Errors

The data in words 10 and 60 (Configuration Error Word) are bit-mapped values with the following definition:

Bit	Code	Description
0	0x0001	Invalid baud rate
1	0x0002	Invalid node address
2	0x0004	Invalid maximum node address
3	0x0008	Invalid token hold factor
4	0x0010	Invalid response timeout
5	0x0020	Invalid status or command error DB offset
6	0x0040	Invalid CIF read count or DB offset
7	0x0080	Invalid CIF write DB offset
8	0x0100	Invalid file size
9	0x0200	Invalid file offset
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

When no configuration errors are present, the words will have a value of 0x0000 (hexadecimal). Configuration errors should be corrected to have the module perform as required by the application.

Additionally, each command in the user command list contains an error status data area. This data can be viewed through the debug/configuration port. This data can be configured to be placed in the module's database with each register containing an error value for each command for each port as shown in the following diagram:

Word#	Description
0	Error code for command index 0.
1	Error code for command index 1.
...	...
99	Error code for command index 99.

The error codes placed in this data area have the following definitions:

5.4.1 DH485 General Error Codes

Error #	Description
0	Operation successful
1	Invalid parameter
2	Device is already open
3	Device is not present
4	Invalid access
5	The function has timed out
6	
7	Unable to configure the requested port
8	Unable to allocate memory for DH-485 driver

5.4.2 DH485 API Specific Error Codes

Error #	Description
0x0800	Command only permitted in master mode
0x0801	Command already active on the port
0x0802	Response to request timed out
0x0803	Unable to allocate memory for the request
0x0804	Illegal command or format
0x0805	Host could not complete request (hardware fault)
0x0806	Out of memory, file or rung does not exist
0x0807	Field has an illegal value
0x0808	Not enough fields in request message
0x0809	Too many fields in request message
0x080A	Symbol not found
0x080B	Symbol 0 or greater than maximum characters permitted in message
0x080C	Does not exist, illegal size
0x080D	File wrong size, address past end of file
0x080E	Data or file too large (memory not available)
0x080F	Request too large to transmit message (size+address > max message)
0x0810	Access denied
0x0811	Command cannot be executed
0x0812	Illegal data type information
0x0813	Illegal parameter, invalid data in search or command block
0x0814	File open by another node
0x0815	Program owned by another node
0x0816	Unknown error returned from host
0x0817	No message active on the port

6 Support, Service & Warranty

In This Chapter

- ❖ How to Contact Us: Technical Support..... 91
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 92
- ❖ LIMITED WARRANTY..... 94

ProSoft Technology, Inc. (ProSoft) 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 contents of file
 - Module Operation
 - Configuration/Debug status information
 - LED patterns
- 2 Information about the processor and user data files as viewed through and LED patterns on the processor.
- 3 Details about the serial devices interfaced, if any.

6.1 How to Contact Us: Technical Support

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

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com
 (<mailto:support.asia@prosoft-technology.com>)

Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com
 (<mailto:support.emea@prosoft-technology.com>)

Languages spoken include: French, English

North America/Latin America (excluding Brasil) (location in California)

+1.661.716.5100, support@prosoft-technology.com (mailto:support@prosoft-technology.com)

Languages spoken include: English, Spanish

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

Brasil (location in Sao Paulo)

+55-11-5084-5178 , eduardo@prosoft-technology.com (mailto:eduardo@prosoft-technology.com)

Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

The following RMA Policies and Conditions (collectively, "RMA Policies") apply to any returned Product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see "Limited Warranty". In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

6.2.1 All Product Returns:

- a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
- b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 91). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
- c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, etc.

6.2.2 Procedures for Return of Units Under Warranty:

A Technical Support Engineer must approve the return of Product under ProSoft's Warranty:

- a) A replacement module will be shipped and invoiced. A purchase order will be required.
- b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

6.2.3 Procedures for Return of Units Out of Warranty:

- a) Customer sends unit in for evaluation
- b) If no defect is found, Customer will be charged the equivalent of \$100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.
- c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

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

6.2.4 Purchasing Warranty Extension:

- a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 94)". The Warranty Period may be extended at the time of equipment purchase for an additional charge, as follows:
 - Additional 1 year = 10% of list price
 - Additional 2 years = 20% of list price
 - Additional 3 years = 30% of list price

6.3 LIMITED WARRANTY

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

6.3.1 *What Is Covered By This Warranty*

- a) *Warranty On New Products:* ProSoft warrants, to the original purchaser, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires three years from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 39 months. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft.
- b) *Warranty On Services:* Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranted in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.3.2 *What Is Not Covered By This Warranty*

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

- b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.
- c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

Product manufactured or supplied by ProSoft is not fault tolerant and is not designed, manufactured or intended for use in hazardous environments requiring fail-safe performance including and without limitation: the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly or indirectly to death, personal injury or severe physical or environmental damage (collectively, "high risk activities"). ProSoft specifically disclaims any express or implied warranty of fitness for high risk activities.

6.3.4 Intellectual Property Indemnity

Buyer shall indemnify and hold harmless ProSoft and its employees from and against all liabilities, losses, claims, costs and expenses (including attorney's fees and expenses) related to any claim, investigation, litigation or proceeding (whether or not ProSoft is a party) which arises or is alleged to arise from Buyer's acts or omissions under these Terms or in any way with respect to the Products. Without limiting the foregoing, Buyer (at its own expense) shall indemnify and hold harmless ProSoft and defend or settle any action brought against such Companies to the extent based on a claim that any Product made to Buyer specifications infringed intellectual property rights of another party. ProSoft makes no warranty that the product is or will be delivered free of any person's claiming of patent, trademark, or similar infringement. The Buyer assumes all risks (including the risk of suit) that the product or any use of the product will infringe existing or subsequently issued patents, trademarks, or copyrights.

- a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated or reproduced in any form without prior written consent from ProSoft.
- b) ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
- c) Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
- d) Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.
- e) Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.
- f) **Additional Restrictions Relating To Software And Other Intellectual Property**

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

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 94) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

6.3.6 Limitation of Remedies **

In no event will ProSoft or its Dealer be liable for any special, incidental or consequential damages based on breach of warranty, breach of contract, negligence, strict tort or any other legal theory. Damages that ProSoft or its Dealer will not be responsible for included, but are not limited to: Loss of profits; loss of savings or revenue; loss of use of the product or any associated equipment; loss of data; cost of capital; cost of any substitute equipment, facilities, or services; downtime; the claims of third parties including, customers of the Purchaser; and, injury to property.

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

6.3.7 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 39 months following shipment of the Product.

6.3.8 No Other Warranties

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

6.3.9 Allocation of Risks

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

6.3.10 Controlling Law and Severability

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

Index

[

[DH485 Port x Commands] • 30
 [DH485 Port x Maps] • 33
 [DH485 Port x] • 28
 [Module] • 26

A

Adding a Module • 23
 Adding a Project • 23
 Adding the Module to an Existing CompactLogix Project • 45
 Adding the Module to an Existing MicroLogix Project • 49
 All Product Returns: • 92
 Allocation of Risks • 97

B

Backplane Data Transfer • 66
 Battery Life Advisory • 2
 Baud Rate • 28
 Block Transfer Size • 27

C

Cable Connections • 83
 Change Command Type Control Block • 72
 CIF Data • 77
 CIF Read Count • 29
 CIF Read DB Offset • 29
 CIF Write DB Offset • 29
 Clearing a Fault Condition • 62
 Cold Boot • 74
 Command Control Blocks • 72
 Command Entry Format • 31
 Command Error DB Offset • 29
 Command List Overview • 30
 Comment Entries • 25
 Configuring RSLinx • 18
 Configuring the MVI69-DH485 Module • 21
 Connect your PC to the Module • 20
 Connect your PC to the Processor • 16
 Control Data Type • 35
 Controlling Law and Severability • 97
 Counter Data Type • 35

D

Data Flow between MVI69-DH485 Module and CompactLogix Processor • 74
 Database Start Address • 31, 34
 Database View Menu • 56
 DB9 to RJ45 Adaptor (Cable 14) • 85
 DH485 API Specific Error Codes • 89
 DH485 Communication Ports • 81

DH485 General Error Codes • 89
 DH485 Port x Menu • 58
 Diagnostics and Troubleshooting • 7, 50, 51
 Disabling the RSLinx Driver for the Com Port on the PC • 81
 Disclaimer of all Other Warranties • 96
 Disclaimer Regarding High Risk Activities • 95
 Displaying the Current Page of Registers Again • 57
 Download the Project to the Module • 38
 Download the Sample Program to the Processor • 17

E

Element • 34
 Element Count • 31
 Element Number • 33
 Enable Type Code • 31
 Error / Status Pointer • 26
 Exiting the Program • 56

F

Failure Flag Count • 27
 File Number • 33, 34
 File Offset • 30
 File Override Entry Format • 34
 File Override Mapping • 34
 File Size • 30
 File Type • 32
 First File • 30
 Function Code • 32
 Functional Overview • 7, 65
 Functional Specifications • 64

G

General Concepts • 65
 General Specifications • 63
 Guide to the MVI69-DH485 User Manual • 7

H

Hardware Specifications • 64
 How to Contact Us
 Technical Support • 91, 92

I

Initialize Output Data • 27, 72
 Install ProSoft Configuration Builder Software • 11
 Install the Module in the Rack • 13
 Intellectual Property Indemnity • 95

K

Keystrokes • 54

L

Ladder Logic • 39
 LED Status Indicators • 7, 61
 Limitation of Remedies ** • 96
 LIMITED WARRANTY • 93, 94

M

Main Logic Loop • 65
Main Menu • 54
Master Driver Mode • 80
Maximum Element Counts for Read/Write Commands • 33
Maximum Node Address • 28
Min Command Delay • 30
Module Data Object (DH485ModuleDef) • 40
Module Entries • 25
Module Name • 26
Module Power Up • 65
Moving Back Through 5 Pages of Registers • 57

N

Navigation • 53
No Other Warranties • 97
Node Address • 28
Node Number • 32
Normal Data Transfer • 68

O

Opening the Command Error List Menu • 58
Opening the Command List Menu • 58, 60
Opening the Database Menu • 55
Opening the DH-485 Port x Menu • 55
Opening the Override File Map List Menu • 58
Opening the Serial Port Menu • 61

P

Package Contents • 10
Pinouts • 83, 84, 85
Please Read This Notice • 2
Poll Interval • 32
Printing a Configuration File • 25
Procedures for Return of Units Out of Warranty: • 93
Procedures for Return of Units Under Warranty: • 93
Product Specifications • 7, 63
ProSoft Configuration Builder • 21
ProSoft® Product Documentation • 3
Protocol Serial DH485 Menu • 60
Purchasing Warranty Extension: • 93

R

Read Block • 68
Read Register Count • 27
Read Register Start • 26
Reading Status Data from the Module • 51
Redisplaying the Menu • 60
Reference • 7, 63
Required Hardware • 51
Response Timeout • 29
Return Material Authorization (RMA) Policies and Conditions • 92
Returning to the Main Menu • 58, 59, 60
RS-232 • 83
RS-232 Configuration/Debug Port • 81
RS-485 • 84

S

Set Module Parameters • 24
Set Up the Project • 22
Setting Jumpers • 12
Skipping 500 Registers of Data • 57
Slave Driver • 75
Start Here • 7, 9
Status Data Definition • 43, 86
Status DB Offset • 29
Status Object (DH485InStat) • 41
Support, Service & Warranty • 7, 91
Swap Code • 32
System Requirements • 9

T

The Configuration/Debug Menu • 52
Time Limit for Bringing Suit • 97
Timer Data Type • 35
Timer, Counter, and Control Data Types • 35
Token Hold Factor • 28
Transferring the Configuration File from MVI69 module to PC • 55
Transferring the Configuration File from PC to MVI69 module • 55
Troubleshooting • 62

U

User Data Objects • 43
Using the Diagnostic Window in ProSoft Configuration Builder • 52

V

Viewing Block Transfer Statistics • 54
Viewing Configuration Information • 60
Viewing Data in ASCII (Text) Format • 58
Viewing Data in Decimal Format • 57
Viewing Data in Floating Point Format • 58
Viewing Data in Hexadecimal Format • 57
Viewing Error and Status Data • 61
Viewing Module Configuration • 55
Viewing Port Configuration • 58
Viewing Port Status Data • 59
Viewing Register Pages • 57
Viewing the Next 100 Registers of Data • 57
Viewing the Previous 100 Registers of Data • 57
Viewing Version Information • 59, 60

W

Warm Boot • 74
Warm Booting the Module • 55
What Is Covered By This Warranty • 94, 96
What Is Not Covered By This Warranty • 94
Word Count (Length) • 34
Write Block • 71
Write Register Count • 27
Write Register Start • 27

Y

Your Feedback Please • 2