Modbus Interface Module

(M/N 57C414)

Industrial CONTROLS

Instruction Manual J2-3017

RELIANCE ELECTRIC

The information in this user's manual is subject to change without notice.

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

WARNING

RELIANCE STRONGLY RECOMMENDS THE USE OF AN EXTERNAL, HARDWIRED EMERGENCY STOP CIRCUIT OUTSIDE THE PROGRAMMABLE CONTROLLER CIRCUITRY. THE EMERGENCY STOP CIRCUIT MUST DISABLE THE SYSTEM IN CASE OF IMPROPER OPERATION. UNCONTROLLED MACHINE MOTION MAY RESULT IF THIS PROCEDURE IS NOT FOLLOWED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.

WARNING

INSERTING OR REMOVING A MODULE MAY RESULT IN UNEXPECTED MACHINE MOTION. POWER TO THE MACHINE SHOULD BE TURNED OFF BEFORE INSERTING OR REMOVING THE MODULE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

CAUTION

THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE.

DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Table of Contents

Appendices

List of Figures

1.0 INTRODUCTION

The products described in this instruction manual are manufactured or distributed by Reliance® Electric Industrial Company.

The Modbus Interface module is used to connect AutoMax to Modicon™ equipment via an RS-232 link utilizing the Modbus communication protocol. The Modbus Interface module turns up to four AutoMax Processors into a single Modicon 584 look-alike, as viewed from the RS-232 link. Figure 1.1 illustrates the AutoMax to Modicon interface.

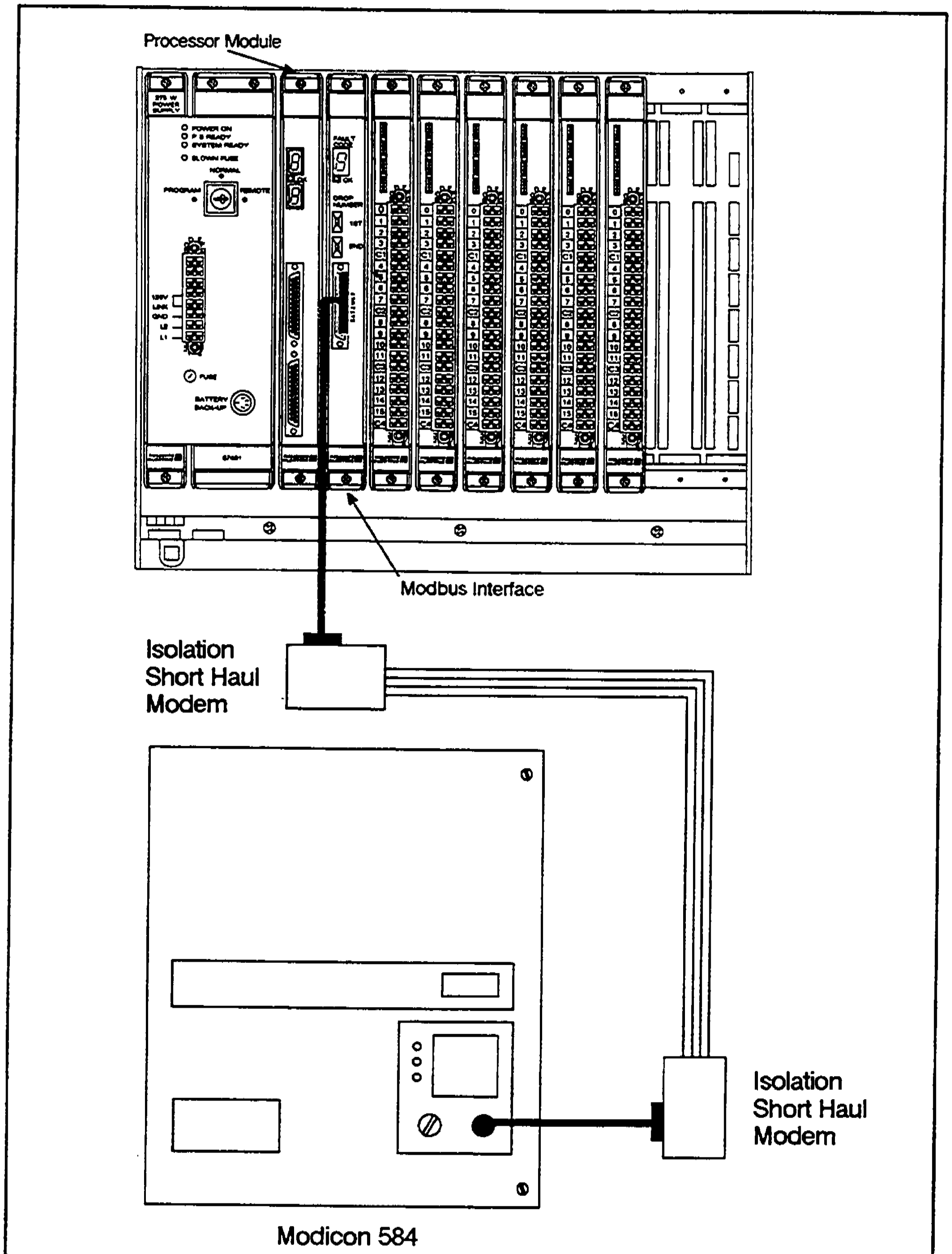


Figure 1.1 – Interfacing AutoMax to Modicon

The Modbus™ Interface module can be placed in any slot in an AutoMax™ rack that contains at least one AutoMax Processor. Each AutoMax Processor has access to the look-alike Modicon 584 registers through the AutoMax rack backplane. The Modbus Interface module will transfer data between the look-alike 584 memory image and remote Modicon-type equipment via a subset of the Modbus functions. Appendix D provides a comparison of the characteristics of Modicon equipment and the Modbus Interface module. Refer to section 4.5 for individual function messages.

The remainder of this manual describes the functions and specifications of the Modbus Interface module. It also includes a detailed description of module installation and servicing procedures, as well as programming methods.

1.1 Additional Information

You must be familiar with all the instruction manuals that describe your system configuration. This may include, but is not limited to, the following:

- J-3616 KERMIT™ COMMUNICATIONS SOFTWARE INSTRUCTION MANUAL
- J-3618 NORTON® EDITOR INSTRUCTION MANUAL
- J-3636 COMMON MEMORY MODULE INSTRUCTION MANUAL
- J-3649 AutoMax CONFIGURATION TASK INSTRUCTION MANUAL
- J-3650 AutoMax PROCESSOR MODULE INSTRUCTION MANUAL
- J-3669 AutoMax Pocket Reference
- J-3675 AutoMax ENHANCED BASIC LANGUAGE INSTRUCTION MANUAL
- J-3676 AutoMax CONTROL BLOCK LANGUAGE INSTRUCTION MANUAL
- J-3677 AutoMax LADDER LOGIC LANGUAGE INSTRUCTION MANUAL
- J-3684 ReSource™ AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL VERSION 2.0
- J-3692 ISOLATION SHORT HAUL MODEM INSTRUCTION MANUAL
- J-3750 ReSource AutoMax PROGRAMMING EXECUTIVE INSTRUCTION MANUAL VERSION 3.0
- IEEE 518 GUIDE FOR THE INSTALLATION OF ELECTRICAL EQUIPMENT TO MINIMIZE ELECTRICAL NOISE INPUTS TO CONTROLLERS
- Your personal computer and DOS operating system manual(s)
- Other instruction manuals applicable to your hardware configuration

1.2 Related Hardware and Software

M/N 57C414 contains one Modbus Interface Module. The module is used with the following hardware and software:

1. M/N 57C430A, 57C431, 57C435 AutoMax Processor.
2. IBM®-AT compatible personal computer running DOS version 3.1 or later.
3. M/N 61C127 RS-232C ReSource Interface Cable. This cable is used to connect the personal computer to the Processor module.
4. M/N 57C413 Common Memory Module. This module is used when there is more than one Processor module in a rack.
5. M/N 57C382 Isolation Short Haul Modem. Two Short Haul Modems are required per application. This modem provides isolation for the communications port on the Modbus Interface module.
6. M/N 57C383 Module Interface Cable. This cable is used to connect the Modbus Interface module to the Short Haul Modem.
7. M/N 57C301 - 57C303 DCS System Operation Manual.
8. M/N 57C391 AutoMax Programming Executive Version 2.
9. M/N 57C395 AutoMax Programming Executive Version 3.

2.0 MECHANICAL/ELECTRICAL DESCRIPTION

This section describes the mechanical and electrical characteristics of the Modbus Interface module.

2.1 Mechanical Description

The Modbus Interface module is a printed circuit assembly that plugs into the backplane of the AutoMax Rack. The module consists of a printed circuit board, faceplate, and protective enclosure. The faceplate contains ejector tabs at the top and bottom to simplify removing the module from the rack. See figure 2.1 for an illustration of the module faceplate.

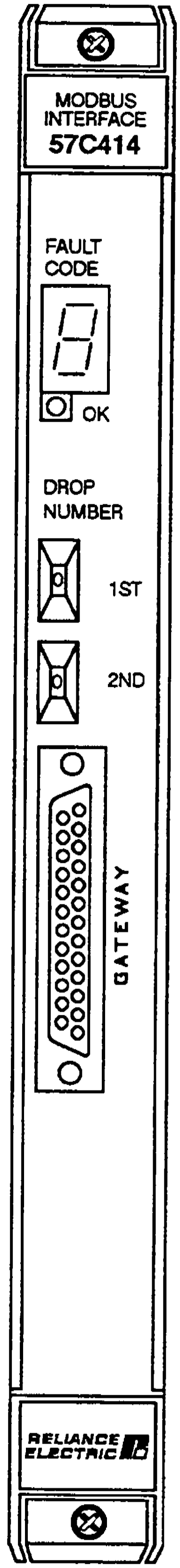


Figure 2.1 – Modbus Interface Module Faceplate

The 25-pin D-shell connector on the faceplate supports the RS-232 compatible serial port. The two thumbwheel switches on the faceplate are used to set the module's device number in decimal. The upper thumbwheel switch represents the most significant digit; the lower thumbwheel switch represents the least significant digit. On the back of the module are two edge connectors that attach to the system backplane.

For diagnostic purposes, the faceplate contains a seven-segment LED which displays error codes. The error codes are defined in Appendix D. A green status LED on the faceplate indicates when the module is operational (ON) or should be replaced (OFF).

2.2 Electrical Description

The Modbus Interface module contains a 4 MHz Z80A microprocessor. The Z80A connects to one port of the module's dual port memory, while the other port interfaces with the AutoMax rack backplane. A block diagram is shown in Appendix B.

The module contains a watchdog timer which is enabled when power is turned on to the module. The on-board Z80A processor must reset the watchdog timer within a specified time or the Z80A will shut down and the status LED on the faceplate will turn off.

At power-up, the on-board processor will run diagnostics on the CPU, EPROM, RAM, serial I/O, memory management unit, and dual port memory, as well as perform system-level diagnostics. As each test is run, a number is written out to the seven-segment display. If there is a fault during the diagnostics, the Z80A will halt, the watchdog will time out, and the seven-segment display will show the code of the failed diagnostic.

3.0 INSTALLATION

This section provides instructions for how to install the Modbus Interface module and its cable assembly.

DANGER
ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

3.1 Wiring

DANGER
THE USER IS RESPONSIBLE FOR CONFORMING WITH THE NATIONAL ELECTRICAL CODE AND ALL OTHER APPLICABLE LOCAL CODES. WIRING PRACTICES, GROUNDING, DISCONNECTS, AND OVER-CURRENT PROTECTION ARE OF PARTICULAR IMPORTANCE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

To reduce the possibility of electrical noise interfering with the proper operation of the control system, exercise care when installing the wiring from the system to the external devices. For detailed recommendations, refer to IEEE 518.

3.2 Initial Installation

Use the following procedure to install the module:

WARNING
INSERTING OR REMOVING A MODULE MAY RESULT IN UNEXPECTED MACHINE MOTION. POWER TO THE MACHINE SHOULD BE TURNED OFF BEFORE INSERTING OR REMOVING THE MODULE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN BODILY INJURY.

CAUTION
THIS MODULE CONTAINS STATIC-SENSITIVE COMPONENTS. CARELESS HANDLING CAN CAUSE SEVERE DAMAGE. DO NOT TOUCH THE CONNECTORS ON THE BACK OF THE MODULE. WHEN NOT IN USE, THE MODULE SHOULD BE STORED IN AN ANTI-STATIC BAG. THE PLASTIC COVER SHOULD NOT BE REMOVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT.

Step 1. Stop any application tasks that may be running.

Step 2. Turn off power to the rack. All power to the rack as well as all power to the wiring leading to the rack should be off.

DANGER

THIS EQUIPMENT IS AT LINE VOLTAGE WHEN AC POWER IS CONNECTED. DISCONNECT AND LOCK OUT ALL UNGROUNDED CONDUCTORS OF THE A-C POWER LINE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

Step 3. Take the module out of its shipping container. Take it out of the anti-static bag, being careful not to touch the connectors on the back of the module.

Step 4. Insert the module into the desired slot in the rack, making sure it is well-seated in the rack. The module may reside in any slot in the rack. Use a screwdriver to secure the module into the rack.

Step 5. Set the device number of the Modbus Interface module (in decimal) using the two thumbwheel switches on the module faceplate. The upper switch is the most significant digit and the lower switch is the least significant digit.

The device number settings are recognized only at power up. Therefore, make sure each device has a unique device number (01-99) before power is applied. If more than one device is given the same number, transmission collisions will occur on the network. Device number 00 is an invalid number.

Step 6. Connect one end of the interface cable (M/N 57C383) to the 25-pin connector on the module faceplate and the other end to a Short Haul Modem (M/N 57C382). Connect the other Short Haul Modem to the Modicon device. Connect the two modems using two twisted pairs of solid or stranded conductors (see J-3692 for modem wiring instructions).

Step 7. Turn on power to the rack. An internal diagnostic routine is automatically executed by the module. If an error is encountered, an error code will be displayed on the seven-segment LED. If the green status light is OFF and no seven-segment error code is displayed, a local watchdog failure has occurred. If a diagnostic fault code 0 through 9 or b is displayed, the Modbus Interface module must be replaced. (Refer to Appendix E for a description of the error codes.)

If the thumbwheel switches are set to an invalid device number, the "A" fault code will be displayed on the seven-segment LED on the module faceplate after power-up. To clear the invalid device number fault code, refer to section 5.2.

3.3 Module Replacement

Use the following procedure to replace the Modbus Interface module.

Step 1. Stop any application tasks that may be running.

> **DANGER**
> **THIS EQUIPMENT IS AT LINE VOLTAGE WHEN AC POWER IS CONNECTED. DISCONNECT AND LOCK OUT ALL UNGROUNDED CONDUCTORS OF THE A-C POWER LINE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

Step 2. Turn off power to the rack.

Step 3. Disconnect the cable from the module faceplate.

Step 4. Use a screwdriver to loosen the screws that hold the module in the rack. Take the module out of the slot in the rack.

Step 5. Take the new module out of its shipping container. Take it out of the anti-static bag, being careful not to touch the connectors on the back of the module.

Step 6. Insert the module into the rack, making sure it is wellseated in the rack.

Step 7. Set the device number of the module to the same number as the old module by using the two thumbwheel switches on the module faceplate. The upper switch is the most significant digit, the lower switch is the least significant digit.

Step 8. Attach the cable to the module faceplate.

Step 9. Turn on power to the rack.

Step 10. Verify the installation. After the powerup diagnostics are completed, the green status LED will go on.

4.0 PROGRAMMING

The Modicon system network is based on a Master/Slave organization. All activity on the Modbus is controlled by the bus Master while the Slave modules simply respond to the requests of the Master. A Modbus Interface module can operate as a Master or a Slave device and can be in any slot in an AutoMax rack that contains at least one Processor module.

When the Modbus Interface module is acting as a Slave device, any incoming messages from the Modbus Master device will cause the registers in the Modbus Interface module's Modicon 584 memory image to be updated automatically without the intervention of any AutoMax application task. Likewise, any incoming messages requesting data from the Modbus Interface module will cause the data to be obtained from the Modbus Interface module's Modicon 584 memory image and be transmitted without the intervention of any AutoMax application task.

When the Modbus Interface module is operating as the Master, an AutoMax application task will initiate data transfers to and from the Slave device by issuing commands to the Modbus Slave, or update the data in the Slave from the Modbus Interface module's Modicon 584 memory image.

This section describes how the data is organized in the module and provides examples of how the module is accessed by the application software. For more detailed information on programming, refer to the AutoMax Programming Reference Binder (J-3686).

4.1 Register Organization

The Modbus Interface module contains a dual port memory that can be accessed through the AutoMax rack backplane by application tasks running on the AutoMax Processor as well as by the Modicon device attached to the 25-pin connector. The dual port memory contains the status and control data, the Modicon 584 image data, and the command message buffer area. The dual port memory map is shown in figure 4.1.

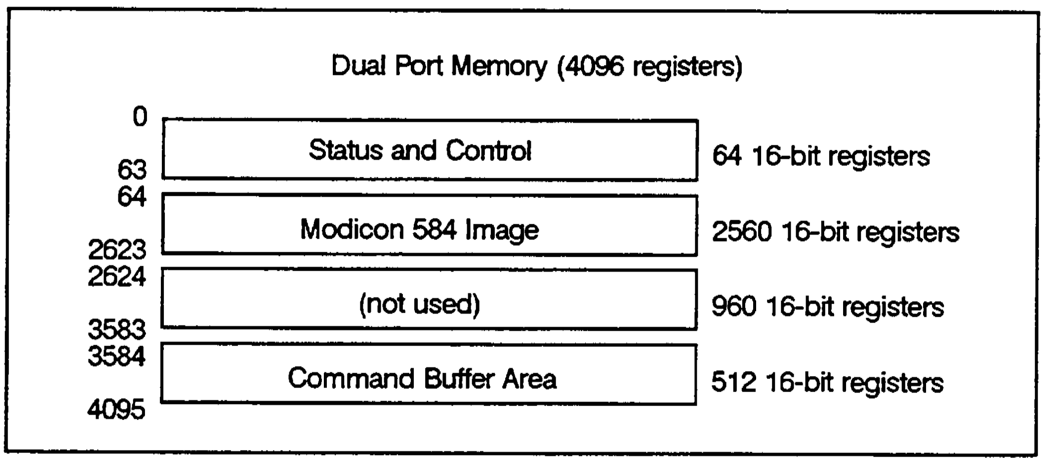


Figure 4.1 - Dual Port Memory Map

4.1.1 Status and Control Registers

All status and control registers are mapped as READ ONLY with the exception of registers 20-25. Refer to figure 4.2 for status and control register assignments.

Status and Control Register Assignments

R/W	Reg. #	Description
R	0	Status and Control Register 1 (System use only)
R	1	Status and Control Register 2 (System use only)
R	2	Status and Control Register 3 (System use only)
R	3	Status and Control Register 4 (System use only)
R	4	Device Status (bit 0)
	5-11	Not used
R	12	Device number
R	13	Program mode (keyswitch position 1 = memory protect, 2 = setup, 3 = program)
R	14	Messages received
R	15	Receive timeouts
R	16	CRC/parity errors
R	17	Overrun errors
R	18	Framing errors
R	19	Messages transmitted
R/W	20	Configuration/update request
R/W	21	Baud rate (1200, 2400, 4800, 9600, 19200)
R/W	22	Response timeout (seconds) (Master mode)
R/W	23	Number of retries (Master mode)
R/W	24	Response turnaround delay (milliseconds) (Slave mode)
R/W	25	Transmit delay (0-5 "character times")
	26-49	Not used
R	50-54	Used for Display Mode
	55-61	Not Used
R	62-63	Interface module identification (ASCII 'GTWY')

Figure 4.2 - Status and Control Register Assignments

Registers 0-3 are used by the GATEWAY_CMD_OK@ function and should not be used by the AutoMax application programs.

Register 4, the device status register, defines the state of the link configuration. Bit 0 will be set to "1" after you have properly configured the module.

Register 12 contains the device number set on the faceplate of the module.

Register 13 contains the status of the keyswitch on the Power Supply in the AutoMax rack.

Registers 14-19 are error and status registers that should be monitored to ensure the communication link is functioning properly.

Register 20 is the link configuration/update request register. The module continually monitors this register and will re-configure the link any time the low byte of the register is non-zero. After the link configuration, the module will zero the low byte to indicate that the link was re-configured. When bit 8 is set, the first 10 bytes of the last message received will be stored in registers 50-54. See Appendix G for how to monitor registers 50-54. Figure 4.3 illustrates register 20.

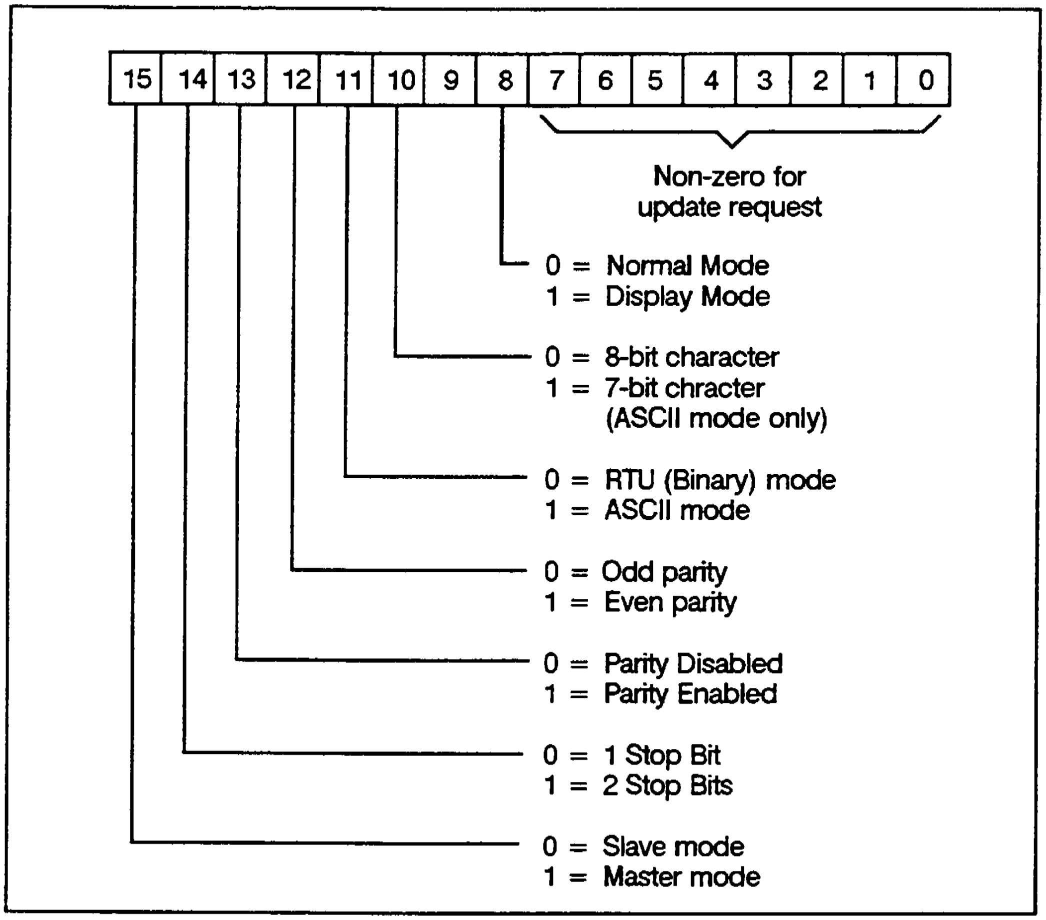


Figure 4.3 - Register 20 Configuration Parameters

Register 21 defines the serial port baud rate. This register must be configured before register 20.

Register 22 (Master mode only) defines the time in which a response must be received from the Slave device before re-transmitting the command. The units are seconds and the range of acceptable values is 1-10 (default = 1).

Note: After storing the command in the dual port buffer on the Modbus Interface module, the GATEWAY_CMD_OK@ function tests the completion bit (15) in the ISCR and, if not complete, delays for a specified time, and then repeats the test for completion. The time used for the delay is calculated as the value in register 22 divided by eight. For example, for a one second response timeout, the GATEWAY_CMD_OK@ function will test for completion every one-eighth of a second (125 milliseconds).

Register 23 (Master mode only) defines the number and frequency of re-transmissions of the command to the Slave device after the initial transmission. The range is 0-10, and the default value is 2.

Register 24 defines the time (in milliseconds) that the Slave device will delay before responding to a message from the Master. The maximum value is 225. The default value is 0.

Register 25 defines the transmit delay. The transmit delay is the interval between the setting of the DTR bit (pin 20 of 25-pin connector) and the transmission of the first character of the message. The interval is specified in "character time." A character time is the time required to transmit a character at the selected baud rate. The value may range from 0 to 5 character times. The default is 0.

4.1.2 Modicon 584 Register Image

The Modicon 584 data image in the dual port memory appears as a subset of the Modicon 584 registers as shown in figure 4.4.

Multibus™ Access (AutoMax application tasks)		Modbus Access (Modicon device)
Read Only	Modicon Registers 00001-04096 Coils/Discrete Outputs (1-bit registers)	Read/Write
Read/Write	Modicon Registers 00001-14096 Discrete Inputs (1-bit registers)	Read Only
Read/Write	Modicon Registers 30001-31024 Input Registers (16-bit registers)	Read Only
Read/Write	Modicon Registers 40001-41024 Holding Registers (16-bit registers)	Read/Write

Figure 4.4 – Modicon 584 Data Image

4.2 Data Update Rate

The time required for the Master to communicate with a Slave depends on the following parameters:

- Transmission time of a request or response message
- Formulation of the request message in the Master
- Slave turn-around time
- Response message processing in the Master

Appendix I describes methods for estimating the time required for transmitting request and response messages.

4.3 AutoMax to Modbus Interface Communication

Application programs communicate with the Modbus Interface module by defining the Modicon 584 image registers in the dual port memory. In AutoMax Version 3.0 and later, you define these registers using the Programming Executive software. If you are using AutoMax Version 2.1 or earlier, this is done using MODDEF statements in the rack configuration task. The format for the MODDEF statement is as follows:

```
nnnn    MODDEF var_name [SLOT=slot number,
        REGISTER=register number]
```

where:

nnnn = configuration task line number.

var_name = integer or boolean variable. Double integer variables can be used but should be avoided because of the possibility that all 32 bits will not transfer in one operation.

> **WARNING**
>
> **IF YOU USE DOUBLE INTEGER VARIABLES IN THIS INSTANCE, YOU MUST IMPLEMENT A SOFTWARE HANDSHAKE BETWEEN THE TRANSMITTER AND RECEIVER TO ENSURE THAT BOTH THE LEAST SIGNIFICANT AND MOST SIGNIFICANT 16 BITS HAVE BEEN TRANSMITTED BEFORE THEY ARE READ BY THE RECEIVING APPLICATION PROGRAM. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY INJURY.**

slot number = slot number of the module. This number may range from 0 to 15.

register number = register number on the module.

Any variables contained in the rack configuration are accessible by any task on any AutoMax Processor in the rack. When AutoMax variables are referenced in AutoMax tasks, the data is directly obtained from or written to the Modicon 584 register image. The data storage for an AutoMax variable mapped to a Modicon 584 register will always exist in the Modbus Interface module's dual port memory.

None of the coils (discrete outputs) or registers will be retentive. All of the registers in the Modicon 584 register image are readable from the Modbus link. However, it will not be possible to write to the discrete inputs (10001-14096) and input registers (30001-31024) from the Modbus link or to write to the discrete outputs (or coils) from the Multibus.

The registers in the Modbus Interface module's Modicon 584 image may be displayed using the AutoMax Monitor I/O function. Refer to Appendix H for the method used to convert Modicon 584 register numbers so they can be viewed using the monitor.

Commands are initiated from a BASIC application program by executing the GATEWAY_CMD_OK@ function:

```
GATEWAY_CMD_OK@(status, cmd_code, slave_drop, &
      slave_reg, master_var, num_regs)
```

where:

status is an integer variable representing the location where the resulting command status is stored; status will contain a zero if the transfer operation is successful and an error code if it is unsuccessful

cmd_code is a variable name or expression of type integer representing the command sent to the Interface module; the commands available are found in Figure 4.5.

slave_drop is a variable name or expression of type integer representing the device number of the hardware the Interface module is communicating with.

slave_reg is a variable name or expression of type string representing the starting register in the device that is to be read from/written to.

master_var is a variable name or expression (via the VARPTRI function) of type double integer, representing the physical address of the first register that is to be read from/written to in the Interface module.

num_regs is variable name or expression of type integer that defines the number of bits or registers (16 bits each) to be transferred; cmd_code determinaes whether the variable represents bits or registers.

The GATEWAY_CMD_OK@ function will be true if the command was successfully completed. If the function is false, the returned status will contain an error code. See Appendix F for the error codes returned by the GATEWAY_CMD_OK@ function.

4.3.1 Link Configuration

The link between the Modbus Interface module and the Modicon 584 is configured through an AutoMax BASIC application program. The status and control registers are defined as 16-bit registers and may be addressed by the application task through the use of IODEF statements. The program must first set the baud rate for the link in status and control register 21 and then store the desired operating mode, transmission characteristics and update request flag in register 20.

The examples that follow show the Configuration task and BASIC statements required to configure the Modbus Interface module for operation as a link Master or as a link Slave. Note that the link configuration register (register 20) must be the last variable defined in the BASIC application program.

Example 1 – Link Master

The Modbus Interface module is the link Master. The link will operate in the ASCII mode with seven bits per character at 9600 baud with no parity and one stop bit. The default value for the number of re-tries is used, but the response timeout is set to two seconds. The response turn-around delay is not applicable in the Master mode.

If you are using AutoMax Version 2.1 or earlier, the following registers must be defined using IODEFs in the rack configuration. If you are using AutoMax Version 3.0 or later, these registers are defined using the Variable Configurator within the Programming Executive.

```
Register 20 - LINK_CONF%
Register 21 - BAUD_RATE%
Register 22 - RESP_TIME%
```

The application task will contain the following statements:

```
:
100  RESP_TIME% = 2
110  BAUD_RATE% = 9600
120  LINK_CONF% = 08CFFH
:
```

Example 2 – Link Slave

The Modbus Interface module is the link Slave. The link will operate in the binary (RTU) mode with eight bits per character at 19200 baud, even parity, and one stop bit. The default values for the number of re-tries and response timeout are used. The response turn-around delay is set to 10 milliseconds.

If you are using AutoMax Version 2.1 or earlier, the following registers must be defined using IODEFs in the rack configuration. If you are using AutoMax Version 3.0 or later, these registers are defined using the Variable Configurator within the Programming Executive.

```
Register 20 – LINK_CONF%
Register 21 – BAUD_RATE%
Register 24 – RESP_DELAY%
```

The application task will contain the following statements:

```
:
100   RESP_DELAY% = 10
110   BAUD_RATE% = 19200
120   LINK_CONF% = 30FFH
:
```

The microprocessor in the Modbus Interface module will continually monitor the configuration register (register 20) and will reconfigure the Modbus link any time the Update Request byte is set. The internal software will reset the Update Request byte during the configuration process and set the Device Status Bit (bit 0 of register 4) to a TRUE or ON state when the Modbus link has been configured successfully. If both the Device Status bit and Update Request byte are reset after a configuration attempt, the specified baud rate was unacceptable.

Example 3 – Application Task Containing GATEWAY Command

Problem: Read 5 registers from Slave node address 12 beginning with register 40005 and store the data in the Master starting at register M_XXX%. Note that the VARPTR! argument returns a double integer value of the address of the specified argument. In this example, the argument is M_XXX%, which is defined in the rack configuration to be register 40010 on the Modbus Interface module.

```
:
500   COMMON   M_XXX%
600   LOCAL    RET_STAT%, MODBUS_RDREG%, SLVDROP%, SLVREG$
610   LOCAL    MSTR_REG!, XFER_SIZE%
:
1000  MODBUS_RDREG% = 3
2000  SLVREG$ = "40005"
2010  IF NOT GATEWAY_CMD_OK@( RET_STAT%, MODBUS_RDREG%, 12,          &
      SLVREG$, VARPTR!(M_XXX%), 5) THEN 20000 {process errors}
:
```

4.3.2 Commands Used with GATEWAY_CMD_OK@

When the Modbus Interface module is used as a Modbus Master by the AutoMax system, the Master mode commands are as follows. Refer to figure 4.5.

GATEWAY_CMD_OK@ Command Number	Command Description	Equivalent Modbus Function
01	Read Discrete data from Slave	01, 02
02	Write discrete data to Slave	05, 15
03	Read register data from Slave	03, 04
04	Write register data to Slave	06, 16
:		
:		
FF	All others are undefined	

Figure 4.5 - GATEWAY_CMD_OK@ Commands

Command 01 transfers discrete (bit) data from the Slave's coil (0xxxx) or Input Status (1xxxx) registers to the Master's coil data or holding register data areas. The Slave register number must be 0xxxx or 1xxxx, while the Master variable must be assigned to the Master's 0xxxx or 4xxxx registers. The transfer size is the number of bits to transfer from the Slave and must not exceed 2000.

Command 02 transfers discrete (bit) data from any data area in the Master to the Slave's coil (0xxxx) registers. The Slave register number must be 0xxxx, while the Master variable may be assigned to any register (0xxxx, 1xxxx, 3xxxx, 4xxxx). The transfer size is the number of bits to transfer to the Slave and must not exceed 800.

Command 03 transfers 16-bit data from the Slave's Input (3xxxx) or holding (4xxxx) registers to the Master's holding register area or coil data areas. The Slave register number must be 3xxxx or 4xxxx, while the Master variable may be assigned to the Master's 4xxxx or 0xxxx registers. The transfer size is the number of 16-bit values to transfer to the Slave and must not exceed 125. Note that if data is being written into the Master's 0xxxx area, the variable name must specify a coil register starting at bit "0" in a 16-bit word, such as 00001, 00017, 00033, 00049, etc.

Command 04 transfers 16-bit data from any Master data area to the Slave's holding (4xxxx) registers. The Slave register number must be 4xxxx, while the Master variable may be assigned to any Master register any master register (0xxxx, 1xxxx, 3xxxx, 4xxxx). The transfer size is the number of 16-bit values to transfer to the Slave and must not exceed 100. Note that if data is being transferred from the Master's 0xxxx or 1xxxx area, the variable name must specify a coil register starting at bit "0" in a 16-bit word, such as 00001, 00017, 00033, 00049, etc.

4.4 Message Format

The Modbus Interface module is capable of communicating in either the ASCII mode or the RTU (binary) mode. This section gives a brief description of the message formats for each mode. The message formats are illustrated in figures 4.7 and 4.8. Details of the RTU message formats for the functions supported by the Modbus Interface module and the exception code response are also provided in this section. The following is a description of the major fields in both message formats.

The **Device Number** is the first field of the message and consists of a single byte. In the Slave mode, the data in this field should correspond to the number selected on the Modbus Interface module thumbwheel switches. The only exception is for a "broadcast" message for which the Device Number is 0. Only functions 05, 06, 15, and 16 accept a broadcast message. No response is given for broadcast messages. In the Master mode, this field should contain the device number of the Slave.

The **Function Code** is a single byte field that defines the action the Modbus Interface module is to perform. If an error is detected in a received message, the high order bit of this field is set in the response message. Only functions 01 through 08, 15, and 16 are supported by the Modbus Interface module.

The **Message Information** field size and content are functiondependent and are described under the specific function. The register/coil number in this field is a maximum of four digits since the most significant digit is uniquely defined by the function. The register/coil number is relative to zero rather than one. Therefore, a register number of zero in Function 03 would specify register 40001 as the starting register.

The **Error Check** characters are the Cyclical Redundancy Check (CRC-16).

The same message transmitted in ASCII will require approximately twice the number of bytes required for the RTU mode. Each byte is broken into two 4-bit nibbles and converted into ASCII digits for transmission. For example, the request to read holding register 40317 through 40340 from the Modbus Interface module having a device number of 62 would appear as shown in figure 4.6 (in hexadecimal):

Message Field	Device Number	Function Code	Address of First Register -1	Number of Registers	Error Check	
# Bytes	1	1	2	2	2	RTU
Hex Data	3E	03	01 3C	00 18	81 3F	RTU
# Bytes	2	2	4	4	2	ASCII
Hex Data	33 45	30 33	30 31 33 43	30 30 31 38	36 41	ASCII

Figure 4.6 - Comparison of RTU and ASCII Message Formats

4.4.1 RTU (Binary) Mode

The format of the message in the RTU mode is shown in figure 4.7. There are actually no Start or End Message characters; the Start and End Messages are three-character times of quiet on the transmission line.

	Start Message	Device Number	Function Code	Message Information	Error Check	End Message
Bytes	(3)	1	2	Function Dependent	2	(3)

Figure 4.7 - RTU Message Format

4.4.2 ASCII Mode

Unlike the RTU mode, ASCII protocol uses specific Start and End message characters. The start of a message is signified by an ASCII colon (:), while the end of a message is denoted by a carriage return (CR) and a line feed (LF). The content of the message is identical in either ASCII or RTU mode; however, ASCII mode uses the Longitudinal Redundancy Check (LRC) error check scheme. The LRC is calculated by taking the two's complement of the sum of the individual bytes of binary data. In other words, the LRC is accumulated before the byte is broken into "nibbles" and converted to ASCII data. The Start and End message characters are not included in the LRC. Figure 4.8 describes the ASCII Message format.

	Start Message	Device Number	Function Code	Message Information	Error Check	End Message
Bytes	:	2	2	Function Dependent	2	CR LF

Figure 4.8 – ASCII Message Format

4.5 Function Codes Supported by Modbus Interface Module

The subset of the Modbus function codes supported by the Modbus Interface module are show in figure 4.9.

Code	Function Name	Register
01	Read coil status	0xxxx
02	Read input status	1xxxx
03	Read holding register	4xxxx
04	Read input register	3xxxx
05	Modify coil status	0xxxx
06	Modify holding register	4xxxx
07	Read Slave Status	00257-00264 (Slave only)
08	Loopback diagnostic	n/a (Slave only)
15	Preset multiple coils	0xxxx
16	Preset holding registers	4xxxx

Figure 4.9 – Modbus Function Codes

4.5.1 Function 01 - Read Coil Status (0xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Register -1		Number of Registers		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	1-99	01	0-4095		1-2000			

Response (RTU Mode)

Message Field	Device Number	Function Code	Byte Count	Coil Status Data	Error Check	
Bytes	1	1	1	Byte Count	2	
Tx Order				See Note	LOB	HOB
Range	As Rcv	01	1-250	1-250		

Note: The Byte Count defines the number of bytes of coil data packed as eight coils per byte. The first byte contains the first eight coils requested, the second byte contains the next eight coils, etc. If the number of coils requested is not a multiple of eight, the remainder of the byte is filled with zeros. The lower number coil bit is the least significant bit of the byte.

4.5.2 Function 02 - Read Input Status (1xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Input -1		Number of Inputs		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	1-99	02	0-4095		1-2000			

Response (RTU Mode)

Message Field	Device Number	Function Code	Byte Count	Input Status Data	Error Check	
Bytes	1	1	1	Byte Count	2	
Tx Order				See Note	LOB	HOB
Range	As Rcv	02	1-250	1-250		

Note: The Byte Count defines the number of bytes of input data packed as eight inputs per byte. The first byte contains the first eight inputs requested, the second byte contains the next eight inputs, etc. If the number of inputs requested is not a multiple of eight, the remainder of the byte is filled with zeros. The lower number coil bit is the least significant bit of the byte.

4.5.3 Function 03 - Read Holding Register (4xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Register -1		Number of Registers		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	1-99	03	0-1023		1-125			

Response (RTU Mode)

Message Field	Device Number	Function Code	Byte Count	Register Data	Error Check	
Bytes	1	1	1	Byte Count	2	
Tx Order				See Note	LOB	HOB
Range	As Rcv	03	2-250	2-250		

Note: The Byte Count indicates the number of bytes of register data to follow that is equal to two times the number of registers requested. The first byte of register data transmitted is the high order byte of the first register requested, followed by the low order byte. The remaining registers are transmitted in high order/low order sequence.

4.5.4 Function 04 - Read Input Register (3xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Register -1		Number of Input Regs		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	1-99	04	0-1023		1-125			

Response (RTU Mode)

Message Field	Device Number	Function Code	Byte Count	Register Data	Error Check	
Bytes	1	1	1	Byte Count	2	
Tx Order				See Note	LOB	HOB
Range	As Rcv	04	2-250	2-250		

Note: The Byte Count indicates the number of bytes of register data to follow that is equal to two times the number of registers requested. The first byte of register data transmitted is the high order byte of the first register requested, followed by the low order byte. The remaining registers are transmitted in high order/low order sequence.

4.5.5 Function 05 - Modify Coil Status (0xxxx)

Request/Response (RTU Mode)

Message Field	Device Number	Function Code	Address of Coil -1		Number of Input Regs		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	0-99	05	0-4095		See Note			

Note: To turn a coil ON, the high order byte of the coil status is all ones, and the low order byte is all zeros. To turn a coil OFF, both bytes are zero.

4.5.6 Function 06 - Modify Holding Register (4xxxx)

Request/Response (RTU Mode)

Message Field	Device Number	Function Code	Address of Register -1		Coil Status		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	0-99	06	0-4095		See Note			

Note: Holding register value is in the range -32768 to +32767. Negative values are expressed in two's complement notation.

4.5.7 Function 07 - Read Slave Status

Request (RTU Mode)

Message Field	Device Number	Function Code	Error Check	
Bytes	1	1	2	
Rx Order			LOB	HOB
Range	1-99	07		

Response (RTU Mode)

Message Field	Device Number	Function Code	Coil Status	Error Check	
Bytes	1	1	1	2	
Tx Order				LOB	HOB
Range	As Rcv	07	0-255		

Note: This function returns the status of discrete coils 00257 through 00264. Coil 00257 is the least significant bit of the coil data.

4.5.8 Function 08 – Loopback Diagnostic

Request/Response (RTU Mode)

Message Field	Device Number	Function Code	Address of Coil -1		Number of Input Regs		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	0-99	08	0		See Note			

Note: Only the diagnostic code 00 is supported on the Modbus Interface module. The message received by the Gateway is transmitted unaltered.

4.5.9 Function 15 – Preset Multiple Coils (0xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Coil -1		Number of Coils		Byte Count	Coil Data	Error Check	
Bytes	1	1	2		2		2	See Note	2	
Rx Order			HOB	LOB	HOB	LOB			LOB	HOB
Range	1-99	15	0-4095		0-800		1-100	1-100		

Note: The Byte Count indicates the number of bytes of coil data that follow. The coil data is packed as eight coils per byte. The first byte contains the first eight coils requested, the second byte contains the next eight coils, etc. If the number of coils requested is not a multiple of eight, the remainder of the byte is filled with zeros. The lower number coil bit is the least significant bit of the byte.

Response (RTU Mode)

Message Field	Device Number	Function Code	Address of First Coil -1		Number of Coils		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	As Rcv	15	As Rcv		As Rcv			

4.5.10 Function 16 – Preset Multiple Holding Registers (4xxxx)

Request (RTU Mode)

Message Field	Device Number	Function Code	Address of First Reg −1		Number of Hold Regs		Byte Count	Reg Data	Error Check	
Bytes	1	1	2		2		1	See Note	2	
Rx Order			HOB	LOB	HOB	LOB			LOB	HOB
Range	0-99	16	0–1023		0–100					

Note: The Byte Count indicates the number of bytes of register data to follow that is equal to two times the number of registers specified. The first byte of register data received is the high order byte of the first register specified, followed by the low order byte. The remaining registers are received in high order/low order sequence.

Response (RTU Mode)

Message Field	Device Number	Function Code	Address of First Coil −1		Number of Coils		Error Check	
Bytes	1	1	2		2		2	
Rx Order			HOB	LOB	HOB	LOB	LOB	HOB
Range	As Rcv	16	As Rcv		As Rcv			

4.5.11 Exception Response

The Exception Response message is transmitted when the Modbus Interface module detects an error in the received message. The received function code is returned with the most significant bit set to signify an error.

Response (RTU Mode)

Message Field	Device Number	Function Code	Coil Status	Error Check	
Bytes	1	1	1	2	
Tx Order				LOB	HOB
Range	As Rcv	FC+80h	1-3		

Code	Error Detected
01	Illegal Modbus Interface module function code
02	Illegal starting coil/register
03	Illegal number of coils/registers, etc.

5.0 DIAGNOSTICS AND TROUBLESHOOTING

This section describes how to troubleshoot the Modbus Interface module. See Appendix E for a list of the error codes that can be displayed by the module. If the problem cannot be corrected using the procedures below, the unit is not user-serviceable.

DANGER

ONLY QUALIFIED ELECTRICAL PERSONNEL FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF THIS EQUIPMENT AND THE HAZARDS INVOLVED SHOULD INSTALL, ADJUST, OPERATE, AND/OR SERVICE THIS EQUIPMENT. READ AND UNDERSTAND THIS MANUAL IN ITS ENTIRETY BEFORE PROCEEDING. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.

5.1 The OK LED is Off

Problem: The green OK status LED on the Modbus Interface module faceplate is off. This LED should be on when the module has passed its internal diagnostics after power-up. If the green OK status LED is off, and no error code is displayed, a local watchdog failure has occurred. Try cycling power to the rack. If the OK status LED remains off, replace the module.

5.2 Invalid Device Number

Problem: Error code A appears on the Modbus Interface module's LED display at power-up. This error code indicates an invalid device number. Use the following procedure to clear the error code:

Step 1. Turn the keyswitch on the faceplate of the Power Supply to the Program mode.

Step 2. Set the correct device number on the Modbus Interface module's thumbwheel switches.

Step 3. Return the keyswitch to the Normal mode.

5.3 Transmission Link Failures

Problem: Error code "09" is returned by the GATEWAY_CMD_OK@ function. This error code indicates a Response Timeout error.

Verify that the cable connections to each device are secure. Check the cable connection at the module faceplate and at the remote device.

See sections 5.3.1 and 5.3.2 for more information on rack failure and remote device failure.

5.3.1 Rack Failure

If the Processor in the rack that contains the Modbus Interface module fails or issues a BOARD RESET command (clears all outputs in the rack), the entire image in the Modbus Interface module's dual port memory will be cleared. The device will remain off-line for at least 1 second to allow the application task to recognize that the device went off-line.

5.3.2 Remote Device Failure

When the Modbus Interface module operates in a slave mode where it only responds to incoming request messages, a failure in the remote device cannot be detected by the module itself. When the Modbus Interface module is initiating request messages, a link failure would be indicated by a Response Timeout error (see 5.3.)

5.4 Bus Error

Problem: Error code "31" or "51" through "58" appears on a Processor module's LED display. These errors indicate the system has a problem accessing a module in the rack though the backplane bus. A bus error may be caused by removal of an I/O module, an I/O module failure, or a rack backplane failure.

Use the following procedure to isolate a bus error:

Step 1. Verify that all modules are in the correct slot.

Verify that the slot number being referenced in the application tasks agrees with the slot number defined during configuration.

Step 2. Verify the device number is correct.

The thumbwheel switches used to set the device number may be changed while the module is online without having any effect on the system. On the next power up, if the thumbwheel switches define a device that is valid and unique on the network, the Processor will accept it as a valid device.

Step 3. Verify that the application software is correct.

Verify that the application software is not attempting to write to READ ONLY registers on the module.

Step 4. Verify that the hardware is working correctly.

> **DANGER**
>
> **THIS EQUIPMENT IS AT LINE VOLTAGE WHEN AC POWER IS CONNECTED. DISCONNECT AND LOCKOUT ALL UNGROUNDED CONDUCTORS OF THE AC POWER LINE. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN SEVERE BODILY INJURY OR LOSS OF LIFE.**

Make certain that power is off before removing any module from the rack. Swap out the Modbus Interface module, the Processor module(s), and the backplane, one at a time. After each swap, if the problem is not corrected, replace the original item before swapping out the next item.

Appendix A

Technical Specifications

Modbus Interface Module M/N 57C414

Ambient Conditions

- Storage Temperature   0°C to 60°C
- Operating Temperature   -40°C to 85°C
- Humidity   5% to 95%, non-condensing
- Altitude   3300 feet (1000 meters) without derating

Dimensions

- Height   11.75 inches (29.845 cm)
- Width   1.25 inches (3.175 cm)
- Depth   7.375 inches (18.733 cm)
- Weight   2 pounds (.9 kg)

Maximum Power Dissipation

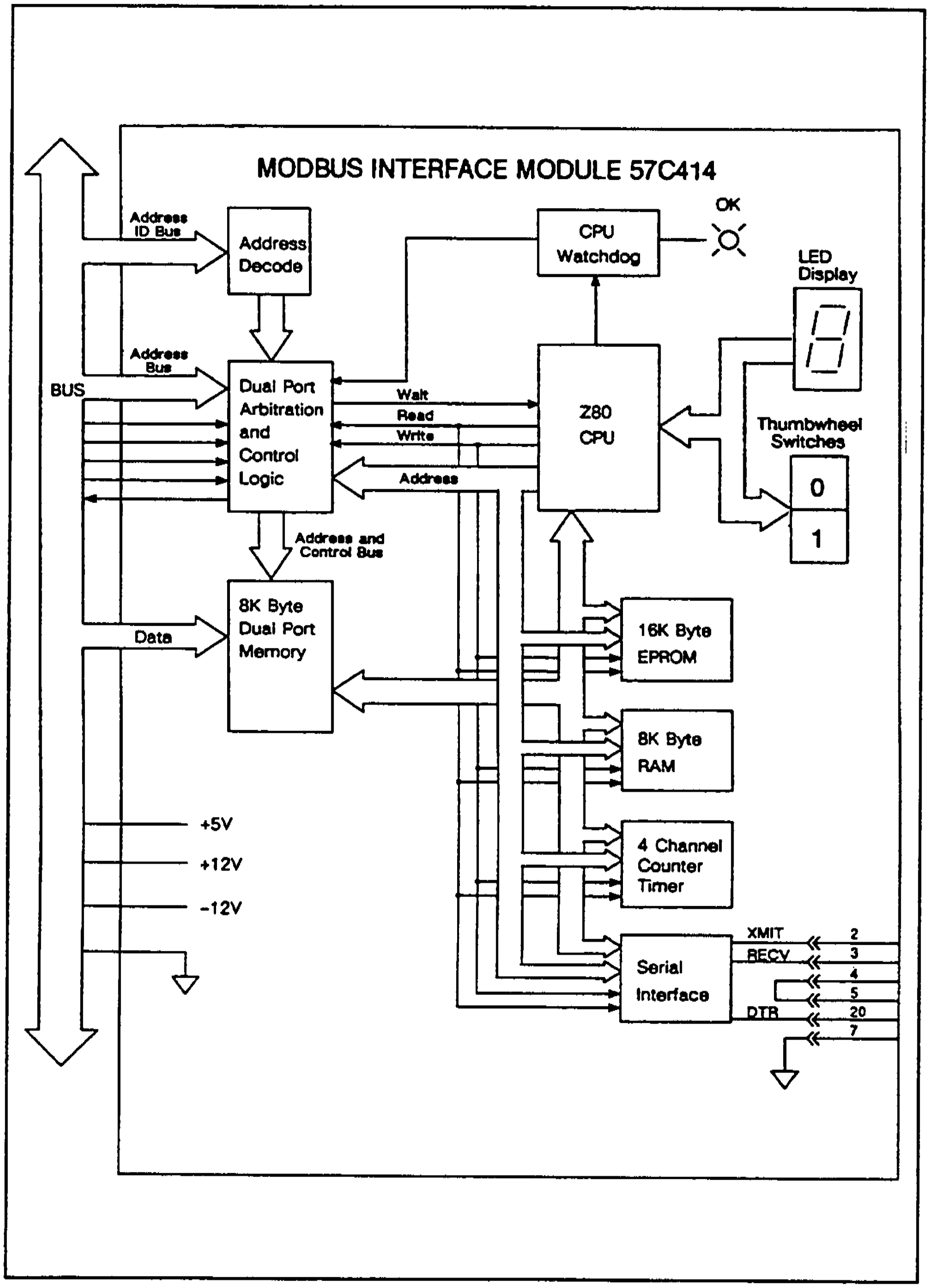
- 13 W

System Power Requirements

- +5 VDC   2.5 A
- +12 VDC   53 mA
- -12 VDC   7.5 mA

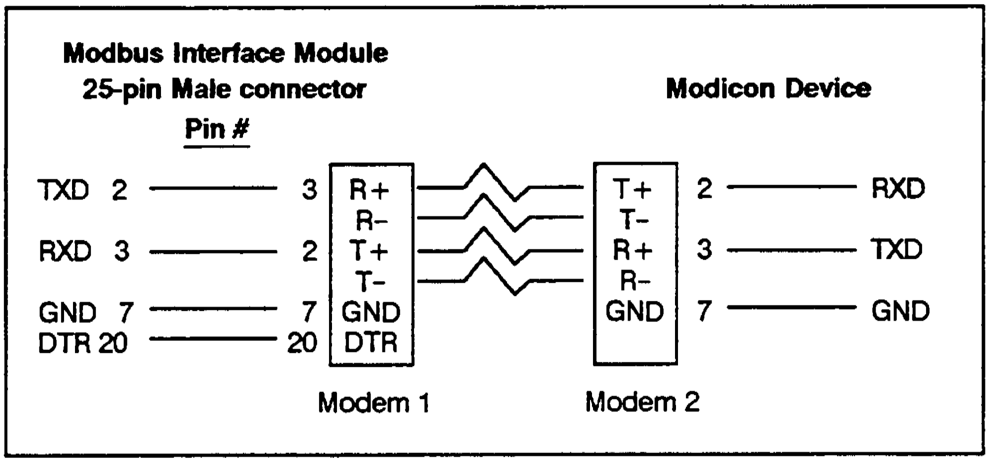
Appendix B

Module Block Diagram



Appendix C

RS-232 Port Pinout



Appendix D

Comparison of Modicon Equipment and Modbus Interface Module

Characteristic	Modicon Equipment	Modbus Interface Module
Transmission Mode	7-bit ASCII with LRC Error Check or 8-bit binary (RTU) with CRC Error Check	7-bit or 8-bit ASCII with LRC Error Check or 8-bit binary (RTU) with CRC Error Check
Parity Bits	1 even, 1 odd, or none	1 even, 1 odd or none
Stop Bits	1 or 2	1 or 2
Baud Rate	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, or 19200	1200, 2400, 4800, 9600, or 19200
Device Mode	Master or Slave	Master or Slave
Request Function Codes	1-8, 11-17	1-8, 15 and 16
Loop Diagnostic Codes	0-4, 10-18	0 only
Max Coils in READ STATUS	2000	2000
Max Registers in READ REGISTERS	125	125
Exception Response Codes	1-7	1-3
Broadcast Mode	Yes	Yes
Device Address	1-247	1-99

Appendix E

Modbus Interface Module Error Codes

0 CPU failed powerup diagnostic

1 EPROM failed powerup diagnostic

2 RAM failed powerup diagnostic

3 CTC failed powerup diagnostic

4 SIO port failed powerup diagnostic

5 DMA failed powerup diagnostic

6 Dual port memory failed powerup diagnostic

7 Memory management unit failed powerup diagnostic

9 Parallel I/O port failed powerup diagnostic

A Invalid device number. Hardware failure.

b Watchdog timer failed powerup diagnostic.

C Communication line failure. Displayed only if the link has not been configured by the AutoMax application program.

d System (backplane) watchdog failed; board is operational but will not transmit or receive any data until the watchdog is reset.

E Power failure. This code is normally present from the time that a low voltage is detected until power is completely lost.

If, at powerup of the rack, any diagnostic fault code remains displayed (0–9, or b), the Modbus Interface module must be replaced.

Appendix F

GATEWAY_CMD_OK@ Error Codes

Modbus Exception Errors

01 Illegal Function Code

02 Illegal Starting Register Number

03 Illegal Data

04 PC Aborted

05 ACK

06 PC Busy

07 NAK

08 Illegal Data in Response Message

09 Response timeout

GATEWAY_CMD_OK@ Errors

20 Dual port address error

21 Interface card not found or not accessible

22 No available Interface channel

23 Illegal register number

24 Illegal number of registers

25 Illegal command number

26 Illegal command number/Register set

27 Illegal register number/Number of registers

28 Illegal Device number

Appendix G

Monitoring Dual Port Registers 50–54

Selecting the Display option (bit 8 = 1) in the link configuration byte of register 20 enables the software to store the first 10 bytes of the incoming message in dual port memory. The dual port registers can then be displayed in hexadecimal using the I/O Monitor function in the AutoMax Programming Executive. The dual port registers will contain the binary representation of the incoming message, and should be interpreted using the RTU formats described in section 4.5. The data is stored in the dual port memory as follows:

Register	Contents	
50	byte 1	byte 2
51	byte 3	byte 4
52	byte 5	byte 6
53	byte 7	byte 8
54	byte 9	byte 10

Bytes 1 and 2 are always the Device Address and Function Code, respectively. The interpretation of the remaining bytes is dependent on the Function Code as well as whether the message is a request or a response. The most significant bit of the Function Code is set if the message is an Exception Response.

Appendix H

Calculating Register Numbers to Monitor

The Modbus registers in the dual port memory of the Modbus Interface module may be displayed by using the Monitor I/O function in the AutoMax Programming Executive. This appendix provides a method for converting the Modbus register number to the register number required by the Monitor I/O function.

Modbus registers 0xxxx and 1xxxx are single-bit registers. Therefore, each Monitor I/O register contains 16 Modbus registers. The lower number Modbus register is stored in the least-significant bit (bit 0) of the Monitor I/O register. Use the following formula to calculate the Monitor I/O register number for Modbus registers 0xxxx or 1xxxx:

$$I.R. = \frac{(xxxx - 1)}{16}$$

where:

xxxx = last 4 digits of Modbus register number
I = integer portion of the quotient
R = fractional portion of the quotient

For 0xxxx registers, the Monitor I/O register = I + 64.
For 1xxxx registers, the Monitor I/O register = I + 320.

The formula to calculate the corresponding bit number within the Monitor I/O register is:

Bit # = 0.R x 16

The conversion process is more straightforward for Modbus integer registers (3xxxx and 4xxxx) since both the Modbus registers and the Monitor I/O registers are 16 bits wide.

For 3xxxx registers, the Monitor I/O register = xxxx + 575.
For 4xxxx registers, the Monitor I/O register = xxxx + 1599.

Example 1 - Modbus register 03547

$$I.R. = \frac{3547 - 1}{16} = 221.625$$

Monitor I/O register = 221 + 64 = 285
Bit # = 0.625 x 16 = 10

Example 2 - Modbus register 11233

$$I.R. = \frac{1233 - 1}{16} = 77.0$$

Monitor I/O register = 77 + 320 = 397
Bit # = 0.0 x 16 = 0

Example 3 - Modbus register 30962

Monitor I/O register = 0962 + 575 = 1537

Example 4 – Modbus register 40251

Monitor I/O register = 0251 + 1599 = 1850

The following table depicts the range of register numbers as stored in the Modbus Interface module's dual port memory.

Monitor I/O Register	**Modbus Register**
64 to 319	00001-00017 to 04081-04096
320 to 575	10001-10017 to 14081-14096
576 to 1599	30001 to 31024
1600 to 2623	40001 to 41024

Appendix I

Calculating Message Transmission Times

This appendix will provide a means of estimating the time required for transmission of messages.

Transmission Time of Request or Response Message

The transmission time in milliseconds for a request or response message can be approximated from the following equation:

TIME (ms) = (C * B * 1000) / Baud Rate

where:

C = Number of characters in the message (Table I.1)
B = Number of bits per character (Table I.2)

The number of characters in a message is dependent on the message type (function code) and the number of registers being transmitted, and the mode of transmission (binary or ASCII). Refer to Table I.1.

Table I.1 - Number of Characters in a Message

Function Code	Binary Transmission Mode[1]		ASCII Transmission Mode[2]	
	Request	Response	Request	Response
1	11	8+(NC/8)	17	11+(NC/4)
2	11	8+(NC/8)	17	11+(NC/4)
3	11	8+(2xNR)	17	11+(4xNR)
4	11	8+(2xNR)	17	11+(4xNR)
5	11	11	17	17
6	11	11	17	17
8	11	11	17	17
15	12+(NC/8)	11	19+(NC/4)	17
16	12+(2xNR)	11	19+(4xNR)	17

NC = Number of Coils
NR = Number of Registers

1 The number of characters shown includes the 3-character timeout.
2 Assumes the message terminates with a carriage return or line feed.

The number of bits in each transmitted character is determined by the link configuration byte. The possible combinations are given in Table I.2.

Table I.2 - Number of Bits per Character

Transmission Mode	Parity Bits	Stop Bits	Data Bits	Total
Binary	No	1	8	10
Binary	Yes	1	8	11
Binary	Yes	2	8	12
ASCII	No	1	8	10
ASCII	Yes	1	8	11
ASCII	Yes	2	8	12
ASCII	No	1	7	9
ASCII	Yes	1	7	10
ASCII	Yes	1	7	11

Formulation of Request Message in the Master

When the Modbus Interface module operates as a Master, the time in milliseconds required for the formulation of the request message is as follows:

Command 01	1.3 ms
Command 02	(0.018 ms x number of coils) + 1.3 ms
Command 03	1.3 ms
Command 04	(0.026 ms x number of registers) + 1.3 ms

When the Modbus Interface module operates as a Slave, the time depends on the remote Master.

Message Processing Turn-Around Time

Turn-around time is the length of time it takes the Slave to formulate a response message to send to the Master. When the Modbus Interface module operates as a Slave, the turn-around time can be found in table I.3:

Table I.3 - Turn-Around Time

Function	Turn-Around Time
01 02 15	(0.018 ms x number of coils) + 0.7 ms
03 04 16	(0.04 ms x number of registers) + 0.7 ms
05 06 08	0.7 ms

(assumes register 24 = 0)

When the Modbus Interface module operates as a Master, the turnaround time is a function of the Slave device.

Response Message Processing

When the Modbus Interface module operates as a Master, the time (in milliseconds) required for processing a response message is as follows:

Command 01	(0.026 ms x number of coils) + 0.7 ms
Command 02	0.2 ms
Command 03	(0.038 ms x number of registers) + 0.2 ms
Command 04	0.2 ms

When the Modbus Interface module operates as a Slave, the time depends on the particular Master.

RELIANCE CONTROLS
DOCUMENTATION IMPROVEMENT FORM

Document Number: ____________________________

Page Number(s): ____________________________

Comments: *(Please give chapters, page numbers or specific paragraphs that the change will affect. Include markeups from the document or attach additional pages if necessary.)*

What will this improvement suggestion provide? ____________________________

Originator: ____________________ City: ____________________ State: ________ ZIP: ____________

Company: ____________________ Phone: (_____) ____________________

Address: ____ Date: ____________________

Technical Writing Internal Use: **Follow-Up Action:**

Writer: ____________________ Date: ____________________

Thank you for your comments . . .

RE 1857LC Printed in U.S.A.

RELIANCE ELECTRIC

For additional information
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124 USA
Tel: (800) 241-2886 or (440) 646-3599
http://www.reliance.com/automax

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe/Middle East/Africa: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 6356-9077, Fax: (65) 6356-9011

Publication J2-3017 - July 1992