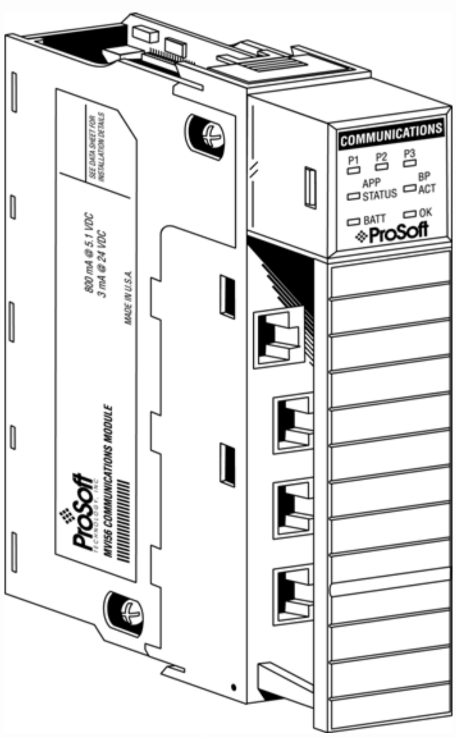
inRAx



MVI56-3964R

ControlLogix
Communications Module
Siemens 3964R Protocol

User Manual

November 12, 2004

ProSoft
TECHNOLOGY

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the ControlLogix Communications Module Siemens 3964R Protocol hardware 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 Allen-Bradley documentation on the operation of the A-B 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.

Notice

This product is a re-packaged version of the original Rockwell Automation 1756-3964 solution.

- All functionality in the original version has been maintained (100% backwards compatible)
- This user manual has been re-formatted to ProSoft Technology style guidelines.

ProSoft Technology, Inc.

1675 Chester Avenue, 2nd Floor

Bakersfield, CA 93301

(661) 716-5100

(661) 716-5101 (Fax)

www.prosoft-technology.com

MVI56-3964R User Manual

November 12, 2004

Table of Contents

PLEASE READ THIS NOTICE 2

1 USING THIS MANUAL 5
1.1 Users 5
1.2 Use of Terms 5
1.3 Products in the Environment 5
1.3.1 Product Compatibility 5

2 INSTALLATION 7
2.1 Scope of Delivery 7
2.2 Installation Instructions 7

3 3964R PROTOCOL 9
3.1 The 3964R Protocol in General 9
3.2 MVI56-3964R Operation 9

4 COMMUNICATIONS 11
4.1 Job Allocation by the 1756-L Series Controller 11
4.2 Header Data 11
4.2.1 Job Identifier 12
4.2.2 Data Block (DB) 12
4.2.3 Data Word (DW) 12
4.2.4 Number of Data Words/Data Bytes 12
4.2.5 Coordination Byte 9/10 13
4.2.6 MVI56-3964R Configuration 13
4.3 Receive Jobs 14
4.4 Send Jobs 15
4.5 Tag Database 16
4.6 Status Messages 17

5 JOB PROCESSING 19
5.1 Job Processing in General 19

5.2 I/O Handshake 20

6 ERROR MESSAGES..21

6.1 Error Processing 21

6.2 Error Processing through LEDs and Status Messages 21

Error Processing Through Response Messages 21

7 MOBY-I/E COMMUNICATION 23

7.1 Purpose of This Chapter 23

7.2 Use of Terms 23

7.3 Hardware Components..23

7.4 Serial Connection..24

7.5 Hardware Configuration ASM 420 24

7.6 Pinout Connections ASM 420 25

7.7 MOBY Instruction Set ASM 420 26

7.8 Hardware Configuration ASM 424 27

7.9 Pinout Connections ASM 424 29

7.10 MOBY Instruction Set ASM 424 30

APPENDIX A – VIEW OF THE MVI56-3964R 31

View of the MVI56-3964R 31

APPENDIX B – PROGRAM DEFINITION..33

Introduction..33

Explanations Concerning the Disk 33

Explanations Concerning 34

the Example Programs..34

The 3964R Communication Program 35

Using the MOBY Instruction Set 35

The MOBY-I/E Communication Program..36

APPENDIX C – REFERENCE DOCUMENTS..37

Allen-Bradley 37

Siemens..37

SUPPORT, SERVICE, AND WARRANTY 39

1 Using this Manual

This manual describes the use of the 3964R protocol with an Allen-Bradley 1756-L series controller. Please refer to this manual for installation, programming and troubleshooting.

1.1 Users

In order to effectively use the 3964R protocol, the reader of this manual must be trained in programming and operating Allen-Bradley 1756-L series controllers and ControlLogix environment.

These skills are presupposed as a basis for the use of this manual and the example programs. Otherwise, the respective programming and operating manuals of the relevant controller and/or ControlLogix environment must be referred to before use.

1.2 Use of Terms

In this manual, the following terms will be used:

- The 1756-L series controller will be referred to as "CLX processor".
- For the communication partner the acronym "CP" will be used.

1.3 Products in the Environment

The 3964R protocol can be installed in all local ControlLogix 1756 I/O chassis with at least one 1756-L series controller. The functionality using a remote chassis can be confirmed if a 1756-CNB(R) or 1756-ENET (firmware revision 2.6 or higher) is used.

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

1.3.1 Product Compatibility

The communication between the MVI56 and the CLX processor is realized through MSG (CIP message) instructions and discrete (direct) I/O. A function test of the 3964R protocol together with the CP 544 communication module of Siemens has been carried out. For further details, please refer to the 3964R specification in the document:

Simatic S5 CP 544 Manual 6ES5 998-2DB11, Issue 01 (Chapters 3 & 4).

2 Installation

2.1 Scope of Delivery

The MVI56-3964R product comprises the following components:

- This manual
- A CD with example programs for the 1756-L series controller
- Two (2) 1454-9F DB9 Female to 5 Pos Screw Terminal Adapter

2.2 Installation Instructions

The following setups/limitations are to be observed:

Please make sure that you are using an MVI56-3964R with the original parameters and image as delivered.

Jumper SETUP:

Please remove to run the MVI56-3964R in application mode.

ATTENTION: Incorrect setting of the jumpers may cause damage to the MVI56-3964R 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 MVI56-3964R.

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 MVI56-3964R 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.

For the complete assignment of the RS-232 and RS-422, please refer to the **Installation Instructions ControlLogix Multi-Vendor Interface Module**, Publication 1756-IN001A-US-P. RS-485 operation is not possible.

3 3964R Protocol

3.1 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.

It is necessary that the user has sufficient knowledge about the 3964R protocol. This is required to understand the operation of the driver for the MVI56-3964R and to make efficient use of the example programs.

3.2 MVI56-3964R Operation

The protocol realized on the MVI56-3964R has the following special features:

- Immediately after power-up and completion of the internal/external initialization procedures the MVI56-3964R with installed 3964R protocol is ready to serve as an interface between a CLX processor and one or two communication partners using 3964R protocol. The parameterization of the module is realized through a MSG instruction of the CLX processor with the relevant entries. 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 Byte.
- The serial transmission parameters are optional and can be allocated independently to each interface. Possible baud rates are: **19200Bd, 9600Bd, 4800Bd, 2400Bd, 1200Bd and 600Bd.** The following parity setups are possible: **Even, Odd and None**.

4 Communications

4.1 Job Allocation by the 1756-L Series 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 MVI56_3964.acd including the tag database is recommended.

Every job which is transferred to or from the MVI56-3964R 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	
..	..	
137	User data	

4.2 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 tag **HEADER_COM2** if the user wants to send to port 2 or the tag **HEADER_COM3** if the user wants 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	00
6	00	00
7	00	00
8	00	00
9	00	00

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

4.2.1 Job Identifier

The job identifiers are defined as follows:

Send jobs to a CP:

Job:	Identifier:
MVI56 Configuration *	00CChex
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

* = Special job, refer to the paragraph “MVI56-3964R Configuration” below.

The MVI56-3964R receives jobs from a CP automatically without extra parameterization. To differentiate the header data the MVI56-3964R adds a header to the user data received and transfers it to the CLX 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

4.2.2 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”).

4.2.3 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”).

4.2.4 Number of Data Words/Data Bytes

Dependent on the type of communication mode used on port 2/3 of the module (refer to page 4-4, Protocol) 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).

4.2.5 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”.

4.2.6 MVI56-3964R Configuration

First of all, the user has to enter the parameters for the serial communication, swapping of the user data, high/low priority and whether he wants to communicate with or without RK512. The appropriate job is called MVI56-3964R configuration (identifier 00CCh) and configures both ports. The user enters the identifier and parameters either in the tag **HEADER_COM2** or **HEADER_COM3**. The table is as follows:

Word:	High Byte:	Low Byte:
0	00CChex	
1	Baud rate port 2	Baud rate port 3
2	Parity port 2	Parity port 3
3	Swapping port 2	Swapping port 3
4	Priority port 2	Priority port 3
5	Protocol port 2	Protocol port 3
6	00	00
7	00	00
8	00	00
9	00	00

The meaning of the entries and possible values are:

4.2.6.1 Baud Rate

Baud Rate:	Value
600	5
1200	4
2400	1
4800	2
9600	0 *
19200	3

* = default

4.2.6.2 Parity

Parity:	Value
Even	0 *
Odd	1
None	2

* = default

4.2.6.3 Swapping

Serves for swapping high byte <-> low byte. The sequence of the user data bytes is to be selected by entering “0” or “1”.

No Swapping = 0 *

Swapping = 1

* = default.

4.2.6.4 High/Low Priority

Defines the behavior of the MVI56-3964R in case of an initialization conflict.

High Priority = 0 *

Low Priority = 1

* = default.

4.2.6.5 Protocol

Defines which communication mode the MVI56-3964R shall use on port 2/3. The ports can be parameterized differently and independently of each other.

3964R with RK512 = 0 *

3964R without RK512 = 1

MOBY-I/E-communication = 3 **

* = default.

** **=** refer to page 4-2, Number of Data Words/Data Bytes.

An MVI56-3964R configuration job is not necessary if the default values are to be used.

4.3 Receive Jobs

If a user wants to receive jobs from a CP only, no additional parameterization is necessary. The user data received is stored in the tag database of the CLX processor. The tag database is explained in the paragraph “Tag Database”.

The table below shows an example of the data stored in Receive_Words_2_1 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

Word:	High Byte:	Low Byte:
6	00	00
7	00	00
8	00	00
9	00	00
10	User data	
..	..	
137	User data	

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

[2] = This is the number of data words stored in Receive_Words_2_1.

the total amount exceeds 128 words, the data words remaining are stored in Receive_Words_2_2.

The user data is extracted from the tags Receive_Words_2_1 (and Receive_Words_2_2) and copied to **SAVE_DATA_COM2**. This is the tag the user should operate with.

For port 3, which is similar to port 2, the tag for storing the user data received is called **SAVE_DATA_COM3**.

4.4 Send Jobs

If a user wants to send a job to a CP, the next step after a possible MVI56-3964R configuration is to define the appropriate header for the job. The table below 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.

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

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

If a user wants 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 tags of the CLX processor.

4.5 Tag Database

In our example programs the tag database of the CLX processor is organized as follows:

Tag Name:	Description:
Dummy	Tag which is used for calculations and/or constants.
Dummy_ONS	Tag which is used to serve the ONS instructions.
HEADER_COM2	Tag for storing the header data for a send job to port 2.
HEADER_COM3	Tag for storing the header data for a send job to port 3.
Local:2:C	Discrete (direct) configuration block of the MVI56-3964R.
Local:2:I	Discrete (direct) input block of the MVI56-3964R.
Local:2:O	Discrete (direct) output block of the MVI56-3964R.
Local:3:C	Discrete configuration block of an input module.
Local:3:I	Discrete input block of an input module. Used to trigger a send job.
MSG_READ_2_1	Message control for the first data block received from the MVI56-3964R. MVI56-3964R source: port 2.
MSG_READ_2_2	Message control for the second data block received from the MVI56-3964R. MVI56-3964R source: port 2.
MSG_READ_3_1	Message control for the first data block received from the MVI56-3964R. MVI56-3964R source: port 3.
MSG_READ_3_2	Message control for the second data block received from the MVI56-3964R. MVI56-3964R source: port 3.
MSG_SEND_2_1	Message control for the first data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 2.
MSG_SEND_2_2	Message control for the second data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 2.
MSG_SEND_3_1	Message control for the first data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 3.
MSG_SEND_3_2	Message control for the second data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 3.
Receive_Words_2_1	Tag for the first data block received from the MVI56-3964R. MVI56-3964R source: port 2.
Receive_Words_2_2	Tag for the second data block received from the MVI56-3964R. MVI56-3964R source: port 2.
Receive_Words_3_1	Tag for the first data block received from the MVI56-3964R. MVI56-3964R source: port 3.
Receive_Words_3_2	Tag for the second data block received from the MVI56-3964R. MVI56-3964R source: port 3.
SAVE_DATA_COM2	Tag for storing the user data extracted from Receive_Words_2_1/2.
SAVE_DATA_COM3	Tag for storing the user data extracted from Receive_Words_3_1/2.
SEND_DATA_COM2	Tag for storing the user data to be copied to Send_Words_2_1/2.
SEND_DATA_COM3	Tag for storing the user data to be copied to Send_Words_3_1/2.

Tag Name:	Description:
Send_Words_2_1	Tag for the first data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 2.
Send_Words_2_2	Tag for the second data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 2.
Send_Words_3_1	Tag for the first data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 3.
Send_Words_3_2	Tag for the second data block to be sent to the MVI56-3964R. MVI56-3964R destination: port 3.

The tags where the user data for a send job is to be entered are called **SEND_DATA_COM2** for sending to a CP connected to port 2 and **SEND_DATA_COM3** for sending to a CP connected to port 3. The example program copies the user data to different tags which are used in the MSG send instructions (e.g.: Send_Words_2_1).

The last step is to trigger the DB-Send job. In the example program this is done with the input bits called “Send Trigger COM2” (port 2) and “Send Trigger COM3” (port 3).

These bits have to be replaced by a user defined instruction which triggers the send job.

4.6 Status Messages

Status messages can occur while a job is processed or after a job is finished. The status is placed in the input data table for the MVI56-3964R.

Local:2:I.Data[x]	Port 2
Word 10:	**Status**
Word 11:	Error from CP *
Word 12:	Error from CP *
Word 13:	Error from CP *
Word 14:	Error from CP *

* refer to chapter 6, Error Processing through Response Messages.

Local:2:I.Data[x]	Port 3
Word 15:	**Status**
Word 16:	Error from CP *
Word 17:	Error from CP *
Word 18:	Error from CP *
Word 19:	Error from CP *

* refer to chapter 6, Error Processing through Response Messages.

Possible values for the **Status** are:

Value:	Description:	Take this action:
0	No error.	None.
1	Connection with CP cannot be established.	Check the physical connection to the CP; Check the serial communication parameters.
2	Communication problem after establishing connection to CP.	Check the physical connection to the CP; Check timeout settings of CP.
3	BCC summation error.	Check the physical connection to the CP.
4	Error in response telegram.	Refer to the installation instructions of the CP.
5	Parameter error: - invalid job identifier. - invalid number of data words.	Check job parameters.
6	Communication error to CLX processor.	Check the parameters of the MSG-instructions. Restart/exchange MVI56-3964R/CLX processor.
7	Internal MVI56-3964R error.	Restart/exchange MVI56-3964R.
8	Initialization conflict.	Refer to chapter 5.

For additional information refer to chapter 6.

5 Job Processing

5.1 Job Processing in General

At this point the reader should be able to program and use a ControlLogix system as well as the 3964R protocol (Chapter 3).

Detailed information about how to set up a job is described in chapter 4.

This knowledge is required to understand the job processing.

A printout of the example program MVI56_3964.acd is recommended.

Part 1(Rung 0-7):

This part is for transferring data received on port 2 from the MVI56-3964R to the CLX processor. Two MSG-Read instructions are used which are interlocked with a classic I/O handshake. Each MSG instruction is capable to transfer a maximum of 128 words.

Usage: 3964R with RK512: Receive a DB-Send job.

3964R without RK512: Receive data.

Part 2 (Rung 8-11):

This part if for sending a 3964R job from the CLX processor to the MVI56-3964R with destination port 2. The bit instruction at the beginning of rung 8 is used to trigger a job. Two MSG-Send instructions are used which are interlocked with a classic I/O handshake to transfer all data necessary. Each MSG instruction is capable of transferring a maximum of 128 words.

Usage: 3964R with RK512: Send a DB-Send job.

3964R without RK512: Send data.

Important: Please be aware that DB-Fetch jobs use both parts corresponding to the definition of a DB-Fetch. If only one part is used for communication, the rungs for the other port can be deleted.

Part 3 (Rung 12-19):

The same as part 1 for port 3.

Part 4 (Rung 20-23):

The same as part 2 for port 3.

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

A possible initialization conflict will be solved as follows:

MVI56-3964R has "High Priority":

The MVI56-3964R 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.

MVI56-3964R has "Low Priority":

The MVI56-3964R rejects its send job and sends the 3964R acknowledge (DLE) to the CP.

Important: If the MVI56-3964R 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).

5.2 I/O Handshake

Bits used for the I/O handshake (Slot dependent):

I/O Bit:	Used for port:	Set/Reset:
Local:2:I.Data[2].0	2	Automatically through MVI.
Local:2:O.Data[2].0	2	Automatically through program.
Local:2:I.Data[3].0	2	Automatically through MVI.
Local:2:O.Data[3].0	2	Automatically through program.
Send Trigger COM 2	2	**To be programmed by the user.**
Local:2:O.Data[6].0	2	Automatically through program.
Local:2:I.Data[6].0	2	Automatically through MVI.
Local:2:I.Data[7].0	2	Automatically through MVI.
Local:2:I.Data[4].0	3	Automatically through MVI.
Local:2:O.Data[4].0	3	Automatically through program.
Local:2:I.Data[5].0	3	Automatically through MVI.
Local:2:O.Data[5].0	3	Automatically through program.
Send Trigger COM 3	3	**To be programmed by the user.**
Local:2:O.Data[8].0	3	Automatically through program.
Local:2:I.Data[8].0	3	Automatically through MVI.
Local:2:I.Data[9].0	3	Automatically through MVI.

6 Error Messages

6.1 Error Processing

The error processing is realized by the two programmable LEDs of the MVI56-3964R (refer to LED U1/U2 of the MVI56-3964R in appendix A) and evaluation of the error using the table at the end of chapter 4, Status Messages.

In addition to this, possible errors in the response messages from the CP can be evaluated as well.

6.2 Error Processing through LEDs and Status Messages

The normal condition of the LEDs is the off state.

When the MVI56-3964R is processing a job, LED U1 will flash if any error occurs using port 2 and LED U2 will flash if any error occurs using port 3. The flash rate is approx. 2 Hertz.

In such a case there is either a fault in the physical/logical connection between the CP and the MVI56-3964R or a communication problem between the CLX processor and the MVI56-3964R. For exact determination of the error occurred, please refer to chapter 4, Status Messages.

Please be aware that the MVI56-3964R remains in "blinking state" until the next valid job for the appropriate port has completed correctly.

Error Processing Through Response Messages

Communication mode: **3964R with RK512.**

As the analysis through LEDs does not allow an exact examination of the error messages received from the CP, the exact error code must be determined by means of the response messages for the **DB-SEND** and the **DB-FETCH** job. The response message for each job comprises 5 words. The first word indicates whether an error occurred or not (refer to the status message table at the end of chapter 4). If there was an error indicated from the CP in a response message, the first word of the status message table will be set to "4". The next 4 words contain the response error from the CP extracted from the response messages. Since a CP error message cannot be larger than one byte, the low byte will be used for representation, whereas the high byte will be filled with 00h.

For a list of the possible error messages of the communication partner CP 544 used in this case, please refer to the corresponding installation manual (appendix C). The table below shows an example of the status message table in case of an error from the CP in the response messages. The CP is connected to port 2.

Local:2:I.Data[x]	**High Byte**	**Low Byte**
Word 10:	00hex	04hex
Word 11:	00hex	0Ahex
Word 12:	00hex	0Ahex
Word 13:	00hex	0Ahex
Word 14:	00hex	0Ahex

7 MOBY-I/E Communication

7.1 Purpose of This Chapter

This chapter describes the requirements for building a peer-to-peer MOBY-I/E Identsystem using a Siemens Interface Module ASM 420/424 and an MVI56-3964R.

7.2 Use of Terms

In this chapter 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“.

7.3 Hardware Components

The hardware components of a MOBY-I/E Identsystem using a MVI56-3964R 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.

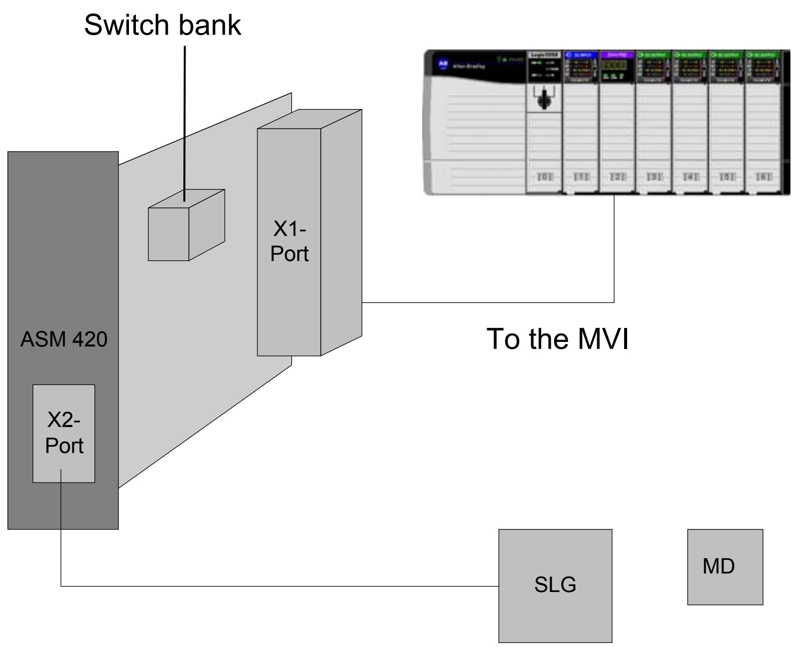
- **Read/Write device (SLG).**
- The SLG communicates wireless to the MDS for executing different MOBY operations. These operations are fixed in the MOBY instruction set.
- **Movable memory devices (MDS).**
- The MDS are the real data carriers of a MOBY-System.
- **MVI56-3964R-3964R56.**
- The MVI56-3964R interfaces as a bridge between the ASM 420/424 and the CLX processor. The connection to the ASM 420/424 is realized via 3964R without RK512.

7.4 Serial Connection

The serial connection between the ASM 420/424 and the MVI56-3964R 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 MVI56-3964R can be configured with a jumper to run either RS-422 or RS-232 (refer to the **Installation Instructions ControlLogix Multi-Vendor Interface Module**, Publication 1756-IN001A-US-P).

7.5 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 is used to interface 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 MVI56-3964R.

The **switch bank (S1)** is for setting the mode of operation. 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

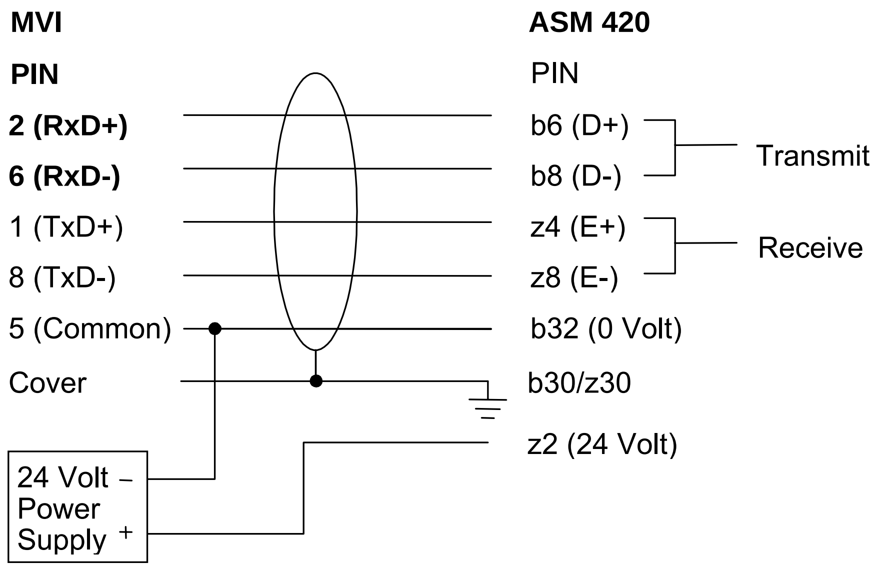
These parameters have to be fixed for the MVI56-3964R to operate with the ASM 420.

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 MVI56-3964R. Adjusting high/low priority is not relevant, because the data exchange between the ASM 420 and the MVI56-3964R is synchronized (Refer to appendix B, 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.

7.6 Pinout Connections ASM 420

The following illustration shows the pins of the ASM 420 (X1-Port) and the MVI56-3964R used for a RS-422 connection. The ASM 420 needs an external 24V DC power supply for operation.



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

7.7 MOBY Instruction Set ASM 420

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 have to be programmed in the CLX processor according to this specification and transferred into the MVI56-3964R. The MVI56-3964R subsequently communicates with the ASM 420 via 3964R without RK512 to manipulate the MDS according to the requirements of the user. It is possible to use either the “normal mode“ or the “ECC special driver“ to interface to the MDS. The CLX processor example program will be explained in Appendix B. The purpose of this section is to give you a short overview about the MOBY instruction set and its realization in the CLX processor example program. As mentioned before, the ASM 420 makes no difference between MOBY-I and MOBY-E, therefore all instructions are valid for both systems.

MOBY-Instruction: RESET

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

MOBY-Instruction: STATUS

Implementation in the CLX processor example program: without restriction.

MOBY-Instruction: DI/DO

Implementation in the CLX processor example program: without restriction.

MOBY-Instruction: NEXT

Implementation in the CLX processor example program: without restriction.

MOBY-Instruction: MDS-INIT

Implementation in the CLX 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

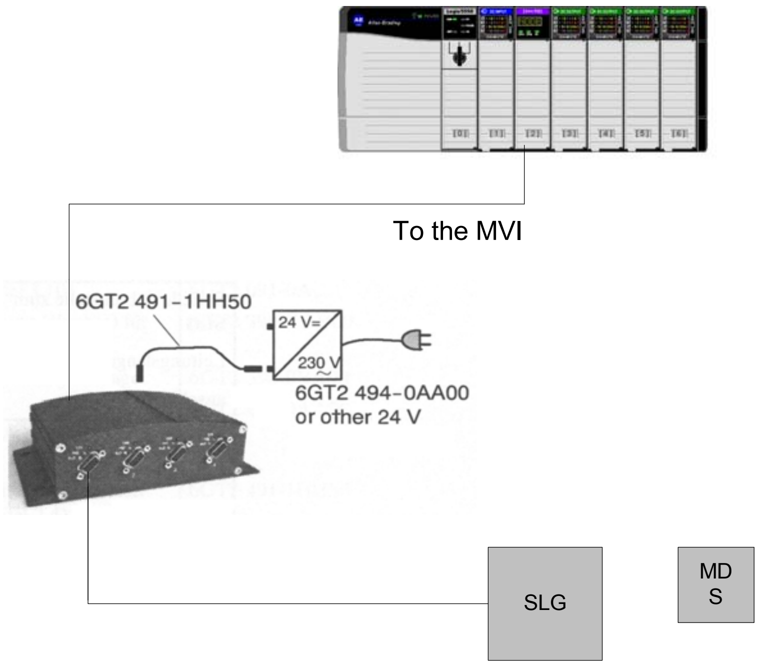
Implementation in the CLX processor example program: This instruction uses the “normal mode“ of the MDS. If you want to use the “ECC special driver“, you have to change the instruction command as explained in the technical description. In our example we write 20 bytes to the MDS.

MOBY-Instruction: DATA READ

Implementation in the CLX 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.

7.8 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 is used to interface to an SLG.

The **RS-232/RS-422 port** interfaces to the MVI56-3964R.

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 of operation. 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 MVI56-3964R (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 MVI56-3964R 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 MVI56-3964R.

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

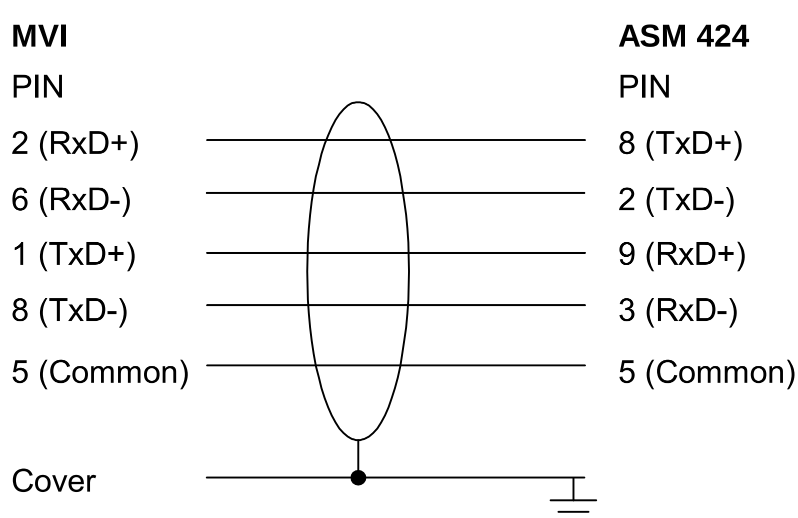
Programming Reference:

MOBY C-Library MOBY API on the CD “Software Moby”

Publication No. 6GT2 080-2AA10.

7.9 Pinout Connections ASM 424

The following illustration shows the pins of the ASM 424 (DB 9 female connector) and the MVI56-3964R used for a RS-422 connection. The ASM 424 needs an external 24V DC power supply for operation.



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

7.10 MOBY Instruction Set ASM 424

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 CLX processor according to this specification and transferred into the MVI56-3964R. The MVI56-3964R subsequently communicates with the ASM 424 via 3964R to manipulate the MDS according to the requirements of the user. It is possible to use either the “normal mode“ or the “ECC special driver“ to interface to the MDS. The CLX processor example program itself will be explained in Appendix B. The purpose of this section is to give you a short overview about the MOBY instruction set and its realization in the CLX 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.

MOBY-Instruction: RESET

Implementation in the CLX processor example program: without restriction.

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

MOBY-Instruction: STATUS

Implementation in the CLX processor example program: without restriction.

MOBY-Instruction: NEXT

Implementation in the CLX processor example program: without restriction.

MOBY-Instruction: MDS-INIT

Implementation in the CLX 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 CLX 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 CLX 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.

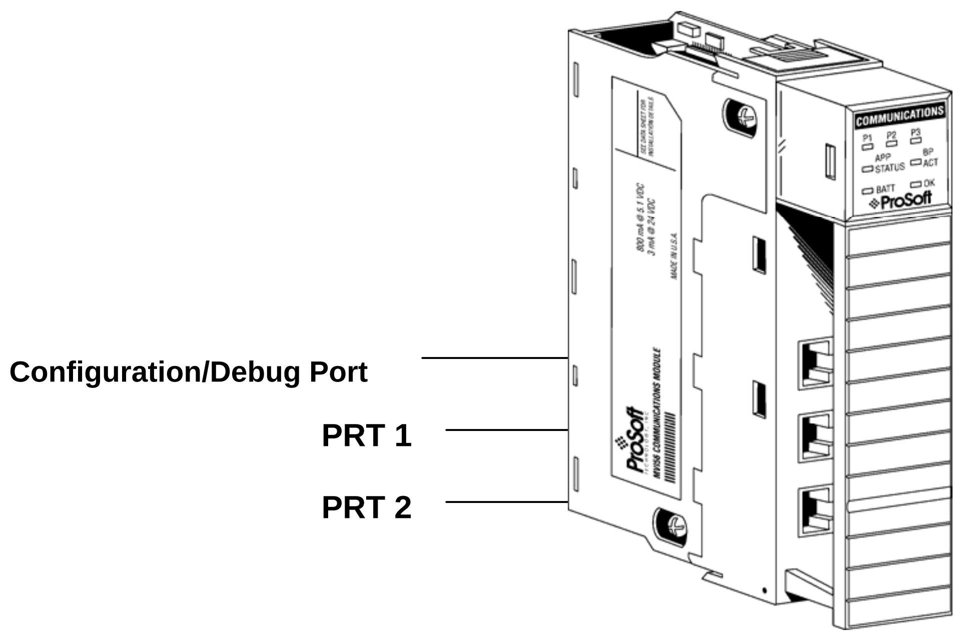
In our example we read 20 bytes from the MDS.

Appendix A – View of the MVI56-3964R

View of the MVI56-3964R

The serial connection to the communication partner is plugged into PRT1 or PRT2. Both ports can be used, i.e. it is possible to connect two CPs.

The LEDs U1/U2 are used for error diagnosis (refer to chapter 6, Error processing through LEDs and Status Messages).



Appendix B – Program Definition

Introduction

Please be aware that the example programs mentioned in the following paragraphs shall only show to the user the principle of how to communicate between an MVI56-3964R and a CLX processor. All programs mentioned were tested with appropriate communication partners.

It is up to the programmer to examine the programs in detail and integrate them into his CLX processor application.

In case of using the MOBY-I/E communication program please notice that instructions like DATA WRITE have to be brought into line according to the amount of user data to be send. Generally, the structure of the instructions can be used as implemented.

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

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

- **Setup:** The MVI56-3964R resides in slot 2 of the ControlLogix chassis.
- **Setup:** The 1756-L series controller resides in slot 0 of the ControlLogix chassis.
- **Limitation:** The example programs must run in a periodic task.

ATTENTION: The user must be trained in programming and operating Allen-Bradley 1756-L series controllers and ControlLogix environment. Otherwise, incorrect use may lead to personal injury or death, property damages or economic loss.

Important: During processing of a MSG instruction the MSG error bit (ER) may be set. In this case it is recommended to evaluate the error code (word ERR of the MSG control), too. Error code 10h (16d) means that no real MSG error has occurred in this case; it is just a feedback of the MVI56-3964R that it has no new data to send. This should be taken into account when the user wants to create an error routine.

Explanations Concerning the Disk

The example programs on the disk explain the handling of the 3964R protocol on the ControlLogix side. At this point the reader should be able to program and use a

ControlLogix system as well as the 3964R protocol (chapter 3) and job processing (chapter 5).

This knowledge is required to understand the example programs.

The disk contains the following files:

- **MVI56_3964.zip : ** General 3964R communication example program.
- **Moby_420.zip : ** MOBY-I/E communication program for ASM 420.
- **Moby_424.zip : ** MOBY-I/E communication program for ASM 424.
- **Pkunzip.exe : ** For decompressing the files above.
- **Readme.txt : ** Contains this chapter in English and German language. It is also used for giving you the latest information concerning late additions and modifications.

All example programs are compressed with „WinZip®“. Please unzip the files with PKUNZIP.EXE (which is also included on the disk) or your own compatible software. If you need additional help about PKUNZIP.EXE, e.g. concerning a destination directory for extracting, start it without declaration of a file. Please read the guidelines about using this shareware too.

If you want to use the example programs in their original form you have to make sure that the CLX processor resides in slot 0 of the Chassis and the MVI56-3964R in slot 2. Using another physical placement of the CLX processor and/or the MVI56-3964R requires appropriate changes to the example programs.

After decompressing the required files the user will be ready to start the integration of the ladder code into his own CLX processor program. Every example program contains documentation about the rungs programmed. For getting a better overview a printout is recommended.

The example programs were created with RSLogix 5000 software, revision: V7.00.00.

Explanations Concerning

the Example Programs

MVI56_3964.acd is the general 3964R communication program used for the following types of 3964R jobs:

3964R with RK 512:

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

3964R:

- Send and receive jobs.

Moby_420.acd is the example program which provides the Moby-I/E communication to an ASM 420.

Moby_424.acd is the example program which provides the Moby-I/E communication to an ASM 424.

If a user wants to build a Moby-I/E System only, the program MVI56_3964.acd will not be needed.

The 3964R Communication Program

The general functionality of the program MVI56_3964.acd is as follows:

Send jobs to a CP:

- Transfer the job data to the MVI56-3964R. Refer to chapter 4 for details.
- Acknowledgement from the MVI56-3964R regarding validity of the job data (refer to the explanations concerning the status messages in chapter 4).
- Depending on the type of send job the transfer of the user data from/to the MVI56-3964R/CLX processor takes place.
- Acknowledgement from the MVI56-3964R 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 MVI56-3964R/CLX processor takes place. The header data generated from the MVI56-3964R will be added automatically. The MVI56-3964R needs no extra job data from the CLX processor for receive jobs, it will be triggered through the STX character sent from the CP to establish the communication.
- Acknowledgement from the MVI56-3964R regarding successful/unsuccessful completion of the receive job.

During the processes described above additional status/error messages may be transferred from the MVI56-3964R to the CLX processor (refer to the explanations concerning the status messages at the end of chapter 4). Error processing is described in chapter 6.

Using the MOBY Instruction Set

The realization of the MOBY instruction set has already been made in the example programs for the CLX processor (Moby_420.zip and Moby_424.zip) on your disk. To get a better overview, a printout of the appropriate program is recommended.

As shown in the technical descriptions of the MOBY-I Identsystems used here, a complete MOBY instruction consists of a command telegram and a response telegram.

The general functionality of transferring a MOBY instruction using the MVI56-3964R subdivides into four points:

1. The CLX processor initiates a 3964R without RK512 send job to the MVI56-3964R. The user data of this job contains the appropriate MOBY instruction.
2. The MVI56-3964R communicates with the ASM using 3964R without RK512 and transfers the MOBY instruction.
3. The ASM transfers the response telegram to the MVI56-3964R.
4. The MVI56-3964R transfers the user data of the response telegram to the CLX processor.

This synchronized communication between the MVI56-3964R and the ASM 420 explains why the high/low priority setting (refer to chapter 7, **Hardware Configuration ASM 420**) is not relevant.

The MOBY-I/E Communication Program

The following paragraph describes additional details within the CLX processor program Moby_420.zip concerning the four points described in the previous section. The explanations can be easily adapted to the other MOBY-I/E communication program (Moby_424.zip). The MOBY-I/E communication programs base upon the general 3964R example program (MVI56_3964.zip).

Moby_420.zip subdivides as follows:

- **Rung 0:** MVI56-3964R configuration initiated through the first scan instruction (S:FS). The configuration is made for port 2 only but can be extended easily to configure port 3 as well (refer to chapter 4).
- The configuration parameters are (refer to chapter 7): 9600 baud, odd parity, no swapping, high priority, protocol (port 2): MOBY-I/E-communication.
- **Rungs 1-7:** These rungs copy the appropriate 3964R header data as well as the data for the MOBY instructions and initiate the send jobs. The jobs will be sent to port 2.
- **Rungs 8-31:** The program MVI56_3964.acd, except for the trigger bits (Send Trigger Com2/Com3) which initiate the send jobs to the MVI56-3964R.

If a user wants to communicate to MOBY devices using port 3 he has to configure port 3 and build the MOBY instruction set as done for port 2 (Rungs 0-7).

Appendix C – Reference Documents

Allen-Bradley

ControlLogix 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

Identsystem MOBY-I

ASM 420

Interface Module ASM 420

Technical Description

Publication No. 6GT2097-3AF00-0DA2

MOBY I Configuration, Installation and Service

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

Programming Reference

MOBY C-Library MOBY API on the CD “Software Moby”

Support, Service, and Warranty

Technical Support

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

Factory/Technical Support

ProSoft Technology, Inc.

1675 Chester Avenue. Second Floor

Bakersfield, CA 93301

(661) 716-5100

(661) 716-5101 (fax)

E-mail address: support@prosoft-technology.com

Web Site: http://www.prosoft-technology.com

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 (you may wish to fax it to us prior to calling):

1. Product Version Number
2. System hierarchy
3. Module Operation
   - Configuration/Debug status information
   - LED patterns
4. Information about the processor and controller tags as viewed through RSLogix and LED patterns on the processor
5. Details about the serial network

An after-hours answering system (on the Bakersfield number) 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 MVI56-3964R card 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 card may require repair.

When purchased from ProSoft Technology, the module has a one-year parts and labor warranty 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 need to return the card for repair, obtain an RMA number from ProSoft

Technology. Please call the factory for this number and display the number prominently on the outside of the shipping carton used to return the card.

General Warranty Policy

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.

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 misapplication or misuse of the Product; (b) failure of Customer to adhere to any of ProSoft's specifications or instructions; (c) neglect of, abuse of, or accident to, the Product; or (d) any associated or complementary equipment or software not furnished by ProSoft.

Limited warranty service may be obtained by delivering the Product to ProSoft and providing proof of purchase or 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.

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 MERCHANT ABILITY 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.

Hardware Product Warranty Details

Warranty Period: ProSoft warranties hardware product for a period of one (1) year.

Warranty Procedure: Upon return of the hardware Product ProSoft will, at its option, repair or replace 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 current ProSoft standard rates for parts and labor, and return the Product freight collect.

- - - - - - - - - - - - - - - - - END OF MANUAL - - - - - - - - - - - - - - - - - -