ProSoft
TECHNOLOGY

Where Automation Connects.

inRAx

MVI94-BSCH

FLEX Platform

Bosch Communication I/O Module

July 06, 2009

USER MANUAL

Important Installation Instructions

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

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

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

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

D "THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

MVI (Multi Vendor Interface) Modules

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

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

CL I Div 2 GPs A, B, C, D

Temp Code T5

II 3 G

Ex nA IIC T5 X

0° C <= Ta <= 60° C

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

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

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.

B Warning - Explosion Hazard - When in Hazardous Locations, turn off power before replacing or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.

C Suitable for use in Class I, division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: -40 to 85°C (-40 to 185°F)
- Shock: 30g Operational; 50g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings

cUL	C22.2 No. 213-1987

C E c(UL)ᵤ

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
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com

Copyright © ProSoft Technology, Inc. 2009. All Rights Reserved.

MVI94-BSCH User Manual
July 06, 2009

ProSoft Technology ®, ProLinx ®, inRAx ®, ProTalk®, and RadioLinx ® are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

ProSoft Technology® Product Documentation

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

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33 (0) 5.3436.87.20

Latin America: +1.281.298.9109

North America: +1.661.716.5100

Contents

1   Quick Start

This section describes the procedure to be followed for installing and configuring the module for communications. These steps should be followed for successful implementation of a module in a user application.

1   Define the communication characteristics of the Bosch master port.
2   Define the command lists to be used on the Bosch master port.
3   Fill in the blank configuration form for application using the data sets defined in steps one and two.
4   Edit the configuration text file **MVI_BSCH.TXT** to reflect the desired data from the configuration form and save the file under a different name. This text file is provided on the program disk supplied with the module.
5   Connect the MVI94-BSCH module's Configuration/Debug Port with a null-modem cable.
6   Connect the module to a 24 VDC power source.
7   Select the directory containing the correct configuration file on the computer.
8   Start Hyperterminal on the computer.
9   Press the [?] key on the terminal to verify that the module is communicating with the computer and that the main menu mode is current.
10 Press **[R]** in Hyperterminal to select the receive option. Press **[Y]** to confirm that you wish to download a new configuration.
11 Press **[ALT] [F3]** and enter the name of the configuration file to load into the module. The configuration will be downloaded, and the module will restart using the new configuration.
12 Connect the module's Bosch master port to the Bosch slave. If everything is configured correctly and the cable connections are correct, communications should be present on the master port.
13 Monitor the communication statistics for the port to verify that everything is working correctly.
14 View the database in the module using the terminal emulator.
15 Edit the ladder logic provided with the module for your system. The ladder logic is responsible for transferring the data between the module and processor.
16 Connect the module to the Flex I/O processor or Control Net processor. If all is configured correctly, the data in the module should be visible in the processor.
17 Use the Configuration/Debug port to view the backplane transfer statistics.

2 Hardware Installation

In This Chapter

2.1 Verify Package Contents

Make sure that you verify the contents of the product before you discard the packing material. The following components should be included with the product:

1 A MVI94 Flex I/O Base
2 A MVI94 Module with 3 jumpers installed
3 One Serial Adapter Cable

2.2   Mounting the MVI 94 Flex I/O Base
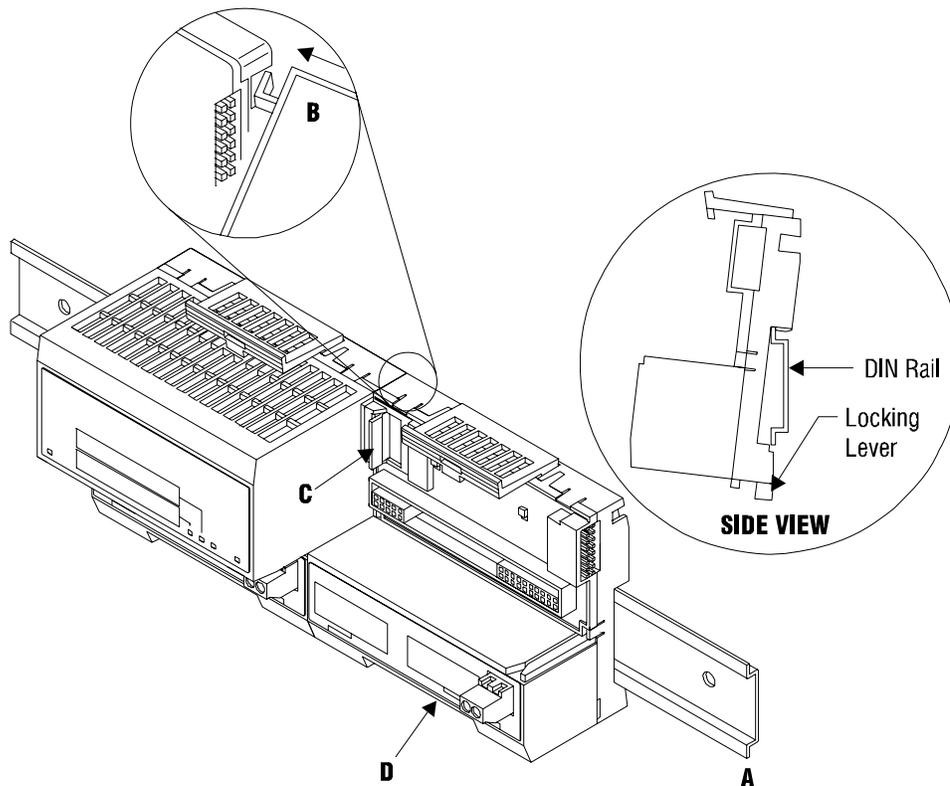
1   Remove the cover plug (if used) in the male connector of the unit to which
     you are connecting this Base.
2   Check to make sure that the 16 pins in the male connector on the adjacent
     device are straight and in line so that the mating female connector on this
     Base will mate correctly.
3   Make certain that the female flexbus connector **C** is **fully retracted** into the
     Base.
4   Position the Base on a 35 x 7.5mm DIN-rail **A** at a slight angle with the hook
     B on the left side of the Base hooked into the right side of the unit on the left.
5   Rotate the Base onto the DIN-rail with the top of the rail hooked under the lip
     on the rear of the Base. Use caution to make sure that the female flexbus
     connector does not strike any of the pins in the mating male connector.
6   Press the terminal base down onto the DIN-rail until flush. The locking tab **D**
     snaps into position and locks the terminal base to the DIN-rail.
7   If the Base does not lock in place, use a screwdriver or similar device to
     move the locking tab down, press the Base flush with the DIN-rail and release
     the locking tab to lock the base in place.
8   **Gently** push the female flexbus **C** connector into the adjacent base or
     adapter male connector to complete the flexbus connections.

2.3   Setting Jumpers

Before installing the MVI94 module onto its base, the module's configuration can be set using the jumpers on the bottom of the module as shown in this figure.
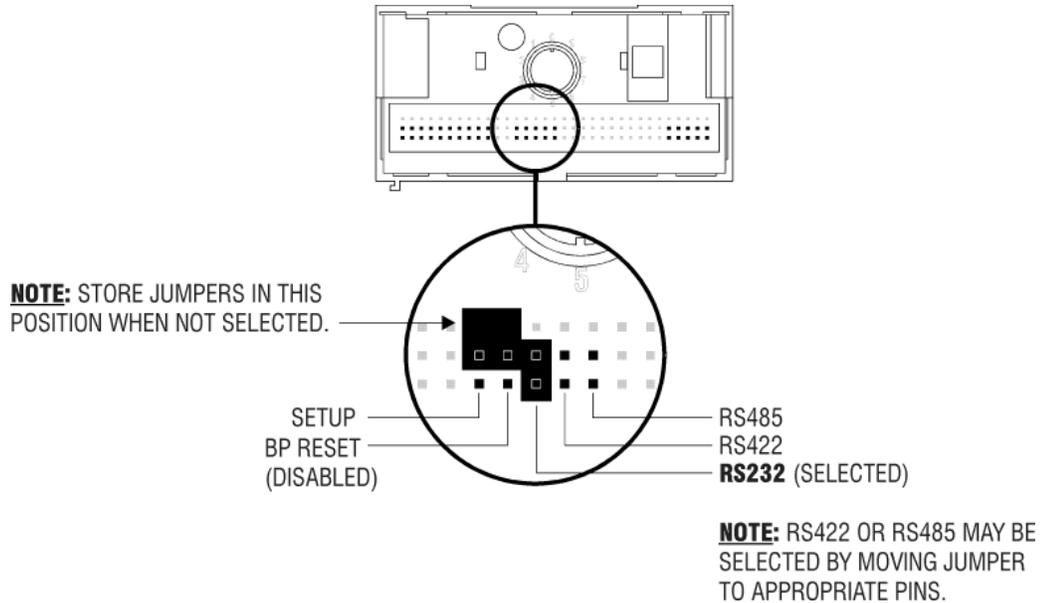
Port 2 RS-232/422/485: Select with jumper (shipped in 232).

BP Reset: If the MVI94 module is to be reset when the Flex Bus is reset, install the BP RESET jumper in the Enabled position.

ATTENTION: Do not remove or replace a base unit when power is applied. Interruption of the flexbus can result in unintended operation or machine motion.

SETUP: To place the module in SETUP mode, install the jumper in the Selected position (DOS default). To prevent the module from being in Setup mode, leave the jumper in the disabled position.

2.4   Installing the Module onto the Base

1   Rotate the keyswitch **1** on the Base clockwise to position #1.
2   Make certain the flexbus connector **3** on the Base is pushed all the way to the left to connect with the neighboring base or adapter. **The Module cannot be installed unless the flexbus connector is fully extended.**
3   Make sure that the pins on the bottom of the Module are straight so they will align properly with the connector socket on the Base.
4   Position the Module with its alignment bar 5 aligned with the groove 6 on the Base.
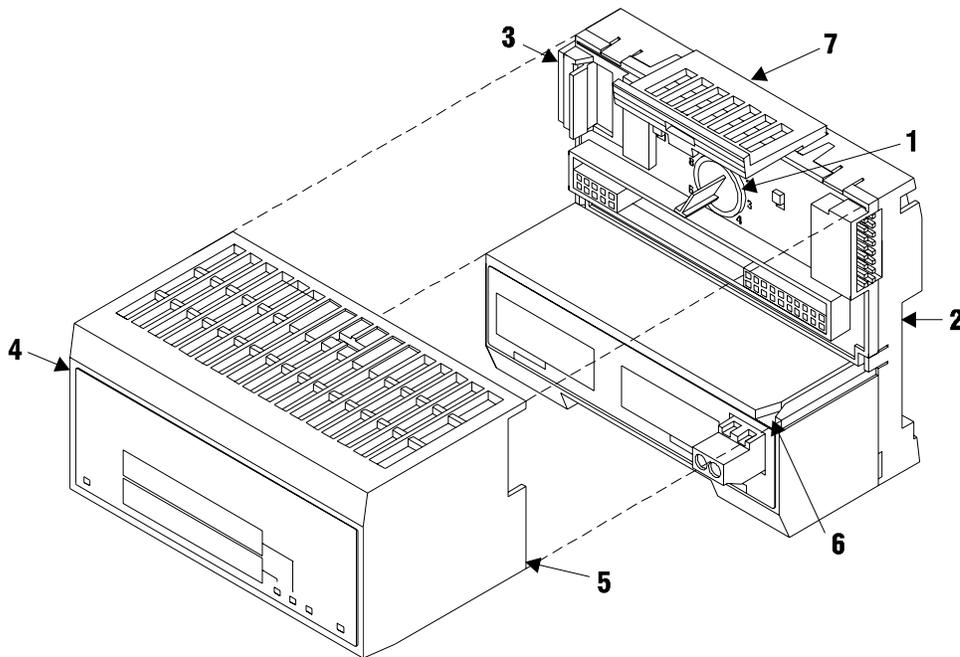
5   Press firmly and evenly to seat the Module in the Base. The Module is seated when the latch **7** on the Base is locked into the Module.

2.5   Installing the Serial Adapter Cables

Two identical serial adapter cables are supplied. Each cable has a locking-type 8 pin Mini-DIN plug on one end and a DB-9 male connector on the other end. The Mini-DIN connector on each cable is inserted into the Mini-DIN receptacles marked **PRT1** and **PRT2** on the Base.

To install the locking-type Mini-DIN connector, slide the spring-loaded sleeve back while inserting the plug into the receptacle on the Base, and then release the sleeve when fully seated. The locking mechanism prevents the cable from being removed during normal operation. To remove the cable, slide the sleeve back and remove the plug.
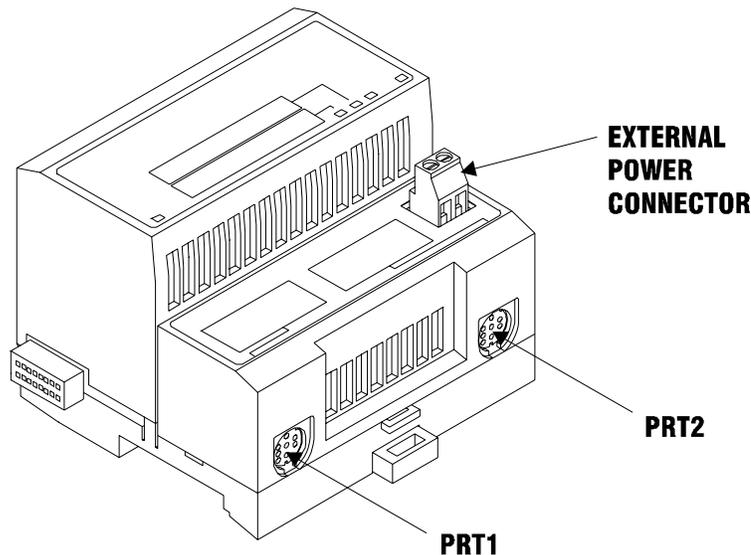
2.6   Wiring the Power Connections

External power is supplied to the Base on the 2 pin screw terminal block. The power supply can be either 24Vdc or 12Vdc, and should be located in close proximity of the base.

- Connect dc common to the **COM** terminal
- Connect +24V dc or +12V dc to the **24VDC** terminal

3   Configuration

3.1   Configuration File

The MVI94-BSCH module stores its configuration in a text file called MVI_BSCH.TXT, located in the module's flash memory. When the module starts up, it reads the configuration file and uses the information to control how the Bosch protocol interacts with the module's application port(s).

The configuration file is arranged in *Sections,* with a heading in **[ ]** characters at the beginning of each section. Each *Section* contains a list of *Parameters* and *Values,* followed by an optional *Comment* that explains the parameter.

The following illustration shows an example of a *Section,* a *Parameter,* a *Value,* and a *Comment.*
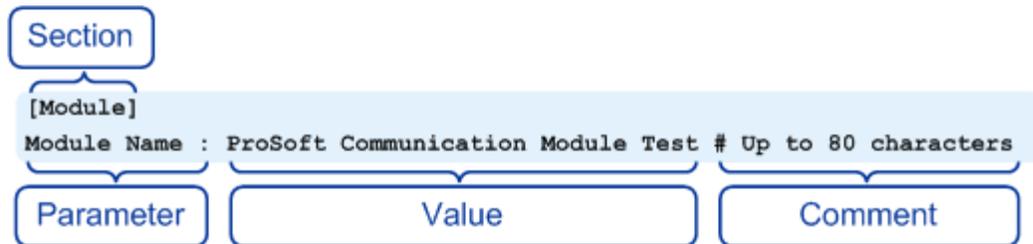
The *Parameter* must be followed by a **[:]** (colon) character. The text following the **[:]** is a *Value.*

The module ignores *"comment"* text following the **[#]** character. Use comments to document your configuration settings.

You can get a sample configuration file for the module in the following places:

- Copy (page 21) the MVI_BSCH.TXT from the module's flash memory to your PC
- Copy the MVI_BSCH.TXT from the ProSoft Solutions CD-ROM supplied with the module
- Download the MVI_BSCH.TXT from the ProSoft Technology web site at www.prosoft-technology.com

3.1.1 Editing the Configuration File

The MVI_BSCH.TXT file is a plain ASCII text file. Use a text editor such as Notepad.exe (included with Microsoft Windows) to open and edit the file.

To open the configuration file in Notepad

1   Click the Start button, and then choose Programs
2   Expand the Programs menu, and then choose Accessories.
3   On the Accessories menu, choose **Notepad.**

4   In Notepad, open the File menu, and then choose Open
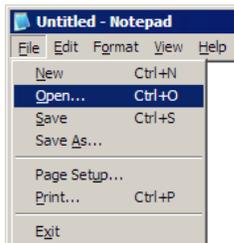
5   In the Open dialog box, select "All Files" in the Files of Type: dropdown list.
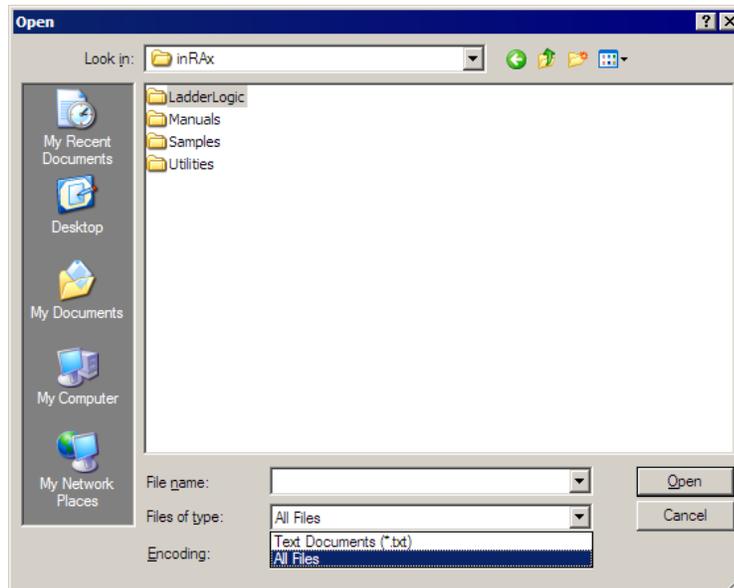
Tip: Sample configuration files are stored under the LadderLogic folder on the ProSoft Solutions CD-ROM.

6   Navigate to the folder containing the configuration file, and then select the file to edit.
7   Click Open to open the file.
8   When you have finished editing, save the file and close Notepad.

Important: Changes to the configuration file will not take effect until you download the file to the module, and then reboot the module.

3.2   Bosch Master

The [BOSCH MASTER] section of the configuration file sets the Bosch master port communication parameters, to define the database, to define the command list specifications for the master port and to specify the Flex backplane data transfer parameters. The following table lists the parameters defined in this section:

[Section]/Item	Value	Range	Description
[BOSCH MASTER]			Configuration header for Master Bosch Port.
Module Name:		Up to 80 chars	Name of the module for use on reports. Use this parameter to identify your module in your system.
Maximum Register:		1 to 3996	This parameter defines the maximum register in the database. You should size the database for your application leaving room for expansion in the future. Requests for registers outside of the range selected will be returned with an error message.
Error/Status Block Pointer:		-1 to 3995	This value represents the relative starting position in the module's internal database where the Error/Status data will be stored. The table can be placed anywhere in the module's data space. The content of the Error/Status table is updated at the frequency defined in the Error/Status Frequency. If a value of -1 is set for the parameter, the data will not be placed in the database.
Error/Status Frequency:		0 to 65535	This parameter specifies the number of program cycles between each update of the Error/Status Block data in the module. If the parameter is set to a value of 0, the data is never updated.
Write Start Register:		0 to 3995	This parameter specifies the starting register in the internal database to write over the backplane.
Write Register Count:		12 to 3996	This parameter specifies the number registers in the internal database to write over the backplane. This parameter computes the number of blocks to transfer from the module to the backplane. The number of blocks must be >= 2 for proper backplane data transfer.
Read Start Register:		0 to 3995	This parameter specifies the starting register in the internal database to fill with data read over the backplane.
Read Register Count:		12 to 3996	This parameter specifies the number registers in the internal database to consider from the read operations over the backplane. This parameter computes the number of blocks to transfer from the backplane to the module. The number of blocks must be >= 2 for proper backplane data transfer.
Protocol:		0 or 1	This parameter specifies the protocol to be used on the port. Valid Protocols are 0=Full Duplex and 1=Half Duplex.

[Section]/Item	Value	Range	Description
Termination Type:		0 or 1	This parameter specifies the termination type to be used on the port. Valid termination types are 0=BCC and 1=CRC.
Baud Rate:		110 to 115K	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entry for this field include: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 and 115.
Parity:		0 to 4	This is the Parity code to be used for the port. The coded values are as follows: 0=None, 1=Odd, 2=Even, 3=Mark and 4=Space.
Data Bits:		5 to 8	This parameter sets the number of data bits for each word used by the protocol.
Stop Bits:		1 or 2	This parameter sets the number of stop bits to be used with each data value sent.
RTS On:		0 to 65535	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted.
RTS Off:		0 to 65535	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.
Use CTS Line:		Y or N	This parameter specifies if the CTS modem control line is to be used. If the parameter is set to N, the CTS line will not be monitored. If the parameter is set to Y, the CTS line will be monitored and must be high before the module will send data. This parameter is normally only required when half-duplex modems are used for communication (2-wire).
ENQ Delay:		0 to 65535	This parameter specifies the number of milliseconds to delay after a response timeout before sending an ENQ.
Command Count:		0 to 100	This parameter specifies the number of commands to be processed by the Bosch Master port.
Command Delay:		0 to 65535	This parameter specifies the number of milliseconds to wait between issuing each command. This delay value is not applied to retries.
Error Block Pointer:		-1 to 3995	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.
Response Timeout:		0 to 65535	This parameter represents the message response timeout period in 1-ms increments. This is the time that a port configured as a master will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending upon the communication network used and the expected response time of the slowest device on the network.
Retry Count:		0 to 10	This parameter specifies the number of times a command will be retried if it fails.

3.3 Commands

The [COMMANDS] section of the configuration file sets the Bosch master port command list. This list polls Bosch slave devices attached to the Bosch master port. The module supports read and write commands. The various Bosch commands are implemented by accessing internal addresses that correspond to different functions.

The command list is formatted differently than the other sections of the configuration file. Commands are present in a block between the labels START and END. These labels inform the program where the list resides. The module's program will parse all commands after the START label until it reaches the END label or until the command count entered for the port is reached.

The function codes used for each command are either a write (0) or a read (1). Each command list record has the same format. The first part of the record contains the information relating to the MVI94-BSCH, communication module and the second part contains information required to interface to the Bosch slave device. The command structure is displayed in the following table for all functions supported:

| Module Information | ←——————→ | Device Information |

BOSCH COMMAND STRUCTURE

Column #	1	2	3	4	5	6	7	8
Function Code	**Enable Code**	**Internal Address**	**Poll Interval Time**	**Count**	**Swap Code**	**Slave Node**	**Function Code**	**Device Address**
fc0	Code	Register	Seconds	Count	Code	Node	0	Register
fc1	Code	Register	Seconds	0	Code	Node	1	Register

node = destination address

3.3.1 Enable Code

0 = Disabled

1 = Enabled with Poll Interval (seconds) utilized

2 = Conditional (executed when point in database changes)

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

To disable the command, set this parameter to 0. You can still execute commands through the processor, using a Command Control block.

To enable the command, set this parameter to 1.

- Set the Poll Interval Time to 0 to execute the command during each scan of the command list.
- Set the Poll Interval Time to a value in seconds, to execute the command at the specified interval.

To execute the command only if the internal data associated with the command changes, set this parameter to 2. This value is valid only for write commands.

3.3.2 Internal Address

This field specifies the database register to be associated with the command. If the command is a read function, the data read from the slave 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 slave device will be sourced from the address specified. Register addresses specified for commands must reside in the range specified by the Maximum Register parameter under the [BOSCH MASTER] section.

3.3.3 Poll Interval Time

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

3.3.4 Count

This parameter specifies the number of registers to be associated with the command. For a write command this will determine the number of registers (words) written to the slave device. For read commands this value must be set to 1.

3.3.5 Swap Code

This parameter defines if the data received from the Bosch slave is to be ordered differently than received from the slave 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 slave 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:

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

3.3.6 Slave Node

This parameter specifies the Bosch slave node address on the network to be considered. Values of 0 to 255 are permitted.

3.3.7 Function Code

This parameter specifies the Bosch function to be executed by the command. The following table defines the purpose of each function supported by the module.

Func	Description
0	Write Parameters
1	Read Status

3.3.8 Device Address

This parameter specifies the starting register address to be considered by the command in the Bosch slave device. Refer to the documentation of each Bosch slave device on the network for their register address assignments.

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 not exactly match the configuration/debug menus in your own module. For specific information about the configuration/debug menus in your module, refer to The Configuration/Debug Menu (page 29).

3.4.1 Required Hardware

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

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

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

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

3.4.3 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.
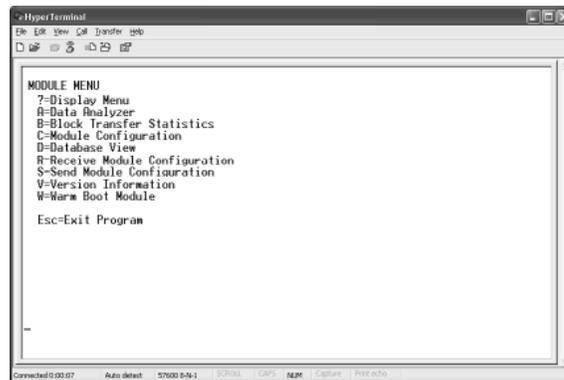
2 Press **[S]** (Send Module Configuration). The message "Press Y key to confirm configuration send!" is displayed at the bottom of the screen.
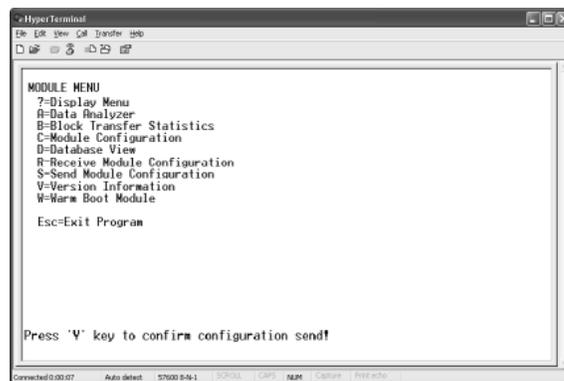
3 Press **[Y]**. The screen now indicates that the module is ready to send.
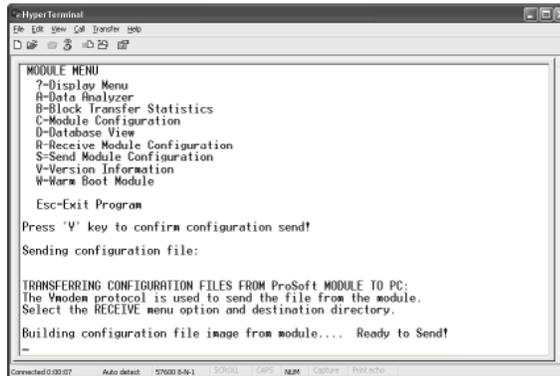
4 From the **Transfer** menu in HyperTerminal, select **Receive File**. This action opens the Receive File dialog box.
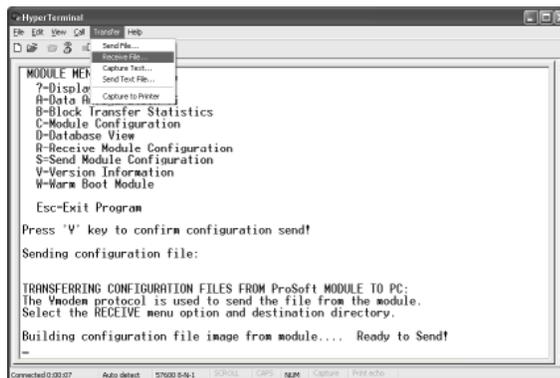
5 Use the Browse button to choose a folder on your computer to save the file.

- ▪ Note: ProSoft Technology suggests that you upload 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 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.
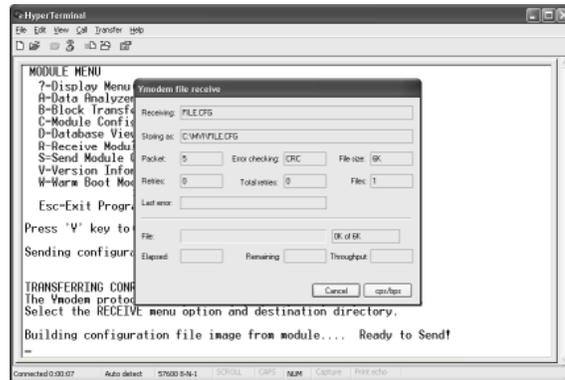
When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.
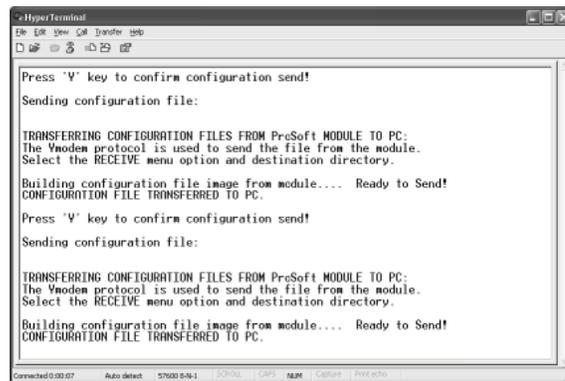
The configuration file is now on your PC at the location you specified.

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.

3.4.4 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.
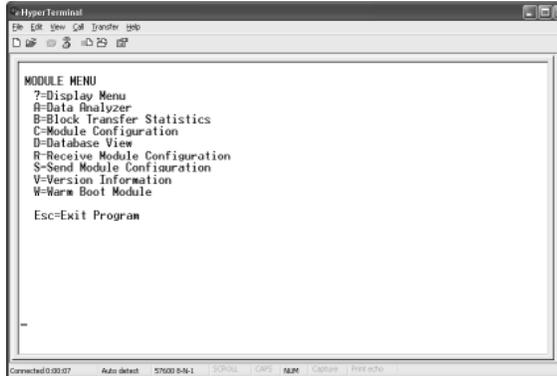
2 Press **[R]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.
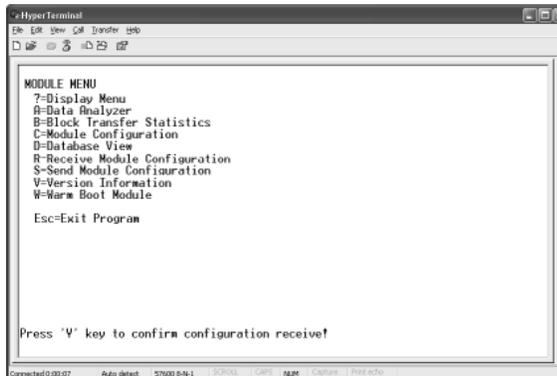
3   Press **[Y]**. The screen now indicates that the PC is ready to send.
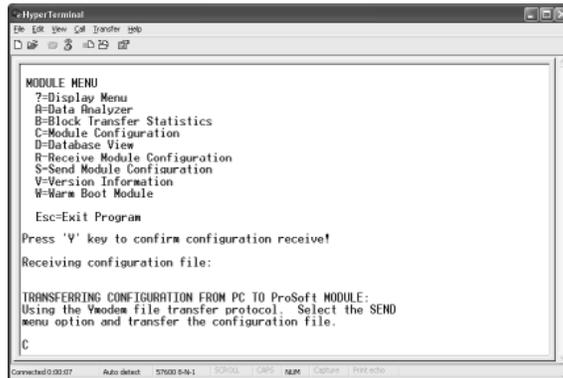
4   From the **Transfer** menu in HyperTerminal, select **Send File**.
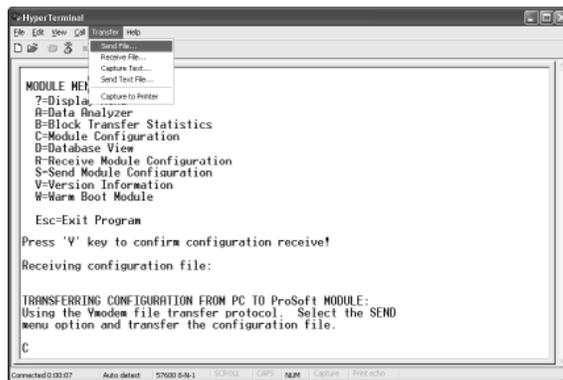
The Send File dialog appears.
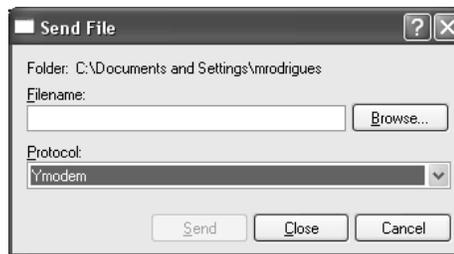
5   Use the Browse button to locate the configuration file your computer.

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 www.prosoft-technology.com.

6   Select **Ymodem** as the protocol.

7   Click the Send button. This action opens the Ymodem File Send dialog box.
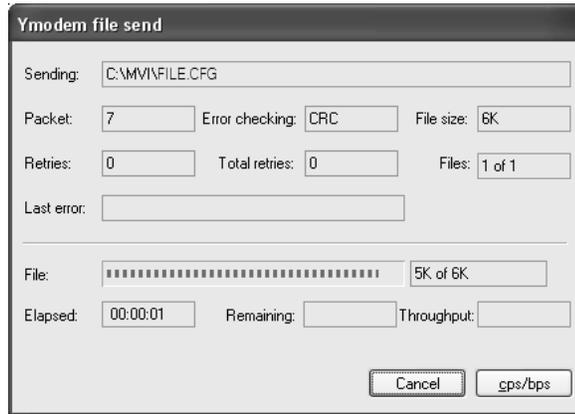
When the file transfer is complete, the module's configuration/debug screen indicates that the module has reloaded program values, and displays information about the module.
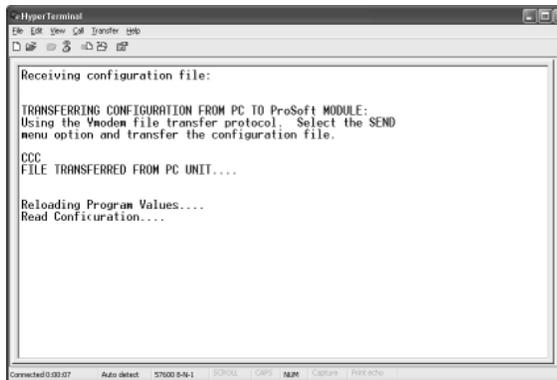
8   Your module now contains the new configuration.

4 Diagnostics and Troubleshooting

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

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

4.1 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.

4.1.1 Navigation

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

The organization of the menu structure is represented in simplified form in the following illustration:
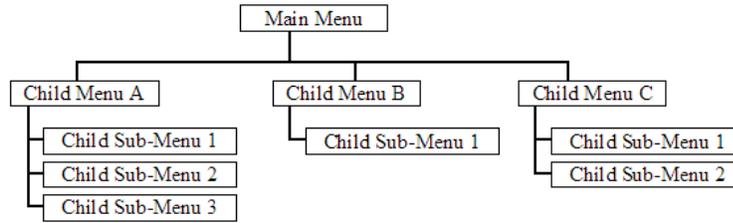
The remainder of this section shows you the menus available for this module, and briefly discusses the commands available to you.

4.1.2 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.

4.2   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 UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

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

4.4   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	None

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

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

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

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

4.5 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:

```
MAIN MENU
 ?=Display Menu
 A=Data Analyzer
 B=Backplane Transfer Statistics
 C=Restore Default Configuration
 D=Database View
 E=Command Errs
 L=Command List
 R=Receive Configuration from Remote
 S=Send Configuration to Remote
 V=Version Information
 W=Warm Boot Module
 1=Port Status          6=Port Cfg
 Esc=Cold Boot Module
```

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

4.5.1 Opening the Data Analyzer Menu

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

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

4.5.2 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.

4.5.3 Restoring the Default Configuration

Press **[C]** to the restore the configuration parameters to the default settings.

4.5.4 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.

4.5.5 Opening the Command Error List Menu

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

4.5.6 Opening the Command List Menu

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

4.5.7 Transferring the Configuration File from MVI94 module to PC

Press **[R]** to receive (upload) 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.

4.5.8 Transferring the Configuration File from PC to MVI94 module

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

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

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

4.5.9 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.

4.5.10 Warm Booting the Module

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

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

4.5.11 Cold Booting the Module

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

4.5.12 Viewing Port Communication Status

Press **[1]** from the Main Menu to view the port communication status for the application port.

Use this command to view communication status and statistics for the selected port. This information can be informative when trouble-shooting communication problems.

4.5.13 Viewing Port Configuration

Press **[6]** from the Main Menu to view configuration information for the application port.

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

4.6   Command List Menu

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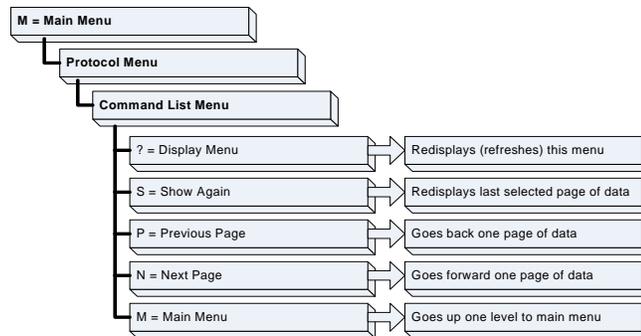
4.6.1 Redisplaying the Menu

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

4.6.2 Redisplaying the Current Page

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

Use this command to display the current page of commands. Ten commands are displayed on each page.

If an enabled command has an error, the EN field will contain a value of -1. This indicates that the command will be re-issued every 30 seconds.

4.6.3 Viewing the Previous Page of Commands

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

4.6.4 Viewing the Next Page of Commands

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

4.6.5 Returning to the Main Menu

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

4.7 Data Analyzer

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

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

4.7.1 Analyzing Data for the first application port

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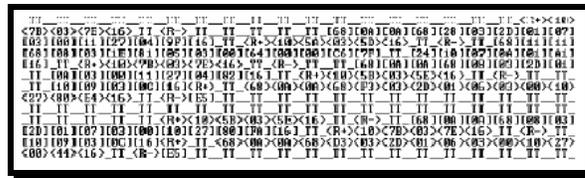
4.7.2 Analyzing Data for the second application port

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

4.7.3 Displaying Timing Marks in the Data Analyzer

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

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

4.7.4 Removing Timing Marks in the Data Analyzer

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

4.7.5 Viewing Data in Hexadecimal Format

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

4.7.6 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.

4.7.7 Starting the Data Analyzer

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

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

The Data Analyzer displays the following special characters:

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

4.7.8 Stopping the Data Analyzer

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

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

4.7.9 Returning to the Main Menu

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

4.8   Data Analyzer Tips

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

```
Data Analyzer Mode Selected
```

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

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

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

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

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

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

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

```
Transfer  Help
  Send File...
  Receive File...
  Capture Text...
  Send Text File...
  Capture to Printer
```
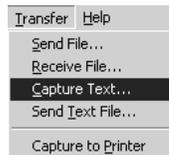
After selecting the above option, the following window will appear:
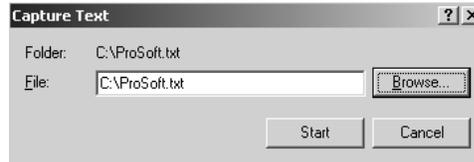
Next name the file, and select a directory to store the file in. In this example, we are creating a file ProSoft.txt and storing this file on our root C: drive. After you have done this, press the [Start] button.

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

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

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

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

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

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

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

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

When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.
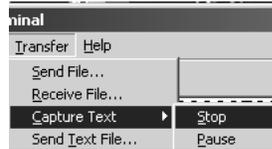
You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

4.9   LED Status Indicators

This section defines the indications provided on the MVI94-BSCH module through LEDs. A description of each LED is given below:

4.9.1 PRT1

This LED indicates data transmit and receive activity on the configuration port. When the TXD or RXD pin is active on the port, the LED will be illuminated green. When the port is not active, the LED will be in the off state.

4.9.2 U1

This LED indicates backplane data transfer operation. When the module is successfully writing data to the FLEX I/O backplane, the LED will be in the off state. When the module is reading a new block of data from the FLEX I/O backplane, the LED will be in the on state (amber). During normal operation of the module, this LED should turn on and off at a very rapid rate. If the LED never turns on, check your ladder logic to verify that the data transfer is set up correctly.

4.9.3 U2

This LED indicates communication errors on the Bosch master port. The LED is illuminated (amber) when no error exists on the port. If a communication error is recognized on the port, the LED will turn off. If the LED is turned off, check for errors in the command list to determine the error condition recognized by the module.

4.9.4 STS

This LED indicates the "health" of the module. When power is applied to the module, the LED will be illuminated. If the LED is green, the program is working correctly and the user configuration is being used. If the LED is red, the program is halted. Try restarting the module by cycling power. This should cause the module to return to its normal state.

4.9.5 PRT2

This LED indicates data transmit and receive activity on the Bosch master port. When the TXD or RXD pin is active on the port, the LED will be illuminated green. When the port is not active, the LED will be in the off state.

5   Reference

In This Chapter

5.1   Product Specifications

The MVI94 Bosch Communication I/O Module is used to interface Bosch SVB/SVK compatible slave devices (Robert Bosch Corporation's line of packaging machinery) with the Flex I/O system.

The module contains a database that is defined by the user. This database is used for the request and command messages sent from the Bosch master port to Bosch slave devices. Data areas in the database can be reserved for status and error information generated by the module under user control.

A Bosch master port is present on the communication module to continuously poll Bosch slave devices. Up to 100 user-defined commands can be defined for the port. Data read from Bosch slave devices are placed in the database. Any write requests for the Bosch slave devices are sourced with data from the database.

Commands can be activated in the module under processor control. This feature permits the processor to issue a command in the command list under program control. When a command is activated, it will be placed in the command queue for immediate execution. Normal command polling will begin after the command queue is completely processed.

The module provides a configuration/debug port for use with an external computer executing a terminal server program. The terminal server program provided with the module permits uploading and downloading of the configuration information required by the module. Additionally, the configuration/debug port provides a view into the database, communication statistics and the configuration.

5.1.1 General Specifications

Some of the general specifications include:

- Operation via simple ladder logic
- Complete setup and monitoring of module through Debug port and user configuration file
- Flex backplane interface via I/O access

5.1.2 FLEX I/O Interfaces

Specification	Description
Form Factor	Single Slot 1794 Backplane compatible
	Locate in any slot of Backplane
Backplane current load	20 mA @ 5 V
External power supply	12V to 24VDC
	340 mA to 170 mA
Operating temperature	0 to 55°C (32 to 131°F)
Storage temperature	-40 to 85°C (-40 to 185°F)
Shock	30g operational
	50g non-operational
	5g from 10150 Hz
Relative humidity	5% to 95% (non-condensing)
LED indicators	Module status
	Backplane transfer status
	Application status
	Serial activity and error LED status
Configuration Serial port (PRT1)	Mini-DIN
	RS-232
	Hardware handshaking
Application serial Port (PRT2)	Mini-DIN
	RS-232/422/485 jumper selectable
	500V optical isolation from backplane
Dimensions (with Module installed in Base)	3.7H x 3.7W x 2.7D inches
	94H x 94W x 69D mm

5.1.3 Functional Specifications

Specification	Description
General Parameters (Software Configurable)	
Internal Database	4000 registers (words) available
Communication parameters	Local Station ID: 0 to 255
	Port 0 Baud Rate: 110 to 115.2K baud
	Port 1, 2, 3 Baud Rate: 110 to 115K baud
	Stop Bits: 1 or 2 Data Size: 7 or 8 bits
	Parity: None, Even, Odd
	RTS Timing delays: 0 to 65535 milliseconds
BOSCH Modes	Full-Duplex
Error Checking	BCC

Specification	Description
BOSCH Master Driver	
General	Full-Duplex Point to Point
Configurable Parameters per Master port	Min Command Delay Number of Commands Response Timeout Retry Count Slave List Error Pointer
BOSCH Commands supported	0x00 Protected Write 0x01 Unprotected Read
Command List	Up to 100 command per master port, each fully configurable for function, slave address, register to/from addressing and word/bit count
Status Data	Error codes available on an individual command basis. In addition, a slave status list is maintained per active master port.
Polling of command list	User configurable polling of commands, including disabled, continuous and on change of data (write only)

5.2 Functional Overview

The MVI94-BSCH communication module interfaces Bosch SVB/SVK compatible slave devices with the Flex I/O system. The module contains a database that is defined by the user. This database is used for the request and command messages sent from the Bosch master port to Bosch slave devices. Data areas in the database can be reserved for status and error information generated by the module under user control.

The database also interfaces with the Flex I/O system using the Flex I/O bus (backplane). Data is made available to the PLC or any processor on a Control Net network using this backplane interface. Input and output image tables in the module present the data in the database to the backplane.

A Bosch master port is present on the communication module to continuously poll Bosch slave devices. Up to 100 user-defined commands can be defined for the port. Data read from Bosch slave devices are placed in the database. Any write requests for the Bosch slave devices are sourced with data from the database.

Commands can be activated in the module under processor control. This feature permits the processor to issue a command in the command list under program control. When a command is activated, it will be placed in the command queue for immediate execution. Normal command polling will begin after the command queue is completely processed.

The module provides a Configuration/Debug port for use with an external computer using a terminal program such as HyperTerminal. Additionally, the Configuration/Debug port provides a view into the database, communication statistics and the configuration.

5.2.1 Database

Central to the functionality of the module is the database. This database is used as the interface between remote Bosch slave devices and the Flex I/O bus. The size, content and structure of the database are completely user defined.

The Flex I/O bus reads data from and write data to the database using the backplane interface. The module interfaces data contained in remote Bosch slave devices to the database using the Bosch master port. User commands are issued out the master port from a command list. These commands gather or control data in the Bosch slave devices. The following illustration shows the relationships discussed above:
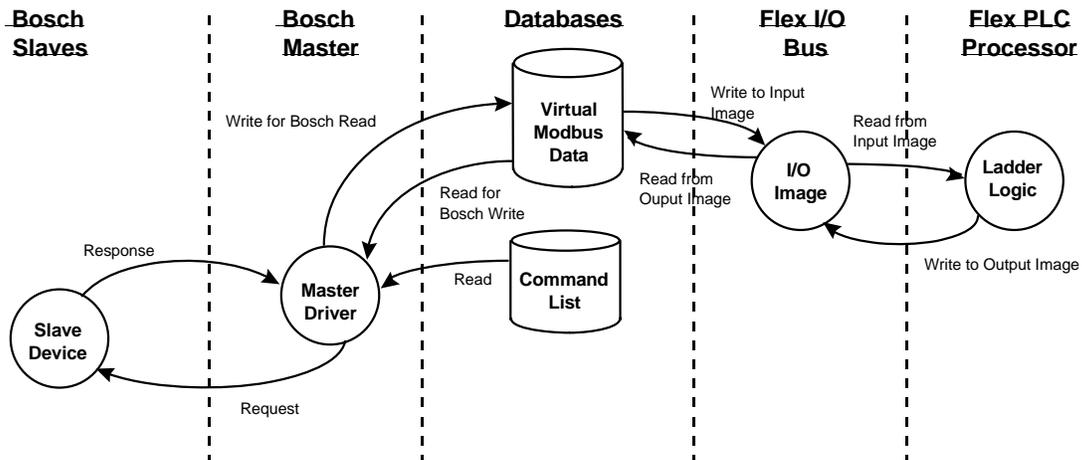
5.2.2 Data Transfer

Data is transferred over the backplane using the module's input and output images. The module is configured with an eight-word input image and a seven-word output image. The module and the Flex processor use these images to page data and commands. The input image is set (written) by the module and is read by the Flex processor. The output image is set (written) by the Flex processor and read by the module. The following illustration shows this relationship:

DATA TRANSFER OPERATION
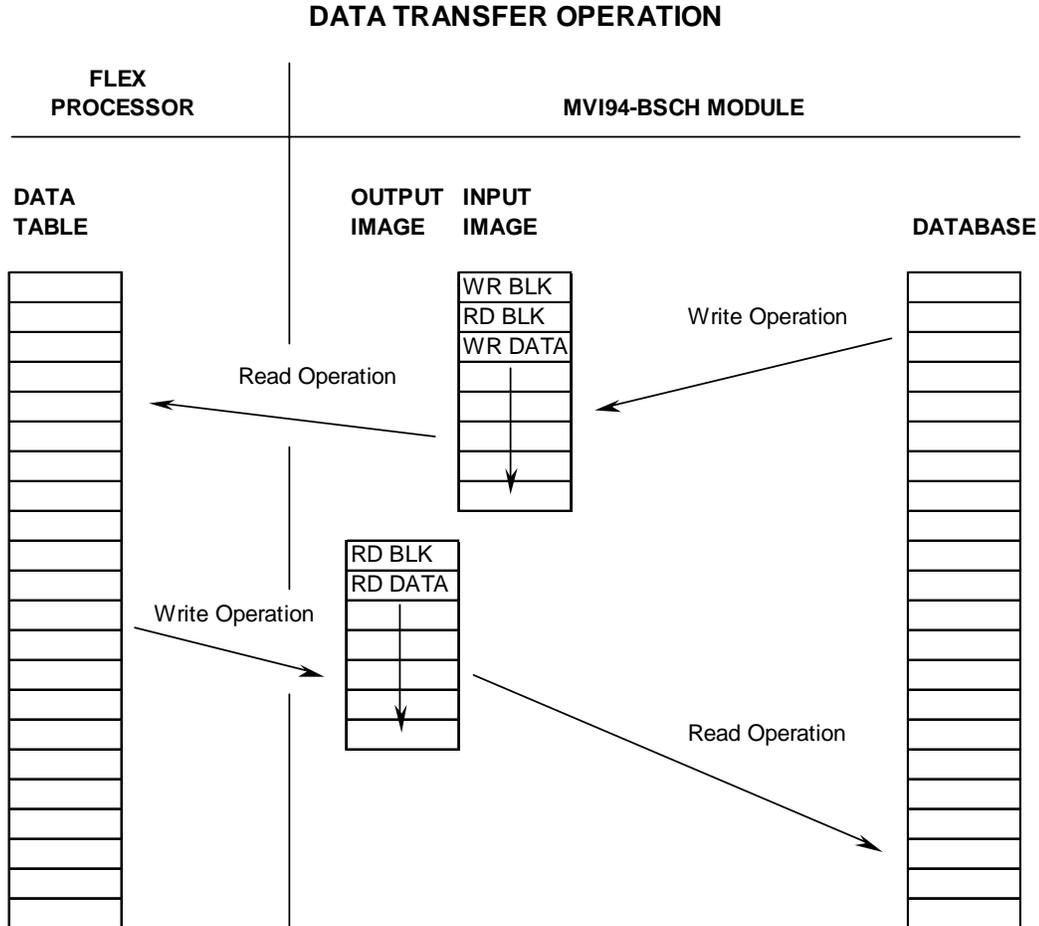
The module's program is responsible for setting the block identification code used to identify the data block written and the block identification code of the block it wants to read from the processor. User configuration information determines the read (Read Start Register) and write (Write Start Register) locations in the database and the amount of data transferred (Read Register Count and Write Register Count). Each read and write operation transfers a six-word data area. The write operation contains a two-word header that defines the block identification code of the write data and the block identification code of the read block requested. These identification codes are in the range of 0 to 666. A value of zero indicates that the block contains no data and should be ignored.

The first valid block identification code is one and refers to the first block of six words to be read or written.

The module and the processor constantly monitor input and output images. How does either one know when a new block of data is available? Recognizing a change in the header information of the image (word 0) solves the problem. For example, when the module recognizes a different value in the first word of the output image, new read data is available. When the processor recognizes a new value in the first word of the input image, new write data is available. This technique requires the storage of the previously processed data block identification code. The following illustration shows the normal sequence of events for data transfer:

NORMAL DATA TRANSFER OPERATION
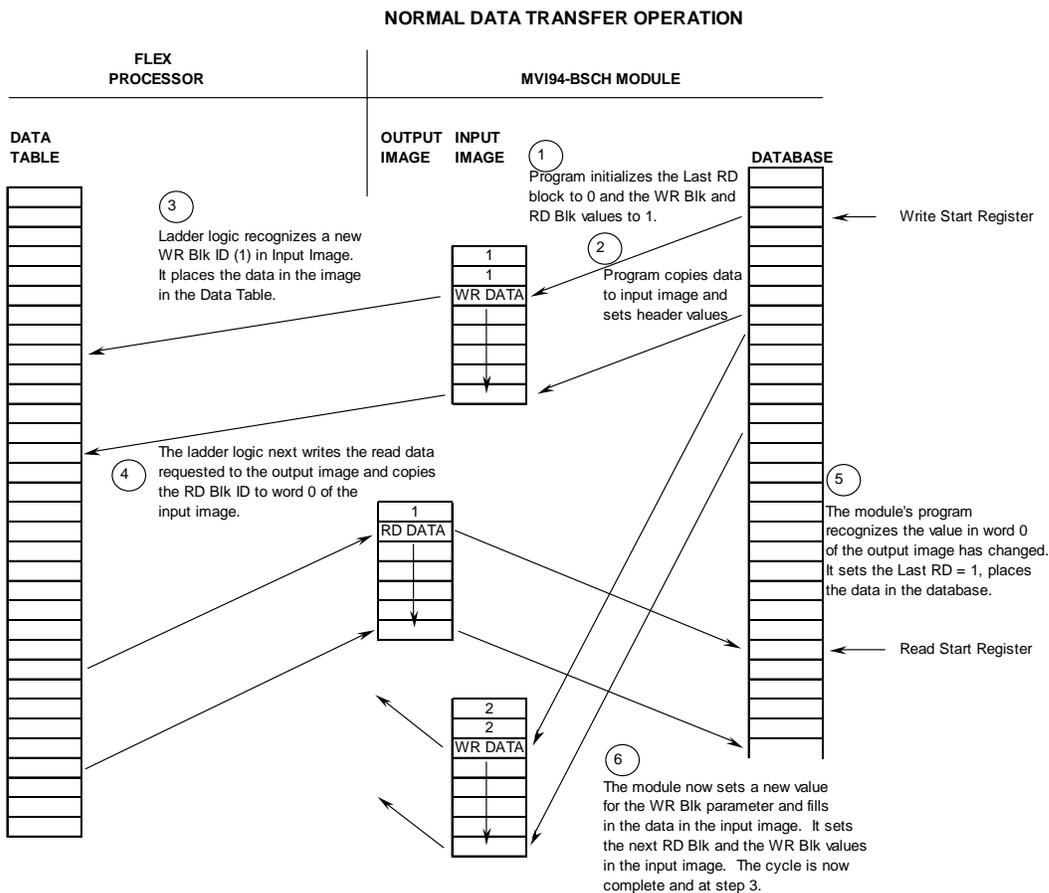
1. The steps outlined in the diagram above are discussed below:
2. During program initialization, the write and read block identification codes are set to one. The last block read variable is set to zero.
3. The program copies the first six-word block of the database starting at the user defined Write Start Register to the input image (words 2 to 7). It then sets the current read block code in word 1 of the input image. To "trigger" the write operation, the program places the current write block code into word 0 of the input image.

4 The Flex processor recognizes a new value in word 0 of the input image (based on the last_write_block_code not equal to write_block_code) in its ladder logic. The ladder logic computes the offset into the file based on the following formula:

```
write_file_offset = (write_block_code - 1) * 6
```

The new data contained in the input image (words 2 to 7) is copied to the offset in the processor's user data file. The last_write_block_code storage register in the processor is updated with the new write_block_code.

Note: If the data area transferred from the module exceeds the size of a single user file in the Flex processor, logic will be required to handle multiple files.

5 The ladder logic next examines the value of the read_block_code and computes the offset into the read data file as follows:

```
read_file_offset = (read_block_code - 1) * 6
```

The required 6-word, read data is copied to the module's output image (words 1 to 6). To "trigger" the transfer operation, the ladder logic moves the read_block_code into word 0 of the output image.

6 The module's program recognizes the new read_block_code. It transfers the data to the correct offset in the database using the following function:

```
Database_offset = Read_Start_Register + (read_block_code - 1) * 6
```

The module sets the last_read_block_code to the value of read_block_code.

7 The module now selects the next read and write blocks. The data for the write operation is placed in the input image and the read_block_code is set. The module "triggers" the transfer operation by setting the new write_block_code in word 0 of the input image. The sequence continues at step 3.

The discussion above is for normal data transfer operation. The following table lists the block identification codes used by the module.

Block Identification Codes

Type	Number	Description
R/W	1 to 666	Data blocks used to transfer data from the module to the backplane and from the backplane to the module. The module's input/output images are used for the data transfers.
R	2001 to 2006	These blocks transfer a list of commands to execute from a device on the backplane to the module. The number of commands to be considered is coded in the block number. For example, to add three commands to the command queue, use block 2003.
R	9998	Warm boot the module. When the module receives this block, it will reset all program values using the configuration data.
R	9999	Cold boot the module. When the module receives this block, it will perform a hardware restart.

Data is transferred between the processor and the module using the block identification codes of 1 to 666. The other block codes control the module from the processors ladder logic. They are implemented when the ladder logic needs

to control the module. In order to use one of the blocks, the ladder logic inserts the data and code in the output image of the module. The data should be set before the code is placed in the block. This operation should be performed after the receipt of a new write block from the module. A discussion of each set of codes is given below:

Commands (Blocks 2001 to 2006)

These control blocks are sent from the processor to the module to execute one or more commands in the module's command list out the Bosch master port. Commands selected for execution need not have the Enable Code set (1 or 2) but must be valid commands. The format for this block is shown below:

Block 2001 to 2006 Structure

Word	Description
0	Command queue block identification code of 2001 to 2006.
1	This word contains the index in the command list for the first command to be entered into the command queue.
2	This word contains the index in the command list for the second command to be entered into the command queue.
3	This word contains the index in the command list for the third command to be entered into the command queue.
4	This word contains the index in the command list for the fourth command to be entered into the command queue.
5	This word contains the index in the command list for the fifth command to be entered into the command queue.
6	This word contains the index in the command list for the sixth command to be entered into the command queue.

When the module receives one of these blocks, it examines word 0 of the output image. This word defines the number of commands contained in the block. The command count is determined by subtracting 2000 from the word value. This permits the controller to set from one to six commands into the command queue. The indexes submitted in the block should be valid for the command list. After the module determines the number of commands to consider, it inserts each command in the command queue. The response message sent from the module to the processor is as follows:

Block 2001 to 2006 Response

Word	Description
0	This word contains the block 2001 to 2006 requested by the processor.
1	This word contains the next read request block identification code.
2	This word contains the number of commands in the block placed in the command queue.
3	Not used.
4	Not used.
5	Not used.
6	Not used.

The ladder logic can examine word 2 of the input image to determine the number of commands placed in the command queue. Any errors associated with the command can be viewed in the command list error table.

Warm-Boot (Block 9998)

This block does not contain any data. When the processor places a value of 9998 in word 0 of the output image, the module will perform a warm-start. This involves clearing the configuration and all program status data. Finally, the program will load in the configuration information from the Flash ROM and begin running. There is no positive response to this message other than the status data being set to zero and the block polling starting over.

Cold-Boot (Block 9999)

This block does not contain any data. When the processor places a value of 9999 in word 0 of the output image, the module will perform a hardware restart. This will cause the module to reboot and reload the program. There is no positive response to this message other than the status data being set to zero and the block polling starting over.

5.2.3 Command List

The MVI94-BSCH communication module's primary services are data concentration and communication gateway. The Bosch master port polls Bosch slave devices based on user-defined commands and places the data in the database. The Flex I/O bus interfaces with this database to a Flex processor. The user is responsible for defining the structure and content of the database.

In order to interface the database with Bosch slave devices, you must construct a command list. The commands in the list specify the Bosch slave device to be utilized, the function to be performed (read or write), the data area in the device to interface with and the position in the database to be associated with the device data. Up to 100 commands can be entered for this purpose. The list is processed from top (command #0) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in seconds between the issuance of a command. If the user specifies a value of 10 for the parameter, the command will be executed no more frequently than every 10 seconds. Additionally, a user specified time delay could be inserted between the issuance of each command. This is useful for slow responding slave devices.

Write commands have a special feature, as they can be set to execute only if the data in the write command changes. If the data in the command has 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 Bosch network. In order to implement this feature; set the enable code for the command to a value of 2.

5.3 Cable Connections

The application ports on the MVI94-BSCH module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

5.3.1 RS-232

When the RS-232 interface is selected, the use of hardware handshaking (control and monitoring of modem signal lines) is user definable. If no hardware handshaking will be used, the cable to connect to the port is as shown below:
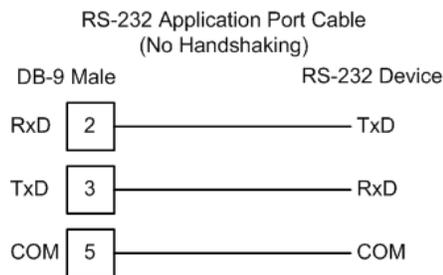
RS-232: Modem Connection

This type of connection is required between the module and a modem or other communication device.
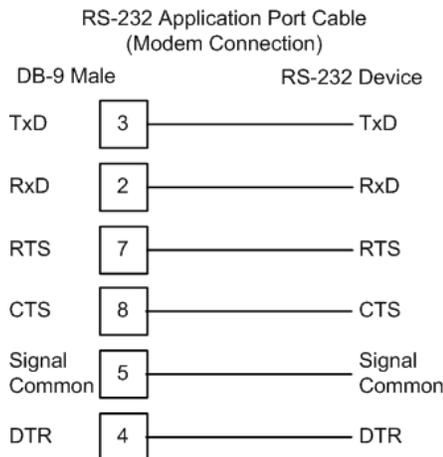
The "Use CTS Line" parameter for the port configuration should be set to 'Y' for most modem applications.

RS-232: Null Modem Connection (Hardware Handshaking)

This type of connection is used when the device connected to the module requires hardware handshaking (control and monitoring of modem signal lines).
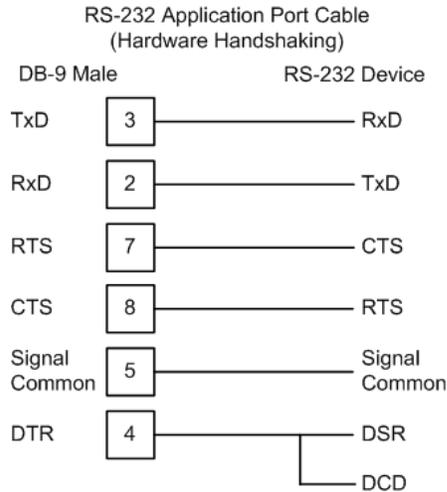
RS-232: Null Modem Connection (No Hardware Handshaking)

This type of connection can be used to connect the module to a computer or field device communication port.
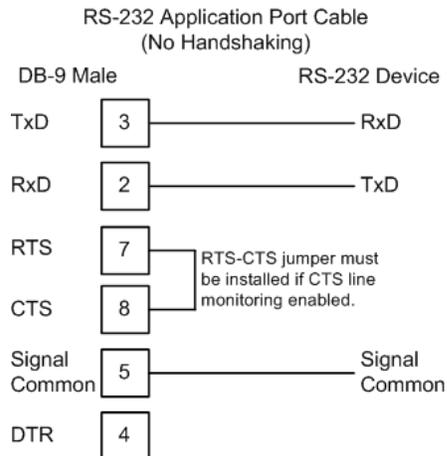
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.2 RS-232 Configuration/Debug Port

This port is physically a Mini-DIN connection. A Mini-DIN 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:

RS-232 Config/Debug Port Cable

DB-9 Male		Config/Debug Port
RxD	2	TxD
TxD	3	RxD
COM	5	COM
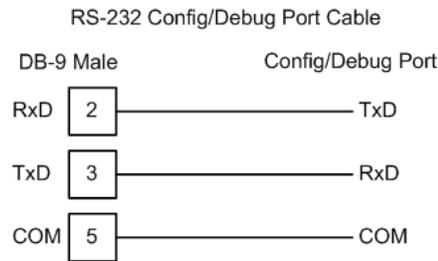
5.3.3 RS-485

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

RS-485 Application Port Cable

DB-9 Male		RS-485 Device
TxD+/RxD+	1	TxD+/RxD+
TxD-/RxD-	8	TxD-/RxD-
Signal Common	5	Signal Common
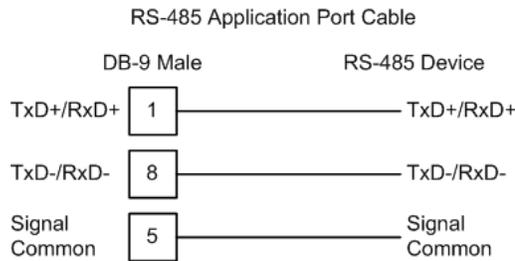
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120-ohm terminating resistor on the RS-485 line.

5.3.4 RS-422

RS-422 Application Port Cable

DB-9 Male		RS-422 Device
TxD+	1	RxD+
TxD-	8	RxD-
Signal Common	5	Signal Common
RxD+	2	TxD+
RxD-	6	TxD-
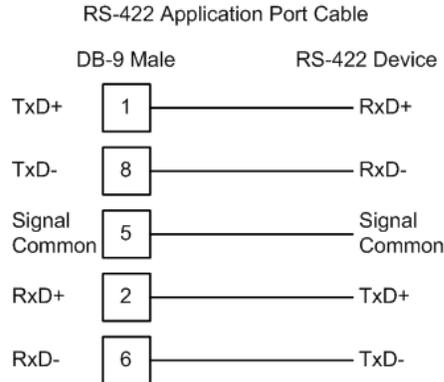
RS-485 and RS-422 Tip

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

5.3.5 DB9 to Mini-DIN Adaptor (Cable 09)

12.0 "

Cable Assembly

J1   J2

J1	RS 232	RS 485	RS 422	J2
6	DSR		RXD-	1
4	DTR			2
7	RTS			3
2	RXD		RXD+	4
1	DCD	TXRXD+	TXD+	5
8	CTS	TXRXD-	TXD-	6
3	TXD			7
5	GND	GND	GND	8
9	N/C			
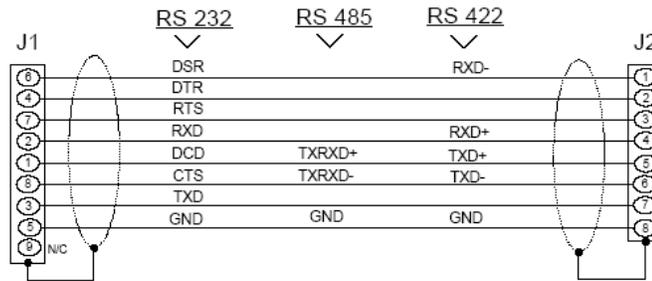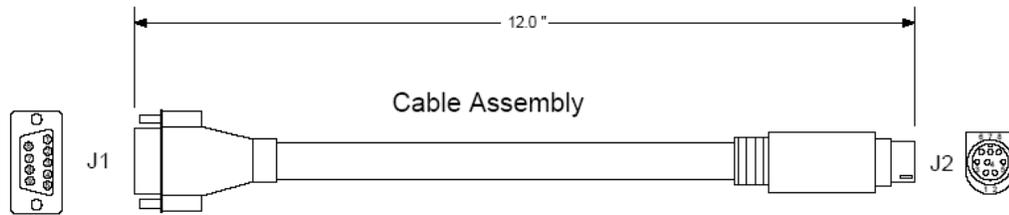
Wiring Diagram

5.4   Error/Status Data

The module error/status data areas are discussed in this section. The module contains two areas related to this data. The user defines the location of these data sets in the database of the module. The error/status data contains module data and the command error list data set contains the errors associated with the command list.

5.4.1   Error/Status Data

The error/status data table is located at the address assigned by the user. If the address is set to -1 or the frequency parameter is set to 0, the data will not be placed in the database. It will only be available through the Configuration/Debug Port. If valid address and frequency values are assigned, the module will update the data area.

The data area will be initialized with zeros whenever the processor is initialized. This occurs during a cold-start (power-on), reset (reset push-button pressed) or a warm-boot operation (commanded or loading of new configuration). The data area is a 32-word register block. The structure of the block is shown below:

System Status

Word	Description
0	Program Cycle Counter
1	Last read block identification code.
2	Current read block identification code.
3	Current write block identification code.
4	Current command index value.
5	Total number of commands to process by the module.
6	Total number of commands in the command queue.
7	Current state code for the Bosch master port.
8	Current communication state code for the Bosch master port.

System Information

Word	Description
9	Product Name (ASCII) = BSCH
10	
11	Revision (ASCII)
13	Operating System Rev(ASCII)
15	Production Run Number (ASCII)

Block Transfer Statistics

Word	Description
26	Total number of read block operations
27	Total number of write block operations
28	Total number of blocks parsed
29	Total number of event blocks received

Word	Description
30	Total number of command blocks received.
31	Total number block transfer errors.

Master Port Statistics

Word	Description
17	Number of Command Requests
18	Number of Command Responses
19	Number of Command Errors
20	Number of Requests
21	Number of Responses
22	Number of Errors Received
23	Number of Errors Sent
24	Reserved
25	Reserved

5.4.2 Command Error List

Each command in the command list has a reserved word value for a status/error code. This error data list can be read using the Configuration/Debug Port. Additionally, the data can be placed in the database of the module. The configuration parameter "Command Error Pointer" defines the register address in the database where the data will be placed.

The first word in the register location defined contains the status/error code for the first command in the port's command list. Each successive word in the command error list is associated with the next command in the list. Therefore, the size of the data area is dependent upon the number of commands defined.

Refer to the following Error Codes section to interpret the status/error codes present in the data area.

5.4.3 Error Codes

The module error codes are listed in this section. Error codes are separated into Bosch error codes and module error codes.

Bosch Error Codes

These error codes are returned to the module from Bosch slave devices attached to the master port. The error codes are listed in the following table.

Code	Description
16	Command Byte Incorrect
80	Illegal Data Address
96	Access Denied
112	Illegal Data Value

Module Error Codes

Module error codes are generated by the module's program. These errors can result due to configuration or communication problems. These errors are stored in the Bosch master, command list error table. A word is allocated for each command in the memory area.

The following table lists the errors returned by the module for communication errors:

Code	Description
-1	CTS modem control line not set before transmit
-2	Timeout while transmitting message
-10	Timeout waiting for ACK
-11	Timeout waiting for response after request
-20	NAK received
-21	Incorrect BCC in response

The following table lists the errors returned by the module for errors found when parsing the command list:

Code	Description
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or > 255)
-44	Count parameter set to 0
-45	Invalid function code
-46	All parameters set to 0

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

5.5   Configuration Form

This section contains configuration forms that will aid in the configuration of the module. If you design your system before trying to directly implement it, you will have a greater chance of success. Fill in the configuration form for your application, and then edit the configuration text file.

[Section]/Item	Value	Range	Description
[BOSCH MASTER]			Configuration header for Master Bosch Port.
Module Name:		Up to 80 chars	Name of the module for use on reports. Use this parameter to identify your module in your system.
Maximum Register:		1 to 3996	This parameter defines the maximum register in the database. You should size the database for your application leaving room for expansion in the future. Requests for registers outside of the range selected will be returned with an error message.

[Section]/Item	Value	Range	Description
Error/Status Block Pointer:		-1 to 3995	This value represents the relative starting position in the module's internal database where the Error/Status data will be stored. The table can be placed anywhere in the module's data space. The content of the Error/Status table is updated at the frequency defined in the Error/Status Frequency. If a value of -1 is set for the parameter, the data will not be placed in the database.
Error/Status Frequency:		0 to 65535	This parameter specifies the number of program cycles between each update of the Error/Status Block data in the module. If the parameter is set to a value of 0, the data is never updated.
Write Start Register:		0 to 3995	This parameter specifies the starting register in the internal database to write over the backplane.
Write Register Count:		12 to 3996	This parameter specifies the number registers in the internal database to write over the backplane. This parameter computes the number of blocks to transfer from the module to the backplane. The number of blocks must be >= 2 for proper backplane data transfer.
Read Start Register:		0 to 3995	This parameter specifies the starting register in the internal database to fill with data read over the backplane.
Read Register Count:		12 to 3996	This parameter specifies the number registers in the internal database to consider from the read operations over the backplane. This parameter computes the number of blocks to transfer from the backplane to the module. The number of blocks must be >= 2 for proper backplane data transfer.
Protocol:		0 or 1	This parameter specifies the protocol to be used on the port. Valid Protocols are 0=Full Duplex and 1=Half Duplex.
Termination Type:		0 or 1	This parameter specifies the termination type to be used on the port. Valid termination types are 0=BCC and 1=CRC.
Baud Rate:		110 to 115K	This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200. Valid entry for this field include: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 and 115.
Parity:		0 to 4	This is the Parity code to be used for the port. The coded values are as follows: 0=None, 1=Odd, 2=Even, 3=Mark and 4=Space.
Data Bits:		5 to 8	This parameter sets the number of data bits for each word used by the protocol.
Stop Bits:		1 or 2	This parameter sets the number of stop bits to be used with each data value sent.
RTS On:		0 to 65535	This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted.

[Section]/Item	Value	Range	Description
RTS Off:		0 to 65535	This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.
Use CTS Line:		Y or N	This parameter specifies if the CTS modem control line is to be used. If the parameter is set to N, the CTS line will not be monitored. If the parameter is set to Y, the CTS line will be monitored and must be high before the module will send data. This parameter is normally only required when half-duplex modems are used for communication (2-wire).
ENQ Delay:		0 to 65535	This parameter specifies the number of milliseconds to delay after a response timeout before sending an ENQ.
Command Count:		0 to 100	This parameter specifies the number of commands to be processed by the Bosch Master port.
Command Delay:		0 to 65535	This parameter specifies the number of milliseconds to wait between issuing each command. This delay value is not applied to retries.
Error Block Pointer:		-1 to 3995	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.
Response Timeout:		0 to 65535	This parameter represents the message response timeout period in 1-ms increments. This is the time that a port configured as a master will wait before re-transmitting a command if no response is received from the addressed slave. The value is set depending upon the communication network used and the expected response time of the slowest device on the network.
Retry Count:		0 to 10	This parameter specifies the number of times a command will be retried if it fails.

Module Information ←——————→ Device Information

BOSCH COMMAND ENTRY FORM

Column #	1	2	3	4	5	6	7	8
	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Slave Node	Function Code	Device Address

6 Support, Service & Warranty

In This Chapter

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

1 Product Version Number
2 System architecture
3 Network details

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

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

6.1 How to Contact Us: Technical Support

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

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com
Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com
Languages spoken include: French, English

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

+1.661.716.5100, support@prosoft-technology.com
Languages spoken include: English, Spanish
For technical support calls within the United States, an after-hours answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer your questions.

Brasil (location in Sao Paulo)

+55-11-5084-5178, eduardo@prosoft-technology.com
Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns:

a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.

b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 63). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.

c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.

d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, and so on.

6.2.2 Procedures for Return of Units Under Warranty:

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

a) A replacement module will be shipped and invoiced. A purchase order will be required.

b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

6.2.3 Procedures for Return of Units Out of Warranty:

a) Customer sends unit in for evaluation

b) If no defect is found, Customer will be charged the equivalent of $100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.

c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

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

6.3   LIMITED WARRANTY

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

6.3.1 What Is Covered By This Warranty

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

b) *Warranty On Services*: Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranteed in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.3.2 What Is Not Covered By This Warranty

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

b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.

c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

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

6.3.4 Intellectual Property Indemnity

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

a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated or reproduced in any form without prior written consent from ProSoft.

b) ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.

c) Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.

d) Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.

e) Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.

f) **Additional Restrictions Relating To Software And Other Intellectual Property**

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

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 65) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

6.3.6 Limitation of Remedies **

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

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

6.3.7 Time Limit for Bringing Suit

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

6.3.8 No Other Warranties

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

6.3.9 Allocation of Risks

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

6.3.10 Controlling Law and Severability

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

Index