SIEMENS

# SIMATIC TI505/TI500

# MODNIM

User Manual

Order Number: PPX:505-8122-1
Manual Assembly Number: 2586546-0083
Original Edition

01/21/92

# MANUAL PUBLICATION HISTORY

SIMATIC TI505/TI500 MODNIM User Manual
Order Manual Number: PPX:505-8122-1

*Refer to this history in all correspondence and/or discussion about this manual.*

| Event | Date | Description |
|---|---|---|
| Original Issue | 03/93 | Original Issue (2801280–0001), Software Release 3.0 |

## LIST OF EFFECTIVE PAGES

| Pages | Description | Pages | Description |
|---|---|---|---|
| Cover/Copyright | Original | | |
| History/Effective Pages | Original | | |
| iii — ix | Original | | |
| 1-1 — 1-20 | Original | | |
| 2-1 — 2-39 | Original | | |
| A-1 — A-4 | Original | | |
| B-1 — B-2 | Original | | |
| C-1 — C-3 | Original | | |
| D-1 — D-6 | Original | | |
| Index-1 — Index-3 | Original | | |
| Registration | Original | | |

# Contents

## Appendix A Network Cable Communications



## Appendix B Exception Codes



## Appendix C Specifications



## Appendix D Using the MODASST Program

## List of Figures

## List of Tables

# Preface

## Purpose of this Manual

This manual provides the following information on the Series 505™ and Series 500™ MODNIM (Modbus™ Network Interface Module), model numbers PPX:505–5184 and PPX:500–5184.

- Basic features of the MODNIM
- Installation and configuration
- Diagnostic self-testing procedures
- Operating instructions
- Modbus commands and error responses
- MODNIM specifications

## Related Manuals

The following publications contain related information which supplement this manual.

**Series 505 Manuals**

- *SIMATIC® TI505™ Programming Reference Manual*
- *SIMATIC TI505 TISOFT™ User Manual*
- *SIMATIC® TI525™/TI535™ Hardware and Installation Manual*
- *SIMATIC® TI545™ System Manual*

**Series 500 Manuals**

- *SIMATIC® TI520C™/TI530C™ Programmable Controller Manual*
- *Series 500 I/O Base User's Manual*
- *SIMATIC® TI560™/TI565™ Programming Manual*

The following Gould Modicon publications are also recommended.

- *Gould Modbus Protocol* (part no. PI-MBUS-300 Rev B)
- *Modbus System Planning* (part no. PI-MBUS-PLN)
- *Gould Modicon J474/J475 Interface* (part no. PI-J475-001 Rev B)

## Technical Assistance

If you need information that is not included in this manual, or if you have problems using the module, contact your Siemens Industrial Automation, Inc. distributor or sales office. If you need assistance in contacting your distributor or sales office in the United States, call 1–800–964-4114.

*Chapter 1*

# Installation and Set-up

## 1.1 Modbus Communications Overview

### Modbus Network Characteristics

The Modbus system is a Local Area Network (LAN) designed to work in an industrial environment. It is a master/slave network in which a number of slave nodes (e.g., Programmable Logic Controllers, or PLCs) are connected to a Master Node, which is a host computer.

### Modbus Protocol

The Modbus protocol determines how messages are passed between the host computer and the secondary nodes. Up to 247 slaves can reside on a single network. Each node in that network must have a unique address ranging from 1 to 247.

Because it is a master/slave network, each request is paired to one response. The exception to this is broadcast mode, when no response is returned. This mode is only valid for Modbus functions 5, 6, 8, 15, and 16.

---

**NOTE:** Only the master can initiate a message in a Modbus network.

---

The following two types of messages are used.

- The Request/Response type, where a single node is addressed.
- The Broadcast/No Response type, where all nodes are addressed but none respond.

The Modbus message consists of an “envelope” which contains different types of data. The envelope enables the data to be directed to the correct address on the network. It also contains information to determine if the contents were received correctly and instructions about what to do with the data.

### Transmission Modes

Two modes of data transmission are available for a given network; however, only one mode can be selected at a given time, and every node on the network must conform to that mode. Usually, the requirements of the host computer determine which mode should be selected. The two modes are the following.

- ASCII (American Standard Code for Information Interchange)
- RTU (Remote Terminal Unit)

The MODNIM can be set to operate in either mode.

Details of the ASCII and RTU modes of transmission are listed in Table 1-1.

**Table 1-1 Transmission Mode Characteristics**

| Characteristic | ASCII | RTU |
|---|---|---|
| *Code used:* | 7 bits hexadecimal (ASCII printable) | 8 bits binary |
| *Bits per character:* | | |
| Start | 1 | 1 |
| Data | 7 | 8 |
| Stop | 1 or 2 | 1 or 2 |
| *Parity* | Optional | Optional |
| *Checksum* | LRC | CRC |

**NOTE:** On multidrop networks, the MODNIM requires a commercial modem to connect to the Modbus network, as shown in Figure 1-1. The MODNIM connected to a Series 505 PLC is equivalent to the use of the Modicon J475 modem interface with a Modicon PLC.

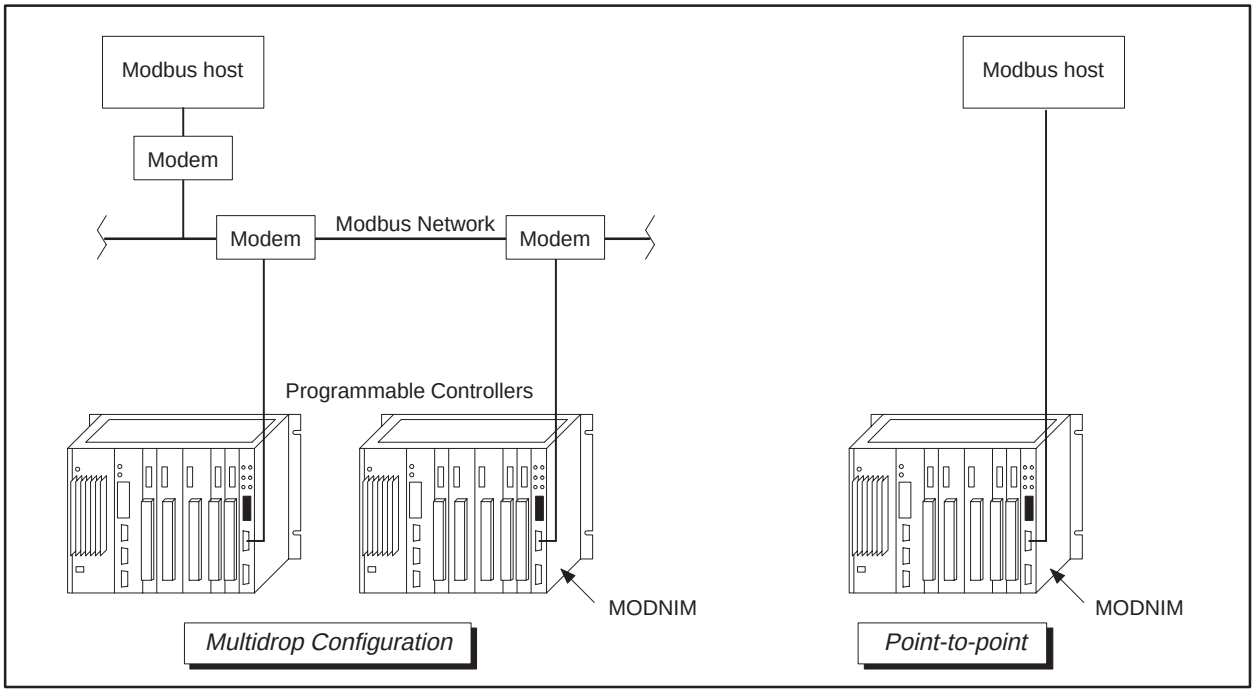



**Figure 1-1 Multidrop and Point-to-point Configurations**

## 1.2 Quick Reference Installation Steps

**General Requirements**

In order to install and use the MODNIM module correctly, you must meet the following requirements.

- Ensure that you have the appropriate MODNIM model.
- The communications configurations of all devices attached to the network must match; i.e., they must all have the same baud rate, stop bits, parity, handshaking, and transmission mode (ASCII or RTU).

**Procedures**

Figure 1-2 provides a quick reference list of the basic steps for installing the MODNIM.

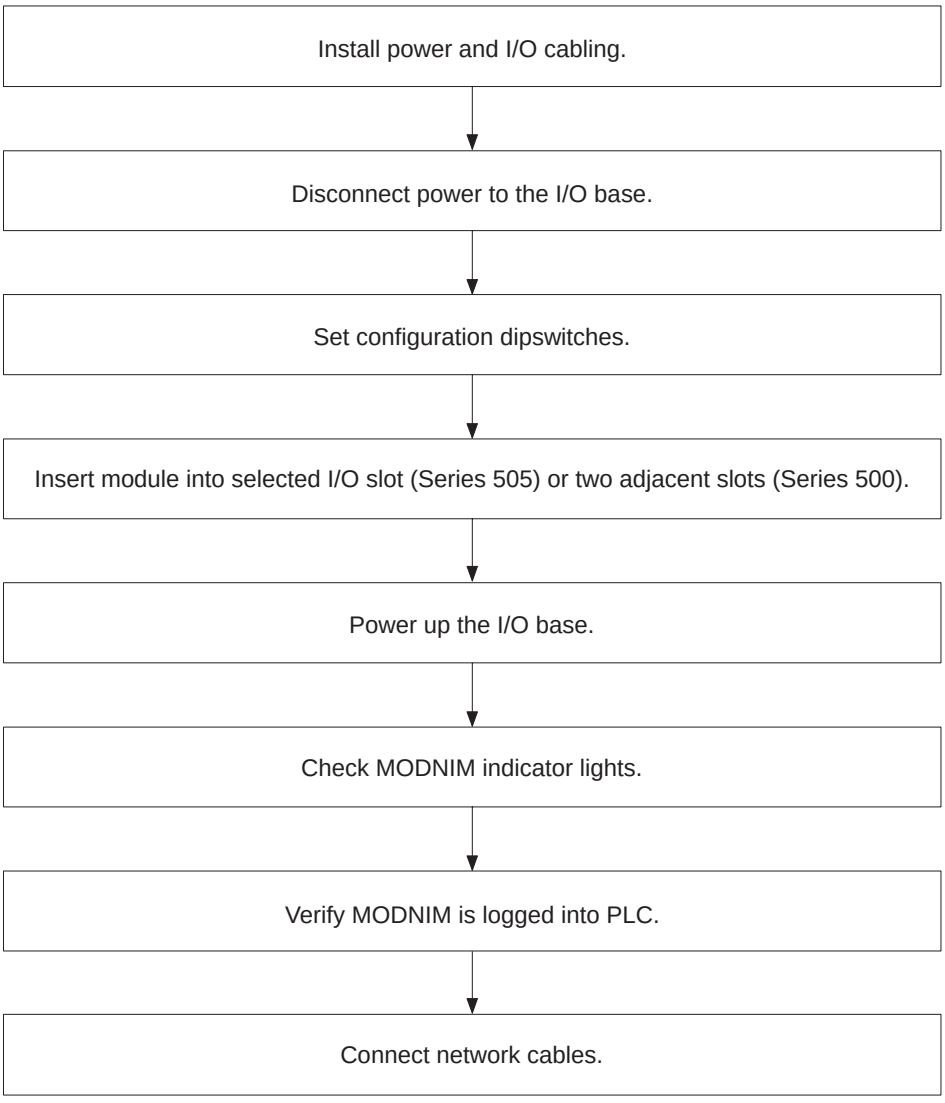



**Figure 1-2 Quick Reference Installation Procedures**

## Handling the Module

Many integrated circuit (IC) devices are susceptible to damage by the discharge of static electricity. Follow the suggestions listed below to reduce the probability of damage to these devices when you are handling this Network Interface module, the PLC, a base controller, or any of the I/O modules.

Both the module and the person handling the module should be at the same ground potential. Also, follow these guidelines.

- Transport the module in an anti-static container or antistatic material.
- Ensure that the work area has a conductive pad with a lead connecting it to a common ground.
- Ground yourself by making contact with the conductive pad and/or by wearing a grounded wrist strap.

## Inspecting the Module

Inspect the module for any visible damage before setting any selectable features. If damage is detected, contact your distributor or sales office for further instructions.

## Getting Started with Configuration and Installation

The following sections describe the procedures for configuring the module, installing the module in the base, connecting the communications cables to the MODNIM, and initializing the system for operation. Before installing the MODNIM, make sure that the PLC is installed and the programming device is connected to the system.

## 1.3 Setting the Dipswitches

### Dipswitch Functions

The MODNIM module has two blocks of dipswitches.

- The block of 8 dipswitches is used to select the MODNIM's address on the network.
- The block of 10 dipswitches is used to configure the network communications parameters.

---

**NOTE:** The configuration and address switch settings are read only once following a power-up or RESET. Be sure to reset the MODNIM following any change in dipswitch settings or PLC memory configuration. In addition, be aware that when the MODNIM is reset, no communication can occur from the host to the module for several seconds.

---

### Setting the Network Address

Each node on a Modbus network must have a unique address. The range of valid addresses is **1** to **247** (0000 0001 to 1111 0111). The address is set in binary numbers using the block of 8 dipswitches.

---

**NOTE:** Addresses 0 or 248 to 255 (0000 0000 or 1111 1000 to 1111 1111) are invalid addresses and, if selected, cause the module to go into Test mode.

---

Figure 1-3 shows examples of network addresses and their corresponding dipswitch settings. (With the Series 505 MODNIM, hold the module with the faceplate pointing upward, as shown in Figure 1-5. For the Series 500 MODNIM, refer to Figure 1-6 for dipswitch orientation.)

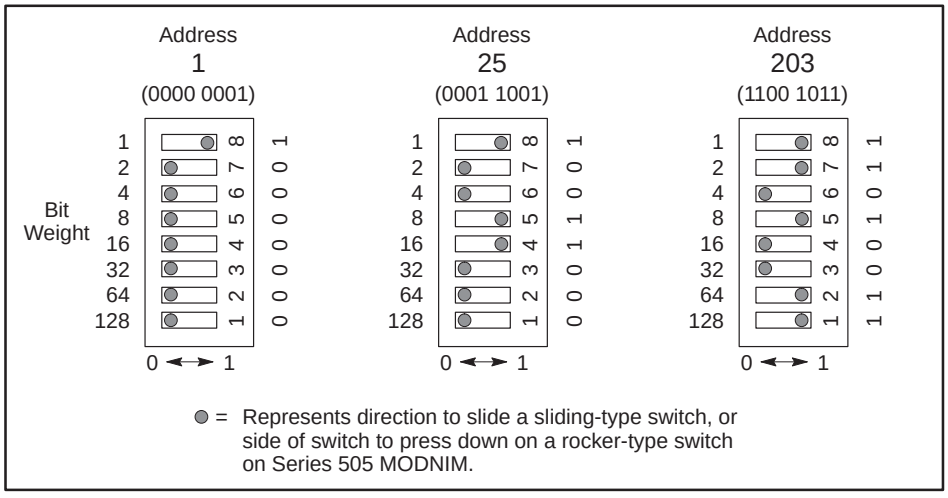



**Figure 1-3 Address Setting Examples**

### Selecting Network Configuration Parameters

The block of 10 dipswitches is used to configure the network communication parameters, which include data transmission rate, stopbit selection, parity, transmission mode, RTS/CTS handshaking, and output coil mapping. (**NOTE:** For all Series 500 MODNIM dipswitch settings, see Figure 1-6.)

### Data Transmission Rate

Switches 1 through 4 are used to set the data transmission rate. All devices on the network must be configured to communicate at the same data rate. Switch settings for each of the available data rates are shown in Figure 1-4.

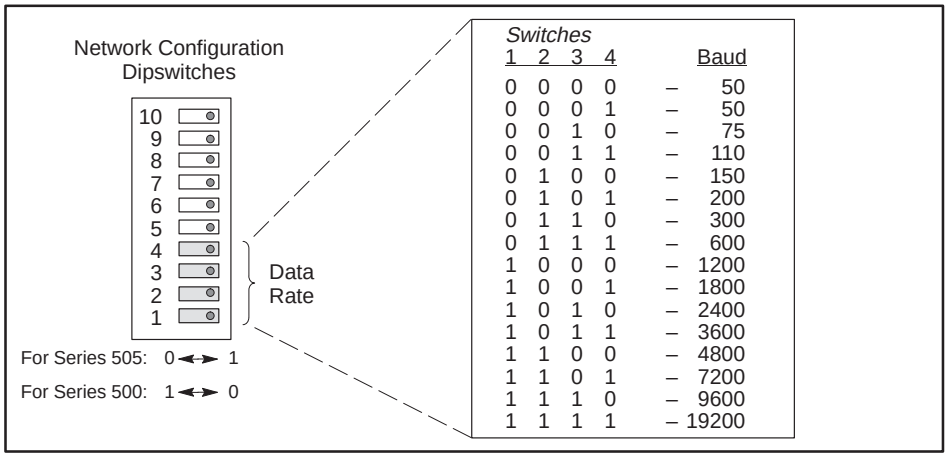



**Figure 1-4 Dipswitch Settings for Network Data Transmission Rates**

### Stopbit Selection

Switch 5 is used to select 1 or 2 stopbits. Set switch to the left for 1 stopbit, to the right for 2 stopbits (on Series 505; for Series 500, see Figure 1-6).

### Parity/No Parity Selection

Switch 6 is used to select Parity or No Parity when communicating over an RS-232-C data link. To select Parity, set switch to the left, then determine whether you need odd or even parity and set switch 7 accordingly.

### Odd/Even Parity Selection

If you selected Parity with switch 6, then you must also select either Odd or Even Parity using switch 7. This switch is active only if switch 6 is set to Parity. Set switch to the left for Even parity, to the right for Odd parity.

### ASCII/RTU Mode Selection

Switch 8 is used to select ASCII or RTU mode of transmission. Slide switch to the left for ASCII mode, to the right for RTU mode.

### RTS/CTS Handshaking

Switch 9 is used to enable or disable RTS/CTS handshaking. If you are using point-to-point connections without using modems, you can disable RTS/CTS handshaking by setting switch 9 to the left. Set switch 9 to the right to select RTS/CTS handshaking for use with modems.

### Y/C Coil Selection

Switch 10 is used to determine whether the MODNIM collects data from the PLC's Discrete Output (Y) memory or the Control Relay (C) memory. Set switch to the left for C coils, to the right for Y outputs.

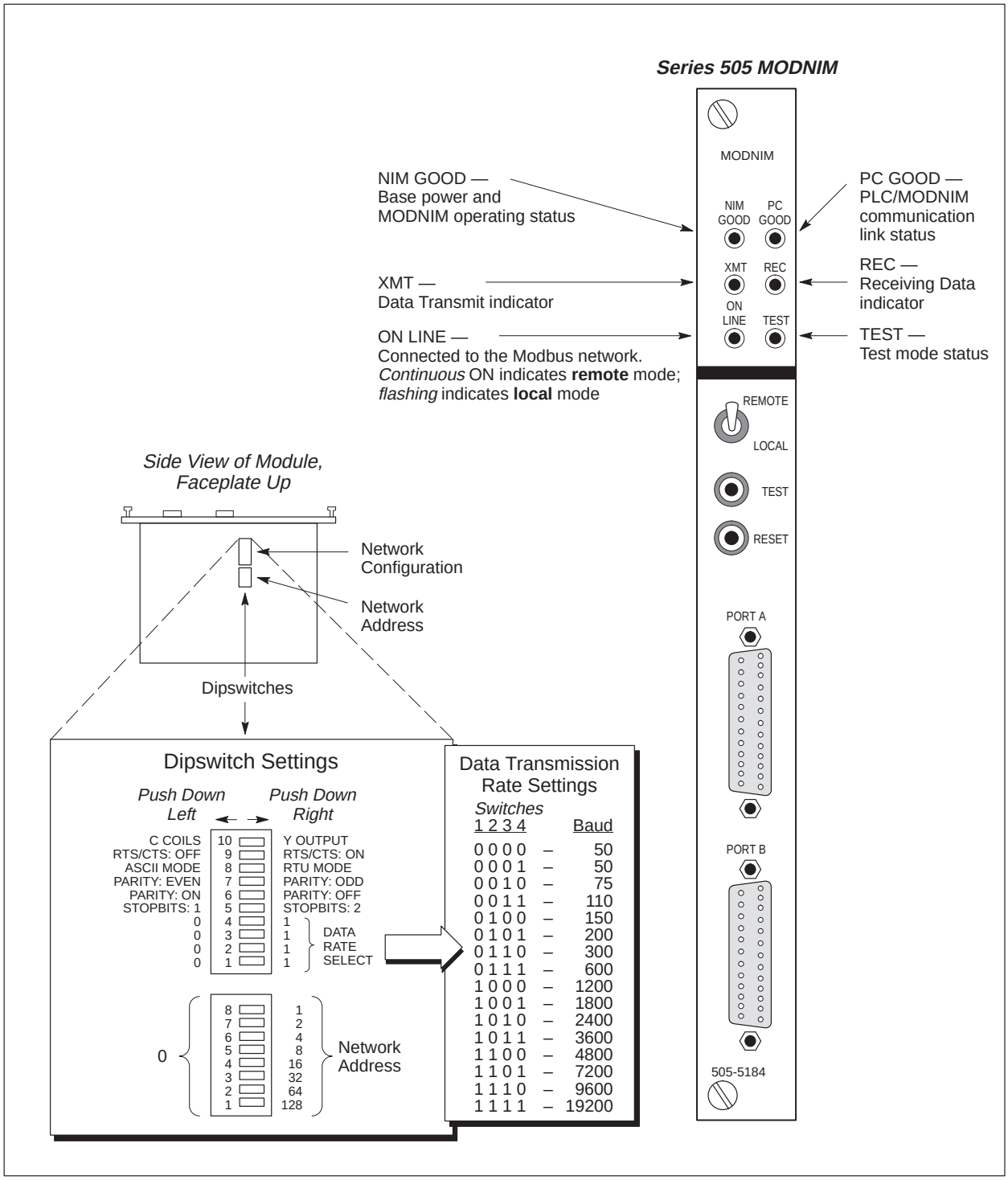



**Figure 1-5 Series 505 MODNIM Switches and LEDs**

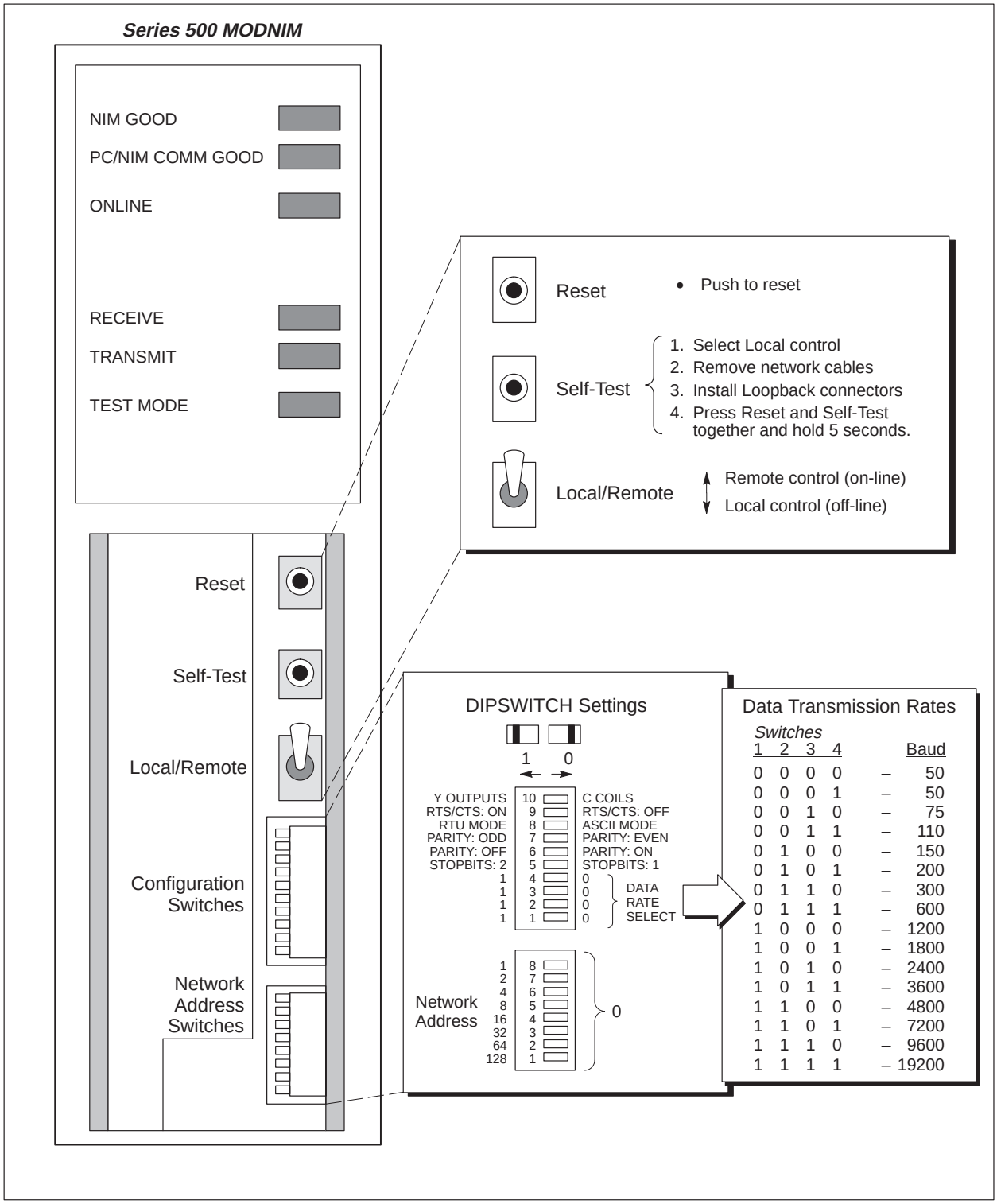



**Figure 1-6 Series 500 MODNIM Switches and LEDs**

## 1.4 Installing the MODNIM

**Installing the Series 505 MODNIM in the I/O Base**

To install the Series 505 MODNIM in the selected slot of the I/O base, follow these steps.

**⚠WARNING**

**To avoid the possibility of personal injury, damage to the module, altering the PLC memory, or causing a PLC fatal error, disconnect power to the base's power supply and to any modules installed in the base before inserting or removing the MODNIM.**

1. Disconnect power to the I/O base.
2. Set the dipswitches, as described in Section 1.3. (See also Figure 1-8 for example dipswitch configuration settings.)
3. Position the module so that the front bezel is facing you.
4. Hold the top and bottom of the bezel and carefully slide the module into the slot, pushing it all the way into the base connector. (See Figure 1-7.)
5. Ensure that the edge card connector is firmly seated in the I/O base connector.
6. Tighten the screws at the top and bottom of the faceplate with a flat-bladed screwdriver. (These screws also ground the module to the base.)

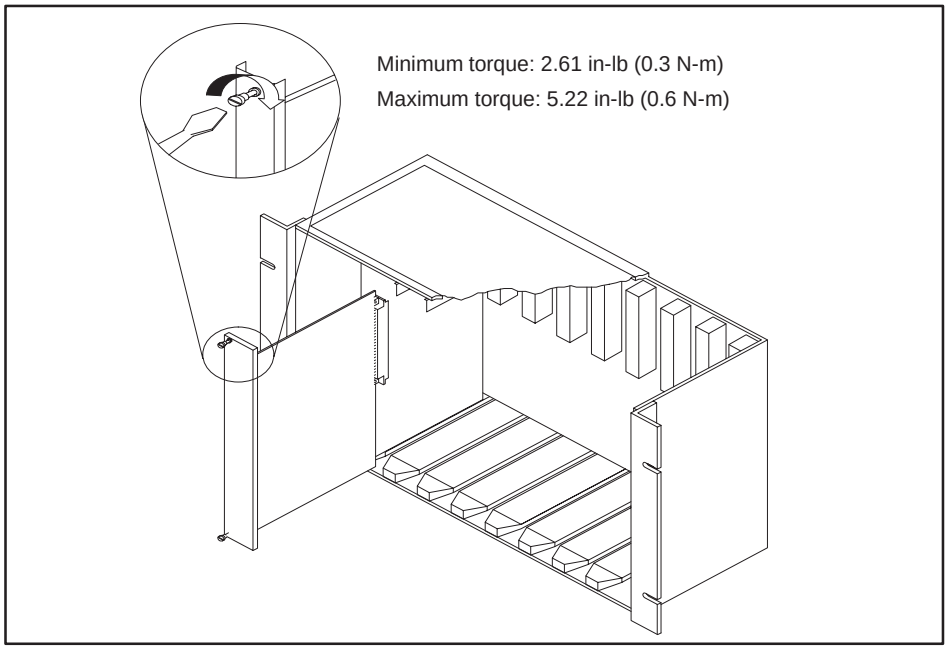



**Figure 1-7 Installing the Series 505 MODNIM in the I/O Base**

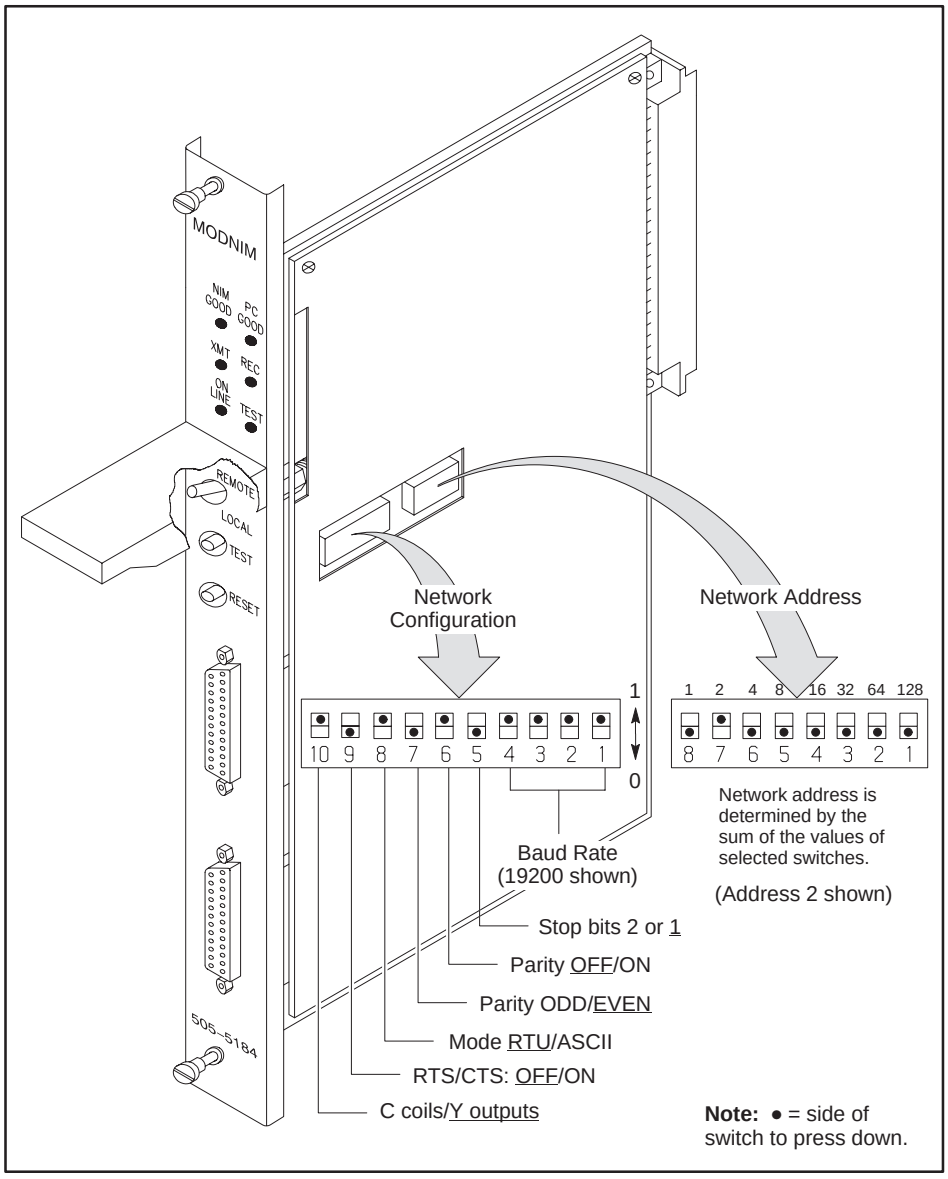



**Figure 1-8 Series 505 MODNIM Example Dipswitch Settings**

### Installing the Series 500 MODNIM in the I/O Base

You can key the Series 500 MODNIM module to prevent another I/O module from being inserted by mistake into the two slots reserved for the MODNIM module.

To do this, insert the three keys provided in the right slot of the two slots occupied by the module so that they fit into the notches in the edge card of the module. (See Figure 1-9.)

**⚠WARNING**

**To avoid the possibility of personal injury, damage to the module, altering the PLC memory, or causing a PLC fatal error, disconnect power to the base's power supply and to any modules installed in the base before inserting or removing the MODNIM or the I/O slot keys.**

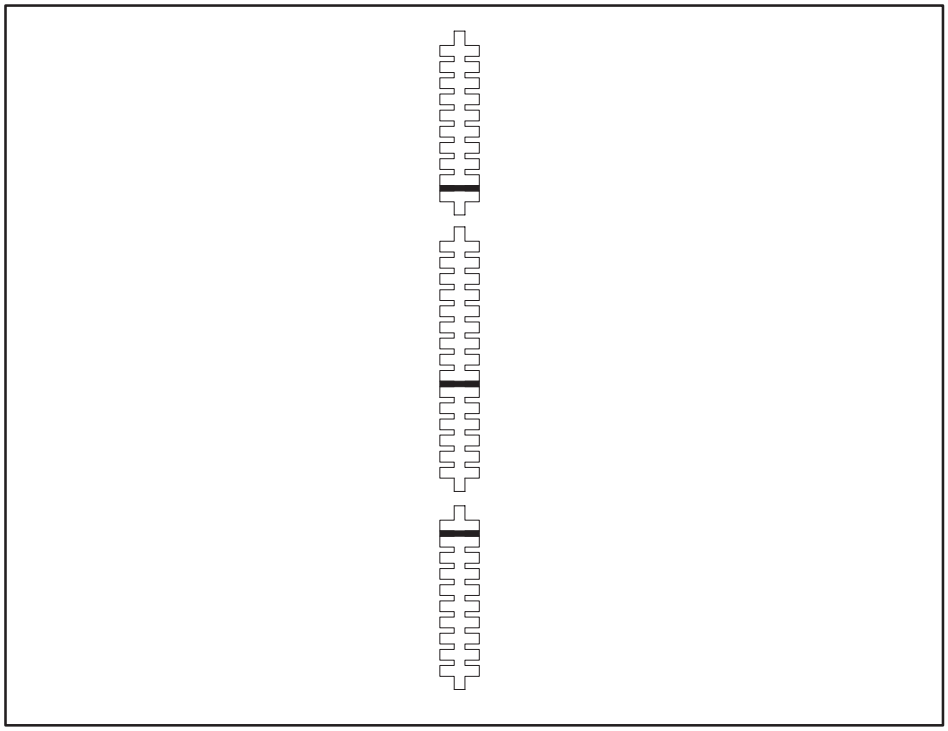


**Figure 1-9 Keying the I/O Base Slot for the Series 500 MODNIM**

Once the I/O slot keys are in place, you can insert the module as follows.

1. Set the dipswitches, described in Section 1.3.

2. Position the module so that the front bezel is facing you.

3. Hold the top and bottom of the bezel and carefully slide the module into the slot, pushing it all the way into the base connector. When the module is fully seated in the I/O base, locking tabs will hold the module in place. (See Figure 1-10.)

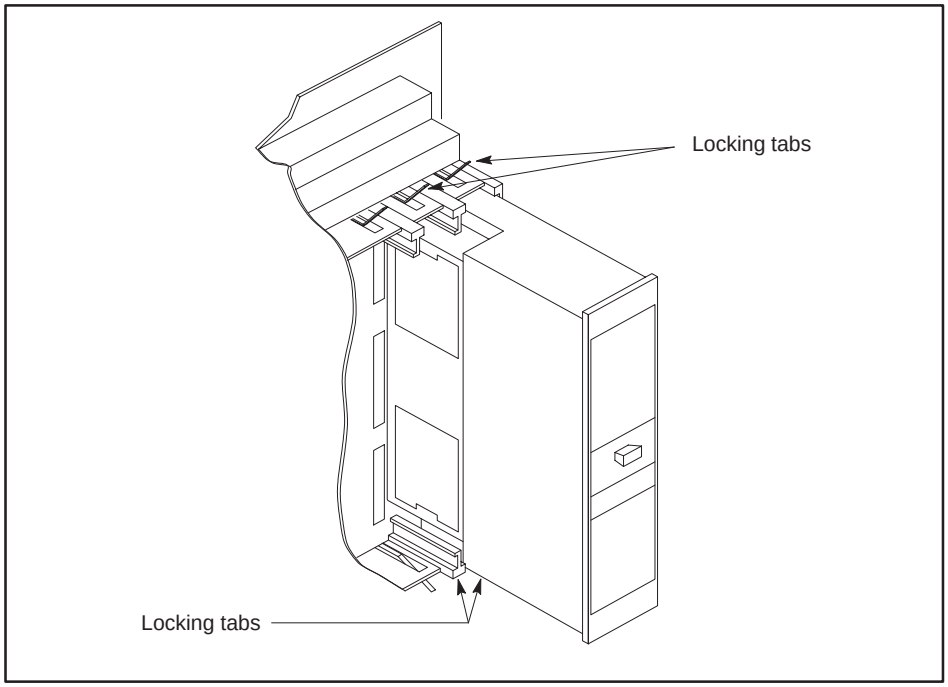



**Figure 1-10 Installing the Series 500 MODNIM in the I/O Base**

4. To remove the module, pull the tabs away from the module and slide it out of the base, being careful not to damage the edge card.

## 1.5 Switches and Indicator Lights

**Switches and Buttons**

The three switches located on the face of the module behind the access door are Reset, Self Test, and Local/Remote, and are described in the following paragraphs.

**Reset Button**

The Reset button is a momentary-contact switch which initializes the MODNIM and initiates the power-up self test. When you press Reset, all the indicators turn on for approximately 1 second. Then, all indicators except TEST go off for about 5 seconds. During this time, the MODNIM runs a series of diagnostic tests to verify that the hardware components of the module are operating properly. In addition, buffers and counters that service the Modbus Diagnostic Functions are initialized.

If the tests have been successfully completed, only the NIM GOOD and the PC GOOD (PC/NIM COMM GOOD) indicators will turn back on, while the TEST indicator turns off. If, however, the TEST indicator *remains* on, run the User Initiated Self-Test to determine the source of the error.

---

**NOTE:** Always press the Reset button after you change any of the configuration dipswitches or address selection dipswitches. In addition, be aware that when the MODNIM is reset, no communication can occur from the host to the module for several seconds.

---

**Test Button**

The Test button initiates a series of diagnostic tests when it is held down for 3 seconds after a Reset operation. Before running the diagnostic tests, disconnect all communications cables and install loopback connectors on the RS-232-C communications ports. These tests are described in more detail in Section 1.6.

**Local/Remote Switch**

When set to Remote, this two-position toggle switch enables the MODNIM to perform write operations to PLC memory. In the Local position, the MODNIM cannot write to the PLC.

In either position, the MODNIM can monitor PLC memory and mode of operation. After the MODNIM is set to Online state, local or remote status is indicated as follows.

- **Local** mode is indicated by a *flashing* ONLINE indicator.
- **Remote** mode is indicated by a *steady* ONLINE indicator.

When set to Local mode, only the Read functions 1, 2, 3, 4, 7, 11, 12, and 17 plus diagnostics function 8 can be performed. All other functions (write operations) will be rejected with Exception Code 01.

### Status Indicator Lights

The MODNIM has six indicator lights (or LEDs) located on the faceplate of the module (see Figure 1-11). These lights indicate the operational or diagnostic status of the module as described below.

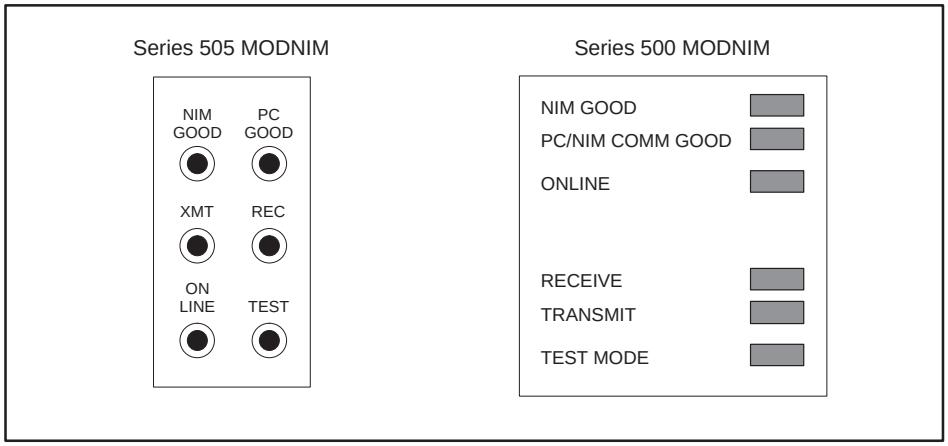



**Figure 1-11 Series 505 and 500 MODNIM Indicator Lights**

Table 1-2 shows how to interpret the status of the six indicator lights on the faceplate of the MODNIM during normal operation. Section 1.6 describes additional interpretations of the indicator lights during diagnostic testing.

**Table 1-2 Status of Indicator Lights**

| Indicator | Status | Description |
|---|---|---|
| NIM GOOD | On | All power-on, reset, or run-time diagnostic tests have been passed successfully; MODNIM operating correctly. |
| PC/NIM COMM GOOD (PC GOOD) | On | Communicating successfully with PLC |
| ONLINE | On | Connected to the network: Remote mode |
| | Flashing | Connected to the network: Local mode |
| RECEIVE (REC) | On | Receiving data over the network |
| TRANSMIT (XMT) | On | Transmitting data over the network |
| TEST MODE (TEST) | On | MODNIM in Test mode |
| | On | Failure detected after power-up diagnostics |
| | Flashing | Tests completed |

## 1.6 Diagnostic Tests

### Built-in Diagnostic Tests

The MODNIM has the following three levels of self tests available.

- Power-up Self Tests
- Run-time Self Tests
- User-initiated Self Tests

### Power-up Self Test

The MODNIM executes a Power-up Self Test in the following cases.

- Immediately after you apply +5 VDC power from the I/O base as part of initialization.
- Any time the Reset button is pressed.

The Power-up Self Test checks the processor and performs a test of on-board RAM and ROM. At the start of the test, all MODNIM indicators are turned on for approximately 1 second, then all except the TEST MODE indicator go off for about 5 seconds.

If the Power-up Self Test is successful, the NIM GOOD light turns on, and the MODNIM attempts to establish communications with the PLC. If this is successful, the PC GOOD (PC/NIM COMM GOOD) light turns on and the MODNIM is ready for normal network control (see Figure 1-12).

If the Power-up Self Test fails, the TEST MODE indicator remains on and the MODNIM will not respond to any communication attempts.

If the PLC does not respond, the PC GOOD (PC/NIM COMM GOOD) indicator remains off, and the MODNIM will issue an exception response with Error Code 04 (failure in associated device) to any host computer initiating an interactive request to this MODNIM. In addition, the TEST MODE indicator flashes at a 3-second interval, remaining on for 1 second.

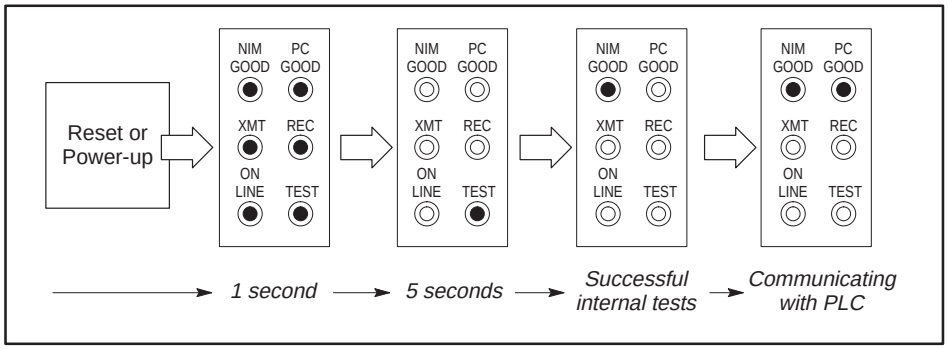



**Figure 1-12 LED Status during Power-Up Self Test**

## Run-time Self Tests

The MODNIM monitors itself continuously during normal operation as follows.

- The operating system continuously performs a ROM integrity test as a background process.
- The MODNIM periodically verifies that it is capable of communicating with the PLC.
- A Watchdog Timer circuit in the MODNIM guards against software lockup.

If any failures are detected in the ROM integrity test, the NIM GOOD indicator light turns off and the MODNIM is forced into the failed state. In this condition, the module goes into the Offline Mode and will not respond to any requests. (See Figure 1-13.)

When the MODNIM is in its normal operating mode but fails to communicate with the PLC, it reports this failure to the host computer with an exception response. Error code 04 (failure in associated device) is sent, and the PC GOOD (PC/NIM COMM GOOD) indicator goes out. (See Figure 1-13.) If communication is re-established, the MODNIM returns to normal mode and the PC GOOD indicator goes back on.

**NOTE:** The first time a valid request is received, the ON LINE indicator goes on and remains on as long as everything is operating properly.

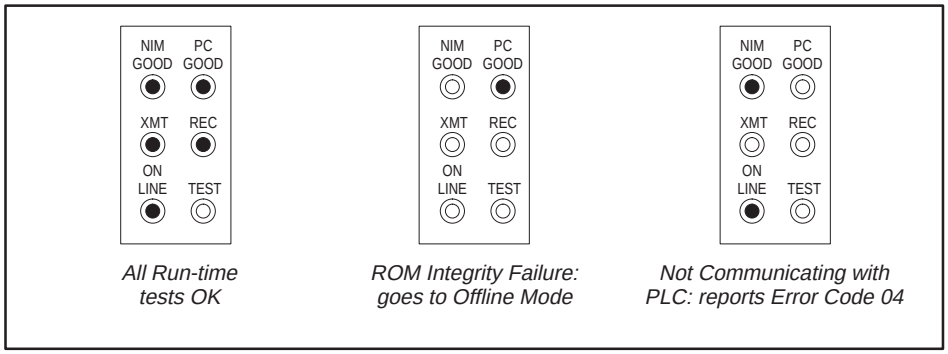



**Figure 1-13 LED Status during Normal Run-Time Operation**

The Watchdog Timer circuit provides an extra measure of protection against network lockup due to a failed MODNIM. This circuit will force a RESET if the operating software fails to execute normally.

### User-Initiated Self Test

The User-Initiated Self Test performs a complete test of the MODNIM hardware, including the communications ports. To run this test, follow the steps listed below.

**NOTE:** Since this test includes the communications ports, be sure to disconnect the MODNIM from the network.

1. Disconnect network cables from both network ports of the MODNIM.
2. Install the loopback connectors supplied with your module on ports A and B of the MODNIM.
3. Press the RESET button and, at the same time, press **and hold** the TEST button for about 5 seconds.

The TEST MODE indicator light turns on to signal the start of the test, while all other indicators go off. After about six seconds, all LED indicators will flash on and off for approximately two seconds. When the test has finished executing, the TEST MODE indicator flashes at a rate of 2 Hz.

Table 1-3 shows how to interpret the results of the user-initiated diagnostic test according to the status of the indicator lights on the MODNIM. The indicators show the **pass** or **fail** status of each of the tests conducted.

**Table 1-3 Indicator Status after User-Initiated Test**

| Indicator | Pass | Fail | Diagnostic Tests Included |
|---|---|---|---|
| NIM GOOD | On | Off | ROM, RAM, watchdog timer |
| PC/NIM COMM GOOD | On | Off | Communications with PLC |
| RECEIVE | On | Off | Network Port A: media interface; loopback connector |
| TRANSMIT | On | Off | Network Port B: media interface; loopback connector |
| TEST MODE | Flashing | Flashing | Tests completed |

If the User-Initiated Self Test is completed successfully, then you can reconnect the network cable to the MODNIM and press the Reset button. The MODNIM will re-initialize and be ready for normal operation.

## 1.7 Establishing Communications

### Verifying that the MODNIM is Logged into the PLC

After installing and configuring the MODNIM, you should verify that the module is properly logged into the PLC I/O map.

**NOTE:** The PLC I/O map is the first thing to check if the PC GOOD (PC/NIM COMM GOOD) indicator light is off.

Connect your programming device to the PLC to verify PLC-to-MODNIM module communication by following these steps.

1. Access the Configure I/O function menu, then execute Show.
2. Execute the Read Base function.
3. Check all the I/O points on the base that the MODNIM is installed in.

   Once you have entered the appropriate commands, the programming device displays a chart listing all slots on the base and the inputs and outputs associated with each slot. If a row on the chart is blank, then the corresponding base slot does not contain a module. Figure 1-14 shows a sample I/O definition chart with a MODNIM installed in Slot 1 of a four-slot base. (Refer to your TISOFT manual for detailed instructions.)

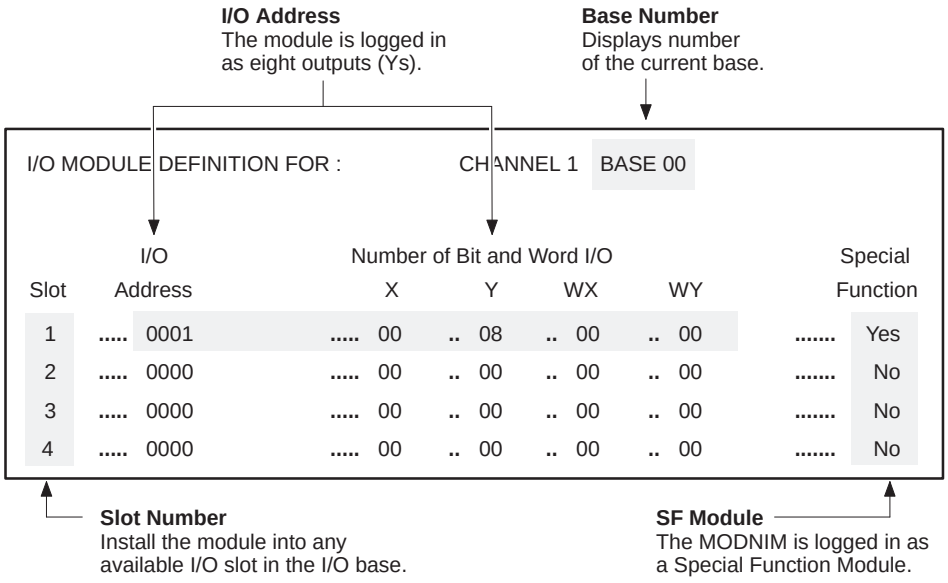



**Figure 1-14 Sample I/O Definition Chart**

4. Look at the chart for the number corresponding to the slot occupied by the MODNIM module. If an S or SF (Special Function) and word memory locations (or 8 Ys) appear on this line, the module is registered in the PLC memory. Assign a unique I/O address to the module and execute a Write PC function. You can now begin to communicate with the MODNIM.

   If the data on the line is incorrect, first check to see if the module is firmly seated in the I/O base and enter the command again. If you still cannot verify the module login, contact your local Siemens Industrial Automation, Inc. distributor.

### Connecting the Network Cables

Network cabling should already be in place before you begin installing the MODNIM.

The MODNIM has two communication ports. Only one port communicates at a time, but two are provided for cabling redundancy.

### Recommended Communication Cables

For high-noise environments, it is recommended that you use a standard Siemens communications cable with your MODNIM. (Refer to Table 1-4.) Either cable is compatible with the MODNIM; the requirements of your host computer may determine which cable you should select. These cables are available through your Siemens Industrial Automation, Inc. distributor.

**Table 1-4 Standard Communication Cables**

| Cable | Description | Cable P/N | Adapter at Module |
|---|---|---|---|
| 9-pin to 9-pin | Standard 9-pin null modem | 2601094–8001 | 9F–to–25M pin port adapter |
| 25-pin serial | Standard 25-pin null modem | VPU200–3605 | None required |

### Building a Cable

If you prefer to build your own 25-pin cable, refer to Appendix A for the pinouts. Ensure that the following requirements are met.

- Cable conductors — 26 AWG, tinned and stranded copper wire, with one uninsulated 26 AWG tinned copper drain wire
- Cable shield — AL foil or aluminum-polyester foil and a 65% minimum tinned copper braid shield
- Outer jacket — polyvinyl chloride (PVC) or equivalent material, with a UL rating of 30V, 60°C minimum
- Connectors — 25-pin male D-connector, with contact pins of copper alloy with gold flashing over nickel plate

*Chapter 2*

# Modbus Commands

## 2.1 Modbus Protocol Overview — ASCII Transmission Mode

**Modbus Protocol**

This chapter examines the Modbus protocol for both ASCII and RTU modes of transmission. The MODNIM-supported functions are then described in detail, with examples of requests and responses for each function code.

**The ASCII Transmission Frame**

Each frame in the ASCII transmission mode begins with a colon ( : ) and ends with CR LF (Carriage Return/Line Feed), shown in Figure 2-1.

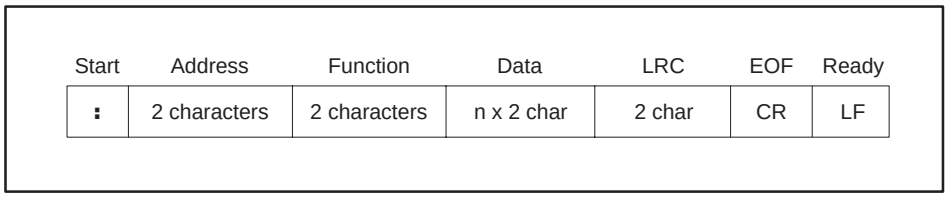



**Figure 2-1 ASCII Transmission Frame**

Each of the fields in the transmission frame is described in the paragraphs below.

**The Start Field**

The Start field is a colon (: = 3A hex), and marks the start of the message. This is the first character transmitted.

**The Address Field**

The Address field identifies the node to which this message is being sent and is a number in the range of 1 to 247. Each node on any one network must have a unique address. Only those nodes addressed will respond unless the broadcast address 0 is used. In that case, each slave will read and act on the message but not respond.

**The Function Field**

The Function field is a 2-character (16-bit) code that determines the action that a slave takes when receiving the message. The function codes supported by the MODNIM are summarized in Table 2-1 and described in more detail with examples in section 2.4.

**Table 2-1 Modbus Functions Supported**

| Code | Type | Description |
|---|---|---|
| 01 | Read Coil Status | Get current status of a group of coils. |
| 02 | Read Input Status | Get current status of a group of discrete inputs. |
| 03 | Read Holding Register | Get current values from holding registers. |
| 04 | Read Input Register | Get current values from input registers. |
| 05 | Force a Single Coil | Change the state of a logic coil to On or Off, forced On or forced Off, or unforced. |
| 06 | Write a Single Register | Write a value into a holding register. |
| 07 | Read Exception Status | Get the 8 internal status coil values. |
| 08 | Execute Diagnostics | Send diagnostic tests to a slave. |
| 11 | Get Communications Event Counter | Enable the success or failure of a query to be determined. |
| 12 | Get Communications Event Log | Get the communications log for Modbus network transactions. |
| 15 | Write Multiple Coils | Change a number of consecutive coils. |
| 16 | Write Multiple Registers | Write values into a series of consecutive holding registers. |
| 17 | Report Slave I.D. | Get the slave type and the condition of its run light. |

**Note:** The highlighted function codes (5, 6, 8, 15, and 16) are supported by Broadcast mode, for which no response is returned.

### The Data Field

The Data field contains the information needed by the specified node to perform the required operation, or the information returned by the node to the host computer. The data is in multiple bytes, with two ASCII character bytes representing a hexadecimal value (1 . . . 9 and A . . . F or a . . . f).

### The LRC Field

The Longitudinal Redundancy Checking (LRC) is the method employed by ASCII mode to ensure that the message transmitted by the host is the same as the one that arrives at the secondary (and vice versa). The characters are passed to a mathematical algorithm creating a checksum of 2 characters in length that can be duplicated at either end for comparison. The calculation of this checksum is described in the *Modbus Protocol Reference Manual*.

### The EOF Field

The End of File (EOF) marker, CR (carriage return) indicates the end of the data and is the point up to which the LRC is generated.

### The Ready Field

The Ready field, LF (line feed) indicates that the sender is now ready for any reply. This character can be changed using Function Code 08, with Diagnostic code 03. Unless there is a compelling reason to change this character, it should be left as the default value (LF).

## 2.2 Modbus Protocol Overview — RTU Transmission Mode

### The RTU Transmission Frame

The RTU mode determines the start and finish of a message based on a time period of silence (no transmission) equivalent to the time it would take to transmit 3.5 characters at the chosen baud rate. The RTU transmission frame is shown in Figure 2-2.

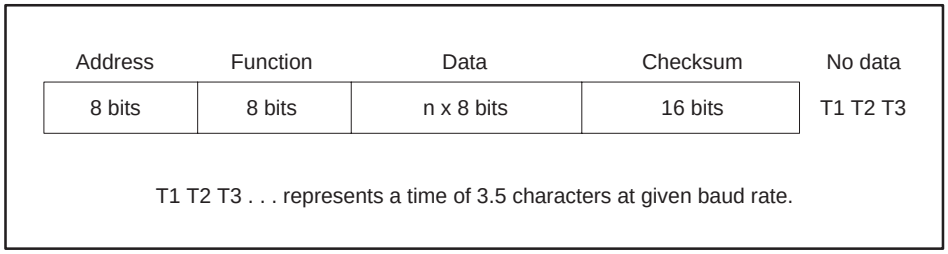



**Figure 2-2 RTU Transmission Frame**

### The RTU Frame Fields

The RTU mode of transmission is more efficient than the ASCII mode as it only uses 8 bits for each field (except for the checksum) and the end of the message is detected by timing. The body of the message (address, function code, and data field) is the same with both modes; therefore, only the message delineation and the checksum are discussed in the following paragraphs.

### Message Delineation

The start and finish of a message is determined by timing in RTU mode. Any time period longer than 3.5 character lengths marks the end of a transmission. The next character after that would then mark the start of the next message. This time period is dependent on the baud rate. Table 2-2 shows the effect of baud rate on the time delay period.

**Table 2-2 RTU Mode Timing**

| Baud Rate | 3.5 Character Time Delay (ms)* |
|---|---|
| 19200 | 2.2 |
| 9600 | 4.4 |
| 7200 | 5.9 |
| 4800 | 8.8 |
| 3600 | 11.7 |
| 2400 | 17.5 |
| 1800 | 23.3 |
| 1200 | 35.0 |
| 600 | 70.0 |
| 300 | 140.0 |
| 200 | 210.0 |
| 150 | 280.0 |
| 110 | 381.8 |
| 75 | 560.0 |
| 50 | 840.0 |
| | *Assuming longest delay with 2 stop bits & parity enabled |

### Checksum

The checksum needs 16 bits and uses the Cyclic Redundancy Checksum (CRC) method. The calculation of this checksum is described in the *Modbus Protocol Reference Manual*.

### Invalid Characters and Messages

The MODNIM ignores messages that contain invalid characters. Messages containing the following errors are also ignored.

- Incorrect checksum
- Parity error
- Framing errors
- Frame too long
- Incomplete transmission

## 2.3 Comparing Modbus Functions with TIWAY I Network Commands

### Modbus Functions vs. TIWAY Commands

This section outlines some of the differences between Modbus and TIWAY™ I network commands.

- Whereas Modbus uses *Functions Codes*, TIWAY I uses *Primitives*. Primitives differ from Function codes in one important way: primitives use TT-types to allow one primitive to address different types of data. With Modbus a different function code is required for each type of data.
- With SIMATIC® TI® PLCs, the first address for each type of memory is always 1. By comparison, Modbus PLCs allow 0 as the first legal address.
- If each system were to return one byte (8 bits) representing eight discrete coils, they would each be positioned as shown in Figure 2-3.

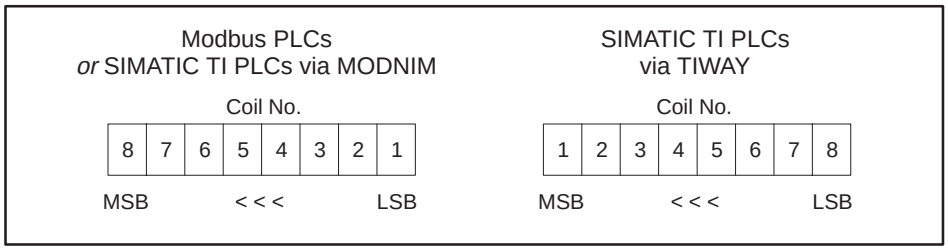



**Figure 2-3 Bit Orientation**

- Terminology is another area in which differences occur. Some of the differences in terms are given in Table 2-3. (Refer also to Appendix C, Table C-3 for a list of the TIWAY primitive equivalents to the Modbus function codes.)

**Table 2-3 Terminology Differences**

| Modbus-based PLCs | SIMATIC TI PLCs |
|---|---|
| Coil | Discrete output (Y) or Control Relay (C) |
| Input Register | Word Image Register (WX/WY) |
| Holding Register | Variable Memory (V-memory) |
| Slave | Secondary Node |
| P/C (Programmable Controller) | PLC |
| Function code | Primitive |
| Modbus | TIWAY I |
| Disabled (coil) | Forced |
| Enabled (coil) | Unforced |

### Addressing

The MODNIM uses the absolute position of the data for the address. That is, if you want to address the 3066th coil, then that is the address you pass with the command (as hexadecimal value BFA). The maximum address is 65535 (FFFF hex). All current PLCs are within this limit.

For example, if you want to access 1000 coils starting at the 703rd coil, the Function Code 01 would be as follows:

: 01 01 02 BF 03 E8 52 CR LF *[ASCII mode]*

01 01 02 BF 03 E8 0C E8 *[RTU mode]*

### Address Limits

Modbus imposes a buffer size limit of 256 bytes. Because of this restriction, MODNIM requests have been limited to the following values.

| Function Code | Quantity |
|---|---|
| 01 | 2000 coils |
| 02 | 2000 discrete inputs |
| 03 | 125 output registers |
| 04 | 125 input registers |
| 15 | 800 coils |
| 16 | 100 registers |

In the process of mapping Modbus commands against those used by SIMATIC TI PLCs, when an appropriate command was absent, it has become necessary to make multiple passes with a single command. In practical terms, this means that when the maximum amount of data for a particular command is requested, the MODNIM must wait for several PLC cycles before sufficient data can be collected. This latent time increases for addresses greater than 1024 (extended addressing).

The command checking will reject requests for data greater than the limits shown above. What the checking does *not* do is check the request against the available PLC memory. When a request goes beyond the memory of a particular PLC, the command will be rejected by the PLC and the MODNIM will notify the host computer with Exception Code 02.

## 2.4 Modbus Function Descriptions

### Introduction

In the descriptions that follow, each function starts on a separate page and includes the TIWAY primitive sent to the secondary in square brackets. For each function, an example of request and response is given.

**NOTE:** As Mode-specific data is not given, each of these examples omits the header ( : ) and tail (CR/LF) of ASCII mode.

### Code 01 — Read Coil Status

Code 01 enables the user to read the On/Off values of logic coils. The data passed with this command is the start address and the number of coils to be read. Addressing is sequential up to the maximum memory size for a particular PLC. The maximum number of coils that can be addressed with one command is 2000. If more are requested, the whole command will be rejected with an Exception code 03.

**⚠ CAUTION**

**SIMATIC TI PLCs number memory locations starting at address 1. Modbus PLCs number areas of memory starting from address 0. Failure to alter host computer application programs may result in the wrong bits being read.**

**Example:**

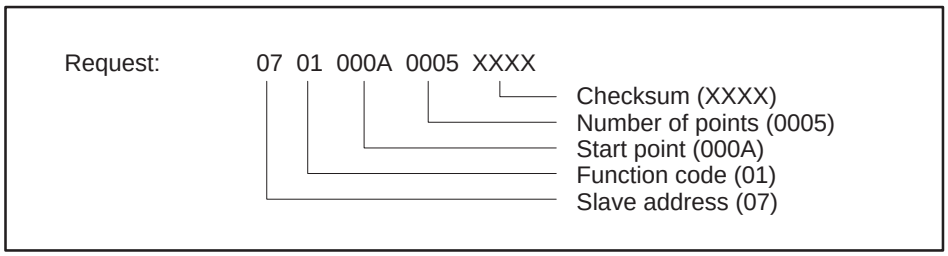



**Figure 2-4 Read Coil Status Example — Request**

This request is asking the MODNIM to return the data from 5 coils (On/Off) starting at the 10th coil.

[This is equivalent to sending a TIWAY Primitive 20 request using TT-type 7 (Packed Discrete Outputs or Ys) or TT-type 8 (Packed Cs), depending on the position of Switch 10 of the network dipswitch bank in the MODNIM.]

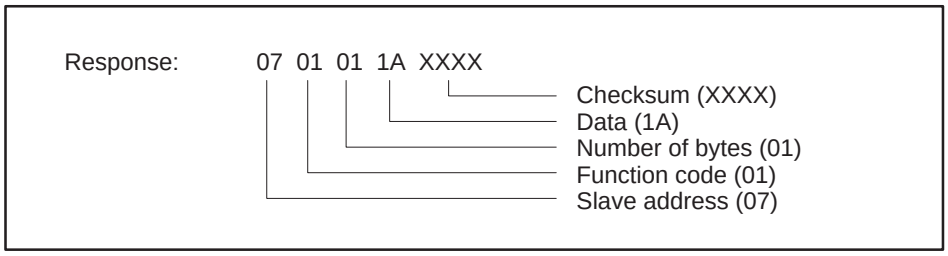



**Figure 2-5 Read Coil Status Example — Response**

The response returns in the data field a single byte (1A) which contains the 1's or 0's for the 5 coils, packed out with zeros for the three high bits.

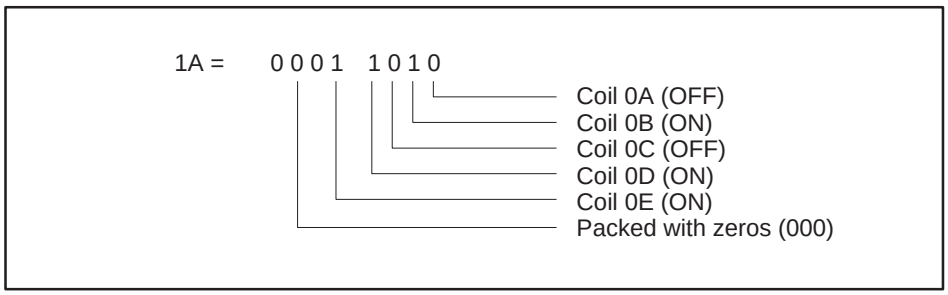



**Figure 2-6 Read Coil Status Example — Data Field**

### Code 02 — Read Input Status

Code 02 allows the application to read a series of discrete inputs. The command includes the start address and the number of points to be read.

Addressing is sequential up to the maximum memory size for a particular PLC. The maximum number of inputs that can be addressed with one command is 2000. If more are requested, the whole command will be rejected with an Exception code 03.

**Example:**

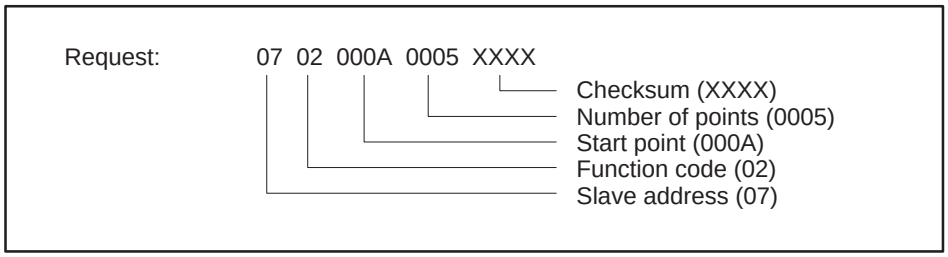



**Figure 2-7 Read Input Status Example — Request**

This request is for the data from 5 discrete inputs (On/Off) starting at the 10th input.

[This command is equivalent to sending a TIWAY I Primitive 20 request using TT-type 6 (Packed Discrete Inputs, or Packed Xs)].

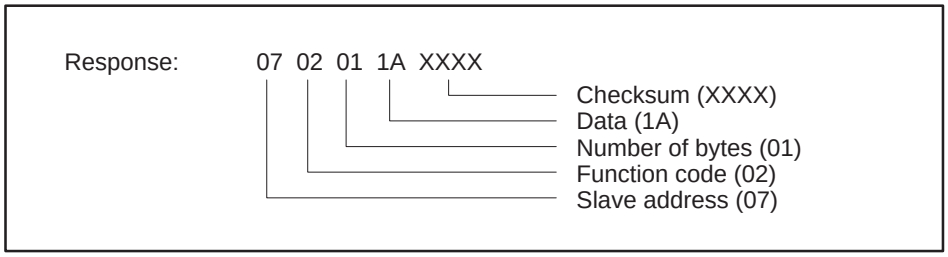



**Figure 2-8 Read Input Status Example — Response**

The response returns in the data field a single byte (1A) containing the 1's or 0's for the 5 discrete inputs, packed out with zeros for the three high bits.

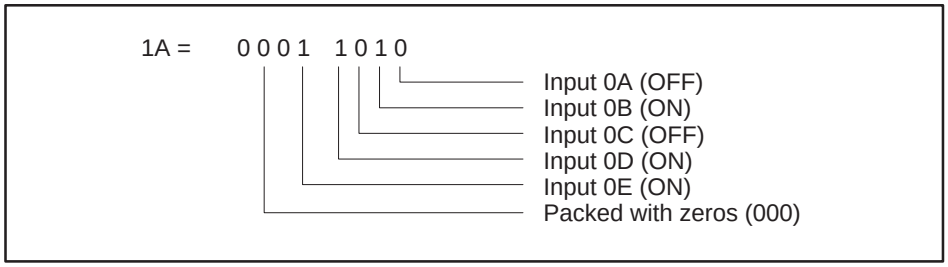



**Figure 2-9 Read Input Status Example — Data Field**

### Code 03 — Read Output Registers

Code 03 allows you to read the contents of holding registers in the attached secondary device. The data field of this command includes the start address of the registers and the number of registers to be read.

Addressing is sequential up to the maximum memory size for a particular PLC. The maximum number of registers that can be addressed with one command is 125. If more are requested, the whole command will be rejected with an Exception code 03.

[This command is equivalent to issuing a TIWAY I Primitive 20 request with a TT-type 01 (Variable Memory)].

**Example:**

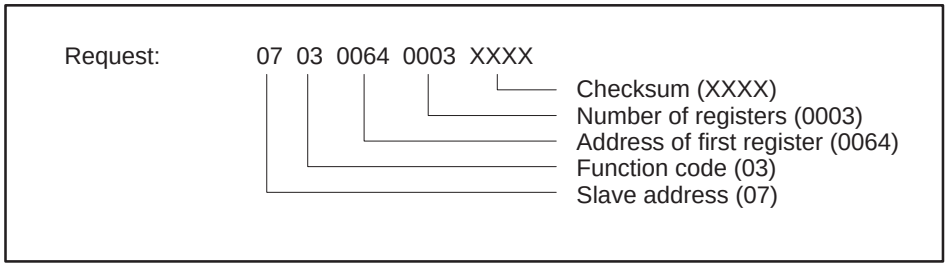



**Figure 2-10 Read Output Register Example — Request**

This request is for secondary address 7 to read the contents of registers 0064, 0065, and 0066 (100, 101, and 102 decimal).

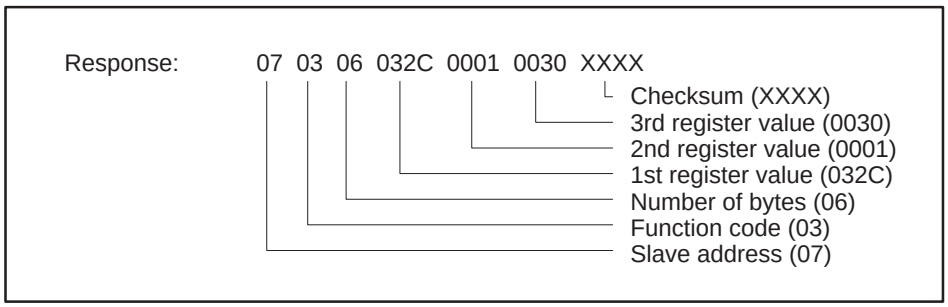



**Figure 2-11 Read Output Register Example — Response**

The response has returned the number of bytes of data (06) and the values for the requested registers; these are 032C, 0001, and 0030 respectively (812, 1, and 48 decimal).

### Code 04 — Read Input Registers

Code 04 allows you to read the contents of input registers in the attached PLC. These registers hold the values returned by the I/O devices. The data field of this command includes the start address of the registers and the number of registers to be read.

Addressing is sequential up to the maximum memory size for a particular PLC. The maximum number of registers that can be addressed with one command is 125. If more are requested, the whole command will be rejected with an Exception code 03.

[This command is equivalent to issuing a TIWAY I Primitive 20 request with a TT-type 09 (Word input WX)].

**Example:**

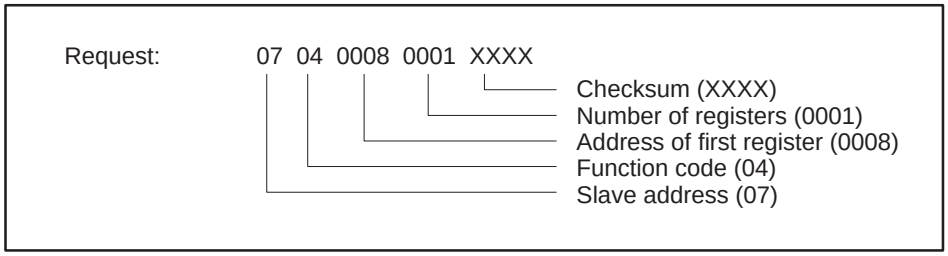



**Figure 2-12 Read Input Register Example — Request**

This request is for secondary address 7 to read and return the value stored in input register 0008.

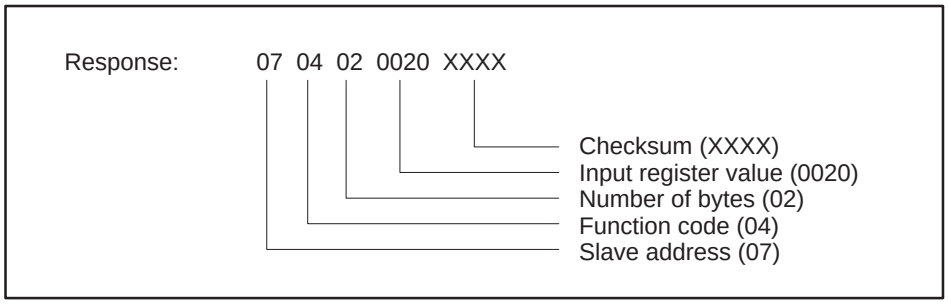



**Figure 2-13 Read Input Register Example — Response**

The response has returned the number of bytes of data (02) and the value 0020 (32 decimal) from the requested input register (0008).

### Code 05 — Write a Single Coil

Code 05 allows a designated coil (discrete output Y or C) to be written. Depending on the code in the data field, it can be changed to On (1) or Off (0), forced On or Off, or unforced. The two-byte codes are the following.

FF00 (hex) ..... ON
0000 (hex) ..... OFF
1100 (hex) ..... forced ON
2200 (hex) ..... forced OFF
3300 (hex) ..... unforced

**NOTE:** All other values are illegal and will result in error response 03.

Where the Broadcast address (00) is used, the MODNIM will send the change coil command to all attached secondaries.

[This command is comparable to Primitive 30, using TT-type 4 (Y coils) or type 5 (Cs), depending on the position of configuration Switch 10.]

**⚠WARNING**

**Because of differences in memory mapping between Modbus PLCs and SIMATIC TI PLCs, you should be quite certain of the effects of the broadcast command before issuing it.**

**Example:**

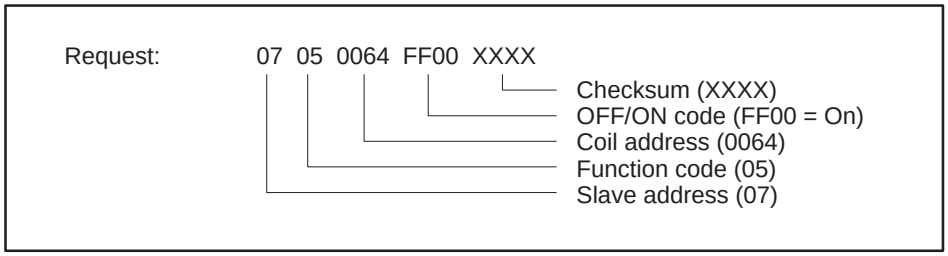



**Figure 2-14 Write a Single Coil Example — Request**

This request to secondary 7 is asking for coil 0064 (100 decimal) to be changed to the ON state.

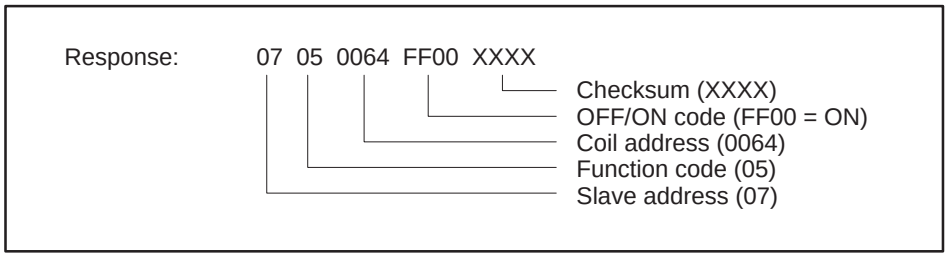



**Figure 2-15 Read Input Register Example — Response**

The response to Function Code 05 is to return the request as received. In the case of a broadcast request, there is no response.

### Code 06 — Write a Single Register

Code 06 allows the contents of a holding register to be changed. Addressing is sequential up to the maximum memory size for a particular PLC. This function is limited to integer values contained in V-memory. The MODNIM will also direct any broadcast address (00) to all attached secondaries.

[This command is comparable to a TIWAY I request using Primitive 30 and TT-type 01 (Variable memory)].

**⚠WARNING**

**Because of differences in memory mapping between Modbus PLCs and SIMATIC TI PLCs, you should be quite certain of the effects of the broadcast command before issuing it.**

**Example:**

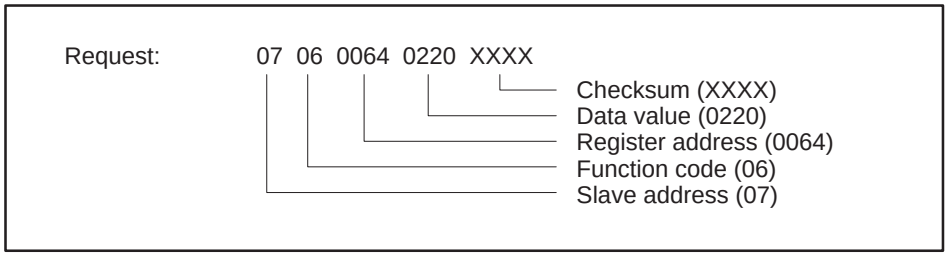



**Figure 2-16 Write a Single Register Example — Request**

This request to secondary 7 is asking for the value of holding register 0064 (100 decimal) to be changed to 0220 (544 decimal).

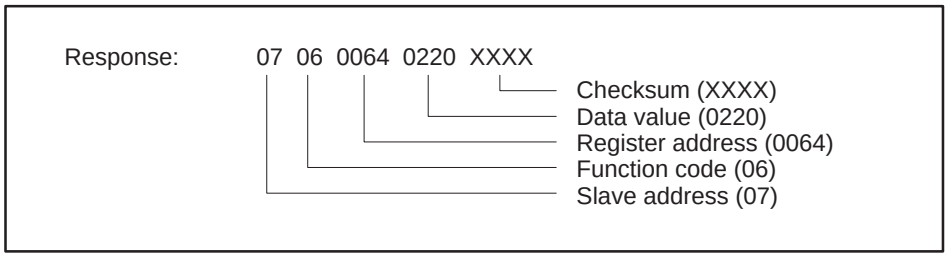



**Figure 2-17 Write a Single Register Example — Response**

The response to Function Code 06 is to return the request as received. In the case of a broadcast request, there is no response.

### Code 07 — Read Exception Status

Code 07 instructs the MODNIM to read 8 predefined coils within a secondary.

This implementation will read the first 8 coils (first 8 Ys or first 8 Cs, depending on the position of the dipswitch) in the discrete image register in a secondary and pack them into one data byte.

---

**NOTE:** The programmer of a non-Modbus PLC must make certain that the application program running within the PLC will place meaningful data in these locations. This data can reflect status information.

---

[This command is equivalent to TIWAY Primitive 20 using TT types 04, 07 (Y unpacked, packed) or TT types 05, 08 (C unpacked, packed), from locations 1 – 8, depending on the position of configuration Switch 10.]

**Example:**


**Figure 2-18 Read Exception Status Example — Request**

This request to secondary 17 (decimal) is asking for exception status data.

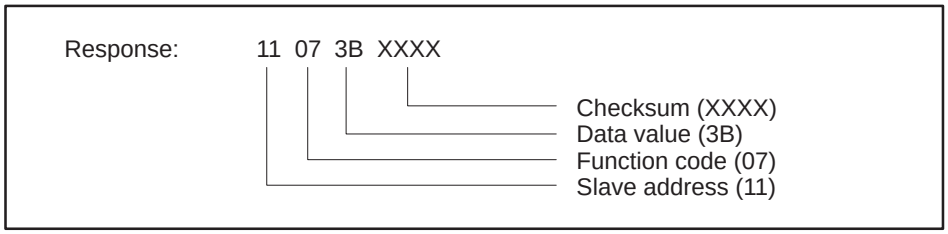



**Figure 2-19 Read Exception Status Example — Response**

The response shows that 3B (hex) has been returned. If the individual bits are examined, they indicate the status of each bit.

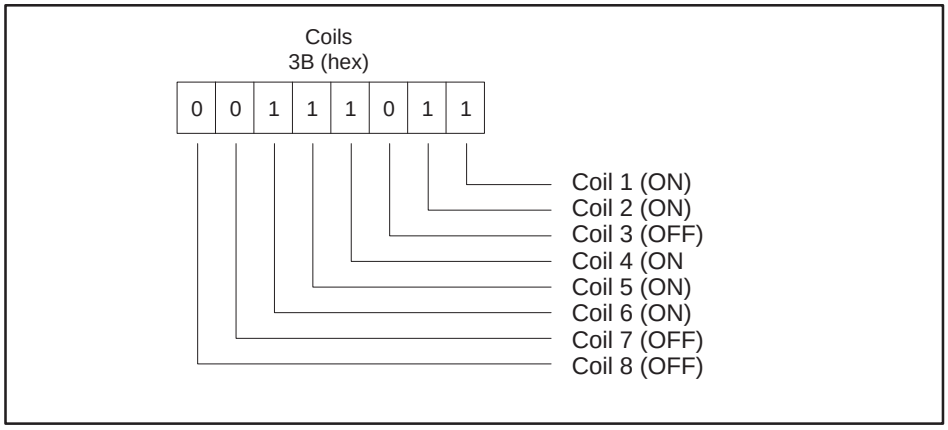



**Figure 2-20 Exception Status Bits**

### Code 08 — Execute Diagnostics

Code 08 enables diagnostic information to be retrieved to conduct network testing. This function does not affect the operation of the PLC, but may have significant effects on the operation of the MODNIM. Where these effects impact upon the operation of the MODNIM, they are described with the details for that diagnostic code.

**⚠WARNING**

**Because of differences in memory mapping between Modbus PLCs and SIMATIC TI PLCs, you should be quite certain of the effects of the broadcast command before issuing it.**

**Example:**

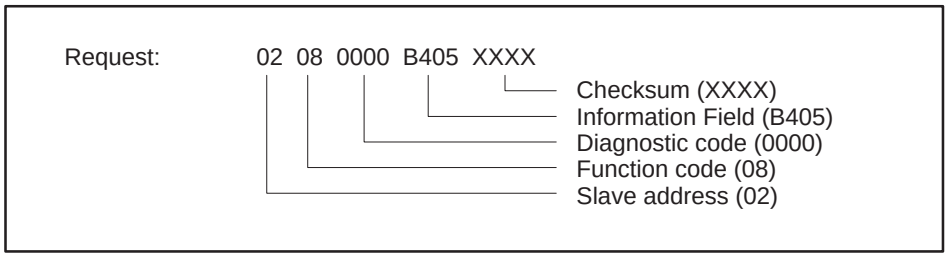



**Figure 2-21 Execute Diagnostics Example — Request**

This example shows a request to conduct diagnostic test 0000. This is the loopback query, where the outgoing message field is returned unchanged.

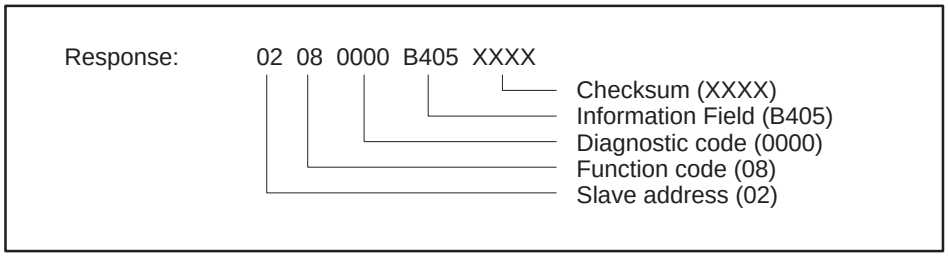



**Figure 2-22 Execute Diagnostics Example — Response**

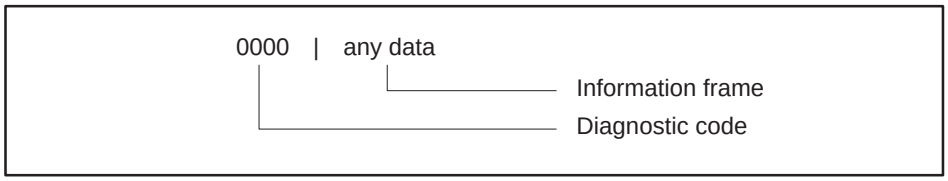
Table 2-4 details the diagnostic codes supported.

**Table 2-4 Diagnostic Codes Supported**

| Code (hex) | Diagnostic Description | Is Code Supported? |
|---|---|---|
| 00 | Return request message | Yes |
| 01 | Restart communications with the slave | Yes |
| 02 | Return diagnostics register | Yes |
| 03 | Change message end character | Yes |
| 04 | Set slave in Listen Only Mode | Yes |
| 0A | Clear counters and diagnostics | Yes |
| 0B | Return message count | Yes |
| 0C | Return checksum error count | Yes |
| 0D | Return exception count | Yes |
| 0E | Return slave message count | Yes |
| 0F | Return slave no-response count | No |
| 10 | Return slave NAK count | No |
| 11 | Return slave busy count | No |
| 12 | Return character overrun count | Yes |
| 13 | Return overrun error count | No |
| 14 | Clear character overrun count | Yes |

The diagnostics codes supported are described in the following paragraphs.

**Diagnostic Code — 00**

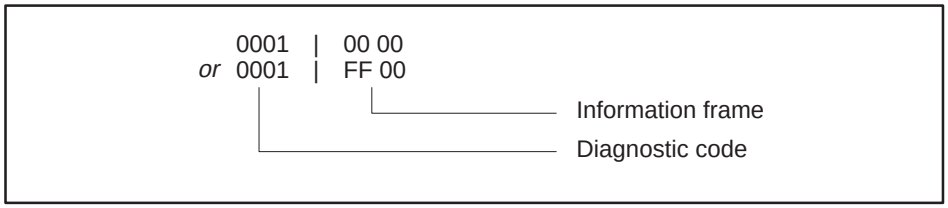



When Function Code 08 uses diagnostic code 0000, any data passed in the information field is returned to the host computer by the addressed secondary node without change.

The purpose of this command is to determine if the communications are functioning correctly, and may be used in conjunction with diagnostic tests described later.

This code has no effect on the functioning of the MODNIM.

**Diagnostic Code — 01**

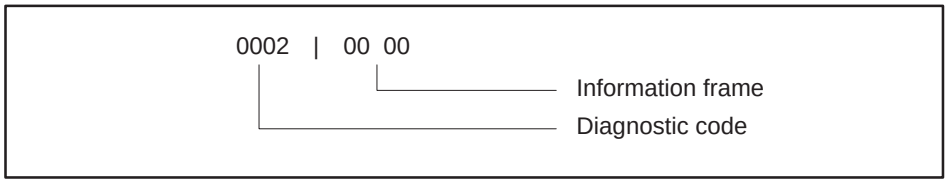



When Function Code 08 uses diagnostic code 0001, the information frame can contain 0000 or FF00.

This function causes the MODNIM to clear all internal counters and registers. If the MODNIM had been in Listen Only Mode (LOM), it will be returned to normal operating mode at the successful conclusion of the command. If the information frame contains FF00, the Communications Event Log will be cleared. If any other value is passed in the information field, the Communications Event Log will not be cleared, and no response will be returned.

This code has significant effect on the functioning of the MODNIM.

### Diagnostic Code — 02




When Function Code 08 uses diagnostic code 0002, the information field may contain any value (00 00 shown). The MODNIM returns a 16-bit word containing diagnostic data about the attached device. The meaning of each bit is given in Table 2-5.

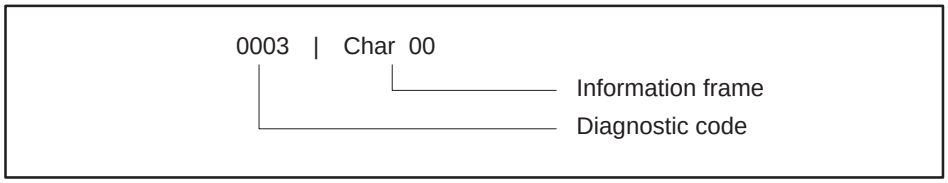
**Table 2-5 Diagnostic Bits**

| Bit | Description |
|---|---|
| 0 | O. S. RAM parity error |
| 1 | Program RAM parity error |
| 2 | O. S. fatal error |
| 3 | Watchdog timeout error |
| 4 | Dynamic program memory diagnostic |
| 5 | Illegal Op code |
| 6 | RAM diagnostics failure |
| 7 | ROM diagnostics failure |
| 8 | Scan diagnostic failure |
| 9 | I/O fatal error |
| 10 | Scan overrun error |
| 11 | Abnormal power loss |
| 12 | Unidentified board failure |
| 13 | EEPROM card RAM download error |
| 14 | Not used |
| 15 | Not used |

If the bit contains a 1, then the error associated with that bit is current.

This code has no effect on the functioning of the MODNIM.

**Diagnostic Code — 03**




When Function Code 08 uses diagnostic code 0003, the ASCII character given in the information field is then used as the delimiter of an ASCII mode message replacing the standard LF (line feed) character.

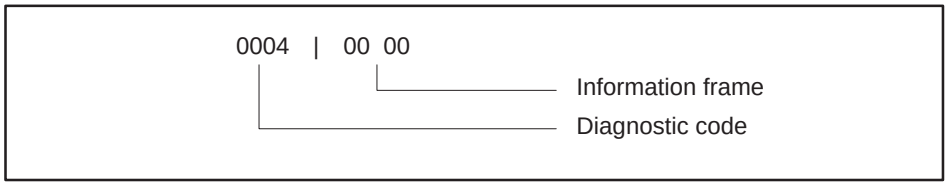
**NOTE:** Avoid using the following characters as alternative 'Ready' field markers:

**:**, **0** to **9**, **a** to **f**, **A** to **F**, or **CR**

You also need to be aware that while the request must be sent with the existing 'ready' character (default: LF), this response, and all future ones will be returned with the new character until a RESET or a power cycle.

This code has significant effects on the operation of the MODNIM.

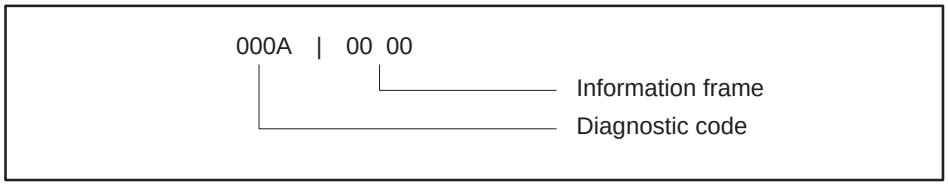
**Diagnostic Code — 04**




When Function Code 08 uses diagnostic code 0004 and any value in the information field (00 00 shown), the secondary node will be forced into Listen Only Mode. In this mode, while the node may be listening to the traffic on the network, it will only react to a Function 8, diagnostic code 01 command, which returns the MODNIM to normal operation. The MODNIM also returns to normal operating mode if the RESET button is pressed or the power is cycled. When the MODNIM enters Listen Only Mode, the ONLINE indicator goes off.

This code has a significant effect on the operation of the MODNIM.

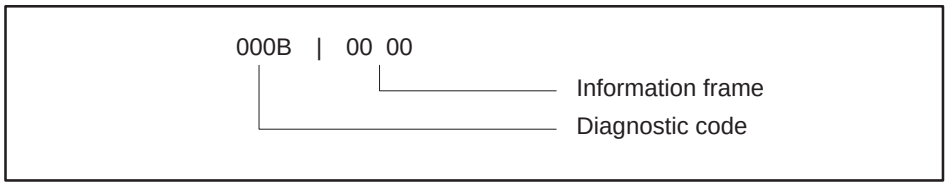
### Diagnostic Code — 0A




When Function Code 08 uses diagnostic code 000A and any value in the information field (00 00 shown), all diagnostic registers and counters within the MODNIM will be cleared. This also happens when the power is cycled or the RESET button is pressed.

This code has a significant effect on the operation of the MODNIM.

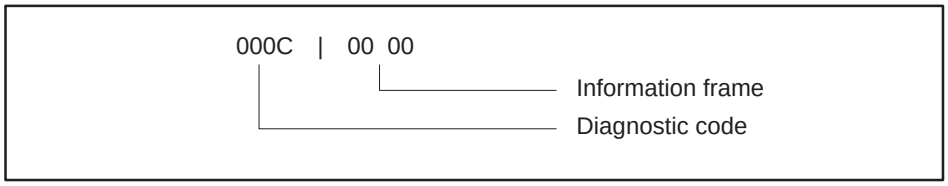
### Diagnostic Code — 0B




When Function Code 08 uses diagnostic code 000B and any value in the information field (00 00 shown), the information field will return the number of processed messages received by the MODNIM since the last RESET or power-up. The counter will increment to a value of FFFF (65535) and will then return to zero (0).

This code has no effect on the operation of the MODNIM.

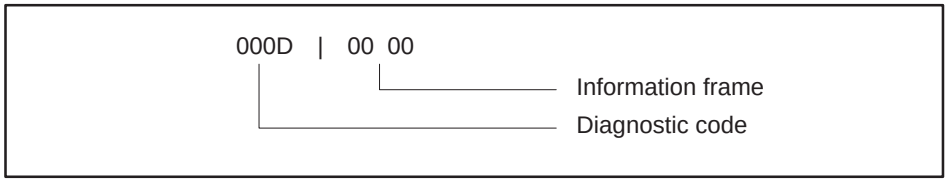
### Diagnostic Code — 0C




When Function Code 08 uses diagnostic code 000C and any value in the information field (00 00 shown), the information field will return the number of checksum errors detected by the MODNIM since the last RESET or power-up. The counter will increment to a value of FFFF (65535) and will then return to zero (0).

This code has no effect on the operation of the MODNIM.

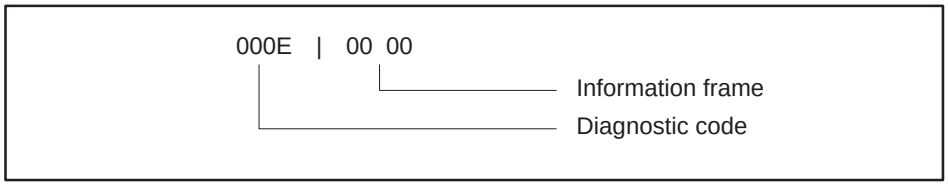
### Diagnostic Code — 0D




When Function Code 08 uses diagnostic code 000D and any value in the information field (00 00 shown), the information field will return the number of exceptions detected by the MODNIM since the last RESET or power-up. The counter will increment to a value of FFFF (65535) and will then return to zero (0).

This code has no effect on the operation of the MODNIM.

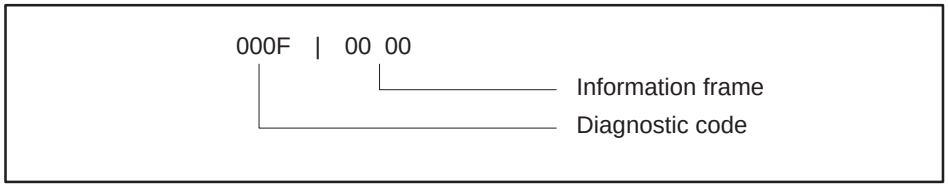
### Diagnostic Code — 0E




When Function Code 08 uses diagnostic code 000E and any value in the information field (00 00 shown), the information field will return the number of messages addressed to the MODNIM since the last RESET or power-up. The counter will increment to a value of FFFF (65535) and will then return to zero (0).

This code has no effect on the operation of the MODNIM.

### Diagnostic Code — 0F

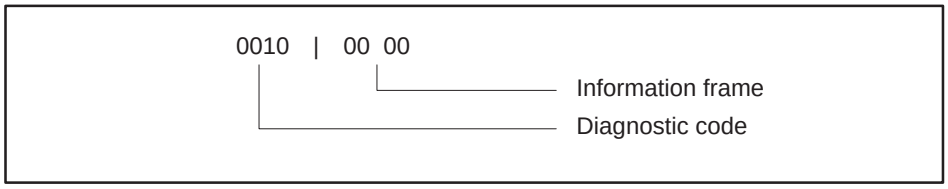



When Function Code 08 uses diagnostic code 000F and any value in the information field (00 00 shown), the MODNIM will return 0000 in the information field.

This code has no effect on the operation of the MODNIM.

This Modbus diagnostic code has no equivalent in Series 500 or Series 505 PLCs. In this application, 0000 is always returned.

### Diagnostic Code — 10

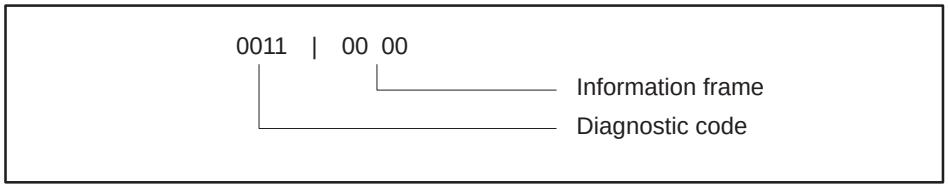



When Function Code 08 uses diagnostic code 0010 and any value in the information field (00 00 shown), the MODNIM will return 0000 in the information field.

This code has no effect on the operation of the MODNIM.

This Modbus diagnostic code has no equivalent in Series 500 or Series 505 PLCs. In this application, 0000 is always returned.

### Diagnostic Code — 11

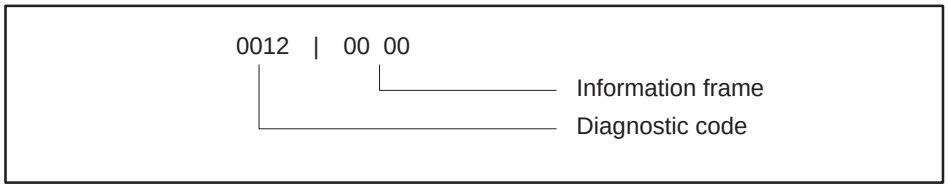



When Function Code 08 uses diagnostic code 0011 and any value in the information field (00 00 shown), the MODNIM will return 0000 in the information field.

This code has no effect on the operation of the MODNIM.

This Modbus diagnostic code has no equivalent in Series 500 or Series 505 PLCs. In this application, 0000 is always returned.

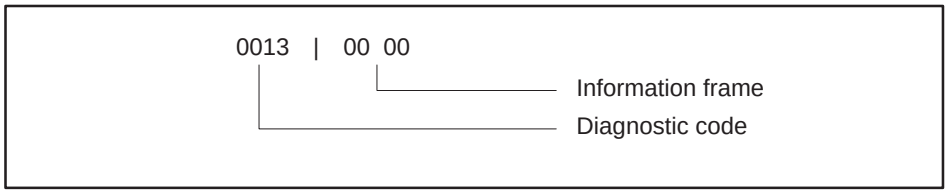
### Diagnostic Code — 12




When Function Code 08 uses diagnostic code 0012 and any value in the information field (00 00 shown), the information field will return the number of character overruns detected by the MODNIM since the last RESET or power-up. This indicates the number of times that characters have arrived faster than the UART can store them without loss. The counter will increment to a value of FFFF (65535) and will then return to zero (0).

This code has no effect on the operation of the MODNIM.

### Diagnostic Code — 13

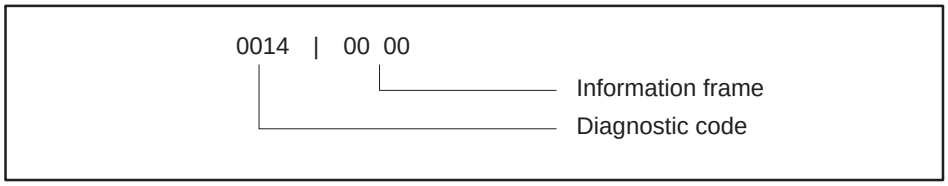



When Function Code 08 uses diagnostic code 0013 and any value in the information field (00 00 shown), the MODNIM will return 0000 in the information field.

This code has no effect on the operation of the MODNIM.

This Modbus diagnostic code has no equivalent in Series 500 or Series 505 PLCs. In this application, 0000 is always returned.

### Diagnostic Code — 14




When Function Code 08 uses diagnostic code 0014 and any value in the information field (00 00 shown), the MODNIM will clear the character overrun counter.

This command affects the operation of the MODNIM.

### Code 11 — Get Comms Event Counter

Code 11 returns a 2-byte status word and an associated 2-byte event counter. The status word is defined in the Modbus Protocol Reference Specification as being 0 for no command in progress and FFFF if a previous command is in progress. The status will always be 0 in the MODNIM.

The event counter is incremented for every successful message; it is not incremented for exceptions or polls, or for this code or Code 12. This code can be used to determine if a single command was successful when a communication error occurred during the request or response.

The event counter will increment to the value FFFF (65535 decimal), then begin again from zero.

**Example:**

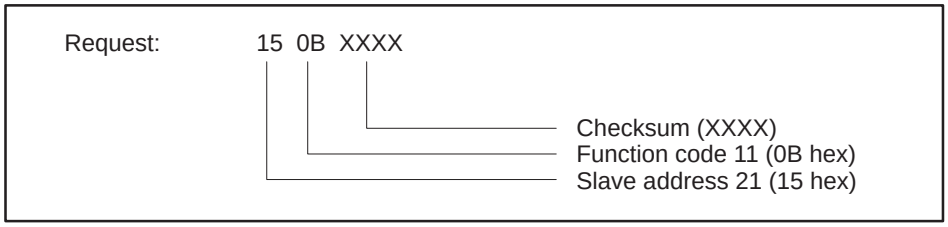



**Figure 2-23 Get Comms Event Counter Example — Request**

In this example, secondary node 21 (15 hex) has been asked to return the event counter and the associated status word.

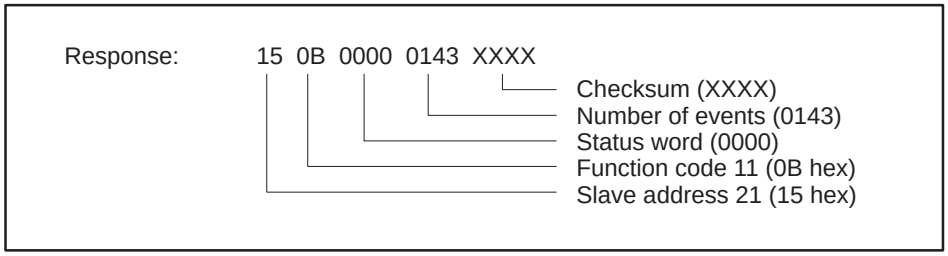



**Figure 2-24 Get Comms Event Counter Example — Response**

The response shows that the status word is 0000 and that 323 (decimal) events have occurred.

### Code 12 — Get Comms Event Log

Code 12 returns the same information as Function Code 11 plus a message count and an additional 64-event byte. This means that a 2-byte status word, a 2-byte event counter, a message count, and the most recent 64 events are returned. A buffer records each send or receive operation and overwrites the oldest event with the latest event. No events are recorded for this code or Code 11. The Event Byte types are shown in Table 2-6.

**Example:**

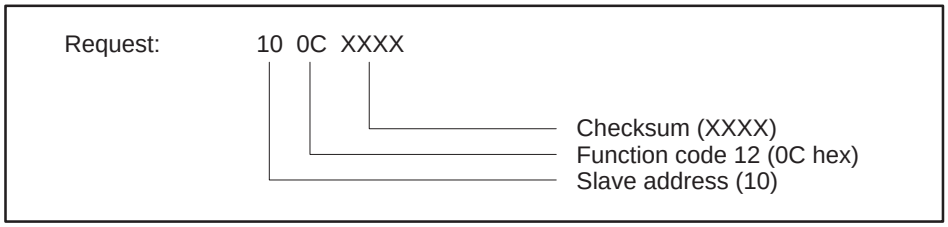



**Figure 2-25 Get Comms Event Log Example — Request**

This request to address 16 (decimal) is for the Exception Event Log.

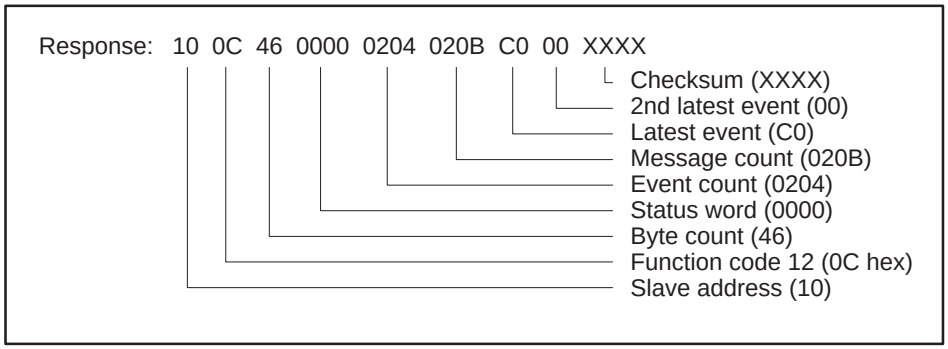



**Figure 2-26 Get Comms Event Log Example — Response**

The response has returned 70 bytes of data (2 status bytes + 2 event counter bytes + 2 message counts + 64 event bytes). Only two of the event bytes are shown; these indicate the secondary node was reset (00) and then received a broadcast message (C0). The most recent events are returned first.

**Table 2-6 Event Byte Types**

| Event Byte | Bit Definitions |
|---|---|
| **Slave Bus Receive.** Byte stored on receipt of message | Bit 0 – Reserved<br>Bit 1 – Set if communications error<br>Bit 2 – Reserved<br>Bit 3 – Reserved<br>Bit 4 – Set if character overrun<br>Bit 5 – Set if in Listen Only mode<br>Bit 6 – Set if Broadcast<br>Bit 7 – 1 |
| **Slave Bus Transmit.** Byte stored when message sent | Bit 0 – Set if an exception (1–3) is sent<br>Bit 1 – Set if secondary abort exception (6) is sent<br>Bit 2 – Set if secondary busy exception (6) is sent<br>Bit 3 – Not used<br>Bit 4 – Set if write timeout occurred<br>Bit 5 – Set if in Listen Only mode<br>Bit 6 – 1<br>Bit 7 – 0 |
| **Entered Listen Only Mode.** Byte stored when LOM mode entered | Bit 0 – 0<br>Bit 1 – 0<br>Bit 2 – 1<br>Bit 3 – 0<br>Bit 4 – 0<br>Bit 5 – 0<br>Bit 6 – 0<br>Bit 7 – 0 |
| **Initiate Communications RESET.** Complete log is set to zeros. | Bit 0 – 0<br>Bit 1 – 0<br>Bit 2 – 0<br>Bit 3 – 0<br>Bit 4 – 0<br>Bit 5 – 0 |

### Code 15 — Write Multiple Coils

Code 15 allows the host computer to write multiple coils. When this command is given, successive coils are changed by an associated bit pattern to the indicated state, where each bit indicates: 1 = ON  0 = OFF

Addressing is sequential up to the maximum memory size for a particular secondary. The maximum number of coils that can be addressed with one command is 800. If more are given, the whole command will be rejected with an Exception code 03.

The broadcast address (00) can be used to change the same coils in all networked programmable controllers.

[This command is equivalent to TIWAY Primitive 30 using data types 04, 07 (Y unpacked, packed) or data types 05, 08 (C unpacked, packed), depending on the position of configuration Switch 10.]

**⚠WARNING**

**Because of differences in memory mapping between Modbus PLCs and SIMATIC TI PLCs, you should be quite certain of the effects of the broadcast command before issuing it.**

**Example:**

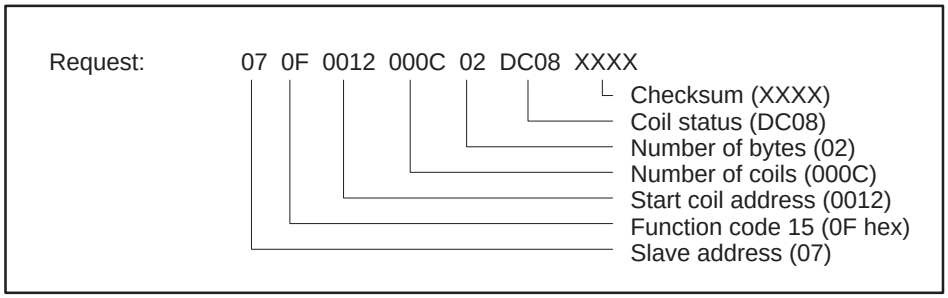



**Figure 2-27 Write Multiple Coils Example — Request**

This request is for the secondary node to change the values of 12 (0C hex) coils starting at address 12 (18 decimal) to the states given by the following bit pattern.

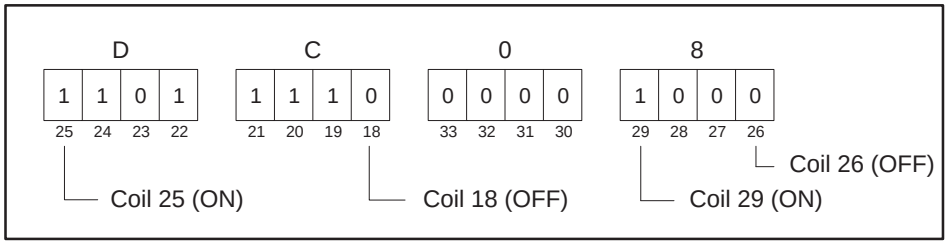



**Figure 2-28 Coil Bit Pattern**

The bit pattern indicates that coils 19, 20, 21, 22, 24, 25, and 29 are to be changed to ON and 18, 23, 26, 27, and 28 changed to OFF.

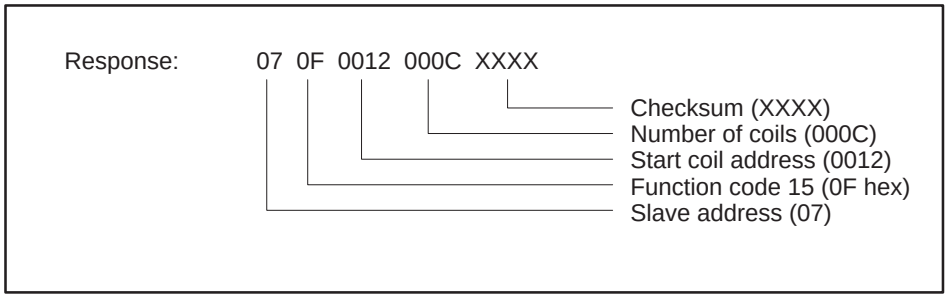



**Figure 2-29 Write Multiple Coils Example — Response**

### Code 16 — Write Multiple Registers

Code 16 allows the host computer to write multiple values into successive registers. Addressing is sequential up to the maximum memory size for a particular PLC. The maximum number of registers that can be addressed with one command is 100. If more are given, the whole command will be rejected with an Exception code 03.

[This command is equivalent to a TIWAY I request using Primitive 30 with TT-type 01 (Variable memory)].

**⚠WARNING**

**Because of differences in memory mapping between Modbus PLCs and SIMATIC TI PLCs, you should be quite certain of the effects of the broadcast command before issuing it.**

**Example:**

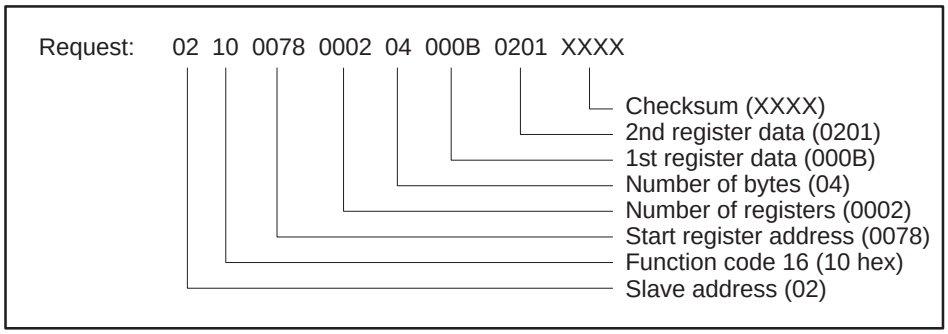



**Figure 2-30 Write Multiple Registers Example — Request**

This request to the secondary node is to change the values of 2 holding registers, starting at address 0078 (120 decimal) to 11 and 513 respectively.

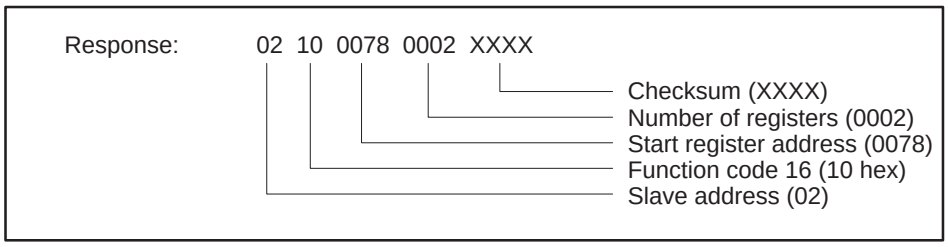



**Figure 2-31 Write Multiple Registers Example — Response**

The response to this command is to return the address, code, starting address, and the numbers of registers to be changed.

### Code 17 — Report Slave ID

Code 17 permits the user to get information from the slave concerning its type, run mode, and other device-dependent data.

[This command has no direct equivalent to a TIWAY I request, but elements of this function are contained in Primitives 02 and 03 (Status primitive and Configuration primitive).]

The MODNIM returns a number in the Slave ID field determined by the type of PLC connected and responds with the state of the RUN light. In addition, it returns the HH status field, the EE auxiliary power source field, the communications port in use, and the software version number.

**Example:**


**Figure 2-32 Report Slave ID Example — Request**

This example shows a request to secondary address 03 to report its ID and associated data.

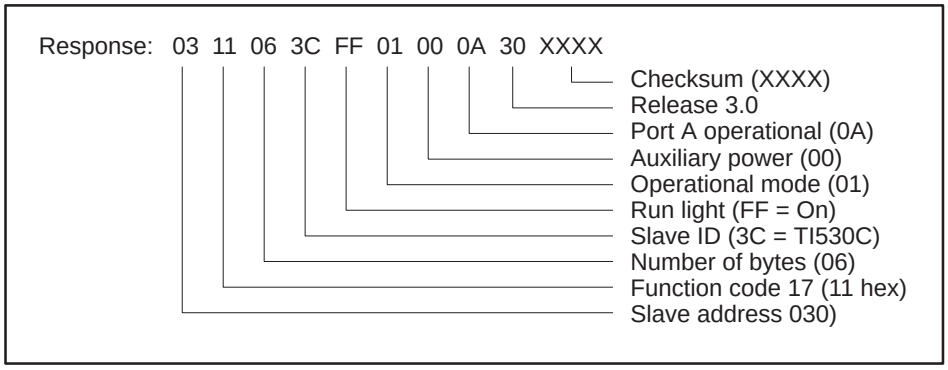



**Figure 2-33 Report Slave ID Example — Response**

This response indicates that 6 bytes have been returned. These bytes show that the Slave ID is a TI530C PLC, the RUN light is On, the operational mode is 01, auxiliary power is good, and the operational port is A. The last byte contains the software version number (see Table 2-7).

The options for each of the fields in the response are listed in Table 2-7.

**Table 2-7 Function Code 17 Response Options**

| Option | Code | Description |
|---|---|---|
| **Slave ID** (Identifies SIMATIC TI PLC) (Note: There can be no guarantee that these codes will not be used by other devices on a Modbus network.) | 20 | SIMATIC® TI520™ |
| | 25 | SIMATIC® TI525™ |
| | 2C | SIMATIC® TI520C™ |
| | 30 | SIMATIC® TI530™ |
| | 3C | SIMATIC® TI530C™ |
| | 35 | SIMATIC® TI535™ |
| | 45 | SIMATIC® TI545™ |
| | 55 | SIMATIC® TI555™ |
| | 60 | SIMATIC® TI560™ |
| | 65 | SIMATIC® TI565™ |
| | 75 | SIMATIC® TI575™ |
| **Run Light** | 00 | OFF |
| | FF | ON |
| **Operational Mode (HH)** | 00 | Operational and performing instruction data type and loop execution (RUN). |
| | 01 | Operational and performing instruction data type and executing with a non-fatal error (RUN with non-fatal error). |
| | 03 | Operational and not performing instruction data type execution or loop execution (PROGRAM). |
| | 05 | Operational and not performing instruction data type execution or loop execution and a non-fatal error is detected (PROGRAM with non-fatal error). |
| | 80 | Not operational due to fatal error condition. |
| **PLC Auxiliary Power Supply Status (EE)** | 00 | Auxiliary power source good. |
| | 80 | Auxiliary power source not good. |
| **MODNIM Operational Port** | 0A | Port A is in use. |
| | 0B | Port B is in use. |
| **Software Version** | 00 | Release 1.0 |
| | 20 | Release 2.0 |
| | 30 | Release 3.0 |

## 2.5 Error Responses

Errors such as illegal addressing or failure of communications with a PLC are reported by exception responses. Exception responses replace the expected response to a request and are recognized by the Function Code with the top bit set high (see examples in Figure 2-34 and Figure 2-35).

**Request:**

| Start | Address | Function | Data | LRC | EOF | Ready |
|---|---|---|---|---|---|---|
| : | 07 | 01 | 00000041 | AD | CR | LF |

**Response:**

| Start | Address | Function | Except. Code | LRC | EOF | Ready |
|---|---|---|---|---|---|---|
| : | 07 | 81 | 02 | 76 | CR | LF |

**Figure 2-34 Exception Response Frame— ASCII Mode**

**Request:**

| Address | Function | Data | CRC |
|---|---|---|---|
| 07 | 01 | 00000041 | XXXX |

**Response:**

| Address | Function | Exception Code | CRC |
|---|---|---|---|
| 07 | 81 | 02 | XXXX |

**Figure 2-35 Exception Response Frame — RTU Mode**

In this example, the request is for secondary node 07 to return the data for 65 (41 hex) coils starting at address 00, which is an illegal address for a SIMATIC TI PLC. The response is an exception indicated by the Function Code 81 (01 with the high bit set), with the exception code 02 indicating that the data address was illegal.

**NOTE:** The TIWAY I network uses Exception Primitives. Exceptions sent by SIMATIC TI PLCs are mapped to Modbus Exception Codes.

Table 2-8 list the exception responses that apply to the functions supported by the MODNIM.

**Table 2-8 Exception Responses**

| Code | Name | Description |
|---|---|---|
| 01 | Illegal Function | The function received is not defined for this application [equivalent to SIMATIC TI exception codes 00, 01, 06, 15]. |
| 02 | Illegal Data Address | The address contained in the data field is not valid for the secondary being addressed [equivalent to SIMATIC TI code 02] |
| 03 | Illegal Data Value | The value passed in the data field is not allowable for the secondary being addressed [equivalent to SIMATIC TI codes 10, 19, 1D]. |
| 04 | Failure in associated device | The secondary address has failed to respond, or a command has aborted [equivalent to SIMATIC TI codes 0A, 0B, 0D, 17, 1A]. |
| 06 | Memory parity error | If during a read of memory, a parity error occurs, this exception response is returned. |

*Appendix A*

# Network Cable Communications

## A.1 Network Cables

### Recommended Communication Cables

The following cables are available through your Siemens distributor.

| Cable P/N | Description | Adapter at Module |
|---|---|---|
| 2601094–8001 | Standard 9-pin null modem | 9-pin (F) to 25-pin (M) port adapter |
| VPU200–3605 | Standard 25-pin null modem | None required |

### Building Cables

If you prefer to build your own 25-pin cable, ensure that the following requirements are met to minimize the possibility of noise interference.

- Cable conductors — 26 AWG, tinned and stranded copper wire, with one uninsulated 26 AWG tinned copper drain wire
- Cable shield — AL foil or aluminum-polyester foil and a 65% minimum tinned copper braid shield
- Outer jacket — polyvinyl chloride (PVC) or equivalent material, with a UL rating of 30V, 60°C minimum
- Connectors — 25-pin male D-connector, with contact pins of copper alloy with gold flashing over nickel plate

### Cable Pinouts

Figure A-1, Figure A-2, and Figure A-3 show the cable connections which can be used for both ASCII and RTU modes of transmission.

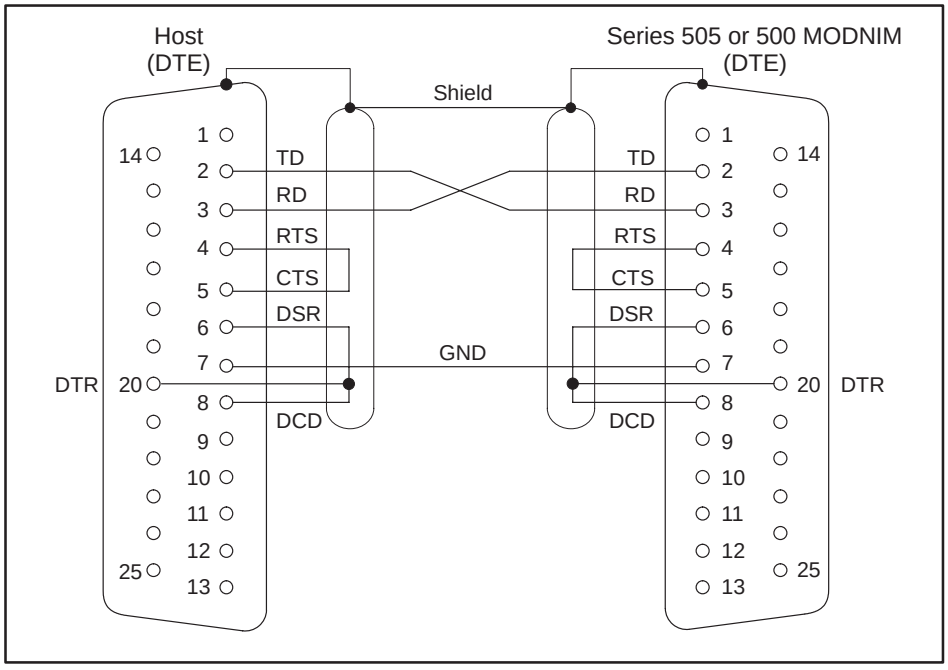



**Figure A-1 MODNIM to Host (without Handshaking)**

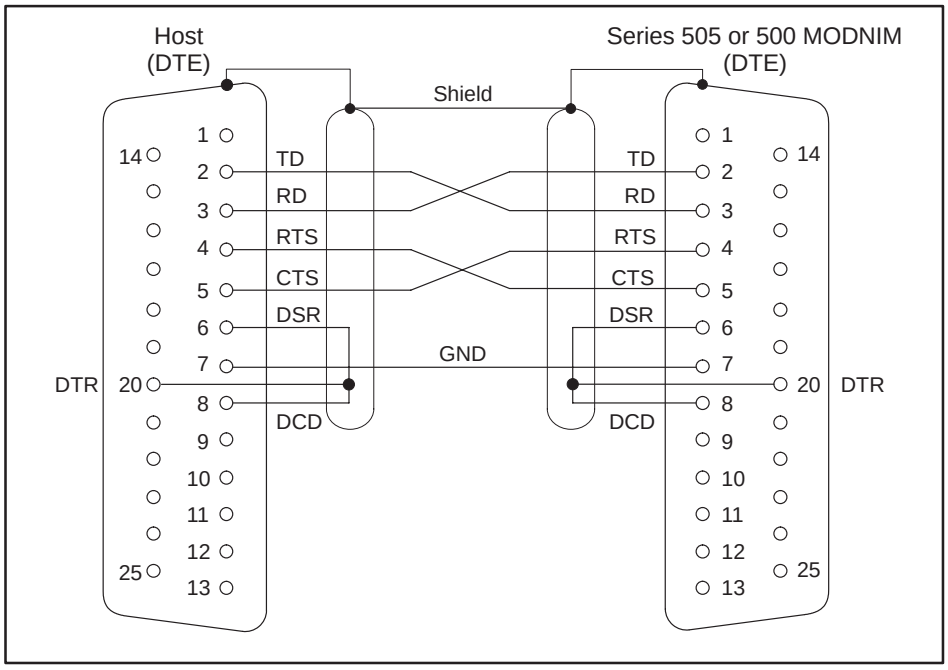



**Figure A-2 MODNIM to Host (with Handshaking)**

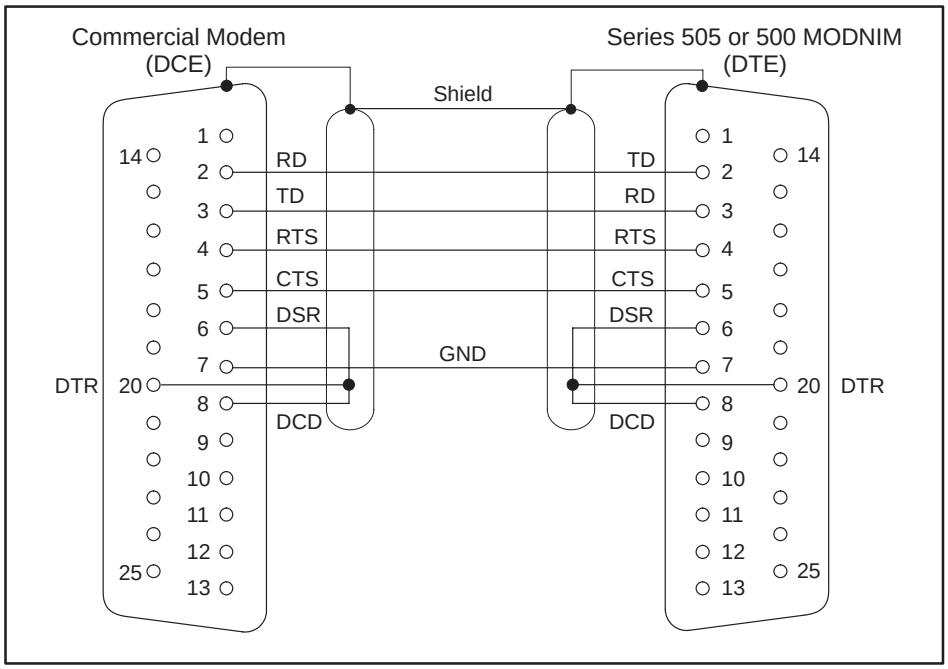



**Figure A-3 MODNIM to Commercial Modem (with Handshaking)**

## A.2 Communications Parameters

### RS-232-C Pin Assignments

The modem interface is a standard Type E DTE configuration as defined in the EIA RS-232-C standard. This interface uses a male 25-pin D-type connector plug on the communication cable. The pin assignments are listed in Table A-1.

**Table A-1 RS-232-C Connector Pin Assignments**

| Pin # | Description |
|---|---|
| 2 | Transmit Data (TXD) |
| 3 | Receive Data (RXD) |
| 4 | Request to Send (RTS) |
| 5 | Clear to Send (CTS) |
| 6 | Data Set Ready (DSR) |
| 7 | Signal Ground |
| 8 | Receive Line Signal Detector/Data Carrier Detect (RLSD/DCD) |
| 20 | Data Terminal Ready (DTR) |

### Recommended Communications Parameters

The following communications parameters are suggested as a starting point for each mode of transmission when first establishing a connection.

**Table A-2 Communications Parameters**

| Parameters | RTU | ASCII |
|---|---|---|
| Baud Rate | 19,200 bps | 19,200 bps |
| Parity | None | Even |
| Stop bits | 1 | 1 |
| Data bits | 8 (auto selected) | 7 (auto selected) |

*Appendix B*

# Exception Codes

## B.1 Exception Codes Supported by the MODNIM

Table B-1 lists the subset of SIMATIC TI Exception Codes that are mapped to the Modbus exception codes.

**Table B-1 Exception Codes**

| Exception Code (hex) | Definition |
|---|---|
| 00 | The primitive is not implemented. |
| 01 | Data type is not defined in the attached device. |
| 02 | Data element location is out of range. |
| 06 | Device in wrong mode for primitive execution. |
| 07 | User program in device has disabled communications to NIM (Lockout Bit) |
| 0A | Device fails to respond. |
| 0B | Primitive aborted due to fatal error condition in attached device. |
| 0D | Error encountered while executing the requested primitive. |
| 10 | The number of locations requested exceeds the maximum allowed. |
| 15 | Primitive not allowed while device is in local mode. |
| 17 | The attached device did not respond properly (communications problem). |
| 19 | The resulting data element location formed by the starting address plus the number of data elements to access, is out of range. |
| 1A | Communications has not been established with the attached device. |
| 1D | The number of locations to access is zero. |

# *Appendix C*
# Specifications

## C.1 Environmental Specifications

The Series 505 I/O subsystem is tested against some of the most stringent standards in the world. Table C-1 lists specifications that are common to Series 505 I/O devices.

**Table C-1 Environmental Specifications**

| | |
|---|---|
| Operating Temperature | 0° to 60°C (32° to 140°F) |
| Storage Temperature | –40° to +70°C (–40° to 158°F) |
| Relative Humidity | 5% to 95% non-condensing |
| Pollution Degree | 2, IEC 664, 664A |
| Vibration<br>Sinusoidal<br><br><br>Random | IEC 68-2-6, Test Fc;<br>0.15 mm, peak-to-peak, 10–57 Hz;<br>1.0 g 57–150 Hz<br><br>NAVMAT P-9492 or IEC 68-2-34<br>Test Fdc with 0.04 $g^2$/Hz, 80–350 Hz, and<br>3 dB/octave rolloff, 80–20 Hz and<br>350–2000 Hz at 10 min/axis |
| Impact Shock | IEC, 68-2-27, Test Ea; Half sine, 15 g 11 ms |
| Electric Noise Immunity<br>Conducted noise:<br><br><br><br><br>Radiated noise:<br><br>Electrostatic discharge: | <br>IEC 801, Part 4, Level 3<br>MIL STD 461B, Part 4; CS01, CS02, CS06<br>IEC 255-4<br>EEC 4517/79 Com(78) 766 Final, Part 4<br>IEEE 472, 2.5 kV<br>IEC 801, Part 3, Level 3<br>MIL STD 461B, Part 4; RS01, RS02<br>IEC 801, Part 2, Level 4, (15 kV) |
| System Isolation | Isolation (user-side to controller-side): 1500 Vrms |
| Torque for bezel screws | 0.3 N-m (2.61 in.-lbs.) minimum<br>0.6 N-m (5.22 in.-lbs.) maximum |
| Corrosion Protection | All parts are of corrosion resistant material or are plated or painted as corrosion protection. |
| Agency Approvals | UL® Listed (UL508 industrial control equipment)<br>CSA® Certified (CSA142 process control equipment)<br>FM Approved (Class I, Div. 2, Hazardous locations) |

## C.2 MODNIM Communications Specifications

Table C-2 lists communication and other specifications of the MODNIM.

**Table C-2 MODNIM Specifications**

| Item | Description |
|---|---|
| Model Number | PPX:505–5184; PPX:500–5184 |
| Media | Dual RS-232-C/423 |
| I/O Slots | 1 slot (Series 505); 2 slots (Series 500) |
| Data Rates | 50, 75, 110, 150, 200, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200 |
| RS-232-C/423 Ports | Configured as DTE, asynchronous, with or without RTS/CTS handshaking; external modem support |
| Data Link Protocol | ASCII and RTU |
| Network Media | Appropriate cable for RS-232-C |
| Maximum power from base | 8 W, +5 VDC |
| Weight, dimensions of unpacked module | 1 lb., 3 oz; 10.5″ × 8.0″ × 0.8″ |
| Weight, dimensions of packed module | 2 lbs., 9 oz; 12.75″ × 13.5″ × 3.75″ |

Table C-3 lists the TIWAY primitives that correspond to the Modbus codes.

**Table C-3 Modbus Codes vs. TIWAY Primitives Chart**

| Code | Modbus Description | Code | TIWAY Primitive Description |
|---|---|---|---|
| 01 | Read Coil Status | TT=7<br>TT=8 | Read discrete outputs packed (Y)<br>Read coils packed (C) |
| 02 | Read Input Status | TT=6 | Read discrete inputs packed (X) |
| 03 | Read Holding Register | TT=1 | Read V memory (V) |
| 04 | Read Input Register | TT=9 | Read word input (WX) |
| 05 | Force a Single Coil | TT=4<br>TT=5 | Write a single discrete output (Y)<br>Write a single discrete coil (C) |
| 06 | Write a Single Register | TT=1 | Write a single V memory (V) |
| 07 | Read Exception Status | TT=4,7<br>TT=5,8 | Read 1st 8 discrete outputs (Y)<br>Read 1st 8 C coils 1 – 8 (C) |
| 08 | Execute Diagnostics | | |
| 11 | Get Comms. Event Cntr. | | |
| 12 | Get Comms. Event Log | | |
| 15 | Write Multiple Coils | TT=4,7<br>TT=5,8 | Write mult. discr. outputs packed (Y)<br>Write mult. discrete coils packed (C) |
| 16 | Write Multiple Registers | TT=1 | Write multiple V memory (V) |
| 17 | Report Slave I.D. | | |

*Appendix D*

# Using the MODASST Program

## D.1 Getting Started

### Overview

MODASST is a Configuration and Diagnostic Assistant software program which can be used to help set up and use your MODNIM or TIWAY Gateway product. This program is provided to assist you in configuring the modules and verifying that the module is communicating. The program can also help verify that your MODNIM or TIWAY Gateway is operational when you are installing it.

The MODASST program provides diagrams of cable pin-outs, dipswitch settings, and a facility to communicate with the MODNIM or TIWAY Gateway.

### Hardware Requirements

You will need the following components to run this program.

- IBM® PC-compatible computer
- The MODASST program
- Serial communications port on your computer. (A mouse is optional.)
- RS-232-C null-modem cable (the same cable as the one used from MODNIM to Host, connected directly without modems).

To set up the hardware to use this program, find out the COM port number you will use to communicate with the module, then follow these steps.

1. Connect the cable from your computer communications port to either one of the 25-pin connectors on the module.
2. When safe to do so, power up the MODNIM and its associated PLC or power up the TIWAY Gateway module.
3. Turn on the power on your computer.

### Running the Program from the Floppy Disk

This program does not require any special installation to use. You may run this program directly from the floppy disk or from a hard drive.

To run this program from the floppy disk, follow these steps.

1. Insert the disk into the appropriate slot in your computer.
2. At the DOS prompt on your machine, type:

   **`a:`** Enter
   **`modasst`** Enter

You will now see the start-up screen. Press Enter to continue. The program starts your configuration setup with the screen as shown in Figure D-1.

## Running the Program from the Hard Disk

To set up this program on your computer's hard drive, follow these steps.

1. Insert the disk into the appropriate slot in your computer.
2. At the DOS prompt on your machine, type:

```
c: Enter
cd\ Enter
md\modasst Enter
cd\modasst Enter
copy a:*.* c:\modasst Enter
```

To run the program on your computer's hard drive, at the DOS prompt type:

```
c: Enter
cd\modasst Enter
modasst Enter
```

You will now see the start-up screen. Press Enter to continue. The program starts your configuration setup with the screen as shown in Figure D-1.

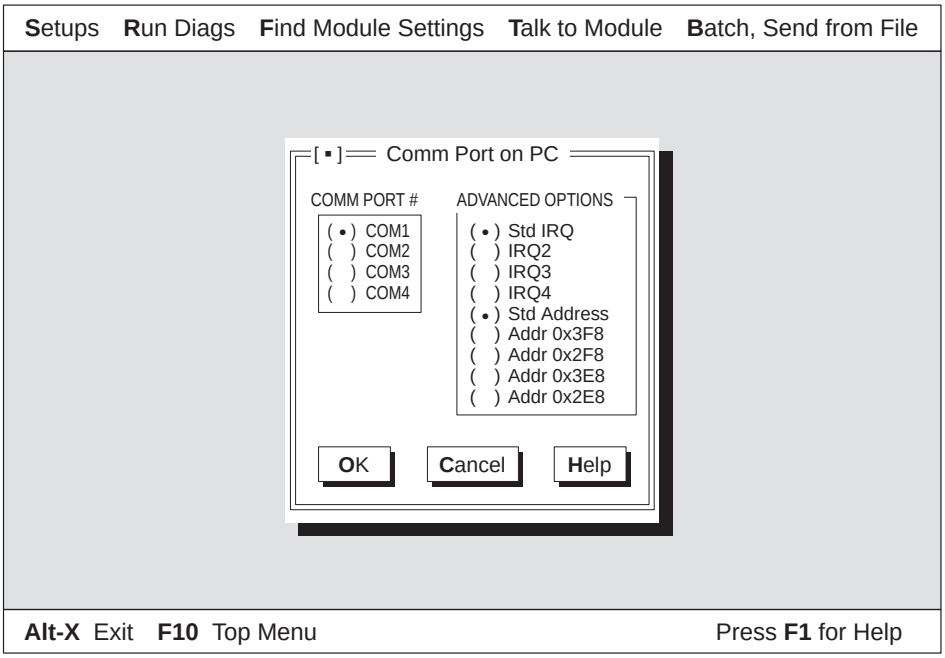



**Figure D-1 First Configuration Screen after Start-Up**

## D.2 Starting Program Operation

### Setting the Communication Port Options

When the program begins, you will be prompted for the communications port number that you wish to use to communicate from your computer to the MODNIM or TIWAY Gateway module.

Use the up and down arrows on your keyboard to select the communications port from COM1 to COM4. If you don't need to set up advanced options for your COM port, press Enter to continue.

If you need to set up advanced options, press TAB to move the cursor from the COM Port box to the Advanced Options box. Use the up/down arrow keys to select the desired IRQ, then press TAB to move the cursor to the address option. Again, select the desired option and press Enter when your selections are complete.

### Setting the Remaining Options

After setting the communication port options, the first of seven additional option windows is displayed to help you complete your configuration settings. These set-up windows are the following.

- Output Type Mapping: Y outputs or C coils
- RTS/CTS Handshaking Selection: Enable or Disable
- Transmission Mode: RTU or ASCII
- Parity Selection: Even, Odd, or No parity
- Stop Bit Selection: 1 or 2 stop bits
- Baud Rate: 50 to 19200
- Network Address: type in the appropriate address number if other than the default address 1.

For each of these set-up windows, use the arrow keys if you want to select a choice other than the highlighted default, then press Enter to continue to the next selection. You can also press TAB to move the highlight cursor to the **OK**, **Cancel**, or **Help** options at the bottom of each window.

In each panel, press TAB to move from one field of the screen to the next and the arrow keys to move the selection dots from one choice to another. Press Enter to accept your choice and exit the screen. To exit a screen without choosing anything or to cancel the selections made, press Esc. You can also use a mouse to select menu options by placing the mouse cursor on the option or selection and pressing the left mouse button. Press F1 at any time to access Help screens.

After entering the last set-up option, the screen is empty except for the top and bottom menus. Press ALT and one of the highlighted letters on the top row of the screen to select the function option you want. To exit the MODASST program, press ALT and **X** together.

## D.3 Using Program Options

### Setup Options

Figure D-2 shows the first menu and the sub-menu available after pressing ALT S to choose the Setups option from the top menu. You can also press F10 to access the top menu, and, since Setups is the default, press Enter.

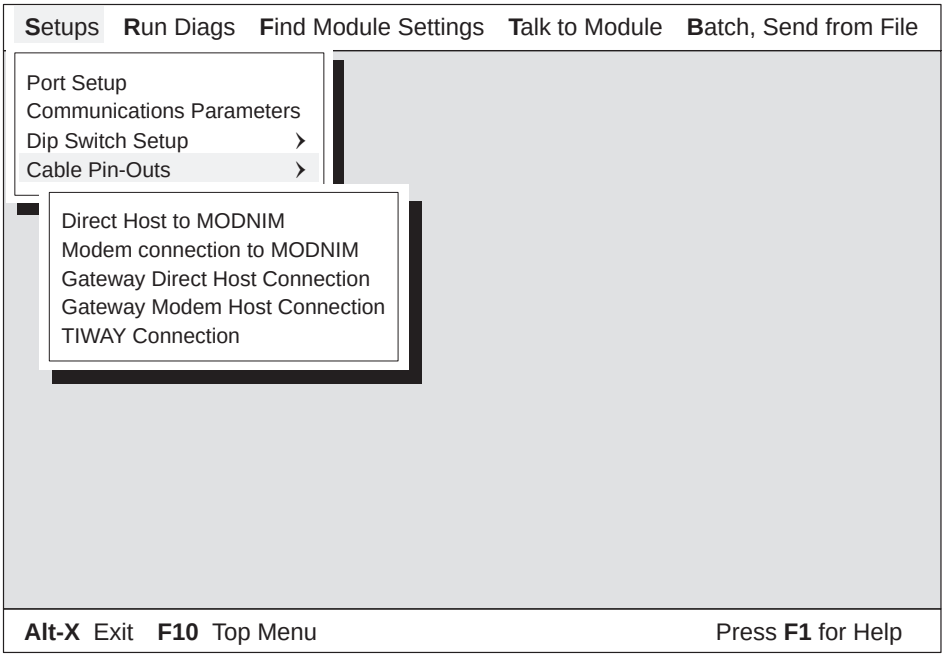



**Figure D-2 MODASST Setup Menu**

Under Setups, you can choose one of the following options.

- **Port Setup** allows you to change the computer Com Port that you selected when the program started.
- **Communications Parameters** allows you to reset the parameters to communicate with the MODNIM or TIWAY Gateway module.
- **Dip Switch Setup** displays the dipswitches that need to be selected to configure the MODNIM or TIWAY Gateway module for the settings chosen using the Parameters option.
- **Cable Pin-Outs** displays the cable pin-outs required to connect the MODNIM or TIWAY Gateway modules to a host computer or a modem.

### Run Diagnostics

The Run Diags option lists the MODNIM diagnostics available using function code 08. You can select and run a diagnostic function by using the up/down arrows and pressing Enter. This option uses the communications parameters selected on the Setup menu item to run diagnostics.

### Find Module Settings

The Find Module option is available when you want to communicate with a MODNIM or TIWAY Gateway module and the communications parameters are not known. This selection tries the various parameter combinations until either a successful match of communications parameters is found, or all combinations have been tried. This option then displays on the screen the communications parameters determined and an option to choose those as the parameters to use for module setup.

### Talk to Module

The Talk to Module option tries to communicate with the MODNIM or the TIWAY Gateway module using one of the following choices.

- **Check Module Communications** sends a few read requests to the module and displays on the screen if communications were successful.
- **Prompt for Info to Send** prompts for the function to send and for each of the function fields. (These are described in Chapter 2 of the user manual.) The function is then packaged properly and sent to the MODNIM or TIWAY Gateway module.
- **Expert, Send Specific Request** allows you to type a complete function request on the screen. Refer to Chapter 2 for request format. (The start field, LRC or CRC field, and Ready field do not need to be entered; they will be computed before being sent.) The program then sends the request to the module. The response from the module is displayed exactly as it is returned from the module. All fields are displayed.

### Batch Send from File

This option will Batch Send a file to the module. It prompts for the input file name and for the output file name. If you do not change them, it assumes **`modasst.tst`** as the input file name and **`modasst.log`** as the output file name.

### Verifying Communications

After using the MODASST program to verify that communications with the MODNIM or Gateway are functional and properly configured, you may still find that network communications are not operating in your installation. In that case, the network cabling from the MODNIM or TIWAY Gateway module to the host computer or the host computer itself may need to be evaluated to determine the source of the problem.

# Index

## I



## K



## L



## M



## N



## O



## P

# Customer Registration

We would like to know what you think about our user manuals so that we can serve you better. How would you rate the quality of our manuals?

| | Excellent | Good | Fair | Poor |
|---|---|---|---|---|
| Accuracy | _____ | _____ | _____ | _____ |
| Organization | _____ | _____ | _____ | _____ |
| Clarity | _____ | _____ | _____ | _____ |
| Completeness | _____ | _____ | _____ | _____ |
| Overall design | _____ | _____ | _____ | _____ |
| Size | _____ | _____ | _____ | _____ |
| Index | _____ | _____ | _____ | _____ |

Would you be interested in giving us more detailed comments about our manuals?

☐ **Yes!** Please send me a questionnaire.

☐ **No.** Thanks anyway.

Your Name: _____

Title: _____

Telephone Number: (____)_____

Company Name: _____

Company Address: _____

_____

_____

**Manual Name:** SIMATIC TI505/TI500 MODNIM User Manual
**Edition:** Original
**Manual Assembly Number:** 2586546-0083
**Date:** 03/93
**Order Number:** PPX:505-8122-1

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO.3 JOHNSON CITY, TN

POSTAGE WILL BE PAID BY ADDRESSEE

SIEMENS INDUSTRIAL AUTOMATION, INC.
3000 BILL GARLAND RD.
P.O. BOX 1255
JOHNSON CITY TN 37605–1255

ATTN: Technical Communications M/S 3519

FOLD