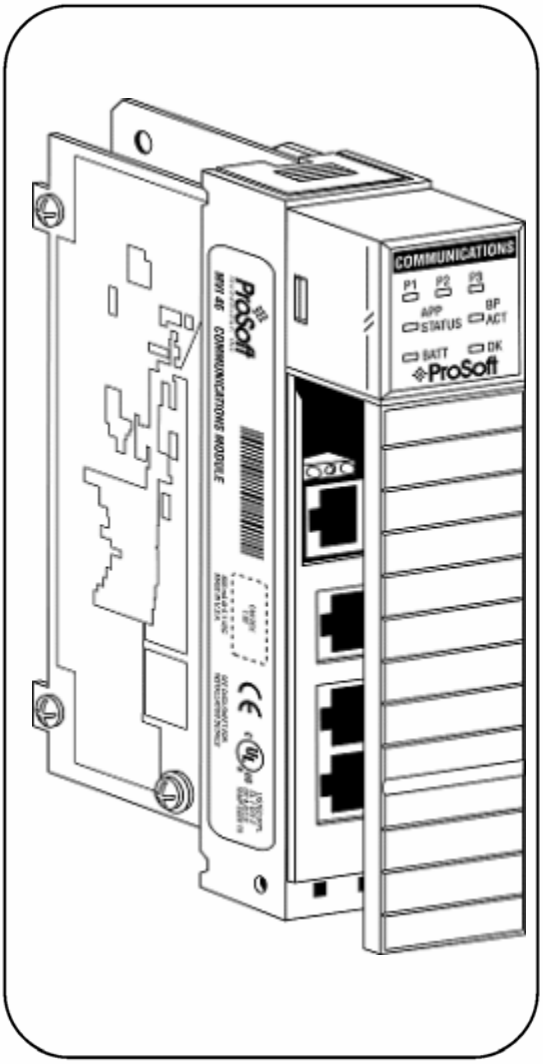
inRAx



MVI46-BDW

SLC Platform

BARDAC Drive Web Interface

User Manual

August 24, 2006

ProSoft TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation SLC hardware, the MVI46-BDW 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 assure that the information provided is accurate and a true reflection of the product's installation requirements. In order to assure 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, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

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

Your Feedback Please

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

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

MVI46-BDW User Manual
August 24, 2006

Contents

1 Product Specifications

In This Chapter

The MVI46-BDW ("BARDAC Drive Web Interface Module") allows Rockwell Automation SLC I/O compatible processors to interface easily with other BARDAC devices.

1.1 General Specifications

The MVI46-BDW module acts as a client collecting and controlling data in a BARDAC-DW compatible device (Four independent clients are contained in the module to interface with BARDAC drives simultaneously. The data is passed from the network to the Rockwell Automation backplane. The data transfer from the SLC processor is asynchronous from the actions on the UDP/IP network. A 5000-word register space in the module exchanges data between the processor and the network.

Some of the general specifications include:

- Support for the storage and transfer of up to 5000 registers to/from the SLC processor's controller tags
- Module memory usage that is completely user-definable
- Ability for the user to define commands to set or get parameters from the BARDAC device
- 10/100 MB Ethernet compatible interface

1.1.1 Client Specifications

A client configured as a BARDAC master device on the MVI46-BDW module will actively issue BARDAC-DW commands to other nodes on the BARDAC-DW network. One hundred commands are supported on each port. The SLC processor can be programmed to control the activity on the client by actively selecting commands from the command list to execute or issuing commands directly from the ladder logic.

1.1.2 Physical

This module is designed by ProSoft Technology and incorporates licensed technology from Rockwell Automation (SLC backplane technology).

- SLC Form Factor - Single Slot
- Connections:
- 1 - RJ45 connector for Ethernet interface
- 1 - RJ45 RS-232 Configuration Tool Connector

1.1.3 SLC Interface

- Operation via simple ladder logic
- Complete set up and monitoring of module through RSLogix 500 software and user constructed configuration file (BARDACDW.CFG)
- SLC backplane interface via I/O access

1.2 Hardware Specifications

The MVI46-BDW module is designed by ProSoft Technology and incorporates licensed technology from Rockwell Automation (SLC backplane technology).

- Current Loads: 800 mA @ 5V (from backplane)
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: –40 to 85°C (–40 to 185°F)
- Relative Humidity: 5 to 95% (non-condensing)
- Ethernet Connector: One RJ45 Connector
- Configuration Connector: RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)

2 Functional Overview

In This Chapter

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

2.1 General Concepts

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

2.1.1 Module Power Up

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

1 Initialize hardware components

2 Initialize SLC backplane driver

3 Test and clear all RAM

4 Read configuration for module from BARDACDW.CFG file on Compact Flash Disk

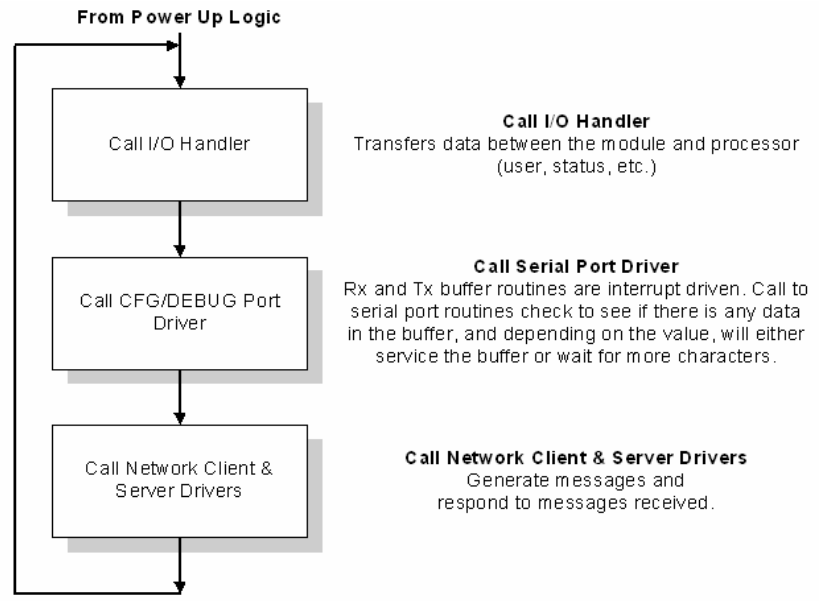
5 Initialize Module Register space

6 Enable Client Driver

After the module has received the configuration, the module will begin communicating with other nodes on the network, depending on the configuration.

2.1.2 Main Logic Loop

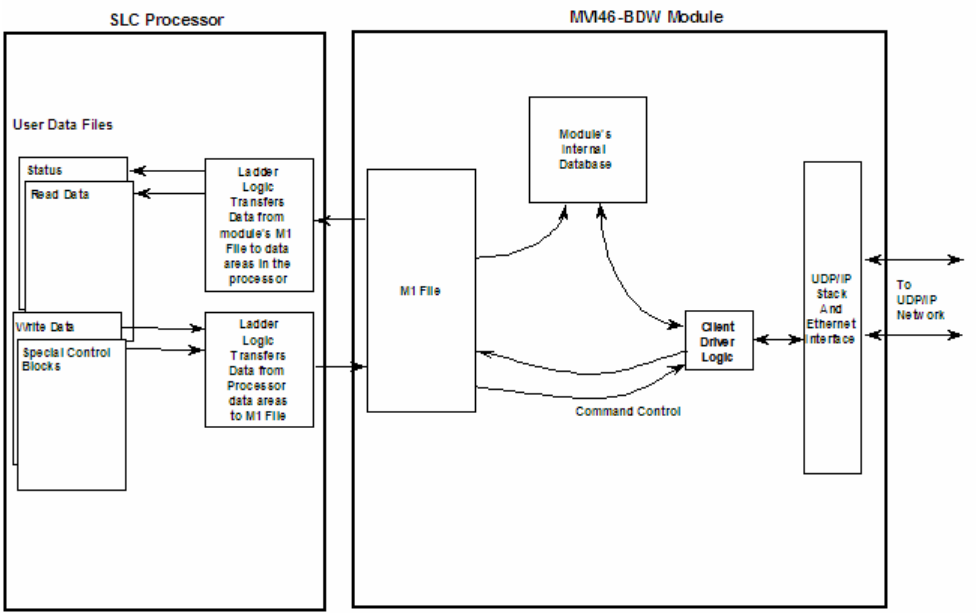
Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.



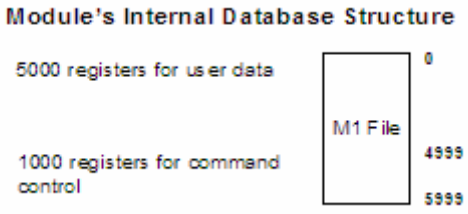
2.1.3 Backplane Data Transfer

The MVI46-BDW module is unique in the way it utilizes the SLC backplane. All data for the module is contained in the module's M1 file. Data is moved between the module and the SLC processor across the backplane using the module's M1 file. The SLC scan rate and the communication load on the module determine the update frequency of the M1 file. The COP instruction can be used to move data between user data files and the module's M1 file.

The following illustration shows the data transfer method used to move data between the SLC processor, the MVI46-BDW module, and the BARDAC-DW Network.



As shown in the diagram above, all data transferred between the module and the processor over the backplane is through the M1 file. Ladder logic must be written in the SLC processor to interface the M1 file data in the module's internal database. All data used by the module is stored in its internal database. This database is defined as a virtual data table with addresses from 0 (40001) to 4999 (45000). The following illustration shows the layout of the database:



Data registers in the module above 4999 are used for command control. When special values are written in this register set, the module will perform specific functions. The following sections define the special functions handled by the module.

Initialize Output Data

When the module performs a restart operation, it will request output data from the processor to initialize the module's output data. Use the **Initialize Output**

Data parameter in the configuration file to bring the module to a known state after a restart operation. The structure of the block used to request the data is displayed in the following table:

Offset	Description/Value	Length
5000	1000	1

The command control value of 1000 is placed in register 5000 of the M2 file to indicate that the module is requesting initialization of the M1 data file. Ladder logic in the processor must recognize this command and place the correct information in the M1file. After the data transfer is complete, the ladder logic should place a value of 1001 in register 5000 of the module's M1 file. The format of the returned write block is shown below:

Offset	Description/Value	Length
5000	1001	1

Command Control Blocks

Command control blocks are special blocks used to control the module. The current version of the software supports four command control blocks: event command control, command control, warm boot and cold boot. Register 5000 of the module's M1 file is used for this feature. The following table lists the command control block numbers recognized by the module.

Block Range	Descriptions
1000	Output Initialization Request from Module
1001	Output Initialization Complete
2000	Event Command
5001 - 5006	Command Control
9998	Warm Boot Control Block
9999	Cold Boot Control Block

Each of the command control blocks are discussed in the following sections.

Event Command

Event command control blocks send BARDAC-DW commands directly from the ladder logic to one of the clients on the module. The format for these blocks is displayed in the following table:

Offset	Description/Value	Length
5000	2000	1
5001 - 5004	IP Address	4
5005	Internal DB Address	1
5006	Swap Code	1
5007	BARDAC DW Function Code	1
5008	Device Database Address	1
5009	Client Number (0-3)	1

The parameters passed with the block construct the command.

The **IP Address** for the node to reach on the network is entered in four registers (1 to 4). Each digit of the IP address is entered in the appropriate register. For example, to interface with node 192.168.0.100, enter the values 192, 168, 0 and 100 in registers 1 to 4.

The **Internal DB Address** parameter specifies the module's database location to associate with the command.

The **Swap Code** is used with functions 3 and 4 requests to change the word or byte order.

The **BARDAC DW Function Code** has one of the following values 1, 2, 3, 4, 5, 6, 15 or 16.

The **Device Database Address** is the register or point in the remote slave device to be associated with the command.

The **Client Number** is the designation of the specified client.

When the module receives the block, it will process it and place it in the command queue. A detailed description of the block is presented in the following table:

Word	Description
5000	This word contains the block 2000 identification code to indicate that this block contains a command to execute by the client driver.
5001 to 5004	These words contain the IP address for the server the message is intended for. Each digit (0 to 255) of the IP address is placed in one of the four registers. For example, to reach IP address 192.168.0.100, enter the following values in words 1 to 4…192, 168, 0, and 100. The module will construct the normal dotted IP address from the values entered. The values entered will be added with the mask 0x00ff to ensure the values are in the range of 0 to 255.
5005	This word contains the internal database address in the module to be used with the command. This word can contain a value from 0 to 4999.
5006	The parameter specifies the swap type for the data; 0 = Do not swap bytes, 1 = swap bytes.
5007	This word contains the BARDAC DW function code to be used with the command; 0 = Get function to retrieve data from unit and 1 = Set function to alter data in the unit.
5008	This word contains the PIN value for the data value or parameter to consider with the command. Please consult the drive manual for a complete list of all PIN values and their meaning.
5009	Client index used with the block. This parameter has a value from 0 to 3.

The module will respond to each event command block with a read block with the following format:

Word	Description
5000	Set to 0 on completion.

Word	Description
5001	This word contains the block identification code 2000 requested by the processor.
5002	This word contains the result of the event request. If a value of one is present, the command was issued. If a value of zero is present, no room was found in the command queue.
5003	Client index used with the block. This parameter has a value from 0 to 3.

Word two of the block can be used by the ladder logic to determine if the command was added to the command queue of the module. The command will only fail if the command queue for the port is full (100 commands for each queue).

Note: Because the BARDAC-DW devices use the UDP protocol, it is suggested that you use an additional GET command to verify if the value was accepted by the device (when a SET command is used).

Command Control

Command control blocks place commands from the command list into the command queue. The client has a command list of up to 100 commands. The module services commands in the queue before the user defined command list. This gives high priority to commands in the queue. Commands placed in the queue through this mechanism must be defined in the module's command list. Under normal command list execution, the module will only execute commands with the Enable parameter set to one or two. If the value is set to zero, the command is skipped. Commands may be placed in the command queue with an Enable parameter set to zero using this feature. These commands can then be executed using the command control blocks.

One to six commands can be placed in the command queue with a single request. The following table describes the format for this block.

Word	Description
5000	Command queue block identification code of 5001 to 5006.
5001	This word contains the index in the command list from the first command to be entered into the command queue.
5002	This word contains the index in the command list for the second command to be entered into the command queue.
5003	This word contains the index in the command list for the third command to be entered into the command queue.
5004	This word contains the index in the command list for the fourth command to be entered into the command queue.
5005	This word contains the index in the command list for the fifth command to be entered into the command queue.
5006	This word contains the index in the command list for the sixth command to be entered into the command queue.
5007	Client index used with the block. This parameter has a value from 0 to 3.

The last digit in the block code defines the number of commands to process in the block. For example, a block code of 5003 contains 3 command indexes that are to be placed in the command queue. The Command index parameters in the block have a range of 0 to 99 and correspond to the module's command list entries.

The module responds to a command control block with a block containing the number of commands added to the command queue for the port. The following table describes the format for this block.

Word	Description
5000	Set to 0 on completion.
5001	This word contains the block 5001 to 5006 requested by the processor.
5002	This word contains the number of commands in the block placed in the command queue.
5003	Client index used with the block. This parameter has a value from 0 to 3.

Note: Because the BARDAC-DW devices use the UDP protocol, it is suggested that you use an additional GET command to verify if the value was accepted by the device (when a SET command is used).

Warm Boot

This block is sent from the SLC processor to the module when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made in the controller tags data area. This will force the module to read the new configuration information and to restart. The structure of the control block is shown in the following table:

Offset	Description/Value	Length
5000	9998	1

Cold Boot

This block is sent from the SLC processor to the module 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 in the following table:

Offset	Description/Value	Length
5000	9999	1

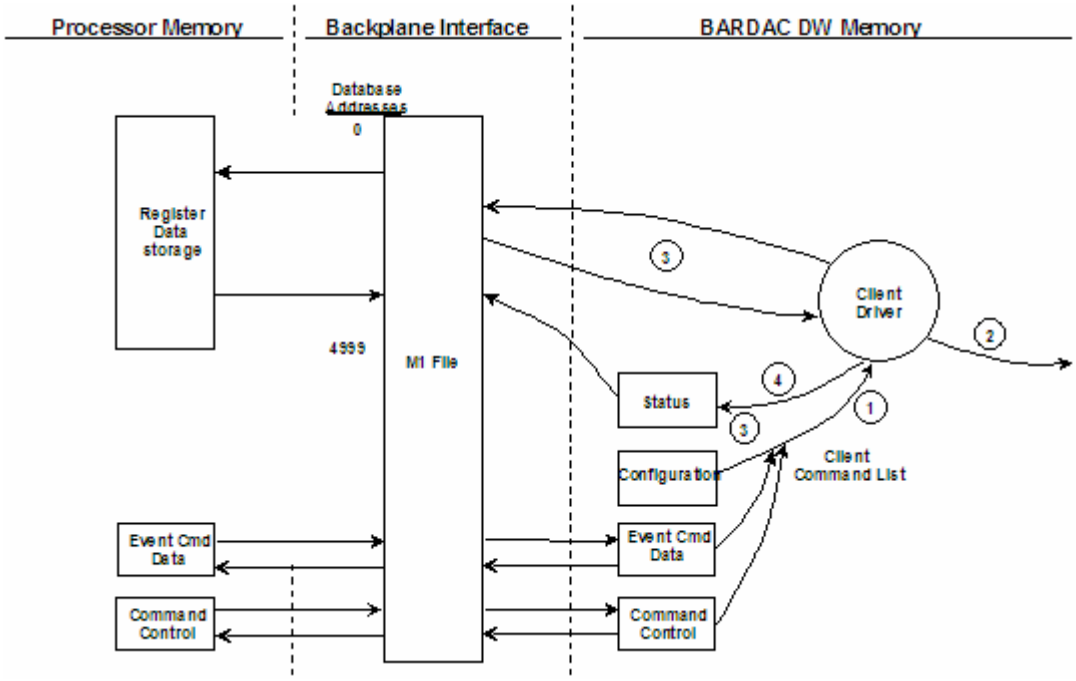
2.2 Data Flow between MVI46-BDW Module and SLC Processor

The following discussion describes the flow of data between the two pieces of hardware (SLC processor and MVI46-BDW module) and other nodes on the network under the module's different operating modes. The module contains four clients. The client can generate UDP requests to the BARDAC-DW device.

The following sections discuss the operation of the client drivers.

2.2.1 Client Driver

In the client driver, the MVI46-BDW module is responsible for issuing read or write commands to servers on the network. These commands are user configured in the module via the Client Command List received from the module's configuration file (BARDACDW.CFG) or issued directly from the SLC 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 client driver obtains configuration data from the BARDACDW.CFG file when the module restarts. The configuration data obtained includes the timeout parameters and the Command List. These values are used by the driver to determine the type of commands to be issued to the other nodes on the (see Module Configuration) network.

Step	Description
2	After configuration, the client 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	Data received from the node on the network is passed into the module's internal database, assuming a read command.
4	Status data is returned to the SLC processor for the client and a Command List error table can be established in the module's internal database.

Client Command List

In order for the client to function, the module's Client Command List must be defined. This list contains up to 100 individual entries, with each entry containing the information required to construct a valid command. This includes the following:

- Command enable mode ((0) disabled, (1) continuous or (2) conditional)
- IP address and service port to connect to on the remote server
- Slave Node Address
- Command Type - Read or Write up to 100 words per command
- Database Source and Destination Register Address - Determines where data will be placed and/or obtained
- Count - Select the number of words to be transferred - 1 to 100
- Poll Delay ($1/10^{th}$ second)

Standard BARDAC DW PL/X Series Errors

Code	Description
0xA501	GET to invalid PIN
0xA702	SET to invalid PIN
0xA703	SET to read-only PIN
0xA704	SET to "restricted in run" PIN
0xA705	SET with out-of-range data
0xyy06	Setup source with invalid PIN
0xyy07	Setup source with invalid index
0xyy08	Setup destination with invalid PIN
0xyy09	Setup destination with invalid index
0xFFFD	Device not ready (initializing)
0xFFFE	Device lock timeout (serious problem with device)
0xFFFF	Unable to obtain device lock (device servicing other transaction)
Others	Please consult drive manual or contact Bardac.

Standard BARDAC DW E-Series Errors

Code	Description
1	Illegal function
2	Illegal data address
3	Illegal data value
6	Busy, reject
7	Negative acknowledge
10	Read-only variable
11	Index out of range
12	Invalid data type
13	Invalid variable address
14	Invalid modbus address
15	Connection not setup
16	Write-only variable
17	EEPROM write fail
0xFFFD	Device not ready (initializing)
0xFFFE	Device lock timeout (serious problem with device)
0xFFFF	Unable to obtain device lock (device servicing other transaction)
Others	Please consult drive manual or contact Bardac.

UDP Error Codes

Code	Description
0xFFDF	Could not find IP on network
0xFFDE	Buffer for UDP socket could not be allocated
0xFFDC	Command response timeout

Command List Entry Errors

Code	Description
40	Too few parameters for command
41	Invalid type code
42	Invalid internal database address
45	Invalid function code
46	Invalid swap code

3 Module Configuration

In This Chapter

This section contains the setup procedure, data, and ladder logic for successful application of the MVI46-BDW module. Each step in the setup procedure is defined in order to simplify the use of the module.

3.1 Installing and Configuring the Module

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

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

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

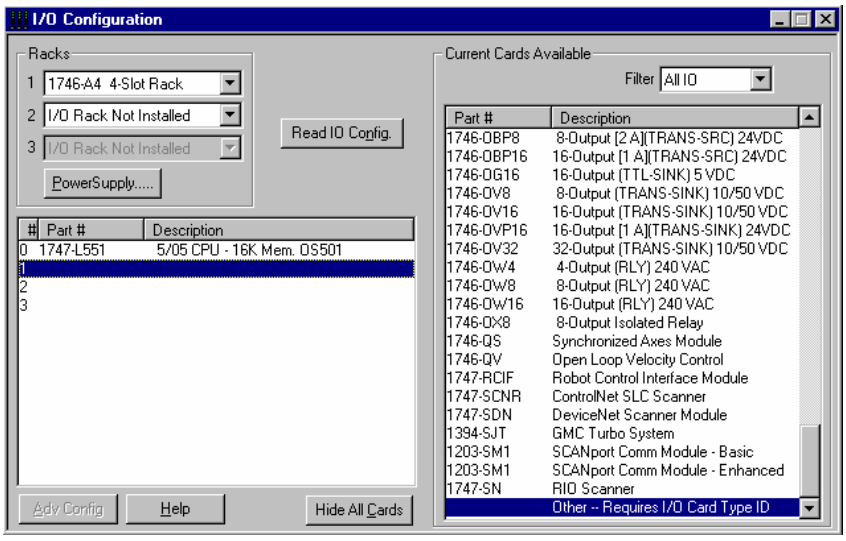
2 Modify the module's configuration files to meet the needs of your application, and copy the updated configuration to the module. Example configuration files are provided on the CD-ROM.

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

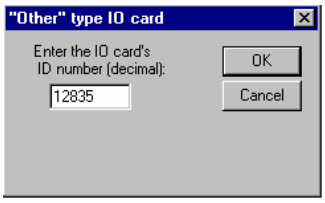
Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

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

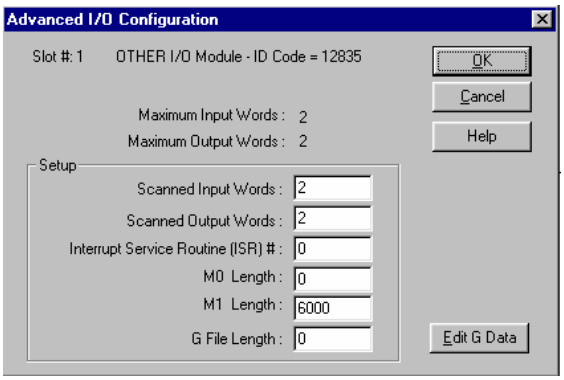
The first step in installing and configuring the module is to define the module to the system. Select the I/O Configuration option from the program screen. The system displays the following window:



Select the Other module from the list. This causes the system to display the following dialog box:



Enter the module I/O card ID number as 12835, and then click OK. Double-click the mouse on the module just added to the rack. Fill in the dialog box as shown:



The next step in the module's setup is to define the user-defined data areas to hold the status and read and write database areas. Edit the BARDACDW.CFG file now for the application to implement. Use any text editor to set the values in the file. You must retain the file name BARDACDW.CFG.

The last step in the module setup is to add the ladder logic. If the example ladder logic is used, adjust the ladder to fit the application. When the ladder example is not used, copy the example ladder logic to your application and alter as necessary.

The module is now ready to be used with your application. Insert the module in the rack (with the power turned off) and attach the serial communication cables. Download the new BARDACDW.CFG file to the module using a terminal emulation program. Download the new application to the controller and place the processor in run mode. If all the configuration parameters are set correctly and the module is attached to a network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink very rapidly. Refer to the Diagnostics and Troubleshooting section if you encounter errors. Attach a terminal to the Debug/Configuration port on the module and check the status of the module using the resident debugger in the module.

3.2 Module Data

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

3.3 Configuration File

In order for the module to operate, a configuration file (BARDACDW.CFG) is required. This configuration file contains information to set the data transfer characteristics between the module and the processor, to configure the module's client and command list. Each parameter in the file must be set carefully in order for the application to be implemented successfully. MVI46-BDW ***Configuration File Example*** (on page 58) contains an example listing of a BARDACDW.CFG file.

The configuration file is separated into sections with topic header names enclosed in the **[ ]** characters. The configuration file consists of the following sections:

[Section]	Description
[Module]	General module configuration information.

[Section]	Description
[BARDAC DW Client 0]	Configuration data for BARDAC DW Client 0
[BARDAC DW Client 0 Commands]	Command list for BARDAC DW Client 0
[BARDAC DW Client 1]	Configuration data for BARDAC DW Client 1
[BARDAC DW Client 1 Commands]	Command list for BARDAC DW Client 1
[BARDAC DW Client 2]	Configuration data for BARDAC DW Client 2
[BARDAC DW Client 2 Commands]	Command list for BARDAC DW Client 2
[BARDAC DW Client 3]	Configuration data for BARDAC DW Client 3
[BARDAC DW Client 3 Commands]	Command list for BARDAC DW Client 3

After each section header, the file contains a set of parameters. Unique labels are used under each section to specify a parameter. Each label in the file must be entered exactly as shown in the file for the parameter to be identified by the program. If the module is not considering a parameter, check the label for the data item. Each parameter's value is separated from the label with the '**:**' character. This character is used by the program to delimit the position in the data record where to start reading data. All data for a parameter must be placed after the '**:**' character. For numeric parameter values any text located after the value will not be used. There must be at least one space character between the end of the parameter value and the following text. An example of a parameter entry is given below:

Error/Status Pointer: 3000 #Database location for Error/Status Data

The parameter label is "Error/Status Pointer" and the parameter value is 3000. The characters after the parameter value are ignored and are used for internal documentation of the configuration file.

Any record that begins with the '#' character is considered to be a comment record. These records can be placed anywhere in the file as long as the '**#**' character is found in the first column of the line. These lines are ignored in the file and can be used to provide documentation within the configuration file. Liberal use of comments within the file can ease the use and interpretation of the data in the file.

The client command list definition section is formatted differently than the other sections. This section contains lists of parameters to be used. Each list begins with the label **START** and when the **END** label is reached. When entering the records into the list, make certain that the first character in each line is left blank.

The [BARDAC DW CLIENT 0 COMMANDS] section defines the BARDAC DW commands to be issued from the module to server devices on the network. These commands can be used for data collection and/or control of devices on the UDP/IP network.

3.3.1 Command List Overview

In order to interface the MVI46-BDW module with UDP/IP server devices, you must construct a command list. The commands in the list specify the server device 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 Client 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 tenths of a second 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 1 second.

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.

3.3.2 Commands Supported by the Module

The format of each command in the list is dependent on the BARDAC DW Function Code being executed. The tables below list the functions supported by the module:

Function Code	Definition	Supported in Client
0	Get (read data)	X
1	Set (write data)	X

Each command list record has the same general format. The first part of the record contains the information relating to the communication module and the second part contains information required to interface to the BARDAC DW UDP/IP server device.

3.3.3 Command Entry Formats

The following table shows the structure of the configuration data necessary for each of the supported commands:

BARDAC DW COMMAND STRUCTURE

Column #	1	2	3	4	5	6	10
Function Code	Enable Code	Internal Address	Poll Interval Time	Swap Code	IP Address	Function Code	Device Address
Get0	Code	Register	1/10th Seconds	0	IP Address	0	Register
Set1	Code	Register	1/10th Seconds	0	IP Address	1	Register

The first part of the record is the Module Information, which relates to the ProLinx module and the second part contains information required to interface to the Server device.

An example of a command list section of the CFG file is displayed below:

```
[BARDAC DW Client 0 Commands]
#
# 1 2 3 4 5 6 7
# DB Poll Swap Cmd Parameter
#Enab Addr Delay Code Node IP Address Code ID
START
 1 110 0 0 192.168.0.67 0 296
 1 111 0 0 192.168.0.67 0 181
 1 112 0 0 192.168.0.67 0 180
END
```

The following table discusses each parameter.

Command Parameter	Range	Description
Enable	0,1,2	This field defines whether or not the command is to be executed and under what conditions. Value: 0 — The command is disabled and will not be executed in the normal polling sequence. Value: 1 — The command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command will be executed, when the interval timer expires. Value: 2 — The command will execute only if the internal data associated with the command changes. This value is valid only for write commands. Because the BARDAC devices use the UDP protocol, the communication between the module and the device is not as reliable as with the TCP/IP protocol. For this reason, ProSoft suggests that the Enable Code of 2 not be used because a specific command may not be transferred. Therefore, it is strongly suggested to use the Enable Code of 1 to send SET commands.

<table>
<tr><th>Command Parameter</th><th>Range</th><th>Description</th></tr>
<tr><td>DB Address</td><td>0 to 4999</td><td>This field specifies the internal database register to be associated with the command.
- If the command is a read function, the data read from the Server device will be placed starting at the register value entered in this field.
If the command is a write function, the data written to the Server device will be sourced from the address specified.
The correct format depends on the parameter type. The following provides examples of the most common types:
Time Secs (XX.X)
SLC = 1 <=> BARDAC DRIVE = 0.1 sec
SLC = 100 <=> BARDAC DRIVE = 10.0 sec
SLC = 1019 <=> BARDAC DRIVE = 101.9 sec
Percentage (X.XX)
SLC = 1 <=> BARDAC DRIVE = 0.01%
SLC = 10 <=> BARDAC DRIVE = 0.10%
SLC = 10000 <=> BARDAC DRIVE = 100.00%
ENABLED/DISABLED
SLC = 0 <=> BARDAC = DISABLED
SLC = 1 <=> BARDAC = ENABLED</td></tr>
<tr><td>Poll Delay</td><td>0 to 65535</td><td>This parameter specifies the minimum interval to execute continuous commands (Enable code of 1). The parameter is entered in units of 1/10th seconds. Therefore, if a value of 10 is entered for a command, the command will execute no more frequently than every 1 second.</td></tr>
<tr><td>Swap Code</td><td>0,1,2,3</td><td>This parameter is used only for functions 3, 4, 6, and 16 to define if the data received (or sent) from the module is to be ordered differently than data received from the server device. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard method of storage of these data types in Server devices. This parameter can be set to order the register data received in an order useful by other applications. The following table defines the values and their associated operations:
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>0</td><td>None - No Change is made in the byte ordering</td></tr>
<tr><td>1</td><td>Words - The words are swapped</td></tr>
<tr><td>2</td><td>Words & Bytes - The words are swapped then the bytes in each word are swapped</td></tr>
<tr><td>3</td><td>Bytes - The bytes in each word are swapped</td></tr>
</table>
When swapping words, make sure you are using an even value in the Count Field. Odd values may generate unexpected results.</td></tr>
<tr><td>Node IP Address</td><td>xxx.xxx.xxx.xxx</td><td>The IP address of the device being addressed by the command.</td></tr>
</table>

Command Parameter	Range	Description
Function Code	0 or 1	This parameter specifies the function to be executed by the command. These function codes are defined in the BARDAC DW protocol.
Parameter ID		This parameter specifies the PIN value for the data value or parameter to be considered by the command. Please refer to the drive manual for a complete listing of valid PINs for your specific application.

Function Code	Description
Get0	Get (read data)
Set1	Set (write data)

3.4 Uploading and Downloading the Configuration File

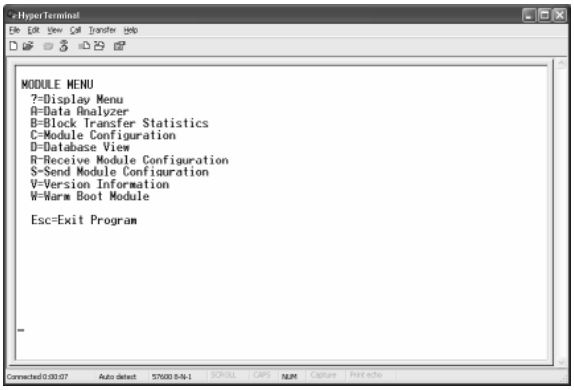
ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you must transfer the file from the module to your PC. After editing, you must transfer the file back to the module.

This section describes these procedures.

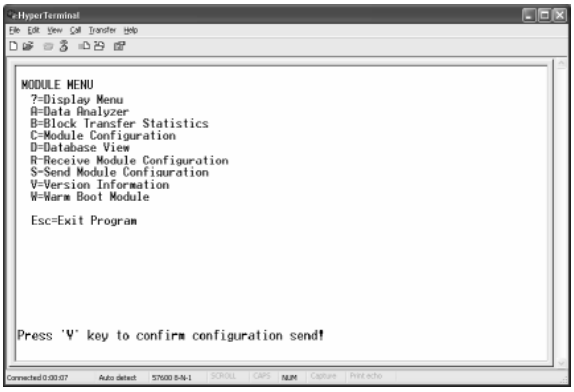
Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may or may not exactly match the configuration/debug menus in your own module. For specific details about the configuration/debug menus in your module, please refer to ***The Configuration/Debug Menu*** (on page 35).

3.4.1 Transferring the Configuration File to Your PC

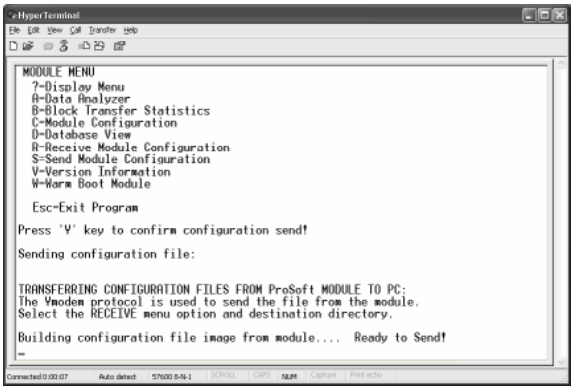
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



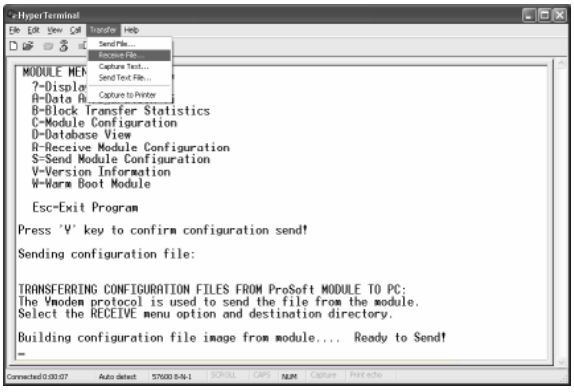
2 Press **[S]** (Send Module Configuration). The message "Press Y key to confirm configuration send!" is displayed at the bottom of the screen.



3 Press **[Y]**. The screen now indicates that the module is ready to send.



4 From the **Transfer** menu in HyperTerminal, select **Receive File**. This action opens the Receive File dialog box.

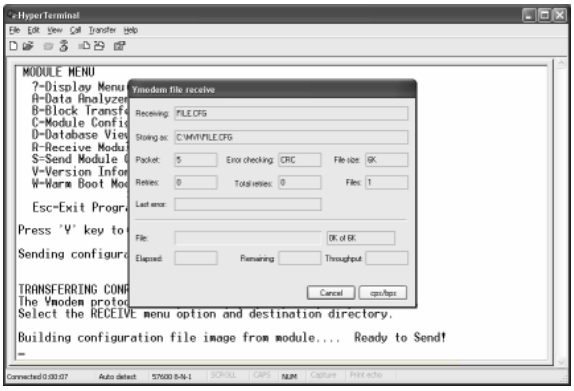


5 Use the Browse button to choose a folder on your computer to save the file, and then click Receive.

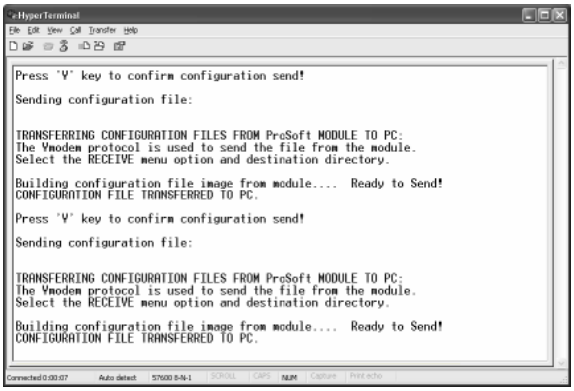
- **Note:** ProSoft Technology suggests that you download the configuration file pre-loaded on your module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at http://www.prosoft-technology.com.

6 Select Ymodem as the receiving protocol.

7 Click the Receive button. This action opens the Ymodem File Receive dialog box, showing the progress of your file transfer.



When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.



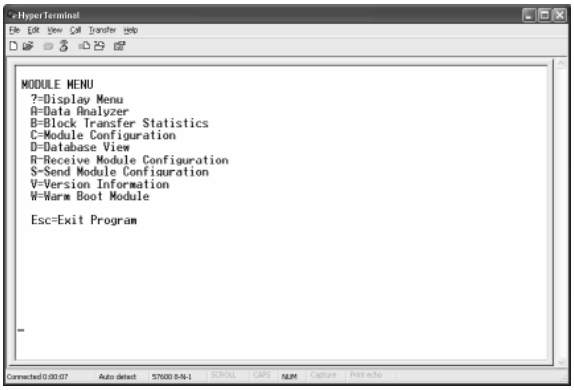
The configuration file is now on your PC at the location you specified.

8 You can now open and edit the file in a text editor such as Notepad. When you have finished editing the file, save it and close Notepad.

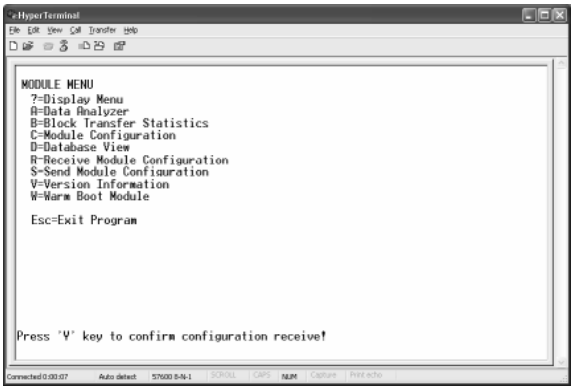
3.4.2 Transferring the Configuration File to the Module

Perform the following steps to transfer a configuration file from your PC to the module.

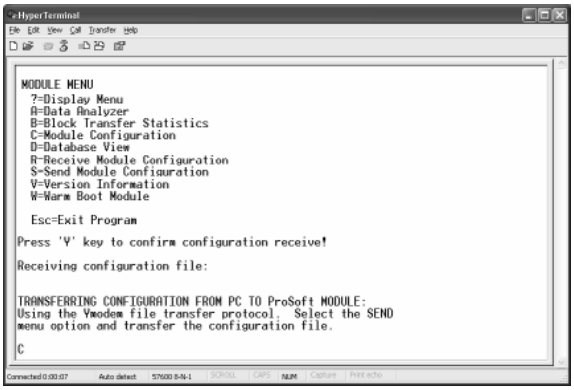
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



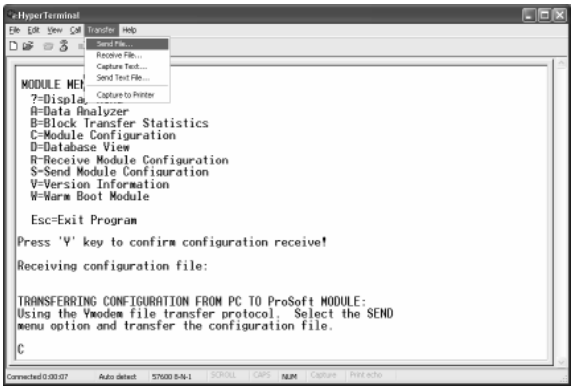
2 Press **[S]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



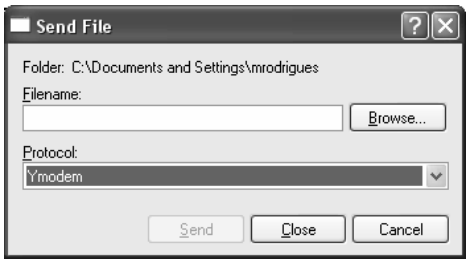
3 Press **[Y]**. The screen now indicates that the PC is ready to send.



4 From the **Transfer** menu in HyperTerminal, select **Send File**.



The Send File dialog appears.

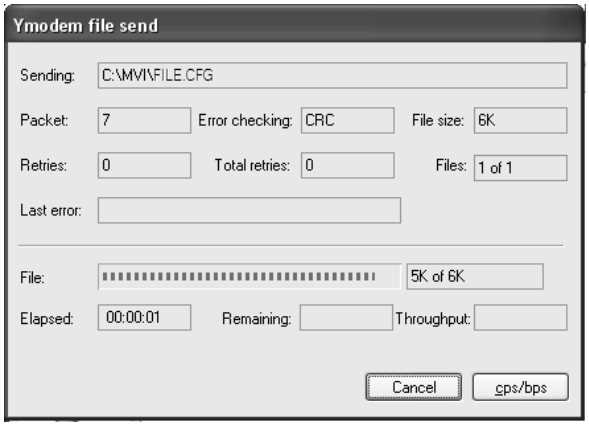


5 Use the Browse button to locate the configuration file your computer.

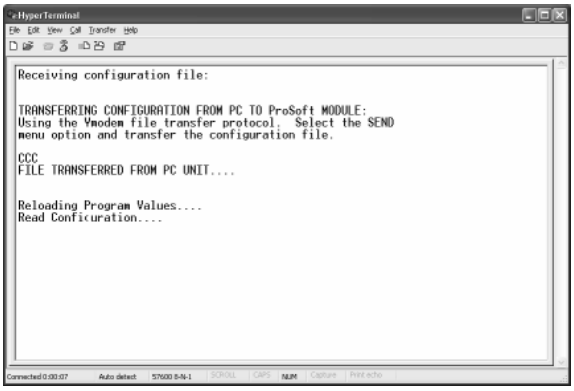
Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at http://www.prosoft-technology.com.

6 Select **Ymodem** as the protocol.

7 Click the Send button. This action opens the Ymodem File Send dialog box.



When the file transfer is complete, the module's configuration/debug screen indicates that the module has reloaded program values, and displays information about the module.



8 Your module now contains the new configuration.

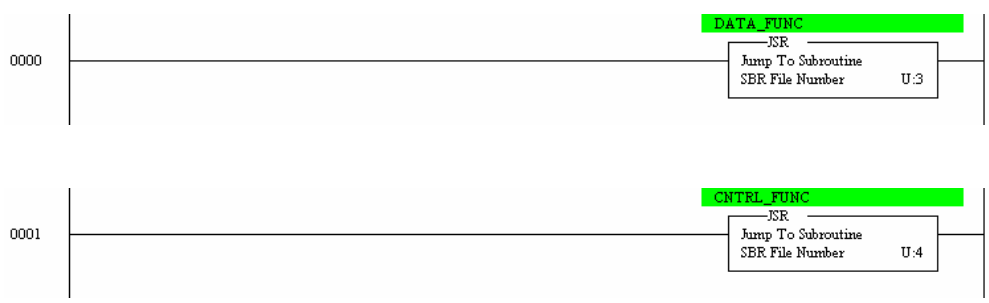
4 Ladder Logic

In This Chapter

Ladder logic is required for application of the MVI46-BDW module. Tasks that must be handled by the ladder logic are data transfer, special block handling, and status data receipt. This section discusses each aspect of the ladder logic as required by the module. Additionally, a power-up handler should be written to handle the initialization of the module's data and to clear any processor fault conditions.

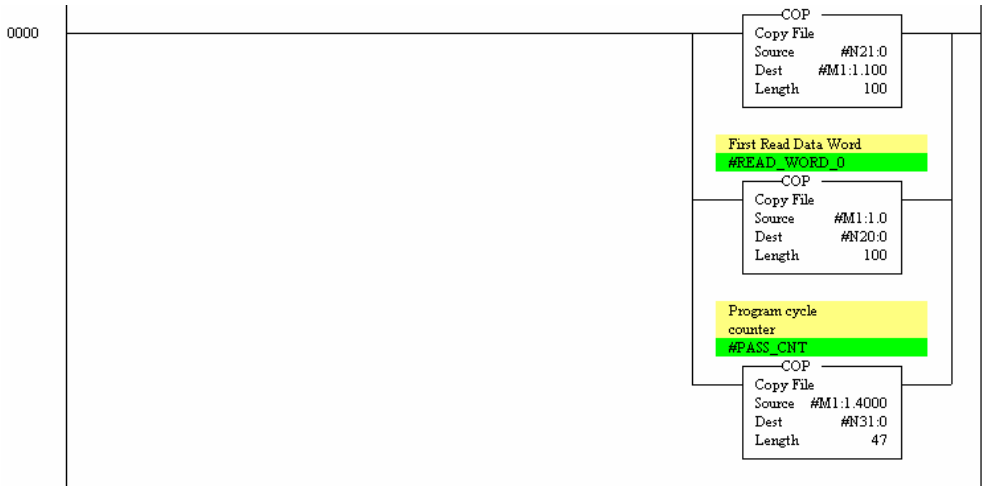
4.1 Main Routine

The Main program file is used to call the data transfer and control subroutines. The following example shows the main routine.



4.2 Data Transfer (U:3)

The data transfer routine is responsible for placing all the input data into the M1 file and for retrieving all the output data from the M1 file. The rung shown in the following diagram transfers the data between the M1 file and the user data files. The first branch transfers input data from the user files to the M1 file. The second branch transfers the output and status data from the M1 file into the user data files.

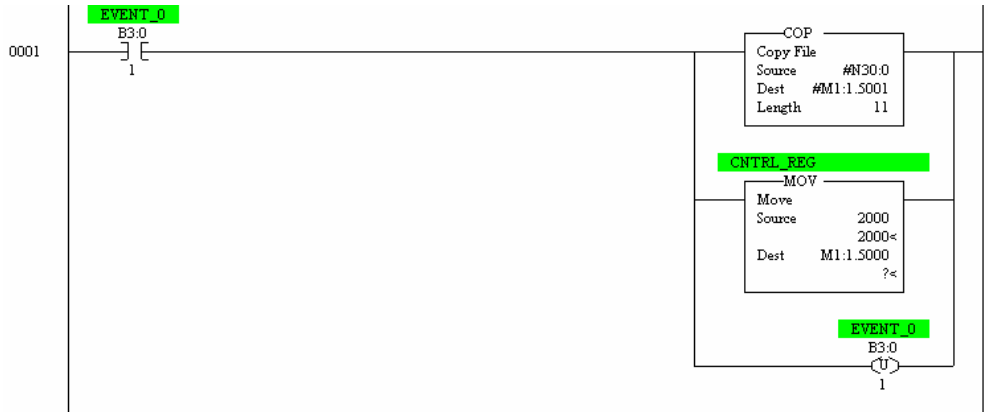


4.3 Control Routine (U:4)

The control routine is responsible for controlling the module or handling requests from the module using the control register (M1:1.5000).

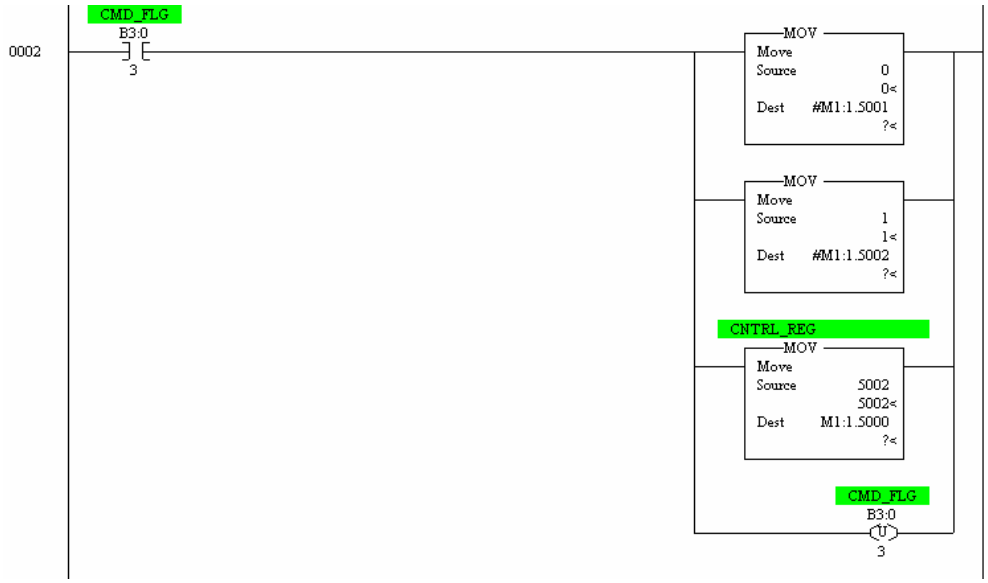
The following rung requests the cold boot operation for the module. Placing the value 9999 in the control register makes this request. When the module recognizes this value in the control register, it performs the cold boot operation. If you want to execute a warm boot instead of a cold boot, replace the value 9999 with 9998.

If the module will be sending event messages to the module to execute on the Ethernet port to other nodes on the network, the following rung may be used:



When the event flag is set, data in the event file is transferred to the module in using a control code of 2000. When the module receives the block, it places the event in the command queue. This is a simple means to send user-generated commands to any node on the network. A file of event commands can exist as a static or dynamic list and can be executed under control of user ladder logic. Make sure that the block is compatible with the ***Event Command Block*** (on page 10).

Ladder logic can also be utilized to execute a select set of commands in the client command list of the module. Remember that the last digit of the value you move to M1:1.5000 should contain the number of commands you want to execute on the queue (1-6). Here, as we are executing two commands, we are moving 5002 to M1:1.5000. The following rung executes the first two commands in the command list:



If the module is configured to receive the processor data set on startup, the following rung is required:

Use this feature to initialize the output data in the module with the values currently held in the processor. This feature is employed to bring the output data to a known or last set state.

5 Diagnostics and Troubleshooting

In This Chapter

This section provides information on diagnostics and troubleshooting in the following forms:

- Status data values are transferred from the module to the processor.
- All 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 modules status.

5.1 Reading Status Data from the Module

The MVI46-BDW module returns a Status Data block that can be used to determine the module's operating status. This data is located in the module's database at a user set location and is viewable using the Configuration/Debug port with a terminal emulation program. The Configuration/Debug port provides the following functionality:

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

5.1.1 The Configuration/Debug Menu

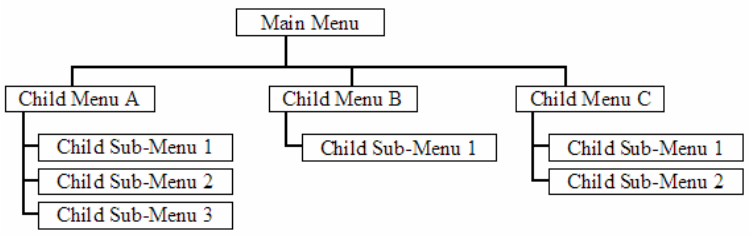
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

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 **[Z]** on your keyboard.

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



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

Keystrokes

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

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

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

5.1.2 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, 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 serial communications port available
- A null modem serial cable.

5.1.3 Required Software

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

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

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

The module uses the Ymodem file transfer protocol to send (download) and receive (upload) configuration files from your computer. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

5.1.4 Using the Configuration/Debug Port

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

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

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

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

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

1. Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2. Verify that your communication software is using the correct settings for baud rate, parity and handshaking.

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

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

5.1.5 Main Menu

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

```
MVI46-BARDAC DW COMMUNICATION MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Modbus Database View
  Client Command List Errors:  E=0  F=1  G=2  H=3
  Client Command List:         I=0  J=1  K=2  L=3
  R=Transfer Configuration from PC to MVI Unit
  S=Transfer Configuration from MVI Unit to PC
  T=Display e-mail setup
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  Communication Status:   1=Client 0   2=Client 1   3=Client 2   4=Client 3
  5=Client Configurations
  @=Network Menu          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, Inc. Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

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 Command Error List Menu

Press **[E], [F], [G]** or **[H]** to open the Command Error List for clients 1 through 4 respectively. 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 **[I], [J], [K]** or **[L]** to open the Command List menu for clients 1 through 4 respectively. Use this command to view the configured command list for the module.

Transferring the Configuration File from PC to MVI46 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 be certain the module is configured correctly.

Transferring the Configuration File from MVI46 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.

Resetting diagnostic data

Press **[U]** to reset the status counters for the client and servers in the module.

Viewing Version Information

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

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

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

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

Warm Booting the Module

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

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

Viewing Client Status

Press **[1], [2], [3]** or **[4]** to display the statistics of clients 1 through 4, respectively.

Opening the Network Menu

Press **[@]** to open the network menu. The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP, gateway and other network specification information. You can find more information about the commands on this menu in the ***Network Menu*** (on page 46) section.

Opening the Network Menu

Press **[@]** to open the network menu. The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP, gateway and other network specification information. You can find more information about the commands on this menu in the ***Network Menu*** (on page 46) section.

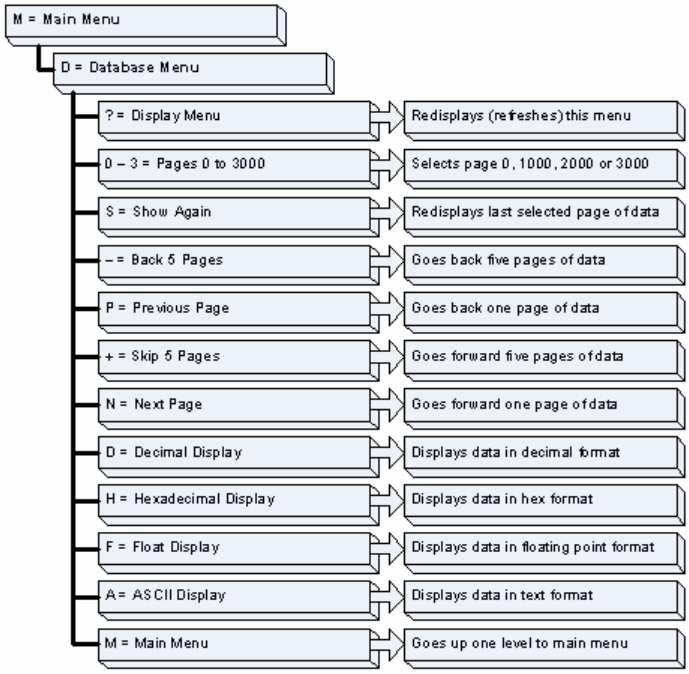
Exiting the Program

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

Press **[Esc]** to exit the program and display the operating system prompt. This command will cause the module to cease operation and stop transferring data between the ports and the module, and between the processor and the module. This could interrupt a currently running process. Only use this command if instructed to do so by the ProSoft Technical Support Group.

5.1.6 Database View Menu

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



Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again

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

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

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

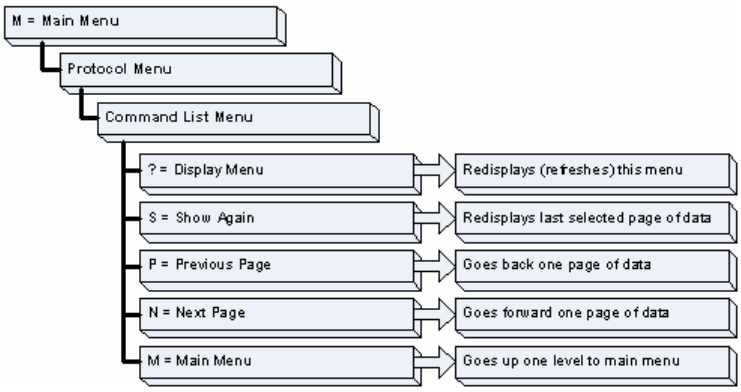
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

5.1.7 Master Command Error List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 20 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 20 Commands

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

Viewing the Next Page of Commands

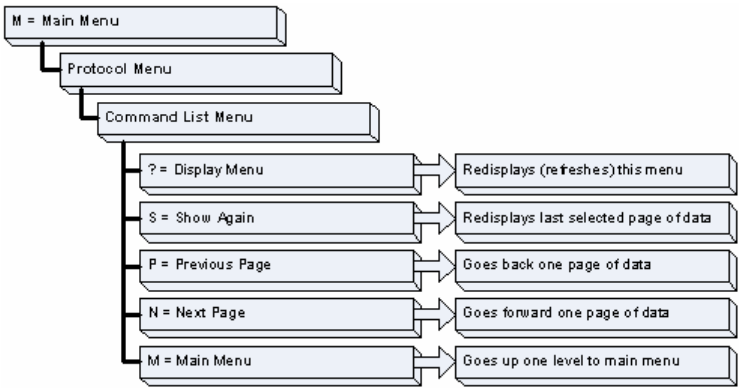
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

5.1.8 Master Command Error List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 50 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 50 Commands

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

Viewing the Next Page of Commands

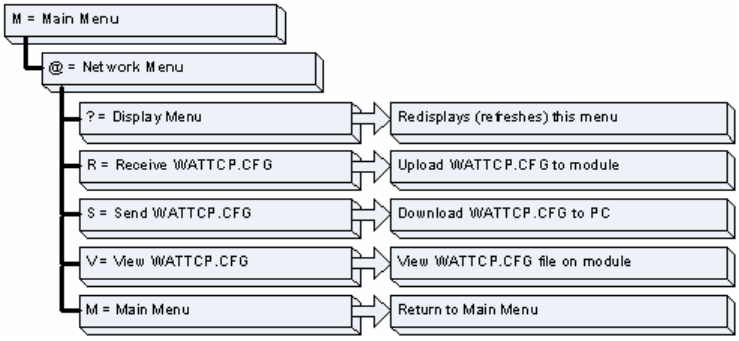
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

5.1.9 Network Menu

The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP and gateway addresses, and other network specification information.



Transferring WATTCP.CFG to the module

Press **[R]** to transfer a new WATTCP.CFG file from the PC to the module. Use this command to change the network configuration for the module (for example, the module's IP address).

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

Transferring WATTCP.CFG to the PC

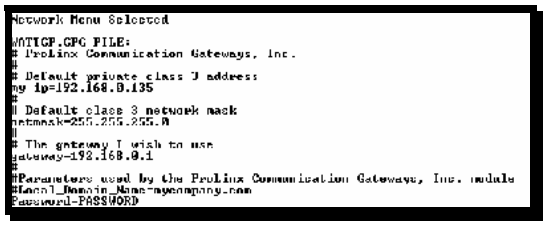
Press **[S]** to transfer the WATTCP.CFG file from the module to your PC.

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 transferred, you can open and edit the file to change the module's network configuration.

Viewing the WATTCP.CFG file on the module

Press **[V]** to view the module's WATTCP.CFG file. Use this command to confirm the module's current network settings.



Returning to the Main Menu

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

5.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
P1	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.
P2	Green	On	Port not used
		Off	Port not used
P3	Green	On	Port not used
		Off	Port not used
APP	Amber	Off	The MVI46-BDW is working normally.
		On	The MVI46-BDW module program has recognized a communication error on one of its Modbus ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or the battery is not present. Replace the battery on the module.

If a configuration error is found for the client, the client configuration error word will have a value other than zero. The configuration error word bits have the following definitions:

Bit	Description	Value
0		0x0001
1		0x0002
2		0x0004
3		0x0008
4	Invalid retry count parameter	0x0010

Bit	Description	Value
5	The float flag parameter is not valid.	0x0020
6	The float start parameter is not valid.	0x0040
7	The float offset parameter is not valid.	0x0080
8		0x0100
9		0x0200
10		0x0400
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

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

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, call ProSoft Technology support. There is a serious problem with the module, and it will have to be sent back to ProSoft.

5.2.1 Clearing a Fault Condition

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

1 Turn the power to the rack off

2 Remove the card from the rack

3 Make certain the Compact Flash is installed and all jumpers are set correctly

4 Re-insert the card in the rack and turn the power back on

5 Verify the configuration data being transferred to the module from the SLC processor

If the module's ACT/FAULT LED does not turn green, make sure the module is inserted completely into the rack. If this does not cure the problem, contact the factory.

5.2.2 Troubleshooting

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

Problem Description	Steps to take
Processor Fault	Be sure that the module is plugged into the slot that has been configured for the MVI46-BDW module. Be sure the ladder logic has been set up correctly
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Use the Configuration/Debug port facility to check this. To establish backplane communications, verify the following items: ▪ 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.
ACT/FLT 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.

6 Reference

In This Chapter

6.1 Cable Connections

The MVI46-BDW module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (RJ45 connector)

6.1.1 Ethernet Connection

The MVI46-BDW module has a single RJ45 plug located on the front of the module for use with the TCP/IP network. The module is connected to the Ethernet network using a cable between the module's RJ45 connector and a hub. Please insure that the proper cable and hub are used with the module. Failure to insure proper installation of the module may cause physical damage to the module. Check with your IT or network support group before installing the module on the network to insure that the module as a valid IP address for the network.

Ethernet Port Configuration: wattcp.cfg

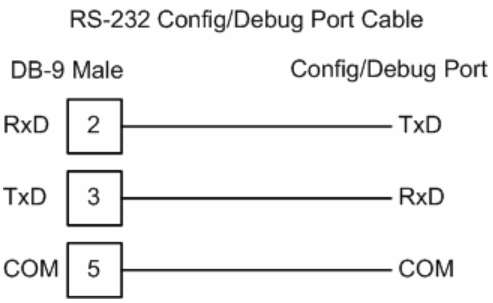
The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection. You can view the current network configuration using an ASCII terminal by selecting "@" (Network Menu) and "V" (View) options when connected to the Debug port.

```
# ProLinx Communication Gateways, Inc.
# Default private class 3 address
my_ip=192.168.0.100
# Default class 3 network mask
netmask=255.255.255.0
# name server 1 up to 9 may be included
# nameserver=xxx.xxx.xxx.xxx
# name server 2
```

```
# nameserver=xxx.xxx.xxx.xxx
# The gateway I wish to use
gateway=192.168.0.1
# some networks (class 2) require all three parameters
# gateway,network,subnetmask
# gateway 192.168.0.1,192.168.0.0,255.255.255.0
# The name of my network
# domainslist="mynetwork.name"
```

6.1.2 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:



6.2 MVI46-BDW Status Data Definition

This section contains a description of the members present in the status data object. This data is transferred from the module to the processor as part of the read data area when the block transfer interface is used. When the side-connect interface is employed, this data set is automatically transferred to the user file by the module. This section also includes the [BARDAC DW Client 0] Error/Status Data table.

6.2.1 Error/Status Data Table

Module Data

Offset	Content	Description
0	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
1	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
2	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.

Offset	Content	Description
3	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
4	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
5	Command Block Count	This field contains the total number of command blocks received from the processor.
6	Error Block Count	This field contains the total number of block errors recognized by the module.

Client 0

Offset	Content	Description
0	Client 0 Cmd Request	This value is incremented each time a command request is issued.
1	Client Cmd Response	This value is incremented each time a command response is received.
2	Client Cmd Error	This value is incremented each time an error message is received from a remote unit or a local error is generated for a command.
3	Client Request Count	This value is incremented each time a request message is issued.
4	Client Response Count	This value is incremented each time a response message is received.
5	Client Error Sent Count	This value is incremented each time an error is sent from the client.
6	Client Error Received Count	This value is incremented each time an error is received from a remote unit.
7	Client Cfg Error Word	This word contains a bit map that defines configuration errors in the configuration file for the client.
8	Client Current Error Code	This value corresponds to the current error code for the client.
9	Client Last Error Code	This value corresponds to the last error code recorded for the client.

Client 1

Offset	Content	Description
0	Client 1 Cmd Request	This value is incremented each time a command request is issued.
1	Client Cmd Response	This value is incremented each time a command response is received.
2	Client Cmd Error	This value is incremented each time an error message is received from a remote unit or a local error is generated for a command.
3	Client Request Count	This value is incremented each time a request message is issued.

Client 1		
4	Client Response Count	This value is incremented each time a response message is received.
5	Client Error Sent Count	This value is incremented each time an error is sent from the client.
6	Client Error Received Count	This value is incremented each time an error is received from a remote unit.
7	Client Cfg Error Word	This word contains a bitmap that defines configuration errors in the configuration file for the client.
8	Client Current Error Code	This value corresponds to the current error code for the client.
9	Client Last Error Code	This value corresponds to the last error code recorded for the client.

Client 2		
Offset	**Content**	**Description**
0	Client 2 Cmd Request	This value is incremented each time a command request is issued.
1	Client Cmd Response	This value is incremented each time a command response is received.
2	Client Cmd Error	This value is incremented each time an error message is received from a remote unit or a local error is generated for a command.
3	Client Request Count	This value is incremented each time a request message is issued.
4	Client Response Count	This value is incremented each time a response message is received.
5	Client Error Sent Count	This value is incremented each time an error is sent from the client.
6	Client Error Received Count	This value is incremented each time an error is received from a remote unit.
7	Client Cfg Error Word	This word contains a bitmap that defines configuration errors in the configuration file for the client.
8	Client Current Error Code	This value corresponds to the current error code for the client.
9	Client Last Error Code	This value corresponds to the last error code recorded for the client.

Client 3		
Offset	**Content**	**Description**
0	Client 3 Cmd Request	This value is incremented each time a command request is issued.
1	Client Cmd Response	This value is incremented each time a command response is received.

Client 3		
2	Client Cmd Error	This value is incremented each time an error message is received from a remote unit or a local error is generated for a command.
3	Client Request Count	This value is incremented each time a request message is issued.
4	Client Response Count	This value is incremented each time a response message is received.
5	Client Error Sent Count	This value is incremented each time an error is sent from the client.
6	Client Error Received Count	This value is incremented each time an error is received from a remote unit.
7	Client Cfg Error Word	This word contains a bitmap that defines configuration errors in the configuration file for the client.
8	Client Current Error Code	This value corresponds to the current error code for the client.
9	Client Last Error Code	This value corresponds to the last error code recorded for the client.

6.3 MVI46-BDW Configuration Data Definition

This section contains a listing of the parameters and their definitions for the MVI46-BDW module configuration file definition.

[Section]/Item	Value	Range	Description
[MODULE]			**Configuration header for general module information**
Module Name:		Up to 80 chars	Name of the module for use on reports. Use this parameter to identify your module in your system.
Password:		Up to 20 chars	This parameter is used to set the password for the module. If the parameter is not set, the module will not be password protected. The module interprets the password as a case-sensitive string. Do not include spaces or special characters in the password. Only alpha and numeric characters should be used.
Error/Status Pointer:		-1 to 4955	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 and all server error/status data.
Write Register Start:		0 to 4999	This parameter specifies the starting register in the module where the data transferred from the processor will be placed. Valid range for this parameter is 0 to 4999.
Write Register Count:		0 to 5000	This parameter specifies the number of registers to transfer from the processor to the module. Valid entry for this parameter is 0 to 5000.

[Section]/Item	Value	Range	Description
[MODULE]			**Configuration header for general module information**
Read Register Start:		0 to 4999	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 4999.
Read Register Count:		0 to 5000	This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 5000.
Failure Flag Count:		0 to 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 (1-65535), communications will cease if the specified number of failures occur.
Initialize Output Data:		0 or 1	This parameter is used to determine if the output data for the module should be initialized with values from the processor. If the value is set to 0, the output data will be initialized to 0. If the value is set to 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.

[Section]/Item	Value	Range	Description
[BARDAC DW CLIENT x]			Start header for Client x (x=0 to 3)
Error/Status Pointer:		-1 to 4990	Starting register location in virtual database for the error/status table for this client. 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.
Minimum Command Delay:		0 to 65535	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.
Command Error Pointer:		-1 to 4999	This parameter sets the address in the internal database where the command error data will be placed. If the value is set to -1, the data will not be transferred to the database.

[Section]/Item	Value	Range	Description
Response Timeout:		0 to 65535	This parameter represents the message response timeout period in 1-ms increments. This is the time that a client will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending upon the communication network used and the expected response time of the slowest device on the network.
Retry Count:		0 to 25	This parameter specifies the number of times a command will be retried if it fails.

The command list for the client in the module is located in the [] section of the file. The table below displays the functions supported by the module and the format of each command:

Module Information ← → Device Information

BARDAC DW COMMAND STRUCTURE

Column #	1	2	3	4	5	6	7
Function Code	Enable Code	Internal Address	Poll Interval Time	Swap Code	IP Address	Function Code	Device Modbus Address
GET 0	Code	Register	1/10 Seconds	0	IP Address	0	Register
SET 1	Code	Register	1/10 Seconds	0	IP Address	1	Register

The form below can be used to design the application's command list:

Module Information ◄—— | ——► Device Information

BARDAC DW COMMAND ENTRY FORM

Column #	1	2	3	4	5	6	7
	Enable Code	Internal Address	Poll Interval Time	Swap Code	IP Address	Function Code	Device Modbus Address

6.4 MVI46-BDW Configuration File Example

This section contains a listing of an example BARDACDW.CFG file for the MVI46-BDW module.

```
# BARDACDW.CFG
#
```

```
# This is an example configuration file for the MVIxx-BARDAC-DW module.
#
# COMPANY  :
# DATE     :
# LOCATION :
#
#
# This section of the file describes the database setup and module level
# parameters.
#

[Module]
Module Name : MVI46-BARDAC-DW Communication Module Test

#Password                : PASSWORD
Local Domain Name        : vcs.com
Error/Status Pointer     : 3000   #Database location for module level status data

Failure Flag Count       : 0      #Determines if BP failure will cause protocol
to be
                                  #disabled (0=Ignore, >0 = failure count to
disable)

Initialize Output Data   : 0      #Read output values from controller (0=No or
1=Yes)

# This section is used to define the configuration for the master device
# simulated on Bardac DW UDP service port 48556 (0xBDAC)
#
[Bardac DW Client 0]
Error/Status Pointer     : 4000   #Database offset for client status data
Command Error Pointer    : 4010   #Database offset for client command error list
Minimum Command Delay    : 20     #Minimum number of msec's between commands
Response Timeout         : 1000   #Response messgage timeout (0-65535 mSec)
Retry Count              : 0      #Response failure retry count

[Bardac DW Client 0 Commands]
#
# This section contains examples for a Bardac DW UDP/IP client using service port
# 48556 (0xBDAC).  Refer to the drive documentation for a full list of Parameter
# ID's.  This section can contain up to 100 commands.
#
#  1     2     3      4         5             6          7
#        DB   Poll   Swap                    Cmd    Parameter
# Enab  Addr  Delay  Code  Node IP Address   Code      ID
START
#
# This is a set of GET instructions
   1     110    0     0     192.168.0.67      0       296
   1     111    0     0     192.168.0.67      0       181
   1     112    0     0     192.168.0.67      0       180
   1     113    0     0     192.168.0.67      0       179
   1     114    0     0     192.168.0.67      0       178
   1     115    0     0     192.168.0.67      0       171
   1     116    0     0     192.168.0.67      0       172
```

```
   1     117    0     0     192.168.0.67      0       173
   1     118    0     0     192.168.0.67      0       174
   1     119    0     0     192.168.0.67      0       175
#
# This is a set of SET instructions (disabled as default)
   0      0     0     0     192.168.0.67      1       281
   0      1     0     0     192.168.0.67      1       282
   0      2     0     0     192.168.0.67      1       250
   0      3     0     0     192.168.0.67      1       251
   0      4     0     0     192.168.0.67      1       252
   0      5     0     0     192.168.0.67      1       253
   0      6     0     0     192.168.0.67      1       254
END

# This section is used to define the configuration for the master device
# simulated on Bardac DW UDP service port 48556 (0xBDAC)
#
[Bardac DW Client 1]
Error/Status Pointer     : 4100   #Database offset for client status data
Command Error Pointer    : 4110   #Database offset for client command error list
Minimum Command Delay    : 20     #Minimum number of msec's between commands
Response Timeout         : 1000   #Response messgage timeout (0-65535 mSec)
Retry Count              : 0      #Response failure retry count

[Bardac DW Client 1 Commands]
#
# This section contains examples for a Bardac DW UDP/IP client using service port
# 48556 (0xBDAC).  Refer to the drive documentation for a full list of Parameter
# ID's.  This section can contain up to 100 commands.
#
#  1     2     3      4         5             6          7
#        DB   Poll   Swap                     Cmd    Parameter
# Enab  Addr  Delay  Code  Node IP Address   Code      ID
START
#
# This is a set of GET instructions for testing
    1   140     0     0     192.168.0.67       0       296
    1   141     0     0     192.168.0.67       0       181
    1   142     0     0     192.168.0.67       0       180
    1   143     0     0     192.168.0.67       0       179
END

# This section is used to define the configuration for the master device
# simulated on Bardac DW UDP service port 48556 (0xBDAC)
#
[Bardac DW Client 2]
Error/Status Pointer     : 4200   #Database offset for client status data
Command Error Pointer    : 4210   #Database offset for client command error list
Minimum Command Delay    : 20     #Minimum number of msec's between commands
Response Timeout         : 1000   #Response messgage timeout (0-65535 mSec)
Retry Count              : 0      #Response failure retry count

[Bardac DW Client 2 Commands]
#
# This section contains examples for a Bardac DW UDP/IP client using service port
```

```
# 48556 (0xBDAC).  Refer to the drive documentation for a full list of Parameter
# ID's.  This section can contain up to 100 commands.
#
#  1     2     3      4         5             6          7
#        DB   Poll   Swap                    Cmd    Parameter
# Enab  Addr  Delay  Code  Node IP Address   Code      ID
START
#
# This is a set of GET instructions for testing
    1   150    0      0     192.168.0.67       0       296
    1   151    0      0     192.168.0.67       0       181
    1   152    0      0     192.168.0.67       0       180
    1   153    0      0     192.168.0.67       0       179
END

# This section is used to define the configuration for the master device
# simulated on Bardac DW UDP service port 48556 (0xBDAC)
#
[Bardac DW Client 3]
Error/Status Pointer     : 4300   #Database offset for client status data
Command Error Pointer    : 4310   #Database offset for client command error list
Minimum Command Delay    : 20     #Minimum number of msec's between commands
Response Timeout         : 1000   #Response messgage timeout (0-65535 mSec)
Retry Count              : 0      #Response failure retry count

[Bardac DW Client 3 Commands]
#
# This section contains examples for a Bardac DW UDP/IP client using service port
# 48556 (0xBDAC).  Refer to the drive documentation for a full list of Parameter
# ID's.  This section can contain up to 100 commands.
#
#  1     2     3      4         5             6          7
#        DB   Poll   Swap                    Cmd    Parameter
# Enab  Addr  Delay  Code  Node IP Address   Code      ID
START
#
# This is a set of GET instructions for testing
    1   160     0     0     192.168.0.67       0       296
    1   161     0     0     192.168.0.67       0       181
    1   162     0     0     192.168.0.67       0       180
    1   163     0     0     192.168.0.67       0       179
    1   170     0     0     192.168.0.67       0       296
    1   171     0     0     192.168.0.67       0       181
    1   172     0     0     192.168.0.67       0       180
    1   173     0     0     192.168.0.67       0       179
END


# This section is used to define e-mail reports to be sent from the module
# to a specified e-mail server/user account based on the value of selected
# user register/value combinations.  When the specified register value contains
# the value defined, the e-mail file will be sent from the module.

[E-MAIL]
#    DB   Trigger     Mail          TO
```

```
#   Reg     Value     Server IP     Name          E-Mail File Name
START
      0      1000     192.168.0.61  rich@vcs.com  stat
END
```

Support, Service & Warranty

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

Internet	Web Site: http://www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Phone	(661) 716-5100 (661) 716-5101 (Fax)
Postal Mail	ProSoft Technology, Inc. 1675 Chester Avenue, Fourth Floor Bakersfield, CA 93301

Before calling for support, please prepare yourself for the call. In order to provide the best and quickest support possible, we will most likely ask for the following information:

1 Product Version Number

2 System architecture

3 Module configuration and contents of BARDACDW.CFG file

4 Module Operation

- o Configuration/Debug status information
- o LED patterns

5 Information about the processor and user data files as viewed through RSLogix 500 and LED patterns on the processor

6 Details about the serial devices interfaced

An after-hours answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer the questions that are important to you.

Module Service and Repair

The MVI46-BDW device is an electronic product, designed and manufactured to function under somewhat adverse conditions. As with any product, through age, misapplication, or any one of many possible problems the device may require repair.

When purchased from ProSoft Technology, Inc., the device has a 1 year parts and labor warranty (3 years for RadioLinx) according to the limits specified in the

warranty. Replacement and/or returns should be directed to the distributor from whom the product was purchased. If you must return the device for repair, obtain an RMA (Returned Material Authorization) number from ProSoft Technology, Inc. Please call the factory for this number, and print the number prominently on the outside of the shipping carton used to return the device.

General Warranty Policy – Terms and Conditions

ProSoft Technology, Inc. (hereinafter referred to as ProSoft) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product. Limited warranty service may be obtained by delivering the Product to ProSoft in accordance with our ***product return procedures*** (on page 65) and providing proof of purchase and receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to ProSoft, and to use the original shipping container or equivalent. Contact ProSoft Customer Service for further information.

This warranty is limited to the repair and/or replacement, at ProSoft's election, of defective or non-conforming Product, and ProSoft shall not be responsible for the failure of the Product to perform specified functions, or any other non-conformance caused by or attributable to: (a) any misuse, misapplication, accidental damage, abnormal or unusually heavy use, neglect, abuse, alteration (b) failure of Customer to adhere to ProSoft's specifications or instructions, (c) any associated or complementary equipment, software, or user-created programming including, but not limited to, programs developed with any EC1131-3 programming languages, "C" for example, and not furnished by ProSoft, (d) improper installation, unauthorized repair or modification (e) improper testing, or causes external to the product such as, but not limited to, excessive heat or humidity, power failure, power surges or natural disaster, compatibility with other hardware and software products introduced after the time of purchase, or products or accessories not manufactured by ProSoft; all of which components, software and products are provided as-is. In no event will ProSoft be held liable for any direct or indirect, incidental consequential damage, loss of data, or other malady arising from the purchase or use of ProSoft products.

ProSoft's software or electronic products are designed and manufactured to function under adverse environmental conditions as described in the hardware specifications for this product. As with any product, however, through age, misapplication, or any one of many possible problems, the device may require repair.

ProSoft warrants its products to be free from defects in material and workmanship and shall conform to and perform in accordance with published technical specifications and the accompanying written materials for up to one year (12 months) from the date of original purchase (3 years for RadioLinx

products) from ProSoft. If you need to return the device for repair, obtain an RMA (Returned Material Authorization) number from ProSoft Technology, Inc. in accordance with the RMA instructions below. Please call the factory for this number, and print the number prominently on the outside of the shipping carton used to return the device.

If the product is received within the warranty period ProSoft will repair or replace the defective product at our option and cost.

Warranty Procedure: Upon return of the hardware product ProSoft will, at its option, repair or replace the product at no additional charge, freight prepaid, except as set forth below. Repair parts and replacement product will be furnished on an exchange basis and will be either reconditioned or new. All replaced product and parts become the property of ProSoft. If ProSoft determines that the Product is not under warranty, it will, at the Customer's option, repair the Product using then current ProSoft standard rates for parts and labor, and return the product freight collect.

Limitation of Liability

EXCEPT AS EXPRESSLY PROVIDED HEREIN, PROSOFT MAKES NO WARRANT OF ANY KIND, EXPRESSED OR IMPLIED, WITH RESPECT TO ANY EQUIPMENT, PARTS OR SERVICES PROVIDED PURSUANT TO THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER PROSOFT OR ITS DEALER SHALL BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION IN CONTRACT OR TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS RESULTING FROM, OR ARISING OUT OF, OR IN CONNECTION WITH THE USE OR FURNISHING OF EQUIPMENT, PARTS OR SERVICES HEREUNDER OR THE PERFORMANCE, USE OR INABILITY TO USE THE SAME, EVEN IF ProSoft OR ITS DEALER'S TOTAL LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT.

Where directed by State Law, some of the above exclusions or limitations may not be applicable in some states. This warranty provides specific legal rights; other rights that vary from state to state may also exist. This warranty shall not be applicable to the extent that any provisions of this warranty are prohibited by any Federal, State or Municipal Law that cannot be preempted. Contact ProSoft Customer Service at (661) 716-5100 for more information.

RMA Procedures

In the event that repairs are required for any reason, contact ProSoft Technical Support at +1 661.716.5100. A Technical Support Engineer will ask you to

perform several tests in an attempt to diagnose the problem. Simply calling and asking for a RMA without following our diagnostic instructions or suggestions will lead to the return request being denied. If, after these tests are completed, the module is found to be defective, we will provide the necessary RMA number with instructions on returning the module for repair.

Index

R

S

T

U

V

W

Y