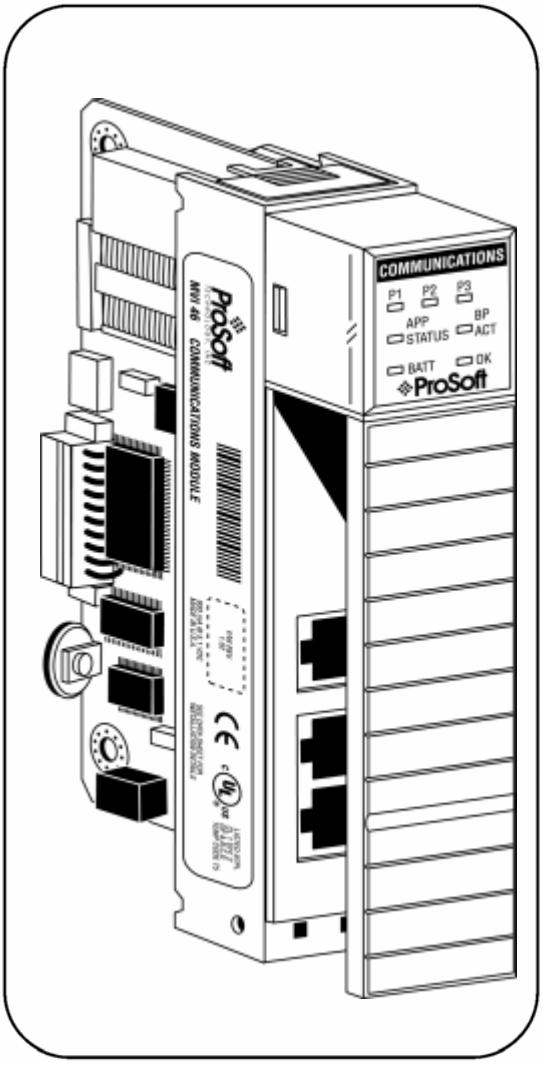
inRAx



MVI46-3964R

SLC Platform

Siemens 3964R Protocol

User Manual

June 08, 2006

ProSoft TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation SLC hardware, the MVI46-3964R Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

Your Feedback Please

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

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

MVI46-3964R User Manual
June 08, 2006

Contents

1 Product Specifications

In This Chapter

The MVI46 3964R module from ProSoft Technology allows point-to-point communication between an SLC controller and a partner with 3964R (with or without RK512) communication capability.

1.1 General Specifications

Specification	Description
Backplane Current Load	800 ma @ 5V (from backplane)
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	–40 to 85°C (–40 to 185°F)
Shock	30g operational, 50g non-operational
Relative Humidity	5 to 95% (non-condensing)
Vibration	5 g from 10150 Hz
LED indicators	Module status, Backplane transfer status, Application status, Serial activity and error LED status
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Application Serial port (PRT1, PRT2) (Serial Modules)	(2) RJ45 RS-232/422/485 Application ports

1.1.1 Functional Specifications

- The MVI46-3964R and the SLC processor communicate via M0 and M1 files.
- Single-slot, SLC backplane compatible
- Multiple modules can be placed in a rack up to the chassis power supply limit
- While in 3964R with RK512 mode on PRT1 and PRT2, DB-SEND and DB-FETCH instructions can be initiated from the SLC processor or received by the communications partner

- While in 3964R without RK512 mode on PRT1 and PRT2, data can be sent from the SLC processor or be received from the communications partner
- The maximum transfer rate is 512 bytes in any mode
- Data byte swapping can be configured to adjust to a different word format
- While in 3964R with RK512 mode on PRT1 and PRT2, evaluation of the header information DB (data block), DW (data word) and coordination bytes 9 and 10 is possible
- Communication activity and diagnostics are available through LEDs and acknowledgment telegrams
- Example programs are provided for using the 3964R protocol with a Siemens ASM 420 / ASM 424 MOBY-I/E communication interface

Ladder Logic

Ladder logic programming in the SLC processor is required in order to enable and support the Siemens 3964R protocol functionality. The ladder program handles the encoding/decoding of data transferred from the module as well as the initiation of protocol-specific functionality in the module. Example ladder programs are provided with the module to ease the implementation of the module in the user application.

2 Functional Overview

In This Chapter

2.1 Introduction

2.1.1 Products in the Environment

The 3964R protocol can be installed in all local SLC chassis with at least one controller.

Further information about SLC and the SLC environment can be obtained through your Rockwell Automation branch office.

Product Compatibility

The communication between the MVI46 and the SLC processor is realized through M0/M1 data transfer.

2.1.2 The 3964R Protocol in General

The 3964R protocol defined by Siemens is used for bi-directional data exchange between two peers through a bit-serial point-to-point connection. This protocol may be additionally embedded in the RK512 telegram level. If 3964R is used with RK512, each participant can send jobs to his partner and has read **(DB-FETCH)** and write access **(DB-SEND)** to the partner's data. The data exchange is realized in the form of messages and response messages. The maximum user data volume is 512 byte per job and 128 byte per message.

3964R without RK512 allows only block wise sending and receiving of data. The detailed handshake procedure through messages and response messages is not applicable in this case.

If both partners want to send a job (3964R with RK512) or data (3964R without RK512) at the same time, the resulting initialization conflict will be solved through the high/low priority setup. In such a case one partner will be allocated high priority and the other one low priority. Thus, in case of an initialization conflict the device with low priority will defer its job whereas the device with the high priority will be able to send a job.

The safety of data transmission on the line is guaranteed by a **BCC** checksum.

You should have sufficient knowledge about the 3964R protocol in order to understand the operation of the driver for the MVI46-3964R and to make efficient use of the example programs.

2.1.3 MVI46-3964R Operation

The protocol realized on the MVI46-3964R has the following special features:

- Immediately after power-up and completion of the internal/external initialization procedures the MVI46-3964R with installed 3964R protocol is ready to serve as an interface between a SLC processor and one or two communication partners using 3964R protocol. The parameterization of the module is realized through a configuration file. This is where a differentiation between the communication modes 3964R with RK512 and 3964R without RK512 takes place. The tables in Chapter 4 show possible entries and configuration examples.
- The communication mode 3964R without RK512 supports processing of up to 512 bytes.
- The serial transmission parameters are optional and can be allocated independently to each interface. Possible baud rates are 300 to 115200**.** The following parity setups are possible: **Even, Odd and None**.

2.2 General Concepts

2.2.1 Module Power Up

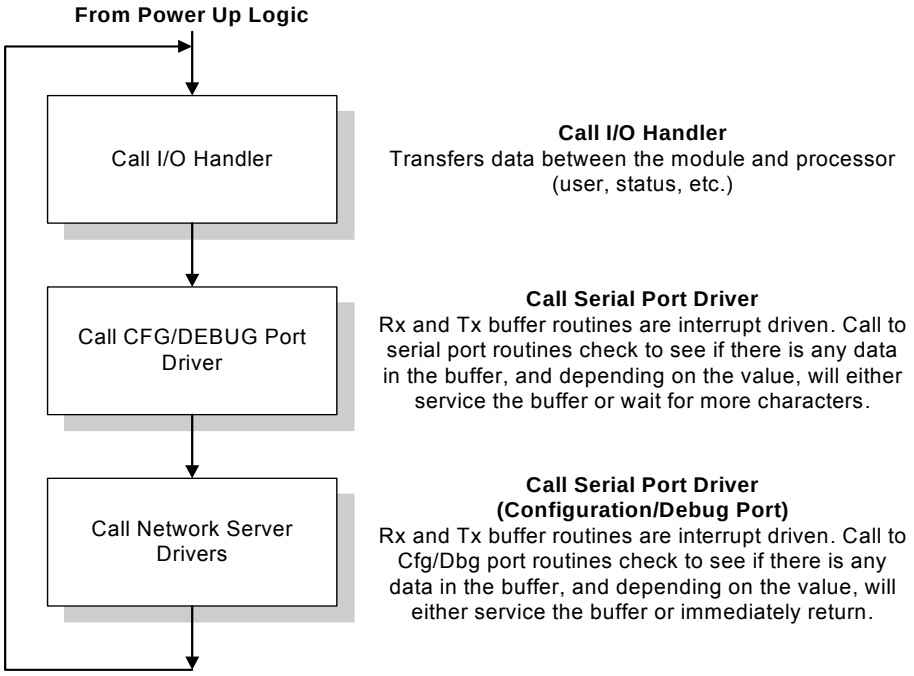
On power up the module begins performing the following logical functions:

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

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

2.2.2 Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the following functions:



2.3 Communications

2.3.1 Job Allocation by the SLC Controller

This chapter describes the different types of jobs and the relevant parameters to be entered. For a better understanding a printout of the example, program MVI46_3964.RSS is recommended.

Every job that is transferred to or from the MVI46-3964R consists of defined header data and user data.

The Job structure in general:

Word:	High Byte:	Low Byte:
0	Header data	
..	..	
9	Header data	
10	User data	
..	..	

Word:	High Byte:	Low Byte:
137	User data	

2.3.2 Header Data

The header data consists of 10 words with a job identifier and parameter data.

The header data is to be entered either in the file **HEADER_P2** to send to port 2 or the file **HEADER_P3** to send to port 3.

The header in general:

Word:	High Byte:	Low Byte:
0	Job identifier (hex)	
1	00	Data Block (DB)
2	00	Data Word (DW)
3	Number of data words/data bytes	
4	Coordination byte 9	Coordination byte 10
5	00	Data Type
6	00	00
7	00	00
8	00	00
9	00	00

The meaning of the entries and possible values are described below.

Job Identifier

The job identifiers are defined as follows:

Send jobs to a CP:

Job:	Identifier:
3964R with RK512: DB-SEND to CP	0041hex
3964R with RK512: DB-FETCH to CP	0045hex
3964R without RK512: Sending to CP (MOBY-I/E)	00FFhex

The MVI46-3964R receives jobs from a CP automatically without extra parameterization. To differentiate the header data the MVI46-3964R adds a header to the user data received and transfers it to the SLC processor.

Receive jobs from a CP:

Job:	Identifier:
3964R with RK512: DB-SEND from CP	1141hex
3964R with RK512: DB-FETCH from CP	1145hex
3964R without RK512: Receiving from CP (MOBY-I/E)	11FFhex

Data Block (DB)

Number of the data block. This data block must exist in the communication partner (3964R with RK512 only, otherwise to be set to "0").

Data Word (DW)

Number of the data word in the selected data block. The selected data block must contain this number (3964R with RK512 only, otherwise to be set to "0").

Number of Data Words/Data Bytes

Dependent on the type of communication mode used on port 2/3 of the module, this value will be interpreted as number of data words or number of data bytes. In any communication mode except for "3" (MOBY-I/E) it means data words. This value defines whether a job will be carried out with or without subsequent telegram. For jobs without subsequent telegram the value will be 1 ... 64 words (1 … 128 bytes), and for jobs with subsequent telegram(s) it will be 65 ... 256 words (129 … 512 bytes).

Coordination Byte 9/10

3964R with RK512 send jobs offer the possibility of entering so called coordination bytes (byte 9 and 10 in the RK512 telegram header). Otherwise to be set to "0".

Data Type

The send message can request a data type of Data Words (44h) or Marker (4Dh). If the value is 0 (zero), then data words is used.

2.3.3 Receive Jobs

To receive jobs from a CP only, no additional parameterization is necessary. The user data received is stored in a file in the SLC processor.

The following table shows an example of the data *stored* in RECVHDR2 in case of a DB-Send from a CP. The CP is connected to port 2.

Word:	High Byte:	Low Byte:
0	1141hex	
1	00	Data Block (DB)
2	00	Data Word (DW)
3	Number of data words (total number) [1]	
4	Number of data words (actual block) [2]	
5	00	00
6	00	00
7	00	00

Word:	High Byte:	Low Byte:
8	00	00
9	00	00

[1] = This is the total number of data words for the whole job.

[2] = This is the number of data words stored in RECVMSG2.

The user data is copied to **RECVMSG2**. This is the file you should operate with.

For port 3, which is similar to port 2, the file for storing the user data received is called **RECVMSG3**.

2.3.4 Send Jobs

To send a job to a CP, the next step is to define the appropriate header for the job. The following table shows an example of the header data for a DB-Send job to a CP with the following parameters:

DB = 10, DW = 0, 128 words, coordination byte 9 = FFhex, coordination byte 10 = FFhex, Data Type = 44hex.

Word:	High Byte:	Low Byte:
0	0041hex	
1	00	10
2	00	0
3	80hex (= 128dec)	
4	FFhex	FFhex
5	00	44hex
6	00	00
7	00	00
8	00	00
9	00	00

A DB-Fetch job will be parameterized similar to the example above.

To send with 3964R without RK512 to a CP (identifier 00FFhex), the values for DB, DW and the coordination bytes will be ignored and can be set to "0".

The next step is to copy the user data into the appropriate files of the SLC processor.

2.4 Job Processing

Both ports can be used independently from each other (multi tasking).

A possible initialization conflict will be solved as follows:

MVI46-3964R has "High Priority":

The MVI46-3964R retries the sending of the 3964R start character (STX) 3 times and waits for the CP to send the 3964R acknowledge (DLE). If the CP does not send the acknowledge as expected the communication will fail and has to be restarted.

MVI46-3964R has "Low Priority":

The MVI46-3964R rejects its send job and sends the 3964R acknowledge (DLE) to the CP.

Important: If the MVI46-3964R has rejected its send job due to "Low Priority", it is necessary to restart its send job completely (starting with the copying of the job data).

2.5 MOBY-I/E Communication

This section describes the requirements for building a peer-to-peer MOBY-I/E Identsystem using a Siemens Interface Module ASM 420/424 and an MVI46-3964R.

2.5.1 Use of Terms

In this section the following terms will be used:

- The Siemens Interface Module ASM 420/424 will be referred to as "ASM 420" respectively "ASM 424".
- Read/Write devices will be called "SLG".
- Movable memory devices will be called "MDS".

2.5.2 Hardware Components

The hardware components of a MOBY-I/E Identsystem using a MVI46-3964R are:

- **Interface Module ASM 420/424.**
- The ASM 420/424 drives the SLG. With the choice of type of SLG and the appropriate MDS you select the type of MOBY-System you want to use. It is possible to connect either a MOBY-I or MOBY-E SLG to the ASM 420/424 but SLG and MDS must be of the same type of MOBY-System.

Important: The firmware revision of the ASM 424 must be version 1.1 or higher.

- **Read/Write device (SLG).**

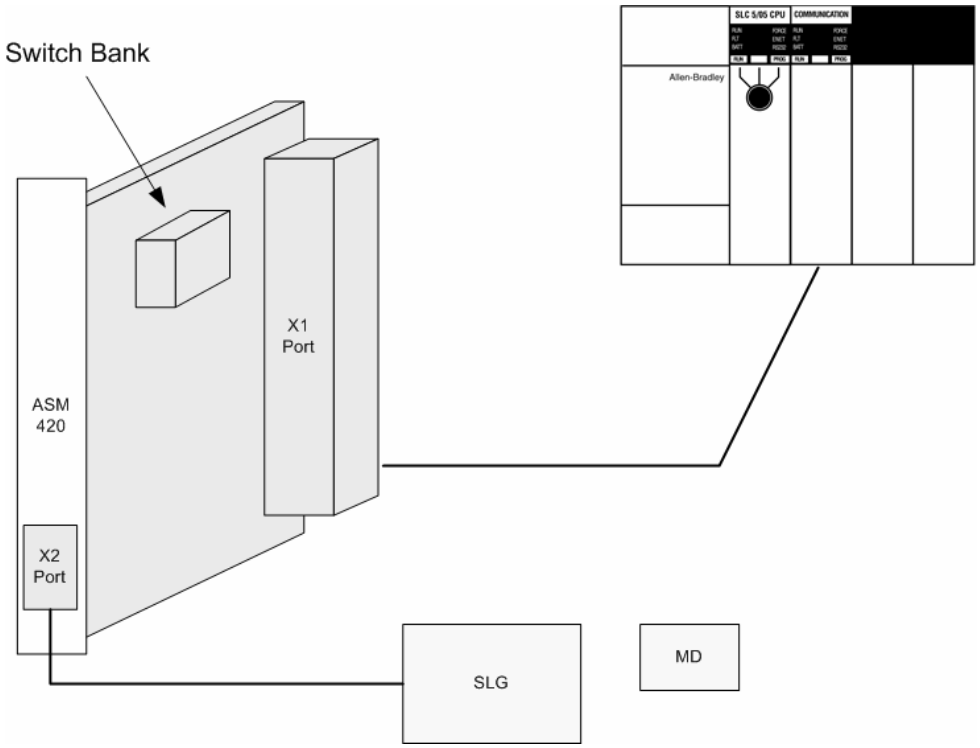
- The SLG communicates wireless to the MDS for executing different MOBY operations. These operations are fixed in the MOBY instruction set.
- **Movable memory devices (MDS).**
- The MDS are the real data carriers of a MOBY-System.
- **MVI46-3964R-3964R56.**
- The MVI46-3964R interfaces as a bridge between the ASM 420/424 and the SLC processor. The connection to the ASM 420/424 is realized via 3964R without RK512.

2.5.3 Serial Connection

The serial connection between the ASM 420/424 and the MVI46-3964R is realized using **RS-422** in our case**.** It is also possible to connect to an ASM 420/424 using RS-232/V.24 but then a different ASM 420/424 module with a RS-232/V.24 interface is necessary. The serial ports of the MVI46-3964R can be configured with a jumper to run either RS-422 or RS-232.

2.5.4 Hardware Configuration ASM 420

The hardware configuration of the peer-to-peer system using an ASM 420 is as follows:



The **X2-Port** (DB 9 female connector) of the ASM 420 interfaces to an SLG.

The **X1-Port** (backplane connector) of the ASM 420 has multiple functionalities. Details are described in the original literature:

Technical Description Identsystem MOBY-I,

Interface Module ASM 420,

Publication No. 6GT2097-3AF00-0DA2.

In our case we use only the connections for the power supply and the RS-422 interface to realize our peer-to-peer connection to the MVI46-3964R.

The **switch bank (S1)** is for setting the mode. For a detailed description of the possible modes, please refer to the technical description mentioned above. The following mode is used:

Switch 1 = Off (0)

Switch 2 = Off (0)

Switch 3 = On (1)

Switch 4 = Off (0)

Switch 5 = Off (0)

Switch 6 = Off (0)

Switch 7 = Off (0)

Switch 8 = On (1)

The switches 1 & 2 determine the baud rate of the ASM 420. 9600 Baud is configured in our case.

The other serial parameters are defined by the ASM 420:

- Data bits: 8
- Parity: odd
- Stop bit: 1

These parameters have to be fixed for the MVI46-3964R to operate with the ASM 420.

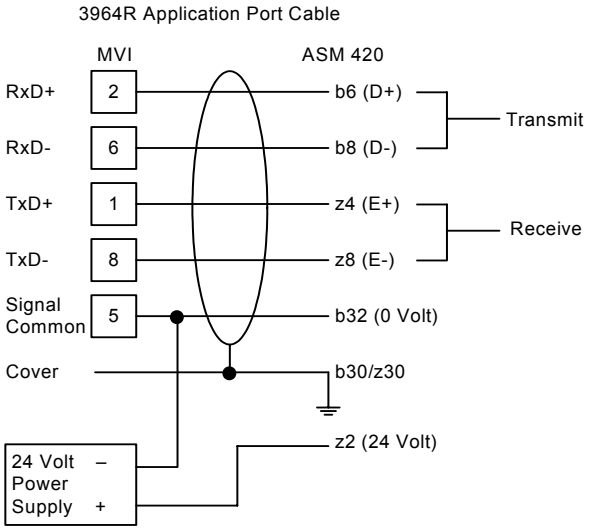
The switches 3-6 select 3964R as the standard communication procedure. There is no other choice in our case, because 3964R is the only procedure offered by the MVI46-3964R. Adjusting high/low priority is not relevant, because the data exchange between the ASM 420 and the MVI46-3964R is synchronized (Refer to Using the Moby Instruction Set). The ASM 420 will be set to "low priority".

The switches 7 & 8 determine the mode of control of the MDS through the ASM 420. All possibilities mentioned in the technical description can be chosen

according to your requirements. "Proximity detection in SIM firmware" is our example choice.

2.5.5 Pinout Connections ASM 420

The following illustration shows the pins of the ASM 420 (X1-Port) and the MVI46-3964R used for a RS-422 connection. The ASM 420 needs an external 24V DC power supply for operation.



The cable must be made according to the RS-422 specifications. In addition to this, you should also check your (ground) cabling to avoid ground loops.

2.5.6 MOBY Instruction Set ASM 420

This section provides a short overview about the MOBY instruction set and its realization in the SLC processor example program. The ASM 420 makes no difference between MOBY-I and MOBY-E, therefore all instructions are valid for both systems.

The MOBY instruction set is described in detail in the Technical Description Identsystem MOBY-I, Interface Module ASM 420, Publication No. 6GT2097-3AF00-0DA2. Please refer to that manual for additional information.

The instructions must be programmed in the SLC processor according to this specification and transferred into the MVI46-3964R. The MVI46-3964R subsequently communicates with the ASM 420 via 3964R without RK512 to manipulate the MDS according to your requirements. It is possible to use either the "normal mode" or the "ECC special driver" to interface to the MDS. The SLC processor example program will be explained in Ladder Logic.

MOBY-Instruction: RESET

Implementation in the SLC processor example program: RESET without parameters. If parameters are required the instruction has to be changed accordingly.

MOBY-Instruction: STATUS

Implementation in the SLC processor example program: without restriction.

MOBY-Instruction: DI/DO

Implementation in the SLC processor example program: without restriction.

MOBY-Instruction: NEXT

Implementation in the SLC processor example program: without restriction.

MOBY-Instruction: MDS-INIT

Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the technical description.

MOBY-Instruction: DATA WRITE

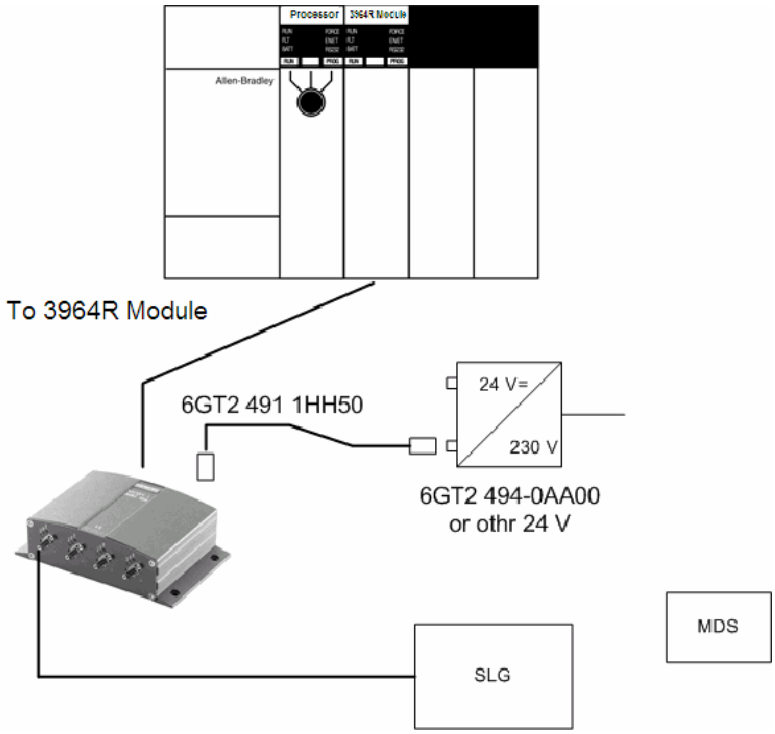
Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the technical description. In our example we write 20 bytes to the MDS.

MOBY-Instruction: DATA READ

Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the technical description. In our example we read 20 bytes from the MDS.

2.5.7 Hardware Configuration ASM 424

The hardware configuration of the peer-to-peer system using an ASM 424 is as follows:



The **Channel 1 connector** of the ASM 424 interfaces to an SLG.

The **RS-232/RS-422 port** interfaces to the MVI46-3964R.

Details are described in the original literature:

MOBY I Configuration, Installation and Service

Publication No. 6GT2 097-4BA00-0EA2.

The **switch bank** is for setting the mode. For a detailed description of the possible modes, please refer to the technical description mentioned above. The following mode is used:

- Switch 1 = Off (0)
- Switch 2 = Off (0)
- Switch 3 = Off (0)
- Switch 4 = Off (0)
- Switch 5 = Off (0)
- Switch 6 = Off (0)
- Switch 7 = Off (0)
- Switch 8 = On (1)

- Switch 9 = On (1)
- Switch 10 = On (1)
- Switch 11 = Off (0)
- Switch 12 = Off (0)

The switches 8–10 are the only ones used for this application. Switch 8 sets the serial port to RS-422. The ASM 424 brings the baud rate automatically into line with the MVI46-3964R (9600 Baud in our example).

The other serial parameters are defined by the ASM 424:

- Data bits: 8
- Parity: Odd
- Stop bit: 1

These parameters have to be fixed for the MVI46-3964R to operate with the ASM 424.

The switches 9 and 10 select 3964R as the standard communication procedure. There is no other choice in our case, because 3964R is the only procedure offered by the MVI46-3964R.

Additional parameters for the ASM 424 can be set through a RESET command. Please refer to:

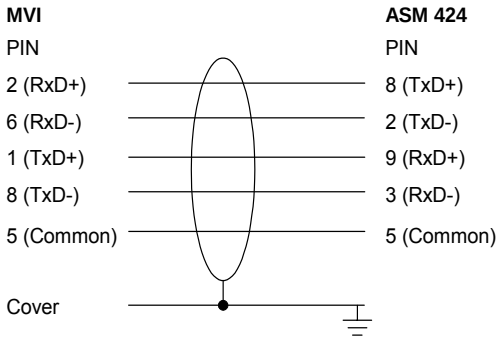
Programming Reference:

MOBY C-Library MOBY API on the CD "Software Moby"

Publication No. 6GT2 080-2AA10.

2.5.8 Pinout Connections ASM 424

The following illustration shows the pins of the ASM 424 (DB 9 female connector) and the MVI46-3964R used for a RS-422 connection. The ASM 424 needs an external 24V DC power supply for operation.



The cable must be made according to the RS-422 specifications. In addition to this, you should also check your (ground) cabling to avoid ground loops.

2.5.9 MOBY Instruction Set ASM 424

This section contains a short overview about the MOBY instruction set and its realization in the SLC processor example program. As mentioned before, the ASM 424 makes no difference between MOBY-I and MOBY-E, therefore all instructions are valid for both systems.

The concrete MOBY instruction set is described in detail in the Programming Reference, MOBY C-Library MOBY API on the CD "Software Moby", Publication No. 6GT2 080-2AA10. Please refer to that manual for additional information.

The instructions have to be programmed in the SLC processor according to this specification and transferred into the MVI46-3964R. The MVI46-3964R subsequently communicates with the ASM 424 via 3964R to manipulate the MDS according to your requirements. It is possible to use either the "normal mode" or the "ECC special driver" to interface to the MDS. The SLC processor example program itself will be explained in Ladder Logic.

MOBY-Instruction: RESET

Implementation in the SLC processor example program: without restriction.

If different parameters are required the instruction has to be changed accordingly.

MOBY-Instruction: STATUS

Implementation in the SLC processor example program: without restriction.

MOBY-Instruction: NEXT

Implementation in the SLC processor example program: without restriction.

MOBY-Instruction: MDS-INIT

Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the programming reference.

MOBY-Instruction: DATA WRITE

Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the programming reference.

In our example we write 20 bytes to the MDS.

MOBY-Instruction: DATA READ

Implementation in the SLC processor example program: This instruction uses the "normal mode" of the MDS. If you want to use the "ECC special driver", you have to change the instruction command as explained in the programming reference.

In our example we read 20 bytes from the MDS.

3 Module Configuration

In This Chapter

3.1 Installation Instructions

The following setups/limitations are to be observed:

	Please make sure that you are using an MVI46-3964R with the original parameters and image as delivered. **Jumper SETUP:** **Please remove to run the MVI46-3964R in application mode.**
	**ATTENTION:** Incorrect setting of the jumpers may cause damage to the MVI46-3964R module.
	The remaining jumpers for PRT2 and PRT3 have to be set according to the requirements of the user. The communication partners are to be connected to the serial interface PRT2 and/or PRT3 of the MVI46-3964R. The serial parameters of the communication partners are to be set to 9600 Baud, even parity, 8 bit and 1 stop bit if the standard parameters of the MVI46-3964R will be used.

The assignment of the RS-232 connection cable between PRT2/3 (9 pole) and CP-544 (25 pole) is as follows:

PRT2/3 Pin	Description	CP 544 Pin	Description
2	RxD	2	TxD
3	TxD	3	RxD
5	Gnd	7	Gnd

Important: The other pins of the RS-232 must not be used.

RS-485 operation is not possible.

3.2 Modifying the Configuration File

3.2.1 [Module]

This section defines the configuration for the Module level data.

```
#
[Module]
Module Name         :  Test Example of MVI69-3964R Communication Module
```

Module Name

0 to 80 characters

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

3.2.2 [Backplane 46]

This section identifies the method of failure for the communications for the module if the processor is not in run.

The following example shows a sample [Backplane Configuration] section:

```
[Backplane 46]
Backplane Fail Count      :    10 #Number of consecutive backplane transfer
failures before halting communications
```

Backplane Fail Count

This parameter specifies the number of consecutive backplane transfer failures that can occur before communications should be halted.

3.2.3 [3964R Port x]

```
# This section is used to define the port 1 configuration for the 3964R device
#
[3964R Port 1]
Enable                :  Yes #No=Port Disabled,Yes=Port Enabled
Baud Rate             : 9600 #Baud rate for port (300, 600, 1200, 2400, 4800,
                             #9600, 19200, 38400, 57600, 115)
Parity                : Even #N=None,O=Odd,E=Even
Data Bits             :    8 #7 or 8 data bits for messages
Stop Bits             :    1 #1 or 2 stop bits for messages
RTS On                :    0 #Delay after RTS set before message sent (mSec)
RTS Off               :    0 #Delay after message before RTS dropped (mSec)
Minimum Response Delay :    0 #Delay before responding to CP
Use CTS Line          :   No #Monitor CTS modem line (Y/N)
Swap                  :    1 #0=No Data Byte Swapping, 1=Data Byte Swapping
Priority              :    0 #0=High Priority, 1=Low Priority
Protocol              :    0 #0=3964 RK512, 1=3964, 3=MOBY I/E
```

```
ACK Delay              : 1000 #Number of mSec to wait for ACK (Default 1000ms)
Setup Attempts         :    6 #Number of times to try to connect to CP
                              #(Default 6)
Transmit Attempts       :    6 #Number of times to try to transmit to CP
                              #(Default 6)
```

Enable

Yes or No

This parameter specifies whether to enable or disable the port. No = Port Disabled, Yes = Port Enabled.

Baud Rate

300 to 115200

This parameter specifies the baud rate to be used on the port. Valid values are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.

Parity

N, O, E

This parameter specifies the parity setting for this port. Valid values are N = None, O = Odd, E = Even.

Data Bits

7 or 8

This parameter specifies the number of message data bits. Valid values are 7 or 8 bits.

Stop Bits

1 or 2

This parameter specifies the number of message stop bits. Valid values are 1 or 2 stop bits.

RTS On

Delay (in milliseconds) after RTS before message is sent.

RTS Off

Delay (in milliseconds) after message before RTS dropped.

Minimum Response Delay

The minimum amount of time in milliseconds to wait before responding to a CP message.

Use CTS Line

Yes or No

This parameter specifies whether or not to monitor CTS modem line.

Swap

0 or 1

This parameter specifies whether bytes should be swapped. 0 = No data byte swapping, 1 = Data byte swapping.

Priority

0 or 1

This parameter specifies whether the data is high priority or low priority. 0 = High priority, 1 = Low Priority.

Protocol

0, 1, or 3

0 = 3964 RK512, 1 = 3964, 3 = MOBY I/E

ACK Delay

Number of milliseconds to wait for ACK. The default is 1000 ms.

Setup Attempts

Number of times to try to connect to CP (Default 6)

Transmit Attempts

Number of times to try to transmit to CP (Default 6)

3.3 Uploading and Downloading the Configuration File

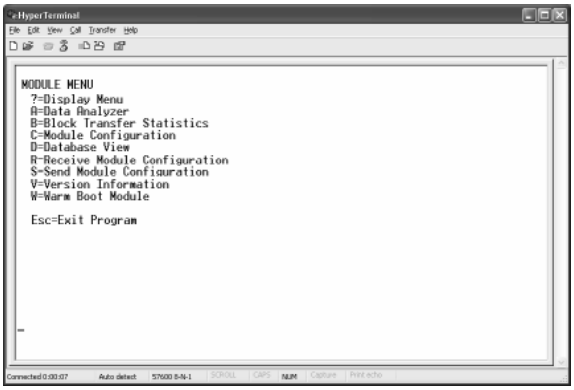
ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you must transfer the file from the module to your PC. After editing, you must transfer the file back to the module.

This section describes these procedures.

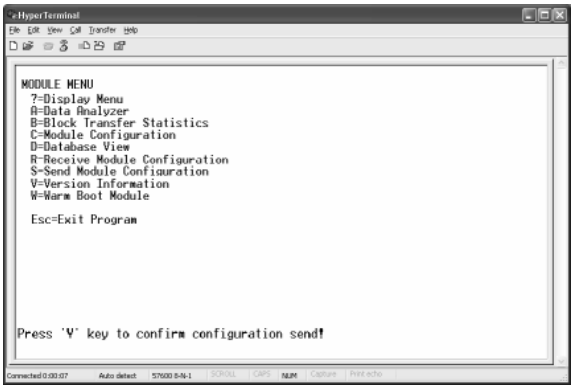
Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may or may not exactly match the configuration/debug menus in your own module. For specific details about the configuration/debug menus in your module, please refer to ***The Configuration/Debug Menu*** (on page 47).

3.3.1 Transferring the Configuration File to Your PC

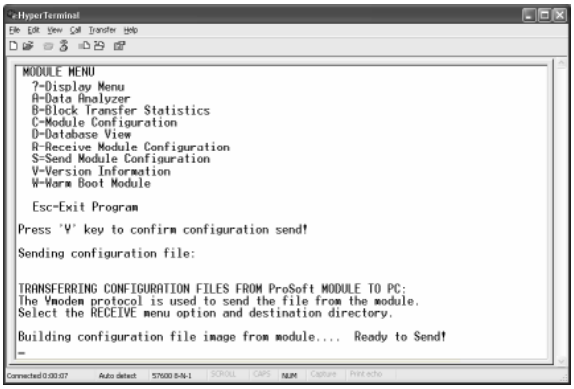
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



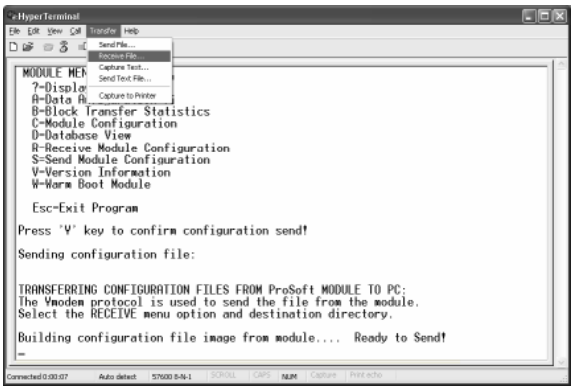
2 Press **[S]** (Send Module Configuration). The message "Press Y key to confirm configuration send!" is displayed at the bottom of the screen.



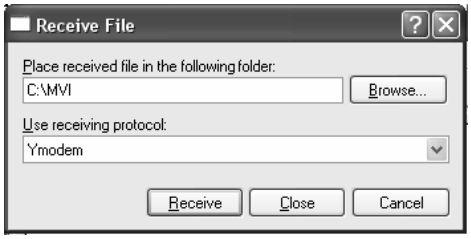
3 Press **[Y]**. The screen now indicates that the module is ready to send.



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



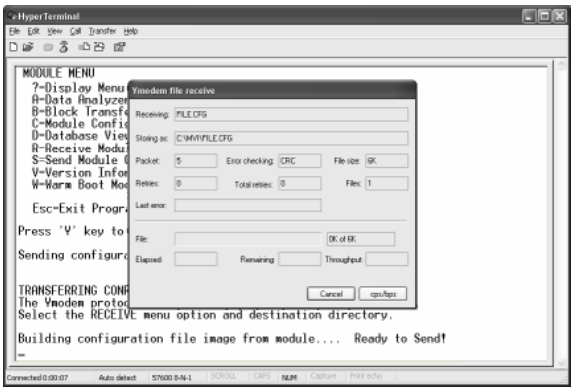
5 Use the Browse button to choose a folder on your computer to save the file, and then click Receive.



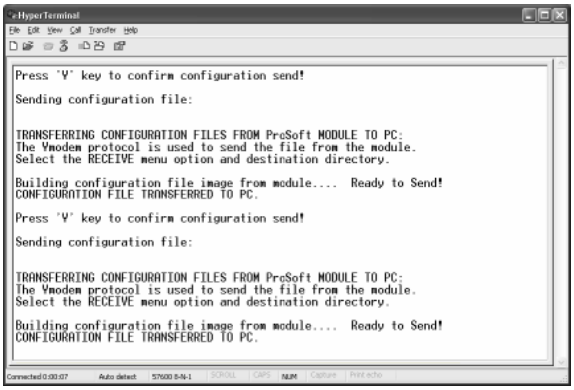
- **Note:** ProSoft Technology suggests that you download the configuration file pre-loaded on your module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at http://www.prosoft-technology.com.

6 Select Ymodem as the receiving protocol.

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



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



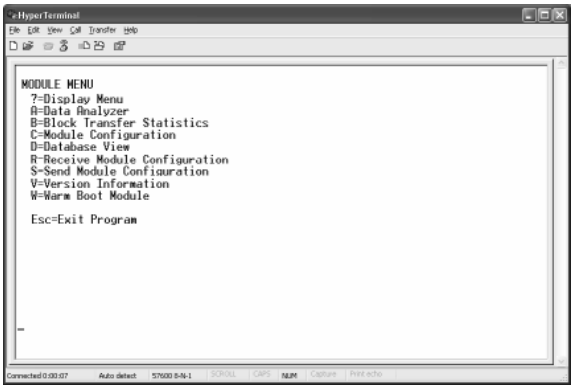
The configuration file is now on your PC at the location you specified.

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

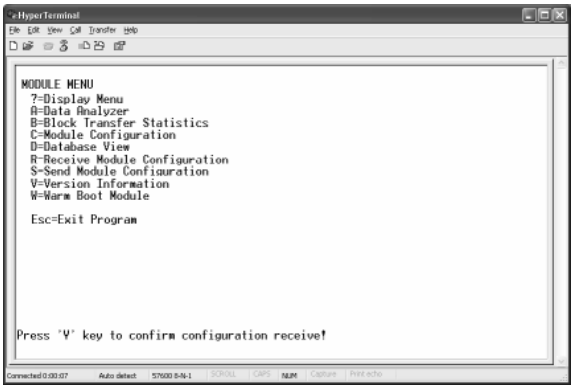
3.3.2 Transferring the Configuration File to the Module

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

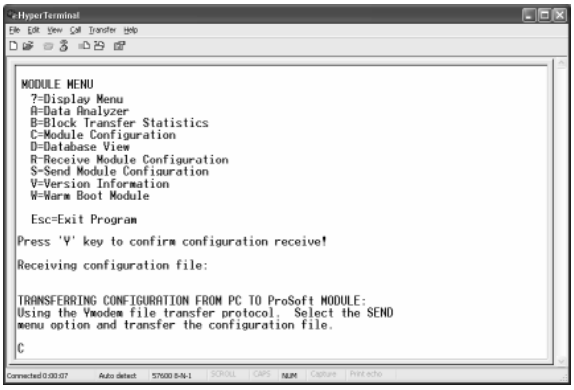
1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



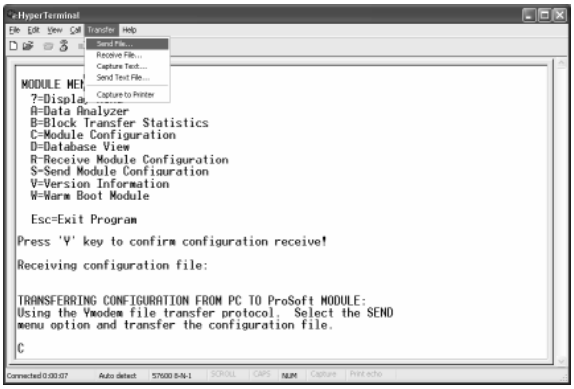
2 Press **[S]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



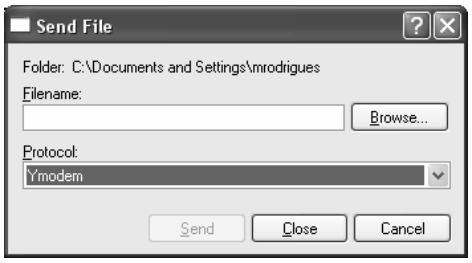
3 Press **[Y]**. The screen now indicates that the PC is ready to send.



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



The Send File dialog appears.

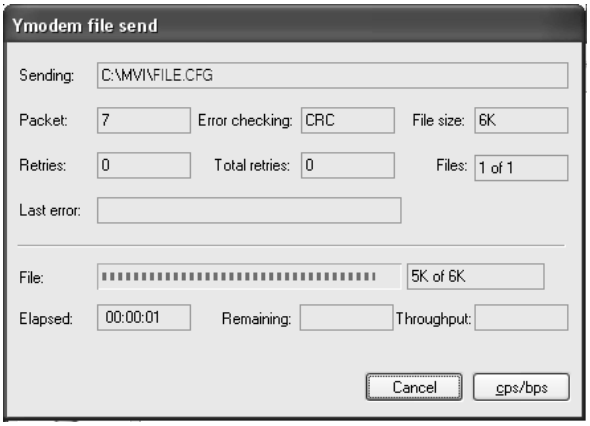


5 Use the Browse button to locate the configuration file your computer.

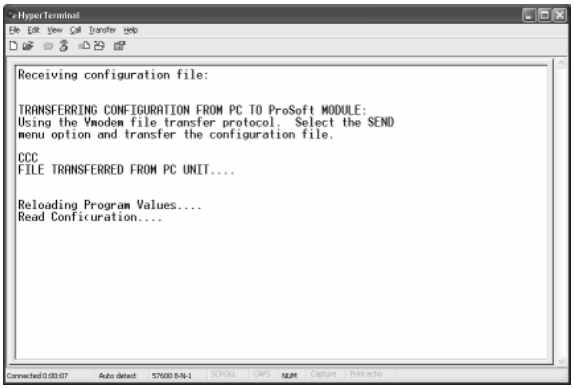
Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at http://www.prosoft-technology.com.

6 Select **Ymodem** as the protocol.

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



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



8 Your module now contains the new configuration.

4 Ladder Logic

In This Chapter

4.1 Introduction

Please be aware that the example programs mentioned in the following paragraphs only show the principle of how to communicate between an MVI46-3964R and a SLC processor. It is up to the programmer to examine the programs in detail and integrate them into his SLC processor application.

The following setups/limitations are to be observed in the example programs:

Changes according to the requirements of your SLC system are recommended for setups only. Limitations must not be changed.

- **Setup:** The MVI46-3964R resides in slot 1 of the SLC chassis.
- **Setup:** The SLC series controller resides in slot 0 of the SLC chassis.

ATTENTION: The user must be trained in programming and operating Rockwell Automation SLC series controllers and SLC environment. Otherwise, incorrect use may lead to personal injury or death, property damages or economic loss.

4.2 Explanations Concerning the Example Programs

MVI46_3964.acd is the general 3964R communication program used for the following types of 3964R jobs:

4.2.1 3964R with RK 512:

- DB-Send jobs (send and receive).
- DB-Fetch jobs (send and receive).

3964R:

- Send and receive jobs.

4.3 The 3964R Communication Program

The general functionality of the program MVI46_3964.RSS is as follows:

4.3.1 Send jobs to a CP:

- Transfer the job data to the MVI46-3964R.
- Acknowledgement from the MVI46-3964R regarding validity of the job data
- Depending on the type of send job the transfer of the user data from/to the MVI46-3964R/SLC processor takes place.
- Acknowledgement from the MVI46-3964R regarding successful/unsuccessful completion of the send job.

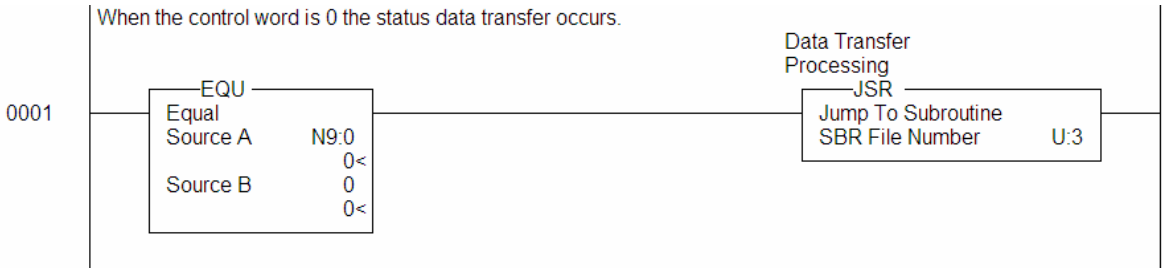
Receive jobs from a CP:

- Depending on the type of receive job the transfer of the user data from/to the MVI46-3964R/SLC processor takes place. The header data generated from the MVI46-3964R will be added automatically. The MVI46-3964R needs no extra job data from the SLC processor for receive jobs, it will be triggered through the STX character sent from the CP to establish the communication.
- Acknowledgement from the MVI46-3964R regarding successful/unsuccessful completion of the receive job.

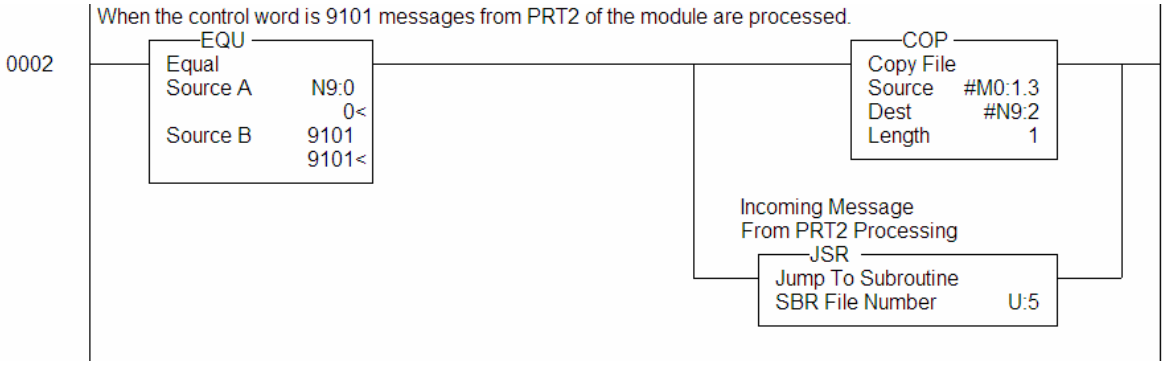
4.4 MainRoutine

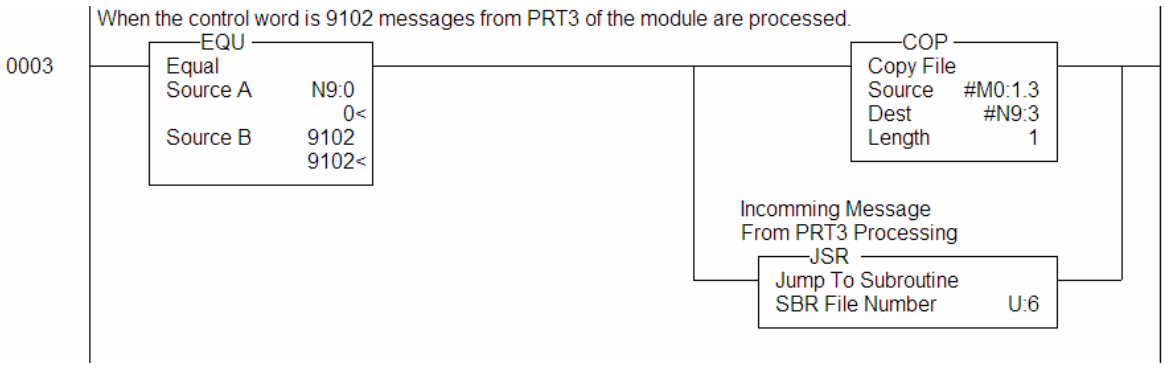
The MainRoutine recognizes the presence of special command request and response messages, and data transfer between the module and the processor. The following run saves the current control word found in the M0 file:

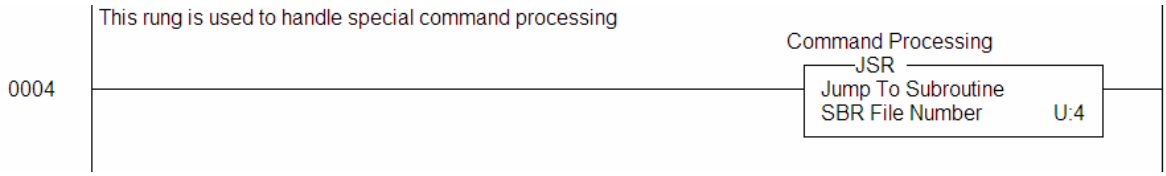
Word 0 of the M0 file is used by the module to indicate a response to a special command instruction. During normal program execution, this register should have a value of 0. If any other value is present, the data transfer function will not be executed. The following rung executes the data transfer function (Trnsf U:3) when the control word is set to zero.



The following rung calls each scan to process any special command request and response messages:

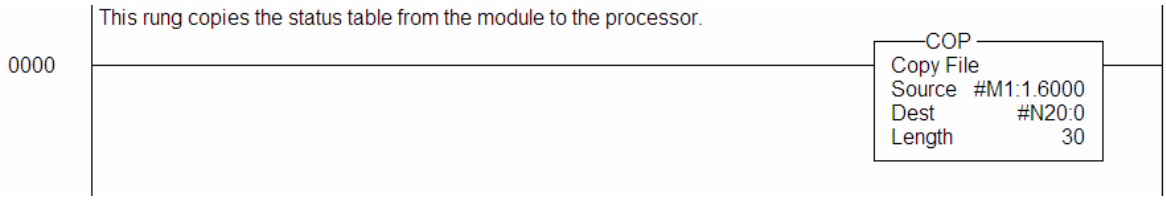






4.5 Transfer Routine

The Data Transfer task is responsible for handling the transfer of data between the processor user defined files and the module's M1 file. In this example, 30 words of Status Data is read from the M1 file and stored in 3964R0.



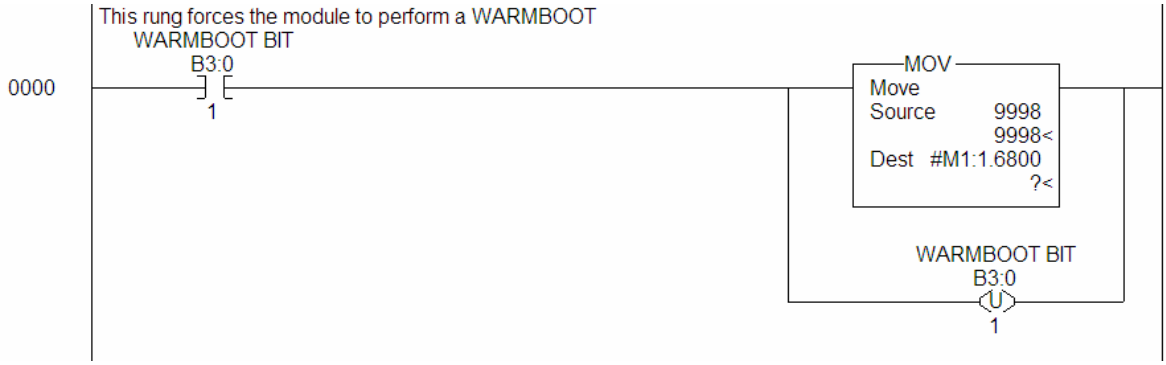
4.6 CMDS

The CMDS sub-routine handles special block processing. Special blocks include:

- 9998 - Warm Boot
- 9999 - Cold Boot
- 9001 - Transfer data to be sent on P2
- 9002 - Transfer data to be sent on P3

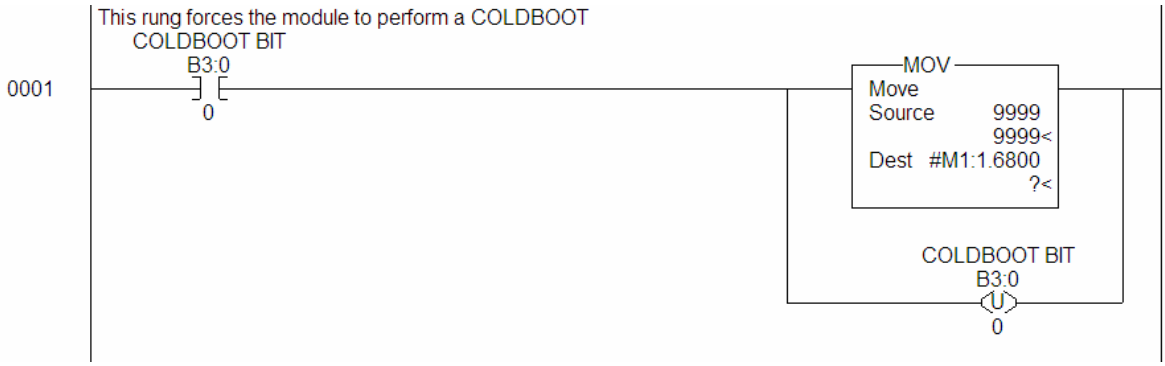
4.6.1 9998 Warm Boot

The SLC processor can request a warm boot operation of the module by placing a value of 9998 is the M1 register 6800 (Command Control Register). Ladder logic to perform this task is shown in the following rung:



4.6.2 9999 Cold Boot

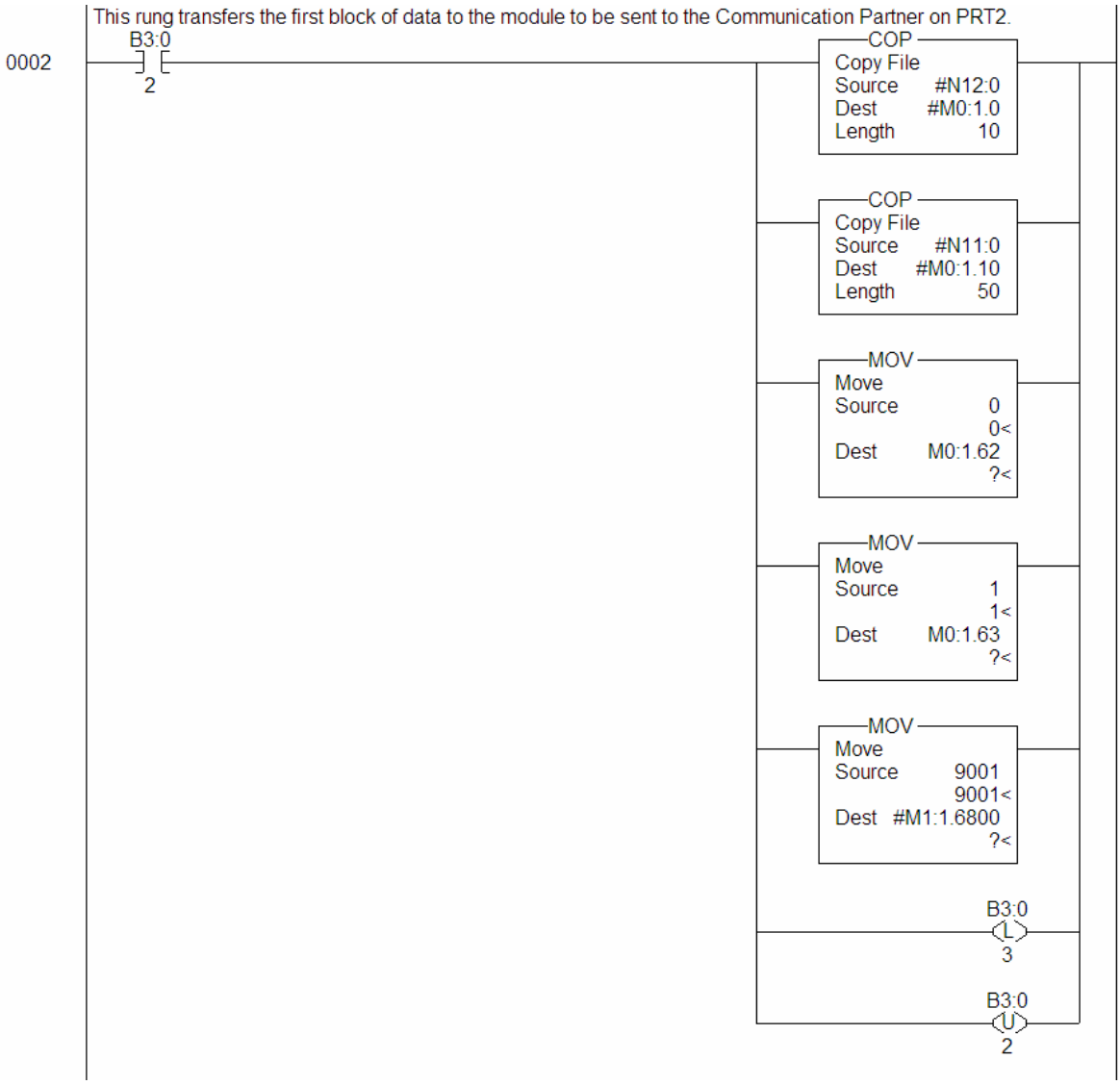
The SLC processor can request a cold boot operation by the module by placing a value of 9999 in the M1 register 6800 (Command Control Register). Ladder logic to perform this task is shown in the following rungs:

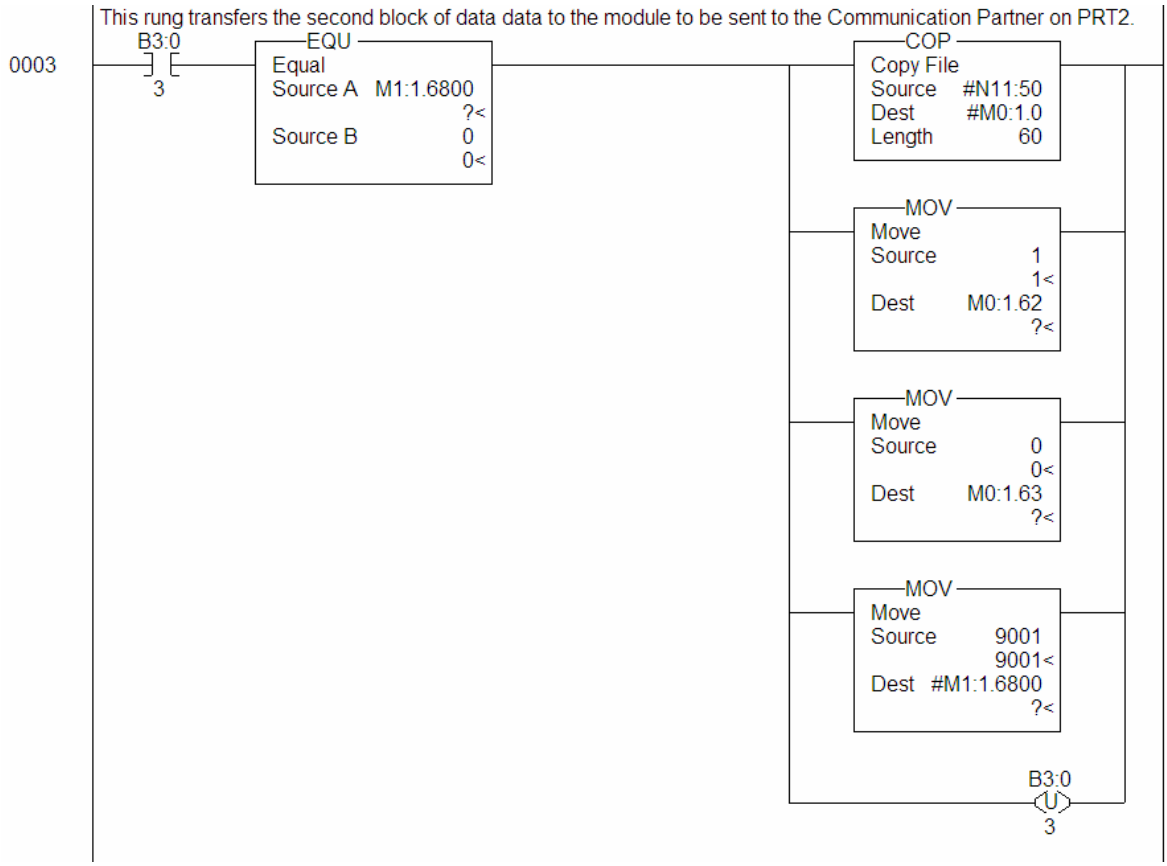


4.6.3 9001 Transfer Data to Be Sent on P2

The following rung builds a block of data with a header to send to the module. The header data resides in N12 and the message data in N11. The next rung transfers the remaining data to the module

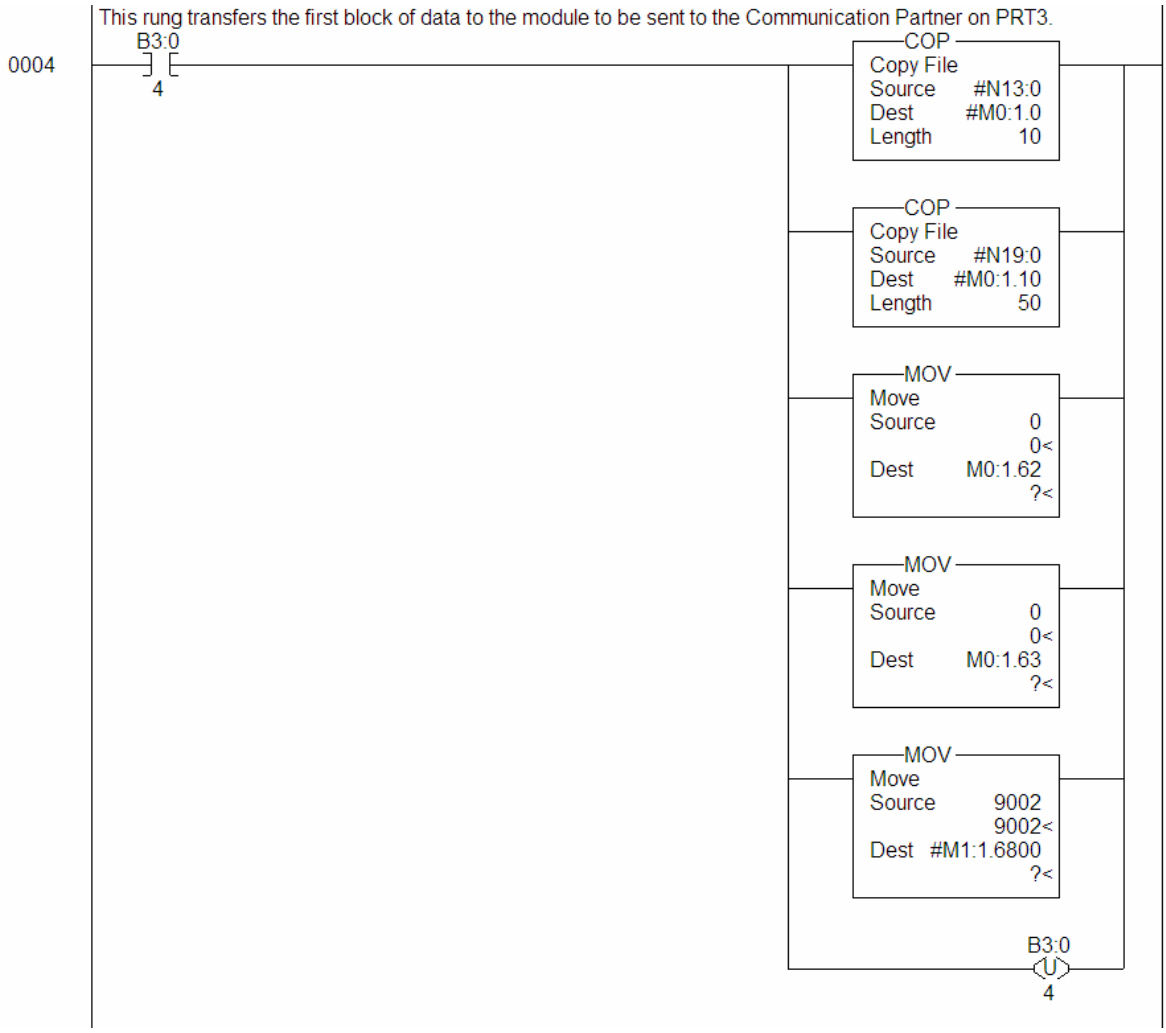
Pass-through from Processor to Module	
Words	Description
0 to 59	Data
60	Reserved
61	Reserved
62	Block Index (0 - 4)
63	Continuation Flag (0 = Last Block, 1 = More Blocks to Come)


This rung transfers the first block of data to the module to be sent to the Communication Partner on PRT2.
0002
B3:0
2
COP
Copy File
Source #N12:0
Dest #M0:1.0
Length 10
COP
Copy File
Source #N11:0
Dest #M0:1.10
Length 50
MOV
Move
Source 0
0<
Dest M0:1.62
?<
MOV
Move
Source 1
1<
Dest M0:1.63
?<
MOV
Move
Source 9001
9001<
Dest #M1:1.6800
?<
B3:0
L
3
B3:0
U
2


This rung transfers the second block of data data to the module to be sent to the Communication Partner on PRT2.
0003
B3:0
3
EQU
Equal
Source A M1:1.6800
?<
Source B 0
0<
COP
Copy File
Source #N11:50
Dest #M0:1.0
Length 60
MOV
Move
Source 1
1<
Dest M0:1.62
?<
MOV
Move
Source 0
0<
Dest M0:1.63
?<
MOV
Move
Source 9001
9001<
Dest #M1:1.6800
?<
B3:0
U
3

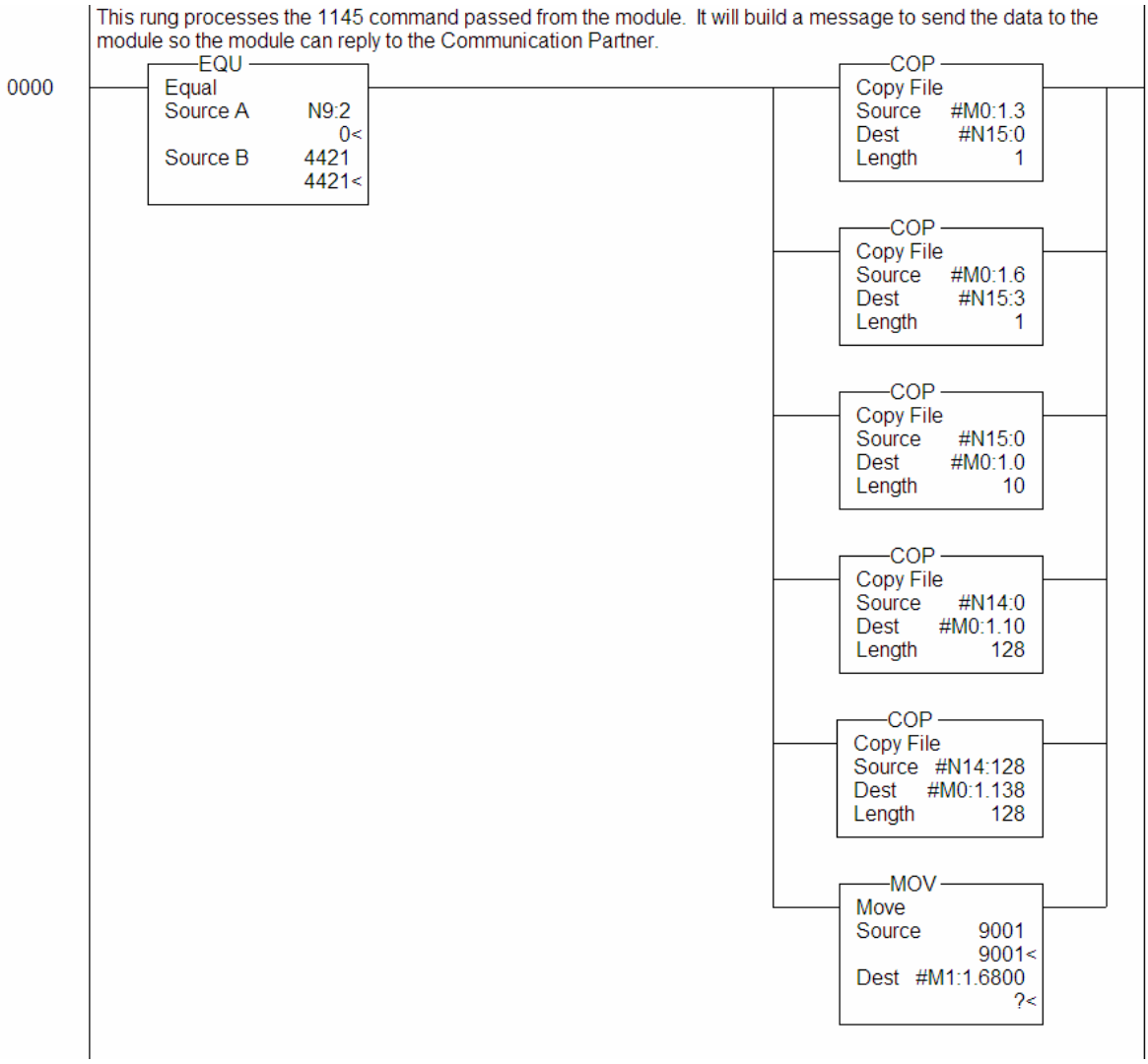
4.6.4 9002 Transfer Data to be Sent on P2

This rung shows a data block comprised of a header from N13 and 50 words of data from N19. If more data is to be transferred, an additional rung similar to rung 003 must be added.



4.6.5 Request for Data (1145h)

This rung processes the request for data (1145h). A Pass-Through block is built containing a header and data which is then sent to the module.

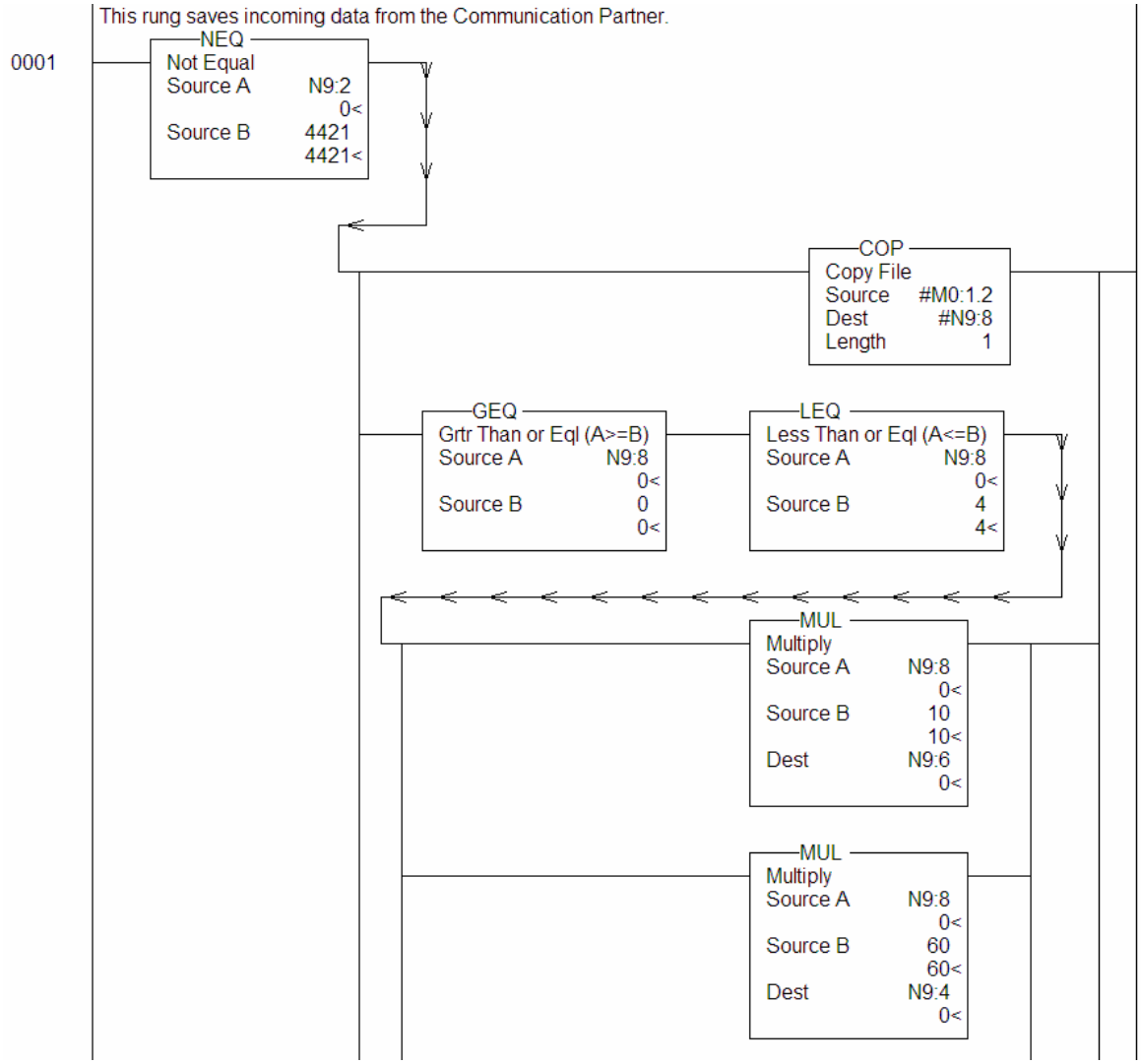


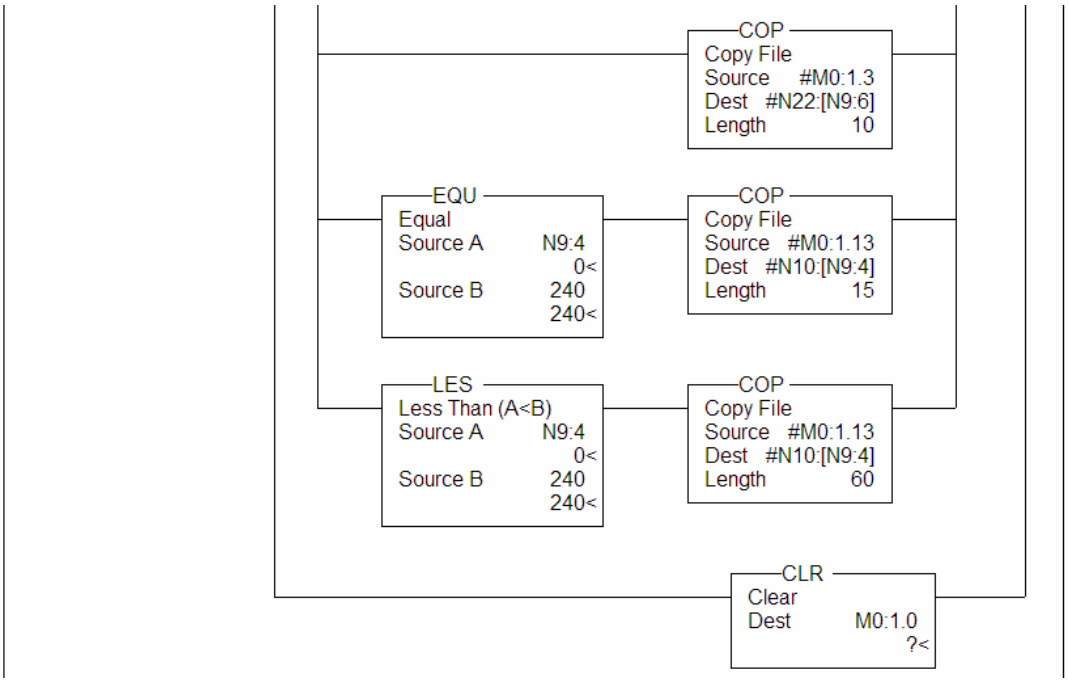
4.6.6 Process Incoming Data

This rung saves the incoming data from the communication partner. The header is saved in N22 and the data in N10. The block index is used to index the saving of the header and data.

Pass-through from Module to Processor	
Words	Description
0	Block ID

Pass-through from Module to Processor	
1	Length
2	Block Index (0 - 4)
3 - 12	Header
13 - 62	Data
63	Reserved




COP
Copy File
Source #M0:1.3
Dest #N22:[N9:6]
Length 10
EQU
Equal
Source A N9:4
0<
Source B 240
240<
COP
Copy File
Source #M0:1.13
Dest #N10:[N9:4]
Length 15
LES
Less Than (A<B)
Source A N9:4
0<
Source B 240
240<
COP
Copy File
Source #M0:1.13
Dest #N10:[N9:4]
Length 60
CLR
Clear
Dest M0:1.0
?<

5 Diagnostics and Troubleshooting

In This Chapter

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

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

5.1 Reading Status Data From the Module

The MVI46-3964R module returns a 30-word Status Data Block that may be used to determine the module's operating status. This data is located in the module's database in registers 6000 through 6029.

This data is transferred to the SLC processor every 100 blocks.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Complete display of the module's internal database (registers 0 to 10000)
- Version Information
- Control over the module (warm boot, cold boot, transfer configuration)

5.1.1 The Configuration/Debug Menu

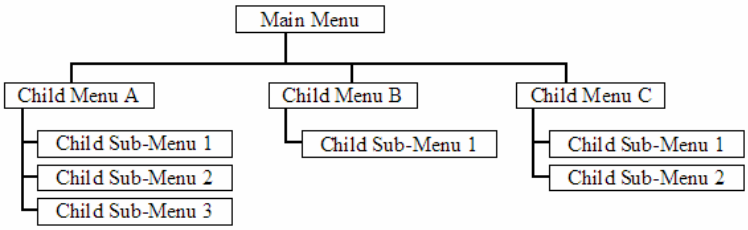
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

Navigation

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

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



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

Keystrokes

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

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

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

5.1.2 Required Hardware

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

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

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one serial communications port available
- A null modem serial cable.

5.1.3 Required Software

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

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

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

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

5.1.4 Using the Configuration/Debug Port

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

1 Connect your computer to the module's port using a null modem cable.

2 Start the communication program on your computer and configure the communication parameters with the following settings:

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

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

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

1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

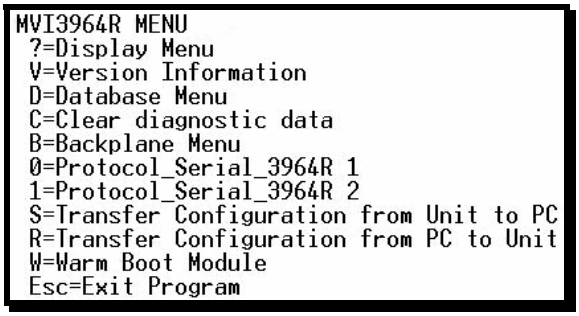
2 Verify that your communication software is using the correct settings for baud rate, parity and handshaking.

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

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

5.1.5 Main Menu

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



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

Redisplaying the Menu

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

Viewing Version Information

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

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

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

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

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.

Clearing Diagnostic Data

Press **[C]** to clear diagnostic data from the module's memory.

Opening the Backplane Menu

Press **[B]** from the Main Menu to view the Backplane Data Exchange List. Use this command to display the configuration and statistics of the backplane data transfer operations.

Tip: Repeat this command at one-second intervals to determine the number of blocks transferred each second.

Opening the Protocol_Serial_3964R Menu

Press **[0]** or **[1]** from the Main Menu to open the Protocol_Serial_3964R menu for 3964R Ports 1 and 2.

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

Transferring the Configuration File from MVI46 module to PC

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

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

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

Transferring the Configuration File from PC to MVI46 module

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

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

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

Warm Booting the Module

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

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

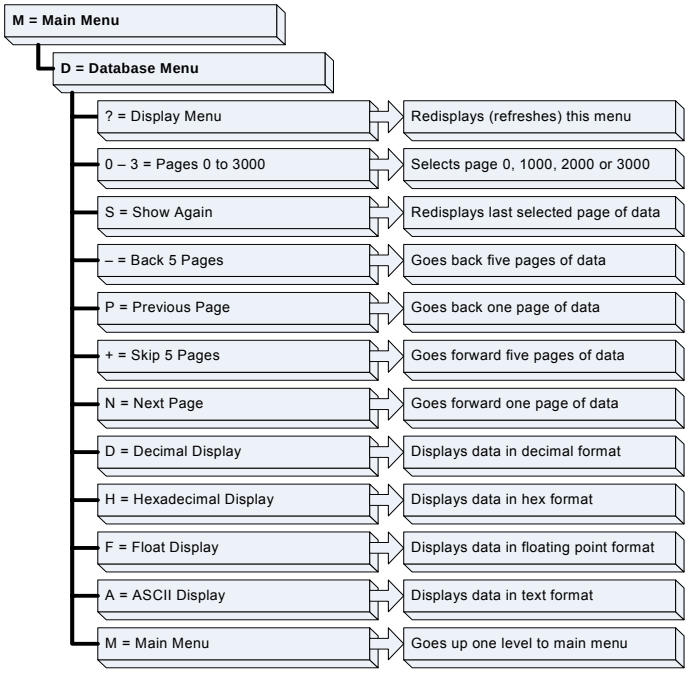
Exiting the Program

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

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

5.1.6 Database View Menu

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



Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again

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

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

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

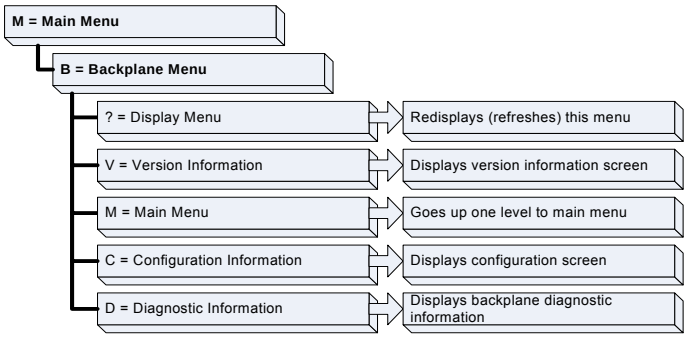
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

5.1.7 Backplane Menu

Press **[B]** from the Main Menu to view the Backplane Data Exchange List. Use this command to display the configuration and statistics of the backplane data transfer operations. Press **[?]** to view a list of commands available on this menu.



Redisplaying the Menu

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

Viewing Version Information

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

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

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

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

Returning to the Main Menu

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

Viewing Configuration Information

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

Viewing Backplane Diagnostic Information

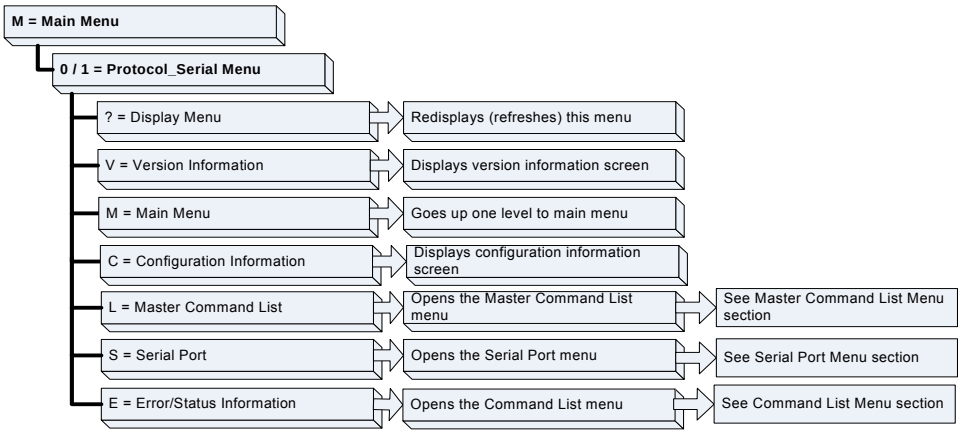
Press **[D]** to view Backplane Diagnostic information.

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: Repeat this command at one-second intervals to determine the number of blocks transferred each second

5.1.8 Protocol_Serial_3964R Menu

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



Redisplaying the Menu

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

Viewing Version Information

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

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

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

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

Returning to the Main Menu

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

Viewing Configuration Information

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

Opening the Command List Menu

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

Opening the Serial Port Menu

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

Viewing Error and Status Data

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

5.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
P1	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P2	Green	On	Data being transferred between master and port 1.
		Off	No data
P3	Green	On	Data being transferred between master and port 2.
		Off	No data
APP Status	Amber	Off	The MVI46-3964R is working normally.
		On	The MVI46-3964R module program has recognized a communication error.

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

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

5.2.1 Clearing a Fault Condition

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

1 Turn the power to the rack off

2 Remove the card from the rack

3 Make certain all jumpers are set correctly

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

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

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

5.2.2 Troubleshooting

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

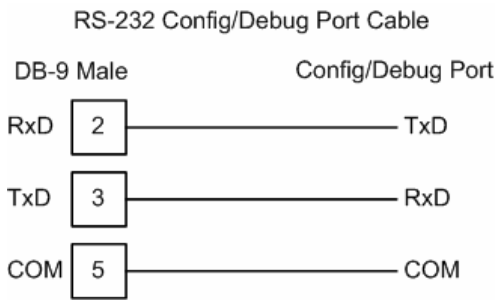
Problem Description	Steps to take
Processor Fault	Be sure that the module is plugged into the slot that has been configured for the MVI46-3964R module. Be sure the ladder logic has been set up correctly
Processor I/O LED flashes	This indicates that there is a problem with backplane communications. Be certain this and all modules in the rack are configured in the processor.
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Use the Configuration/Debug port facility to check this. To establish backplane communications, verify the following items: ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
ACT/FLT LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, remove the card from the rack, then re-insert.

6 Reference

In This Chapter

6.1 RS-232 Configuration/Debug Port

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



6.2 Reference Documents

- ***Rockwell Automation***

SLC Multi-Vendor Interface Module

(Cat. Number 1756-MVI)

Installation Instructions

Publication 1756-IN001A-US-P

➢ ***Siemens***

Simatic CP544

Handbuch zum Kommunikationspartner CP544

Ausgabe 01

Bestell-Nr. 6ES5 998-2DB11

Siemens

Identsystem MOBY-I

ASM 420

Interface Module ASM 420

Technical Description

Publication No. 6GT2097-3AF00-0DA2

➢ ***Siemens***

MOBY I Configuration, Installation and Service

Publication No. 6GT2 097-4BA00-0EA2

➢ ***Siemens***

Programming Reference

MOBY C-Library MOBY API on the CD "Software Moby"

6.3 Error Codes

Error Code		Description
–2	0xFFFE	DLE could not be sent
–10	0xFFF6	DLE not received in time
–11	0xFFF5	Communication Timeout occurred
–20	0xFFEC	Invalid character or acknowledge received
–30	0xFFE2	Amount of user data is 0 or greater than 256 words
–31	0xFFE1	Reject pending job from the processor due to low priority
–32	0xFFE0	No reaction (STX) from Communication Partner
–33	0xFFDF	Communication Partner error occurred
–34	0xFFDE	Job code not defined

Error Code		Description
–35	0xFFDD	High/High (priority) initialization conflict occurred
–36	0xFFDC	First Header Byte does not match telegram type
–37	0xFFDB	Character received was not an STX
–38	0xFFDA	3964R problem occurred after connection to Communication Partner was established
–39	0xFFD9	3964R communication could not be established/finished

6.4 MVI46-3964R Configuration File Example

```
# 463964R.CFG
#
# This file contains the configuration for the MVI46-3964R communication module.
#
# LOCATION      : Test Bench
# DATE          : 07/01/2004
# CONFIGURED BY : KDH
# MODIFIED      :
#
# This section is used to define the configuration for the Module level
# data.
#
[Module]
Module Name           : Test Example of MVI46-3964R Communication Module
#
#
[Backplane 46]
Backplane Fail Count  :    10 #Number of consecutive backplane transfer failures
                              #before halting communications
#
#
# This section is used to define the port 1 configuration for the 3964R device
#
[3964R Port 1]
Enable        :   Yes #No=Port Disabled,Yes=Port Enabled
Baud Rate     :  9600 #Baud rate for port (300, 600, 1200, 2400, 4800, 9600,
                      #19200, 38400, 57600, 115200)
Parity        :  Even #N=None,O=Odd,E=Even
Data Bits     :     8 #7 or 8 data bits for messages
Stop Bits     :     1 #1 or 2 stop bits for messages
RTS On        :     0 #Delay after RTS set before message sent (mSec)
RTS Off       :     1 #Delay after message before RTS dropped (mSec)
Use CTS Line  :    No #Monitor CTS modem line (Y/N)
Swap          :     1 #0=No Data Byte Swapping, 1=Data Byte Swapping
Priority      :     0 #0=High Priority, 1=Low Priority
Protocol      :     1 #0=3964 RK512, 1=3964, 3=MOBY I/E

# This section is used to define the port 2 configuration for the N2 device
#
[3964R Port 2]
Enable        :   Yes #No=Port Disabled,Yes=Port Enabled
Baud Rate     :  9600 #Baud rate for port (300, 600, 1200, 2400, 4800, 9600,
                      #19200, 38400, 57600, 115200)
```

```
Parity        :  Even #N=None,O=Odd,E=Even
Data Bits     :     8 #7 or 8 data bits for messages
Stop Bits     :     1 #1 or 2 stop bits for messages
RTS On        :     0 #Delay after RTS set before message sent (mSec)
RTS Off       :     1 #Delay after message before RTS dropped (mSec)
Use CTS Line  :    No #Monitor CTS modem line (Y/N)
Swap          :     1 #0=No Data Byte Swapping, 1=Data Byte Swapping
Priority      :     0 #0=High Priority, 1=Low Priority
Protocol      :     1 #0=3964 RK512, 1=3964, 3=MOBY I/E
```

Support, Service & Warranty

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

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

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

1 Product Version Number

2 System architecture

3 Module configuration and contents of MVI46_3964R.CFG file

4 Module Operation

- o Configuration/Debug status information
- o LED patterns

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

6 Details about the serial devices interfaced

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

Module Service and Repair

The MVI46-3964R device is an electronic product, designed and manufactured to function under somewhat adverse conditions. As with any product, through age, misapplication, or any one of many possible problems the device may require repair.

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

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

General Warranty Policy – Terms and Conditions

ProSoft Technology, Inc. (hereinafter referred to as ProSoft) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product. Limited warranty service may be obtained by delivering the Product to ProSoft in accordance with our ***product return procedures*** (on page 67) and providing proof of purchase and receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to ProSoft, and to use the original shipping container or equivalent. Contact ProSoft Customer Service for further information.

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

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

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

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

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

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

Limitation of Liability

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

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

RMA Procedures

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

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

Index