2573-MOD and 2573-TCM2
SERIAL DEVICE INTERFACE ADAPTER
INSTALLATION AND OPERATION GUIDE
Version 2.4

CTI Part # 062-00187-024

DOCUMENT DISCLAIMER STATEMENT

Every effort has been made to ensure the accuracy of this document; however, errors do occasionally occur. CTI provides this document on an "as is" basis and assumes no responsibility for direct or consequential damages resulting from the use of this document. This document is provided without express or implied warranty of any kind, including but not limited to the warranties of merchantability or fitness for a particular purpose. This document and the products it references are subject to change without notice. If you have a comment or discover an error, please call us toll-free at 1-800-537-8398.

REVISION HISTORY

Ver 1.0	6/12/95	Original Release
Ver 1.1	10/27/95	Added description of Modem Support Feature
		Added error code descriptions missing from the previous version.
		Added Modbus ASCII protocol manager description.
		Corrected the Modbus RTU example command block.
		Corrected miscellaneous typographical errors.
Ver 2.0	11/18/96	Added Memory Exchange Command description for CAMP master
		Created separate chapters for Modbus protocol managers
		Documented Modbus Slave Protocol Managers (new)
		Documented Modbus Master Initiate Poll List command (new)
		Documented Modem Control Feature for Modbus Master (new)
		Revised Modem Control Feature chapter to improve clarity
		Added Appendix D - PLC Command Interface
		Added missing error codes 2877, 2977
Ver 2.1	2/17/97	Documented RS-422/485 Disconnect Switches for 2573 rev. "D" boards
		Corrected errors in the description of flags for the Modbus Initiate Poll command and the Query Block
Ver 2.2	3/12/98	Updated RS-422 2573 MOD to PLC serial cable section to include pinout configuration for newer PLCs.
		Corrected the Modem Configuration Table definition for Offset 6
		Added missing Modem error codes 04EA, 04EB, 04EC
		Added Siemens® generated general memory transfer error codes 0201-0261 to Appendix B. Corrected Offset 8 definition in the Create Connection Command Block - Modbus RTU Master.
Ver 2.3	6/15/98	Added RTS/CTS handshake option description for ver 6.2 firmware. Added Accept Early Response configuration bit description.
Ver 2.4	10/03/01	Added Modem Handshake Option description for ver 7.0 firmware.
		Added Modbus Enron Option
		Clarified command codes for Send Modbus Query and Initiate Poll List
		Corrected errors in Modbus Master ladder logic examples
		Moved hardware handshake to Appendix B and Revised Appendix E (Modbus)
		Combined 2573-MOD and 2573-TCM2 manuals and updated copyright

PREFACE

This ***Installation and Operation Guide*** provides installation and operation instructions for the CTI 2573-MOD and the 2573-TCM2. The information in this manual is directed to individuals who will be installing, maintaining, and troubleshooting the module.

This manual assumes that you are familiar with the installation and operation of SIMATIC® 505 programmable controllers. Please refer to the appropriate SIMATIC® user documentation for specific information on SIMATIC® 505 programmable controllers and I/O modules.

This manual also assumes that, if you intend to use the Modbus Master or Slave protocol managers, that you have an understanding of the Modbus Protocol. This protocol is fully specified in the Modicon Modbus Protocol Reference Guide (PI-MBUS-300 Rev E) published by:

> MODICON Inc. Industrial Automation Systems
> One High Street
> North Andover
> MA 01485

NOTE:
Only the 2573-MOD is capable of running the Modbus protocol managers.

This manual describes the use of NITP/ CAMP Slave, Task Code Master, CAMP Master and Modbus protocol managers. Instructions for using the General ASCII Support (GAS) Protocol Manager are contained in a separate manual, the *General ASCII Support Protocol Manager Reference Manual* (CTI Part # 62-182).

USAGE CONVENTIONS

> *NOTE:*
> *Notes alert the user to special features or procedures.*

> *CAUTION:*
> *Cautions alert the user to procedures that could damage equipment.*

> **WARNING:**
> **Warnings alert the user to procedures that could damage equipment and endanger the user.**

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1. DESCRIPTION

1.1. Introduction

The CTI 2573-MOD and 2573-TCM2 Serial Interface Adapters are a microprocessor-based communications modules for SIMATIC® 505 controllers. Both the 2573-MOD and 2573-TCM2 can enable communications with bar code readers, scales, displays, motor drives, printers, and other PLCs In addition, the 2573-MOD provides the capability to interface the PLC to devices which support serial data communications using the Modbus RTU protocol. Typical devices include flow meters, motor drives, power monitors, and PLCs.

CTI 2573 modules provide four serial ports. Each port is independently configurable for electrical interface, communications parameters, and protocol. Each serial port supports RS-422, RS-485, and a subset of RS-232-C electrical interfaces. The operative electrical interface is determined by choosing which pins are connected to the serial communications cable.

The module firmware includes a set of Protocol Managers which process messages sent to and from the serial ports and transfer message data between the module and the PLC. Each serial port can be individually configured to use a particular protocol manager. Ports 1 and 3 can use any available protocol manager. Ports 2 and 4 can use any available protocol manager except Modbus RTU. Once configured, each serial port operates independently and concurrently.

The 2573-TCM2 and the 2573-MOD support CAMP/NITP (Slave, Task Code (NITP) Master, Common ASCII Message Protocol (CAMP) Slave, and CAMP Master. They also feature a General ASCII Support (GAS) protocol manager which can be tailored to support a wide variety of ASCII protocols. The 2573-MOD adds protocol managers for Modbus RTU Master, Modbus ASCII Master, Modbus RTU Slave, and Modbus ASCII Slave.

The module provides an Active LED, which indicates module status, and four sets of transmit and receive LEDs which indicate when a signal is transmitted and received.

Figure 1. 2573 Front Panel

1.2. Serial Ports

The 2573 module provides four serial ports. All four ports can operate concurrently. Each serial port can be independently configured. Thus, you could have up to four different types of devices communicating with the PLC at the same time.

The serial ports use a male DB9 connector which incorporates the most common serial electrical interfaces, RS-232-C, RS-422, and RS-485-A. You select the electrical interface by connecting the appropriate connector pins to the communications cable.

The most commonly used electrical interface for point-to-point serial communications is RS-232C. The module supports the standard PC/AT RS-232 pinout configuration for RECEIVE DATA (RxD- pin 2), TRANSMIT DATA (TxD - pin 3), SIGNAL GROUND (SG - pin 5), REQUEST TO SEND (RTS - pin 7), and CLEAR TO SEND (CTS - pin 8). ***Other pins normally used for other RS-232 handshake lines, including DATA SET READY (DSR), DATA TERMINAL READY (DTR), and CARRIER DETECT (CD), are dedicated to the other electrical interfaces.***

NOTE:
When using a cable that straps DSR and DTR, you must disconnect the RS-422 pins. See Setting the RS-422/485 Disconnect Switches *on page 10.*

The RS-422 interface is typically used to improve noise immunity and/or to extend the length of cable that can be reliably used. RS-422 uses two sets of twisted pair wires, one for transmit and one for receive. Within each set of twisted pair, one wire carries the signal in positive polarity and the other in negative polarity. This configuration, often called *balanced line*, tends to cancel interference. RS-422 is normally used for point to point communications. The 2573 modules use pin 1 for RECEIVE + (Rx+) and pin 6 for RECEIVE - (Rx-). Pin 4 is used for TRANSMIT + (Tx+) and Pin 9 is used for TRANSMIT - (Tx-).

The RS-485-A interface is similar to the RS-422 interface, except that only one twisted pair is used. The single twisted pair is used for both transmitting and receiving data. The 2573 modules use Pin 4 for TRANSMIT/RECEIVE + (Tx/Rx +) and Pin 9 for TRANSMIT/RECEIVE - (Tx/Rx-).

RS-422-A and RS-485 circuits may require the use of termination resistors at the module connection. The 2573 modules provide jumpers which allow you to insert or remove termination resistance from the circuit.

NOTE:
See Appendix A for details on port configuration, cabling, and jumper settings.

The ports may be configured for baud rates of 300, 600, 1200, 2400, 4800, 9600, and 19,200 baud. If you use the module dipswitches to select the port protocol, you can set the baud rate to 1200, 2400, 9600, and 19,200 baud using the switches. See *Chapter 2. Installation*, page 9. Otherwise the baud rate and other communication parameters are set by PLC logic.

1.3. LED Indicators

Active LED

The *Active* LED will illuminate steady when the module hardware is performing properly. Should a hardware problem be detected by the diagnostic software, the LED will cycle off and on about once every two seconds. The LED will also flash about once a second when a Listen Only Modbus command sent to the 2573 while in Modbus Slave mode. The LED is off when no power is present. It may also cycle off for a few seconds when the module power is first applied.

Transmit / Receive LEDs

The module provides a pair of Transmit and Receive LEDs for each serial port which flash when a signal is present. The Transmit LED will illuminate when an output signal is present; the Receive LED will illuminate when a signal is received. During normal data communications activity the LEDs will appear to flicker at a rate which corresponds to the baud rate. The LEDs are a positive indication that the serial ports are operating and serve as a valuable problem determination tool.

1.4. Functional Overview

The CTI 2573 modules may be used for a wide variety of applications, including:
- Communicating with devices which use the Modbus protocol (2573-MOD only),
- Attaching an NITP compatible operator interface to the PLC,
- Transferring data to/from another PLC (SIMATIC® Series TI500 or 505),
- Communicating with other CTI products including another 2573 module,
- Sending command to and/or obtaining data from serial ASCII devices (such as bar code readers, scales, etc.).

The CTI 2573 modules use software routines called *Protocol Managers* to accomplish these functions. Protocol Managers create outgoing messages and process incoming messages under control of the PLC. The 2573 modules provide several different protocol managers whose functions are described below.

MODBUS Master (2573-MOD Only)

When a 2573-MOD serial port is configured to use the Modbus RTU Master or the Modbus ASCII Master protocol manager, PLC logic can be used to send queries to a Modbus slave device. The protocol manager will place the responses from the slave device in a specified V memory location.

MODBUS Slave (2573-MOD Only)

When a 2573-MOD serial port is configured to use the Modbus RTU Slave or the Modbus ASCII Slave protocol manager, it will allow Modbus master devices to read and write to PLC memory.

NITP/CAMP Slave

When a 2573-MOD or 2573-TCM2 serial port is configured to use the *CAMP/NITP Slave* protocol manager, it will support both the *Non-Intelligent Terminal Protocol* (NITP) used by SIMATIC® TI500 and 505 Series PLCs and the *Common ASCII Message Protocol* (CAMP) used by most CTI communications modules. This protocol manager acts as a slave, responding to commands from the

attached device. No PLC logic is required to implement this protocol manager unless you want to use Modem TRS/CTS handshaking. See the example in *Chapter 7. Application Examples* on page 66.

The NITP support emulates the program port of a SIMATIC® 505 series PLC. A device which can communicate using NITP (Non-Intelligent Terminal Protocol), such as an operator interface panel, can access the PLC using this port. When an NITP message containing a task code is received by the module, it retrieves the task code from the message and routes it to the PLC processor via the backplane. When the PLC processor replies to the task code, the 2573 module sends an NITP message containing the response back to the device.

CAMP (Common ASCII Message Protocol) is used by most CTI communications modules to transfer data. As compared to NITP, CAMP allows much more data to be sent in a single message. Where NITP will allow only 16 words per message, CAMP allows 256 words per message. For applications requiring large amounts of data transfer, the increased message size results in improved performance. See the section on CAMP Master below for additional information.

Task Code (NITP) Master

When a 2573-MOD or 2573-TCM2 serial port is configured to use the Task Code Master protocol manager, the PLC can *initiate* NITP messages containing task codes. Since NITP is used, the message can be processed directly by the program ports of both the SIMATIC® 505 and TI500 series PLCs. You can attach a cable from the serial port to the program port of the PLC. See the example application in *Chapter 7. Application Examples* on page 71.

This function allows the PLC to read data from or write data to another PLC (called a *remote* PLC) under control of the local PLC logic. On command from the local PLC, the 2573 module will build the task code message and send it to the remote PLC attached to the serial port. The response from the remote PLC will be processed by the protocol manager and any data retrieved will be stored in local PLC memory.

CAMP Master

When a 2573-MOD or 2573-TCM2 serial port is configured to use the CAMP Master protocol manager, the PLC can *initiate* messages using the Common ASCII Messaging Protocol format. Because CAMP allows up to 256 words of data to be sent in one message, it offers improved performance over NITP. CAMP is typically used to transfer data among CTI communications modules. For example, in an application where you need to transfer large amounts of data between two PLCs, you might configure both PLCs with a 2573 module in each rack, cable the 2573 ports together, and use CAMP to transfer the data.

General ASCII Support (GAS)

When a 2573-MOD or 2573-TCM2 serial port is configured to use the General ASCII Support (GAS) protocol manager, it can be used to communicate with a wide variety of devices which adhere to the ASCII data encoding standard. GAS allows the application developer to specify how output messages will be constructed and how input messages will be processed. PLC logic can be used to control sending and receiving device messages. The GAS protocol manager is described in a separate reference manual. See the *General ASCII Support Protocol Manager Reference Manual* for comprehensive information.

CHAPTER 2. INSTALLATION

The installation of the Module consists of the following steps:

1. Planning the installation,
2. Unpacking and configuring the module,
3. Physical installation,
4. Connecting cables,
5. Checking the module operation.

2.1. Installation Planning

The CTI 2573-MOD and 2573-TCM2 are single wide modules which draws 6.0 watts at 5 VDC from the backplane. You should ensure that you have adequate space and power available for the module.

The serial ports on the 2573 module provide electrical interfaces for RS-422-A, RS-485, and a subset of RS-232-C. The electrical interface used is determined by how the pins on the DB-9 connector are connected to the cable. You should determine which interface you wish to use (typically determined by the capability of the attached devices) and construct/obtain cables which implement the desired interface. See Appendix A for port configuration and cabling details.

The serial ports can be configured for a variety of different protocols. Before installation, you should review your requirements and determine which protocols you wish to use. If you are using the Modbus, CAMP, or NITP protocols, refer to the appropriate chapter in this manual. If you are using the General ASCII Support (GAS) protocol manager, you should read the material in the *General ASCII Support Protocol Manager Reference Manual*.

2.2. Unpacking the Module

Open the shipping carton and remove the special anti-static bag that contains the module. After discharging any static build-up, remove the module from the static bag. Do not discard the static bag. Always use this bag for protection against static damage when the module is not inserted into the I/O base.

<div style="border:1px solid black; padding:10px;">

CAUTION:
The components on the 2573 module can be damaged by static electricity discharge. To prevent this damage, the module is shipped in a special anti-static bag. Static control precautions should be followed when removing the module from the bag and when handling the printed circuit card during configuration

</div>

2.2.1. Setting the 2573 Switches

There are four switchblocks on the 2573 module, one for each serial port. Each switchblock contains a set of eight switches. You may use the switches on the 2573 to select the baud rate, handshaking, and protocol for each serial port. Alternately, you may set the switches so that PLC logic is used to select the above.

Figure 2 illustrates the location of the switches.
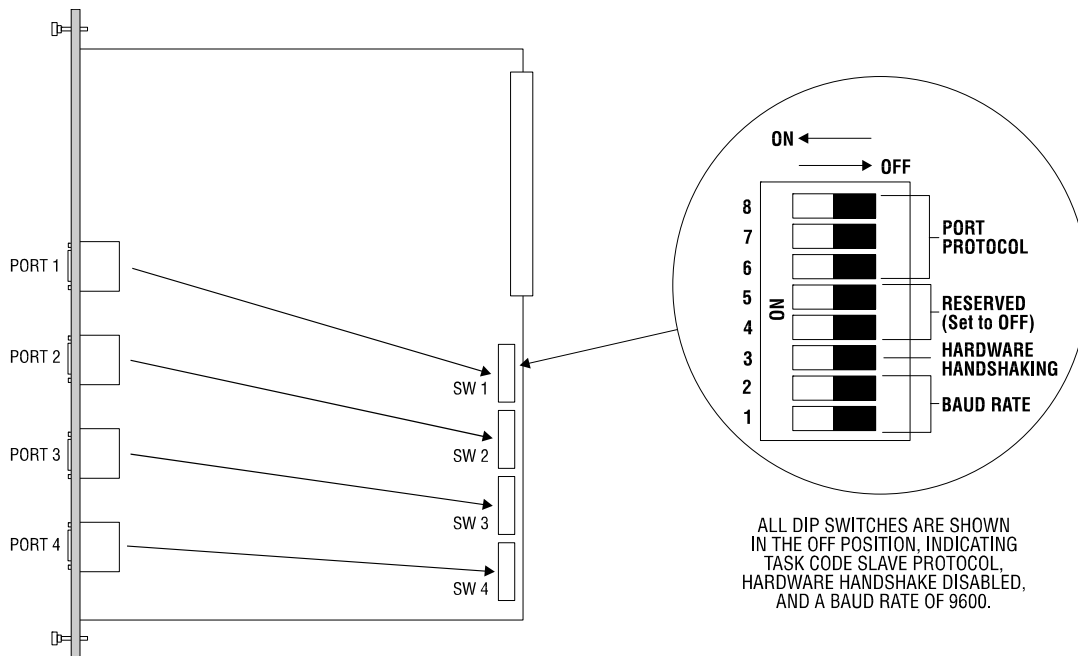
Figure 2. 2573 Switch Locations

NOTE:
The switch settings are read by the module at startup only. If you change the switch settings, you must reset the module by removing and restoring power before the new settings will take effect.

NOTE:
Switches on some versions may be labeled OPEN and CLOSED rather than OFF and ON. OPEN is equivalent to OFF and CLOSED is equivalent to ON.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Serial Port Protocol

Switches 6, 7, and 8 on each switch block are used to set the protocol used by the port. You have the following options:

Protocol	Description	Use
NITP/CAMP Slave	Supports both NITP (Non-Intelligent Terminal Protocol) and CAMP (Common ASCII Message Protocol). This protocol manager responds to commands sent from another device acting as a master.	Used to access PLC memory. NITP slave emulates a PLC programming port. Typically used to connect operator interface devices, MMI software, or PCs running PLC programming software. CAMP supports higher density memory transfer and is used between CTI modules. It is also used for special module specific commands.
NITP Only Slave	Similar to the function above except that is supports only NITP (not CAMP).	Used when you need strict NITP compatibility with the the SIMATIC® 505 program port. This is usually not required.
Task Code Master	Allows PLC logic to send task codes using NITP (Non-Intelligent Terminal Protocol).	Used to transfer data between PLCs. You can attach to the remote PLC via the program port.
Loopback	Loopback generates a series of ASCII characters which may be directed back to the serial port via a loopback connector.	Used to Test serial ports and cables. See Appendix A for construction of loopback connectors.
PLC Select	Allows the PLC logic to select the port protocol.	Used to load other port protocol managers including Modbus Master and Modbus Slave protocol managers.

See the figure below for switch settings used for protocol selection:

Port Protocol	Switch 6 Position	Switch 7 Position	Switch 8 Position
CAMP/NITP Slave	OFF	OFF	OFF
NITP Only Slave	OFF	OFF	ON
Task Code Master	OFF	ON	OFF
Loopback	ON	OFF	OFF
PLC Select	ON	ON	ON
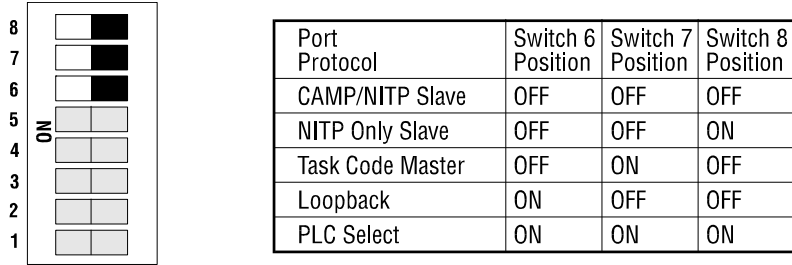
Figure 3. Port Protocol Settings

All unused positions will select the loopback function.

> *NOTE:*
> *If you are not using the port for other purposes, you should set the switches to select the NITP/CAMP Slave protocol manager (Switch 6, 7, and 8 in OFF position). This is the factory setting.*

> *NOTE:*
> *When you use the module switches to select the Task Code Slave protocol or Task Code Master protocol, the baud rate is determined by switch settings. Other communications parameters such as data bits, parity, and stop bits default to those used by the SIMATIC® 505 program port.*

Serial Port Baud Rate

You can individually set the default baud rate settings for each serial port. The diagram below applies to all switchblocks (SW1-SW4).
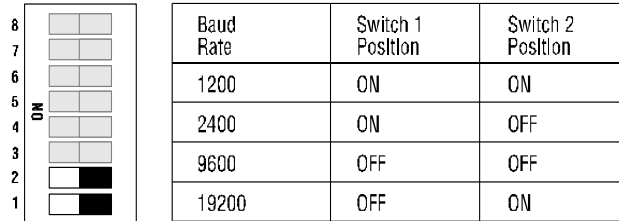
Figure 4. Baud Rate Settings

Hardware Handshaking

The figure below shows the switch settings for hardware handshake.
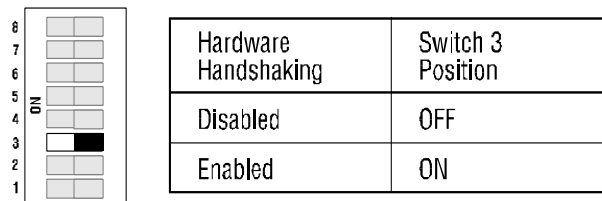
Figure 5. Handshake Settings

2.3. RS-422/485 Disconnect Switches (Board Revision D and above)

In order to use a standard 9-pin 'D' shell connector, the 2573 routes the lines used for RS-422 and RS-485 communications to pins normally used for the following RS-232 hardware handshaking lines:

- Data Carrier Detect (DCD)
- Data Terminal Ready (DTR)
- Data Set Ready (DSR).

This arrangement can cause communications problems when you use a cable that straps these handshake lines together (such as a typical PLC programming cable). Revision D and above of the 2573 printed circuit board contains a switch for each port which allows you to disconnect the RS-422 and RS-485 lines from the 9 pin connector. Putting the switch in the 'OFF' (disconnected) position allows you to use unmodified PLC programming cables with the 2573 module.

For board revisions C and below, you will need to construct new cables or build an adapter cable to eliminate the problem. Appendix A describes how to construct these cables.

Setting the RS-422/485 Disconnect Switches

The RS-422/485 Disconnect switches are located adjacent to the ports they serve, as seen in the following figure. When the interface for an individual port is RS-232, the associated Disconnect switch may be left in the 'ON' (connected) position, unless transition problems arise from that setting. For RS-422/485 operation, the Disconnect switch *must* be in the 'ON' (connected) position.
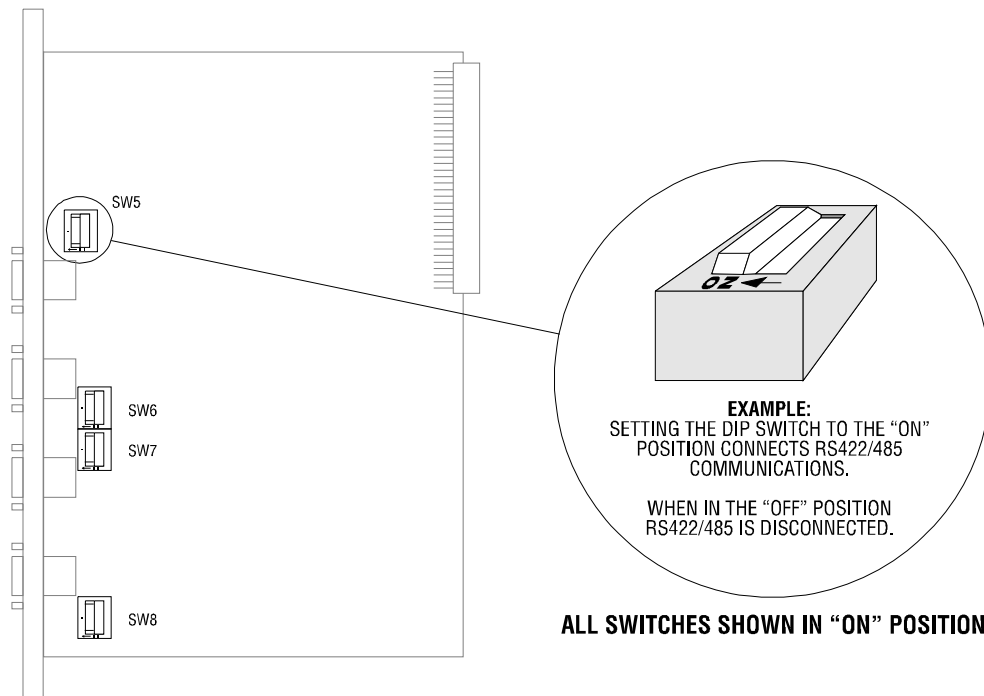
Figure 6. 2573 RS-422/485 Disconnect Switch Locations

CTI 2573-MOD2/TCM2 Installation and Operation Guide

2.4. Termination Resistor Jumpers

RS-422C and RS-485 electrical interfaces may use termination resistors to reduce circuit reflections and to improve signal quality. For RS-422A, the termination resistor is optional. It is usually not used at baud rates below 20kb. For RS-485, termination resistance is required on both ends of the line. See the documentation accompanying your device for particular installation information.

The 2573 module uses a set of jumpers to insert or remove the termination resistance from the circuit. The module is shipped from the factory with the jumpers positioned so that the termination resistance is disabled (not installed in the circuit). If you need to change the factory setting, Appendix A, page 109 illustrates the jumper locations and configurations.

> *NOTE:*
> *Jumpers for RS-422 and RS-485 are shipped in the disabled position. If you are using an RS-485 circuit you will probably need to change the setting. See Appendix A.*

2.5. Physical Installation

Inserting the Module into the I/O Base

Ensure that power to the base is off. Hold the top and bottom of the bezel and slide the module carefully into the slot, pushing it all the way into the base. If you have inserted the module correctly, you will feel a slight increase in resistance as the module mates with the base backplane connector. Once the module is fully seated in the slot, tighten the captive screws at the top and bottom to hold the module in place. To remove the module from the I/O base, remove power from the base, loosen the captive screws, then remove the module. Take care not to damage the connector at the back of the module when inserting or removing the module.

Power Up

First turn on the base power supply. If the diagnostics on the Model 2573 detect no problems, the ACTIVE indicator will illuminate. If the indicator does not illuminate or the module continually resets, the module has detected a component failure. See *Chapter 9. Troubleshooting* of this manual for troubleshooting information.

Checking PLC Login

Next, check that the module is configured in the memory of the PLC. The 2573 logs in as a special function module with 2 WX and 6 WY words. To view the PLC I/O configuration chart refer to your PLC programming software programming manual.

```
I/O MODULE DEFINITION FOR CHANNEL  . . . 1   BASE . . . . . . . 00

I/O                        NUMBER OF BIT AND WORD I/O               SPECIAL
SLOT   ADDRESS      X              Y            WX          WY       FUNCTION
01     00001        00             00           02          06       YES
02     00000        00             00           00          00       NO
..     ......       ..             ..           ..          ..       ...
15     00000        00             00           00          00       NO
16     00000        00             00           00          00       NO
```

Figure 7. Sample I/O Configuration

In this example, the Model 2573 module is installed into slot 1 on I/O base 0. For your particular installation, look in the chart at the slot address occupied by the module in your configuration. If the WX and WY values are the same as those shown in the example above and Special function = YES, then the module is logged into the PLC memory and is ready for operation. If the line is blank or erroneous, re-check the module to ensure that it is firmly seated in the I/O base slot. Generate the PLC I/O configuration chart again by reading the I/O base. If the problem persists, contact your distributor or contact CTI at 1-800-537-8398.

2.6. Connecting Cables

Connect the serial cabling from the serial port(s) to the device(s) with which you wish to communicate. Your cables should have been wired to select the electrical interface you wish to use. See Appendix A for port pin-out and cable wiring examples.

2.7. Module Checkout

Apply power to the module. After a few seconds the Active LED should remain lit. If the Active LED cycles off and on, the module diagnostics may have detected a hardware error. If the Active LED is off, either power is not applied or module hardware has failed. See *Chapter 9. Troubleshooting* for more information.

If you wish to check out each serial port, you can set the module dipswitches for the loopback function. If you have selected the loopback setting for a port, you should see the XMT LED for the port flash, indicating data is being sent to the port. If you install a loopback connector (see *Appendix A*), you should see the XMT and RCV indicators flash.

CHAPTER 3. OPERATION

3.1. General Module Operation

Under normal conditions, the red Active LED should be lit. This indicates that the module is receiving power and that no hardware problems have been detected by the module diagnostic software. If the Active LED is not lit or if it is blinking, please refer to *Chapter 9. Troubleshooting* for more information.

3.2. Protocol Managers

The 2573 uses software routines called Protocol Managers to send and receive data via the serial ports. Ports configured via module dip switches as a CAMP/NITP or NITP slave do not require PLC logic in order to work. If you are using other protocol managers, you will need to use PLC logic to control the operation.

The following chapters describe the capabilities and implementation of the 2573 Protocol Managers.

3.3. PLC Command Interface

The 2573-MOD and 2573-TCM2 modules provide a special command interface which allows you to use standard SIMATIC® 505 PLC ladder logic to send messages and process responses. The interface consists of two structures, the module WX/WY words and the Command Block. The WX/WY words are used to control command execution and the Command Blocks contain the command parameters. Please refer to *Appendix D. PLC Command Interface* for a more complete description of the command block, WX/WY usage, and command timing diagrams.

CHAPTER 4. MODBUS SLAVE

4.1. Before Proceeding

> *NOTE:*
> *The use of the Modbus Slave protocol managers requires a CTI 2573-MOD module. The 2573-TCM2 module does not support these protocol managers.*

An understanding of the Modbus Protocol is required in order to use the information contained within this chapter. This protocol is fully specified in the Modicon Modbus Protocol Reference Guide (PI-MBUS-300 Rev E). This chapter assumes that you have access to this document or the equivalent information. See *Appendix E. Modbus Reference* for additional information about Modbus.

In addition, Enron has established a de-facto standard for transferring 32 bit words and floating point numbers. The 2573 Modbus Slave protocol managers will support this extension when the appropriate option bit is set. Please see *Appendix E. Modbus Reference* for more information about Modbus Enron extensions.

The command block illustrations in this chapter show their contents in both Hexadecimal and Decimal (integer) format. You can enter and display values in either format using your PLC programming software. The hexadecimal presentation format is usually used to display the error word.

> *NOTE:*
> *In Command Block Descriptions, a boldface entry in the value column designates a required value. Other entries are recommended values.*

4.2. Introduction

When operating as a Modbus slave, the 2573 module will accept a Modbus query from a master device. As shown below, the Modbus Slave Protocol Manager will interpret the query and send a request to the PLC to access PLC memory. Once the PLC acts upon this request and returns a response back to the module, it will build the appropriate Modbus response and send it back to the master. Both Modbus RTU and Modbus ASCII protocols are supported.

Using Modbus Protocol with SIMATIC® 505 PLCs does necessitate taking differences in conventions and architectures into account. These include differences in variable types, addressing schemes, and nomenclature.
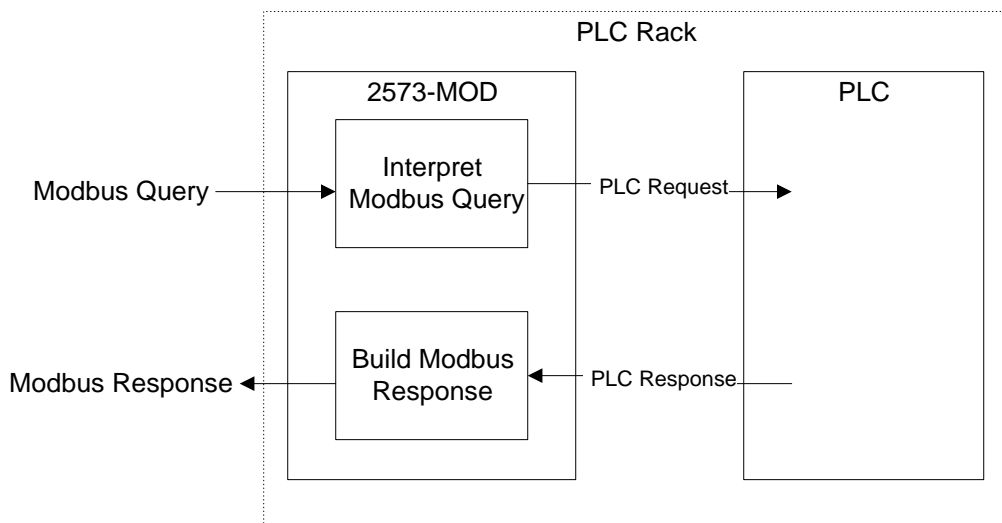
Figure 8. Modbus Slave Command Flow

4.3. Differences in Variable Names

The terminology for variables differs between SIMATIC® 505 PLCs and Modicon PLCs. Below is a table showing Modbus variables and the equivalent variables that are used by the SIMATIC® 505 PLC.

Modbus Variable	505 PLC Equivalent Variable
Coil	Output (Y) or Control Relay (C) See Note below.
(Discrete) Input	X
Holding Register	V Memory location
Input Register	WX

NOTE:
Whether coils refer to Y or C variables depends upon the option selected in the Create Connection Command Block. When using the Enron extensions option, you can select only C variables.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

4.4. Relative Addressing

The relative addressing base used by the MODNIM differs from the Modbus standard. The Modbus Slave protocol manager supports either addressing base. You may select the relative address base to be used by setting the *Compatibility Option* bit in the CREATE CONNECTION Command Block. You may choose either Modbus Compatibility or MODNIM Compatibility. The two addressing methods are explained below.

Modbus Compatible Addressing

Data addresses in Modbus compatible messages are referenced to zero. The first occurrence of a data item is addressed as item number zero. For instance, when using Function Code 02 to read discrete inputs, X1 would be referenced as 0000 in the data address field of the Modbus message. Similarly, when using Function Code 03 to read a holding register, V100 would be referenced as 0x0063 (decimal value 99).

MODNIM Compatible Addressing

Data addresses in MODNIM compatible messages are referenced to one. The first occurrence of a data item is addressed as item number one. For instance, when using Function Code 02 to read discrete inputs, X1 would be referenced as 0001 in the data address field of the Modbus message. Similarly, when using Function Code 03 to read a holding register, V100 would be referenced as 0x0064 (decimal value 100).

Determining which Addressing Method to Use

It is highly desirable that all devices on a Modbus network use the same relative addressing method. This is especially important if you intend for the master to issue broadcast messages to write all slaves on the network.

Your choice of relative addressing will probably depend upon the method used by other devices on the network:
- If the other devices (masters and/or slaves) use the Modbus convention, you will want to choose Modbus Compatibility.
- If the module is on a network with MODNIM modules or a master devices expects you to emulate MODNIM slave, you should choose MODNIM compatibility.

Unless you explicitly require MODNIM compatibility, you should choose the Modbus Compatibility option.

4.5. Bit Ordering for Discrete Items

Certain Modbus function codes read or write multiple discrete elements such as inputs or coils. These elements are packed into the data bytes of the Modbus message. The least significant bit (LSB) of the first data byte contains the first element addressed in the query. Additional discrete elements are placed in order up to the most significant bit (MSB) of this byte. The pattern repeats from 'low order' to 'high order' bits in subsequent bytes. If the returned item quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte).

EXAMPLE: Below is an example of what might be returned in an RTU response message from a query for 10 items from the slave PLC. The first requested element is located in the LSB of the first Data Field byte. The "1" in the bit location shows that the element in ON. Elements 9 and 10 also happen to be ON. Since the requested number of elements is not a multiple of eight, the six most-significant bits of the second byte are padded with zeros.

1st Byte									2nd Byte							
1	0	0	1	0	1	1	1		0	0	0	0	0	0	1	1

Item #8   Item #1   Item #10   Item #9
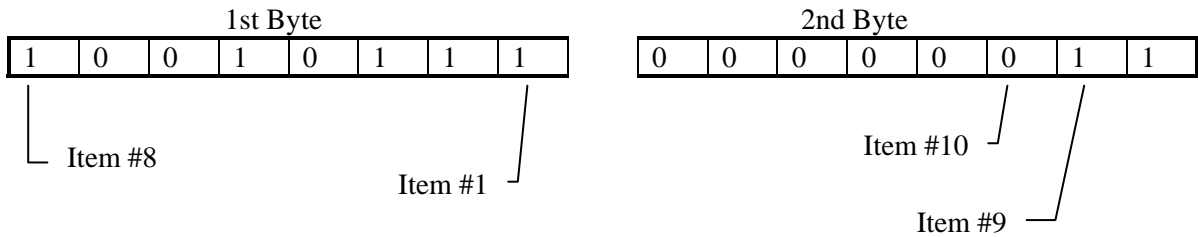
The above message would correspond to the hex value 9703 when transmitted. Thus as ASCII message representing the above would contain the four characters 9 7 0 3 (0x39 0x37 0x30 0x33).

4.6. Supported Function Codes

The Modbus Slave Protocol manager supports the following Modbus Function Codes. See your Modbus documentation for a complete description of these commands.

FC (dec)	FC (hex)	Function Codes	505 Variable	Comments
01	01	Read Coil Status	Y or C*	Max number of coils = 2000
02	02	Read Input Status	X	Max number of inputs = 2000
03	03	Read Holding Register	V	Max = 125
04	04	Read Input Register	WX	Max = 125
05	05	Force Single Coil	Y or C*	
06	06	Preset Single Register	V	
07	07	Read Exception Status	Y or C*	Reads coils 1-8
08	08	Diagnostics	N/A	See Diagnostic Sub-functions
11	0B	Fetch Comm. Event Counter	N/A	
15	0F	Force Multiple Coils	Y or C*	Max = 800
16	10	Preset Multiple Registers	V	Max = 100
17	11	Report Slave ID	N/A	See next section

* Selection of Y or C depends upon the option selected in the Create Connection Command Block.

Function Code 17 Responses

Function Code 17 (Hex 11) will return a Slave ID according to the table below:

PLC Description	Slave Address Returned
SIMATIC® TI525	20
SIMATIC® 545-1101/-1102/-1104	45
SIMATIC® 555	55
SIMATIC® 565	60
SIMATIC® 575	75
SIMATIC® 545-1103	80

4.7. Diagnostic Subfunctions

Modbus Function Code 08 (Diagnostics) in the table above, provides a set of diagnostic functions which may be used to determine status or to set certain operational modes. The function uses a two-byte subfunction code field in the query to define the type of function to be performed. Below is a list of supported diagnostic sub-functions.

Code (Dec)	Diagnostic Subfunction	Returns
00	Return Query Data	Modbus Standard
01	Restart Communications Option	Modbus Standard
02	Return Diagnostic Register	2573-MOD Unique
03	Change ASCII Input Delimiter (N/A to Modbus RTU)	Modbus Standard
04	Force Listen Only Mode	Modbus Standard *
10	Clear Counters and Diagnostic Register	Modbus Standard
11	Return Bus Message Count	Modbus Standard
12	Return Bus Communication Error Count	Modbus Standard
13	Return Bus Exception Error Count	Modbus Standard
14	Return Slave Message Count	Modbus Standard
15	Return Slave No Response Count	Modbus Standard
16	Return NAK Count	Returns 0000
17	Return Slave Busy Count	Returns 0000
18	Return Slave Character Overrun Count	Modbus Standard
19	Return IOP Overrun Count	Returns 0000
20	Clear Overrun Counter and Flag	Modbus Standard

* Placing the 2573 in Listen Only mode will cause the module's Active LED to blink

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Non-Standard Diagnostic Subfunction Responses

Because of the differences in SIMATIC® 505 PLC architecture, the responses to certain diagnostic subfunctions varies from the Modicon standard. These non-standard responses are listed below.

Subfunction 02 - Return Diagnostic Register

The Modbus response message will return a 16-bit diagnostic status word. This word will be sent as two bytes in the data field for an RTU message. For ASCII this word will be converted to four ASCII bytes.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

Where,
A = Fatal error
B = Other hardware error
C = EEPROM card RAM download error
D = Unidentified board failure
E = Abnormal power loss
F = Scan overrun error
G = I/O fatal error
H = Scan diagnostic failure
I = ROM diagnostics failure
J = RAM diagnostics failure
K = Illegal Op code
L = Dynamic program memory diagnostic
M = Watchdog time-out error
N = O. S. fatal error
O = Program RAM parity error
P = O. S. RAM parity error

Subfunction 16 - Return NAK Count

Since NAK count is not implemented, the value returned will always be 0000.

Subfunction 17 - Return Slave Busy Count

Since Slave Busy count is not implemented, the value returned will always be 0000.

Subfunction 19 - Return IOP Overrun Count

Since IOP Overrun count is not implemented, the value returned will always be 0000.

4.8. Exception Codes

Exception codes are used in slave exception responses to indicate the nature of the problem that a slave encountered in processing a query from a master. The exception code returned by the Modbus Slave protocol manager will not only depend upon the error that occurred, but also whether you are using Modbus or MODNIM exception codes. You may select which set of exception codes to use by setting the *Compatibility Option* bit in the CREATE CONNECTION Command Block. If Modbus compatibility has been chosen, then the exception code returned will be taken from the Modbus Exception Code table below. If MODNIM compatibility has been chosen, then the exception code returned will be taken from the MODNIM Exception Code table.

Modbus Exception Codes

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the slave.
02	Illegal Data Address	The data address received in the query is not an allowable address for the slave.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for the slave.
04	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action.
08	Memory Parity Error	The slave attempted to read extended memory, but detected a parity error in the memory.

MODNIM Exception Codes

Code	Meaning
00	The function code received in the query is not supported by the slave.
01	The data type is not defined in the slave device.
02	The data address received in the query is not an allowable address for the slave.
0B	An unrecoverable error occurred while the slave was attempting to perform the requested action.
10	The number of locations in the query data field exceeds the allowable range for the slave.
15	The function is not allowed while the slave is in Write Protect mode.
19	The requested starting location plus the number of items requested leads to locations that are out of range for the slave device.
1D	The number of locations to access is zero.

4.9. Modbus RTU Slave Command Block

Create Connection Command - Modbus RTU Slave

Before the 2573 can act as a Modbus RTU slave, the CREATE CONNECTION command must be executed. This command starts a copy (instance) of the Modbus RTU Slave Protocol Manager and associates this copy with a physical port (either port 1 or 3). The CREATE CONNECTION command block for Modbus RTU slave is shown below.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	0000	0
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221- 19299)		
3	Protocol Manager Number (Modbus RTU Slave)	**002A**	**42**
4	Physical Port Number (1 or 3)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (RTU requires 8 bits/character)	**0008**	**8**
7	Parity (0=None, 1=Odd, 2= Even)		
8	Stop Bits (0 or 1)	0000	0
9	Handshake (0=None, 4=RS-485, 6=Modem RTS/CTS)		
10	Startup Option Bits (see description below)		
11	Modbus Slave Address (1-247)		
12	RTU End-of-Message Interval - in milliseconds (5-100)	*0000*	*0*
13	Enron Extensions: Control Relay Address of Modbus Coil 1000	0000	0
14	Enron Extensions: V Memory Address of Modbus Register 7000 (Most Significant Word) Note: If the starting V memory location is greater than 65535, enter the V memory location here as a double word.	0000	0
15	Enron Extensions: V Memory Address of Modbus Register 7000 (Least Significant Word) Note: If the V memory location is less than 65535, enter 0 into offset 14 and enter the V memory location here as an integer value.	0000	0

Offset 0   *Command Error Word* - If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command Code for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**3** for port 3).

Offset 3   *Protocol Manager Number* - Set to 42 (Hex 002A) to select the Modbus RTU Slave protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1 or 3) you wish to use.

Offset 5   *Baud Rate* - The baud rate must match the setting for the master device.

Offset 6   *Bits per Character* - Modbus RTU standards requires 8 bits per character.

Offset 7   *Parity* - The parity setting must match the setting for the master device.

Offset 8   *Stop Bits* - Use 0 for the Modbus default. This will automatically set the number of stop bits according to the parity setting. For no parity, stop bits will be set to 1; for even or odd parity, stop bits will be set to 0. Setting this value to 1 explicitly specifies 1 stop bit; you might need to set this bit to 1 with certain modem configurations.

Offset 9   *Handshake*- Set this to 4 if you are using an RS-485 line or if you wish to raise the RTS line before sending data. Set this to 6 to enable RTS/CTS modem flow control. When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this to 0.

Offset 10   *Option Bits* - These are used to select special protocol manager options. The diagram below shows how the 16 bit word is defined. Bit 1 is the Most Significant Bit while Bit 16 is the Least Significant Bit.

MSB   LSB

Bits 1 – 10	Bit 11	Bit 12	Bit 13	Bit 14	Bit 15	Bit 16

Bit Description	Bit(s)	Definitions
Unused - Reserved	1 - 10	These bits are reserved for future use. Set these bits to 0.
Enron Protocol Used	11	This bit determines whether to use the Enron extensions. 0 = Use Standard Modbus 1 = Use Enron Extensions
End of Message Interval Timing	12	Determines whether to use a non-standard time interval to indicate the end of a message: 0 = Use standard (3.5 character time) end of message timing 1 = End of Message Interval defined by user. Use the value specified in offset 12 of the command block.
Coil Mapping	13	Determines whether references to coils by Modbus master will refer locally to Y (outputs) or C (control relays): 0 = coil addresses refer to C addresses 1 = coil addresses refer to Y addresses
Write Protect	14	Determines whether Modbus can write to local PLC memory 0 = Not Protected. PLC memory can be written to 1= Write Protected. PLC memory cannot be written to
Compatibility Option	15	See *4.4. Relative* Addressing on page 16 and *4.8.* Exception Codes on page 22. 0 = Modbus compatible, base addresses start at 0, Use Modbus Exception Codes. 1 = MODNIM compatible, base addresses start at 1. Use MODNIM Exception Codes.
Modem Used	16	Set this bit to 1 when you are using a modem on this port.

                                               CTI 2573-MOD2/TCM2 Installation and Operation Guide

Offset 11   *Modbus Slave Address* - Used to set the Modbus Slave Address of the module (1-247). The module will process only messages matching this user defined slave address and broadcast messages (slave address 0).

Offset 12   *End of Message Interval* - Used to specifically override the standard RTU end of message timing (3.5 character times). Enter a value from 5 -100 milliseconds. Bit 13 in the startup Option Bits must be set before this value will be examined by the protocol manager. The interval applies to all messages received by the protocol manager. Transmitted messages will continue to use standard Modbus inter-character timing.

This option may be useful when communicating via data links that cannot maintain the constant bit rate required by Modbus RTU. For example, some modems, especially radio modems, may not be able to maintain a constant bit rate. Thus standard RTU will prematurely terminate a message, determine that a protocol error has occurred, and ignore the message. This feature can only be used for rates above 300 baud.

Ensure that the value you select is greater than 3.5 character times or your results will be unpredictable. See the table below for minimum times in milliseconds for each baud rate.

Baud Rate	Minimum time (ms)
300	Feature not supported at this baud rate.
600	64
1200	32
2400	16
4800	8
9600	5
19200	5

Offset 13   *Starting Coil Address* – When the Enron extensions are being used, this value specifies
                   the address of the SIMATIC® 505 control relay (C) element that will be mapped to the
                   Modbus Coil 1000. This applies to Modbus Function Code 01 (Read Coil Status),
                   Function Code 05 (Force Single Coil) and Function Code 15 (Force Multiple Coils). For
                   example, an entry of 2000 specifies that Modbus Coil 1000 references C2000.

Offset 14-15   *Modbus Register 7000 V memory Address* – When the Enron extensions are being used,
                   this specifies the V Memory address that will contain the Modbus register 7000. This
                   applies to Modbus Function Code 03 (Read Holding Register), Function Code 06 (Preset
                   Single Register) and Function Code 16 (Preset Multiple Register). An entry of 10000
                   specifies that Modbus Register 7000 references V Memory 10000. V memory addresses
                   greater than 65535 can be entered in offset 14 as a double word. For V memory addresses
                   less than 65535, enter a 0 into offset 14 and enter the starting V memory location in
                   offset 15 as an integer value.

NOTE:
All Modbus register addresses will be translated to V memory addresses based on this starting point.
Register 5000 will start in a V memory address that is 4000 below the Modbus Register 7000 address,
since it consists of 2000 long integers. Register 3000 will start at 2000 less than the V memory address of
Register 5000, since it consists of short (16 b it) integers. See Appendix E. Modbus Reference.

Once the command block has been entered, you must create the PLC logic which causes the module to
read and execute the CREATE CONNECTION command block. If you are interested only in understanding
how to setup the Modbus RTU protocol manager, then skip the next section about Modbus ASCII and
jump to *4.11. Startup Ladder Logic* on page 30.

4.10. Modbus ASCII Slave Command Block

Create Connection Command

Before the 2573 can act as a Modbus ASCII slave, the CREATE CONNECTION command must be initiated. This command starts a copy (instance) of the Modbus ASCII Slave Protocol Manager and associates this copy with a physical port. The CREATE CONNECTION command block is shown below.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	0000	0
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 -- 19299)		
3	Protocol Manager Number (Modbus ASCII Slave)	**002B**	**43**
4	Physical Port Number (1, 2, 3, or 4)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (7 or 8)		
7	Parity (0=None, 1=Odd, 2= Even)		
8	Stop Bits (1 or 2)		
9	Handshake (0=None, 1=Software, 2=Hardware, 4=RS-485, 6=Modem RTS/CTS)		
10	Startup Option Bits (See definitions below)		
11	Modbus Slave Address (1-247)		
12	Reserved for Future Use (Set to 0)	**0000**	**0**
13	Enron Extensions: Control Relay Address of Modbus Coil 1000	0000	0
14	Enron Extensions: V Memory Address of Modbus Register 7000 (Most Significant Word) Note: If the starting V memory location is greater than 65535, enter the V memory location here as a double word.	0000	0
15	Enron Extensions: V Memory Address of Modbus Register 7000 (Least Significant Word) Note: If the V memory location is less than 65535, enter 0 into offset 14 and enter the V memory location here as an integer value.	0000	0

Offset 0   *Command Error Word* - If a connection error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**2** for port 2).

Offset 3   *Protocol Manager Number* - Set to 43 (Hex 002B) to select the Modbus ASCII Slave protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1-4) you wish to use.

Offset 5   *Baud Rate* - The baud rate must match the setting for the master device.

Offset 6   *Bits per Character* - Modbus ASCII standards require 7 bits per character.

Offset 7   *Parity* - The parity setting must match the setting for the master device.

Offset 8   *Stop Bits* - This must match the characteristics of your master device. The Stop Bits setting is determined by the Parity setting. If you chose even or odd parity, the number of stop bits should equal 1. If no parity is selected, the number of stop bits is set to 2.

Offset 9   *Handshake*- If you are using an RS-485 (two wire) electrical interface, set this value to 4. Set this to 6 to enable RTS/CTS modem flow control (*requires firmware Version 7.0 and above*). When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this value to match your device requirements.

Offset 10   *Option Bits* - These are used to select special protocol manager options. Settings are determined by the value of individual bits within the 16 bit word. The diagram below shows how the 16 bit word is defined. Bit 1 is the Most Significant Bit while Bit 16 is the Least Significant Bit.

MSB   LSB

Bits 1 – 10	Bit 11	Bit 12	Bit 13	Bit 14	Bit 15	Bit 16

Bit Description	Bit(s)	Definitions
Unused - Reserved	1 - 10	These bits are reserved for future use. Set bits to 0.
Enron Protocol Used	11	This bit determines whether to use the Enron extensions. 0 = Use Standard Modbus 1 = Use Enron Extensions
Unused - Reserved	12	This bit is reserved for future use. Set the bit to 0.
Coil Mapping	13	Determines whether references to coils by Modbus master will refer locally to Y (outputs) or C (control relays): 0 = coil addresses refer to C addresses 1 = coil addresses refer to Y addresses
Write Protect	14	Determines whether Modbus can write to local PLC memory 0 = Not Protected. PLC memory can be written to 1= Write Protected. PLC memory cannot be written to
Compatibility Option	15	See *4.4. Relative Addressing* on page 17 and *4.8. Exception Codes* on page 22. 0 = Modbus compatible, base addresses start at 0, Use Modbus Exception Codes. 1 = MODNIM compatible, base addresses start at 1. Use MODNIM Exception Codes.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Bit Description	Bit(s)	Definitions
Modem Used	16	Set this bit to 1 when you are using a modem on this port.

Offset 11   *Modbus Slave Address* - Used to set the Modbus Slave Address of the module (1-247). The module will only process broadcast messages (slave address 0) or messages matching this user defined slave address.

Offset 12   *Unused* - Reserved for future use. Set to 0.

Offset 13   *Starting Coil Address* – When the Enron extensions are being used, this value specifies the address of the 505 control relay (C) element that will be mapped to the Modbus Coil 1000. This applies to Modbus Function Code 01 (Read Coil Status), Function Code 05 (Force Single Coil) and Function Code 15 (Force Multiple Coils). For example, an entry of 2000 specifies that Modbus Coil 1000 references C2000.

Offset 14-15   *Modbus Register 7000 V memory Address* – When the Enron extensions are being used, this specifies the V Memory address that will contain the Modbus register 7000. This applies to Modbus Function Code 03 (Read Holding Register), Function Code 06 (Preset Single Register) and Function Code 16 (Preset Multiple Register). An entry of 10000 specifies that Modbus Register 7000 references V Memory 10000. V memory addresses greater than 65535 can be entered in offset 14 as a double word. For V memory addresses less than 65535, enter a 0 into offset 14 and enter the starting V memory location in offset 15 as an integer value.

NOTE:
All Modbus register addresses will be translated to V memory addresses based on this starting point.
Register 5000 will start in a V memory address that is 4000 below the Modbus Register 7000 address,
since it consists of 2000 long integers. Register 3000 will start at 2000 less than the V memory address of
Register 5000, since it consists of short (16 b it) integers.
See Appendix E. Modbus Reference.

4.11. Startup Ladder Logic

To start the Modbus Slave Protocol Manager using the information stored in the command block in V memory, some PLC logic is required. The logic places the starting address of the command block into one of the module's four Command Slots. It then sets the corresponding CMD Trigger and CMD Mode bits in WX2 to execute the command. See the following example. See *Appendix D* for details on the PLC command interface.

Example:

In the example ladder logic below, the Create Connection command block starts in V memory location 500. The module is assumed to be logged in starting at WX1. Command Slot 1 is used to trigger and monitor commands to the module.

When the program is first started, C1 will be in the off state and Rung 1 will place the V memory address
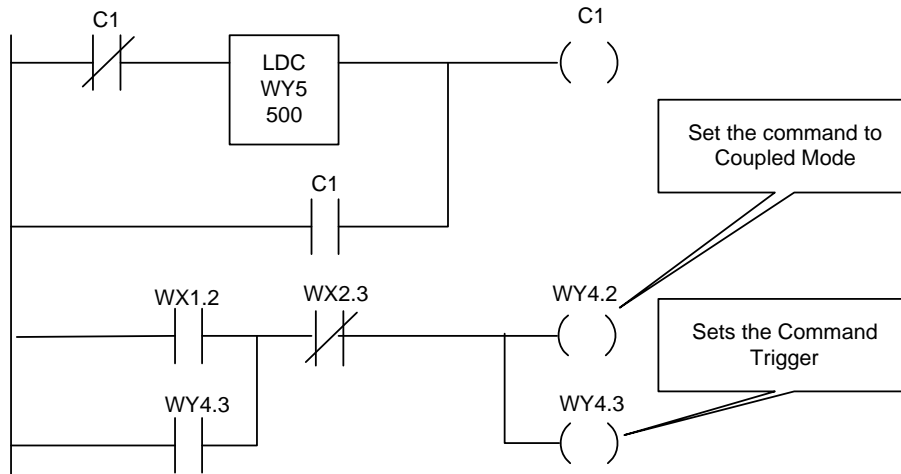
of the command block into the module's Command Slot 1 (WY5). Once C1 has been set, this function will be bypassed on subsequent scans.

When WX1.2 is on (which indicates that the module serial ports are not configured) and the Command Busy off (WX2.3), Rung 2 sets the Command Trigger (WY4.3) and the Coupled Mode bit (WY4.2). Setting WY4.2 on informs the module to execute this command only once per command trigger transition. WY4.3 is used to hold the trigger on until Command Busy (WX2.3) comes on, indicating that the module is executing the command. Once WX2.3 is on, the Command Trigger and Command Mode bits are turned off.

CHAPTER 5. MODBUS MASTER

5.1. Before Proceeding

> *NOTE:*
> *The use of the Modbus Master protocol managers requires a CTI 2573-MOD module. The 2573-TCM2 module does not support these protocol managers.*

An understanding of the Modbus Protocol is required in order to use the information contained within this chapter. This protocol is fully specified in the Modicon Modbus Protocol Reference Guide (PI-MBUS-300 Rev E). This chapter assumes that you have access to this document or the equivalent information. See *Appendix E. Modbus Reference* for additional information about Modbus.

The command block illustrations in this chapter show their contents in both Hexadecimal and Decimal (integer) format.   You can enter and display values in either format using your PLC programming software. The hexadecimal presentation format is usually used to display the error word.

> *NOTE:*
> *In Command Block Descriptions, a boldface entry in the value column designates a required value. Other entries are recommended values.*

5.2. Introduction

When operating as a Modbus master, all messages sent from the module are initiated by PLC logic. The Modbus Master protocol manager will build the output message, called a *Query*, based on the data you supply in a V memory block for the address, function, and data fields. It will then calculate the applicable error check field and send the message via the serial port.

When the response is returned, the protocol manager will validate the check characters, the slave address, and the response function code. Finally, it will then place the response data in a user specified PLC V memory block.
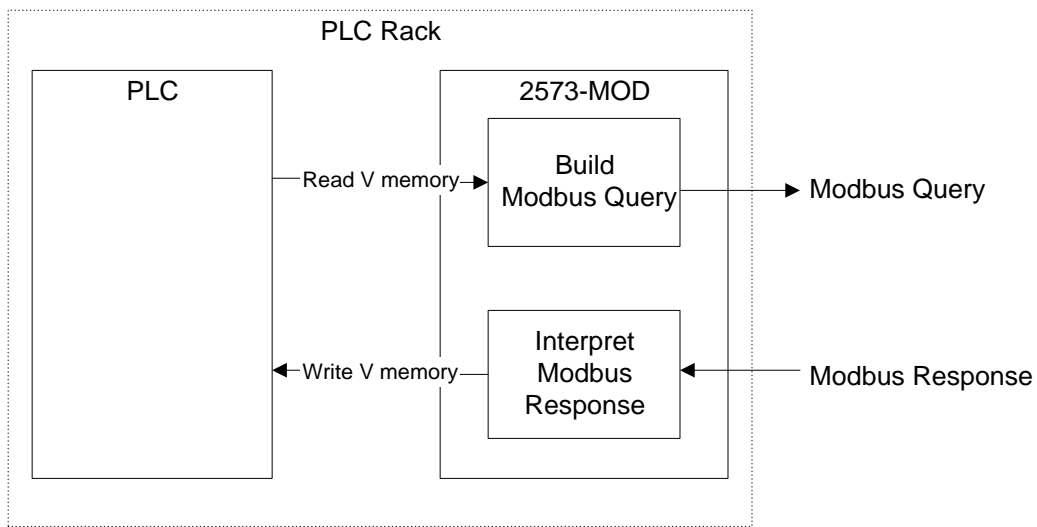
Figure 9. Modbus Master Command Flow

5.3. Create Connection Command

Before the 2573 can act as a Modbus master, the CREATE CONNECTION command must be initiated. This command is used to start a copy (instance) of either the RTU or ASCII Modbus Master Protocol Manager and associates this copy with a physical port. The choice of using either the RTU or ASCII protocol is determined by the information entered into the CREATE CONNECTION Command block. After you have successfully completed this command, you then refer to this instance by the logical connection number in all subsequent commands.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Create Connection Command Block - Modbus RTU Master

You can create up to two connections for Modbus RTU, one on Port 1 and one on Port 3, assuming you have not created any previous connections or automatically started any protocol managers via dipswitch settings on these ports. Ensure that the physical port you select in offset 4 has been enabled for *PLC Select*. See *Chapter 2. Installation* for information regarding switch settings. The following command block is used to create a connection for the Modbus RTU Master Protocol Manager.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 -- 19299)		
3	Protocol Manager Number (Modbus RTU Master)	**0028**	**40**
4	Physical Port Number (1 or 3)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Reserved (Unused)	**0000**	**0**
7	Parity (0=None, 1=Odd, 2= Even)		
8	Stop Bits (0=default, 1=1 stop bit)	**0000**	**0**
9	Handshake (0=None, 4=RS-485, 6=Modem RTS/CTS)	0000	0
10	Option Bits (see definitions below)		
11	V memory Address of Modem Configuration Table		
12	RTU End-of-Message Interval - in milliseconds (5-100)	*0000*	*0*
13-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**3** for port 3).

Offset 3   *Protocol Manager Number* - Set to 40 (Hex 0028) to select the Modbus Master RTU protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1 or 3) you wish to use.

Offset 5   *Baud Rate* - The baud rate must match the settings for the slave device.

Offset 6   *Reserved-* Not used in this command. Set to 0.

Offset 7   *Parity-* Enter a value which matches your slave device requirements. When no parity is selected, the protocol manager automatically sets the number of stop bits to 2. When even or odd parity is selected, the number of stop bits is set to 1.

Offset 8   *Stop Bits* - Use 0 for the Modbus default. This will automatically set the number of stop bits according to the parity setting. For no parity, stop bits will be set to 2; for even or odd parity, stop bits will be set to 0. Setting this value to 1 explicitly specifies 1 stop bit; you might need to set this bit to 1 with certain modem configurations.

Offset 9   *Handshake-* If you are using an RS-485 (two wire) electrical interface, set this value to 4. Set this to 6 to enable RTS/CTS modem flow control. When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this value to 0.

Offset 10   *Option Bits* - These are used to select special protocol manager options. The diagram below shows how the 16 bit word is defined. Bit 1 is the Most Significant Bit while Bit 16 is the Least Significant Bit.

MSB   LSB

Bits 1 - 10	Bit 11	Bit 12	Bit 13	Bit 14	Bit 15	Bit 16

Bit Description	**Bit(s)**	**Definitions**
Unused - Reserved	1 - 10	These are the most significant bits in the word. They are unused and are reserved for future use. Set bits to 0.
Accept Early Response	11	Enables the module to accept replies returned before the standard 3.5 character inter-message gap. 0 = Use the standard 3.5 character inter-message gap 1 = Accept a fast reply
End of Message Interval Timing	12	Determines whether to use a non-standard time interval to indicate the end of a message: 0 = Use standard (3.5 character time) end of message timing 1 = End of Message Interval defined by user. Use the value specified in offset 12 of the command block.
Unused - Reserved	13 - 15	These bits are reserved for future use and should be set to 0.
Modem Control Feature	16	This bit determines whether the Modem Control Feature is used. See *Chapter 8. 2573 Modem Control Feature* 0 = Modem Control Feature not used 1 = Modem Control Feature is used. See offset 11.

Offset 11   *V Memory Address of Modem Configuration Table* - Used with the 2573 Modem Control Feature.

Offset 12   *End of Message Interval* - Used to specifically override the standard RTU inter-message timing of 3.5 character times. Enter a value up to 100 milliseconds. Bit 12 in the startup Option Bits must be set before this value will be used by the protocol manager. If bit 12 is set to 0, the timing value is not read. If bit 12 is set, a value of zero milliseconds will be invalid. This option applies only to messages received by the module. Messages transmitted by the module will continue to use Modbus standard character timing.

You may use this option when communicating via data links that cannot maintain the constant bit rate required by Modbus RTU. For example, some modems, especially radio modems, may not be able to maintain a constant bit rate. Thus standard RTU will prematurely terminate a message, determine that a protocol error has occurred, and ignore the message.

NOTE:
Where possible, when character timing is a problem, the preferable solution is to use Modbus ASCII. In many cases, using ASCII will result in better performance than using RTU with long inter-message timing because of the delay before a message can be processed. If you choose to override the Modbus standard, you should set the delay as short as possible while maintaining reliable operation.

Ensure that the value you select is greater than 3.5 character times or your results will be unpredictable. See the table below for minimum times in milliseconds for each baud rate.

Baud Rate	Minimum time (ms)
300	Feature not supported at this baud rate.
600	64
1200	32
2400	16
4800	8
9600	5
19200	5

Offset 13-15   *Unused* - Reserved for future use. Set to 0.

NOTE:
The Modbus Master RTU protocol manager can be used with module ports 1 and 3 only.

Create Connection Command Block - Modbus ASCII Master

This command starts a copy (instance) of the Modbus ASCII Master protocol manager and associates this copy with a physical port. After you have successfully completed this command, you then refer to this instance by the logical connection number in all subsequent commands.

You can use any port that is not being used for another protocol manager for Modbus ASCII. Ensure that the physical port you select in offset 4 has been enabled for *PLC Select*. See *Chapter 2. Installation* for information regarding switch settings.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 - 19299)		
3	Protocol Manager Number (Modbus ASCII Master)	**0029**	**41**
4	Physical Port Number (1, 2, 3, or 4)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (7 or 8)	**0007**	**7**
7	Parity (0=None, 1=Odd, 2= Even)		
8	Stop Bits (1 or 2)		
9	Handshake (0=None, 1=Software, 2=Hardware, 4=RS-485, 6 = Modem RTS/CTS)	0000	0
10	Startup Option Bits (see definitions below)	0000	0
11	V memory Address of Modem Configuration Table		
12-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**3** for port 3).

Offset 3   *Protocol Manager Number* - Set to 41 (Hex 0029) to select the Modbus Master ASCII protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1 - 4) you wish to use.

Offset 5   *Baud Rate* - The baud rate must match the settings for the slave device.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Offset 6   *Bits per Character* - Modbus ASCII may use either 7 or 8 bits per character.

Offset 7   *Parity*- Enter a value which matches your slave device requirements.

Offset 8   *Stop bits* - Enter a value which matches your slave device requirements. If you chose
                  even or odd parity, the number of stop bits is usually set to 1. If no parity is selected, the
                  number of stop bits is usually set to 2.

Offset 9   *Handshake*- If you are using an RS-485 (two wire) electrical interface, set this value to 4.
                  Set this to 6 to enable RTS/CTS modem flow control (*requires firmware Version 7.0 and
                  above*). When this option is selected, the module will raise RTS when it needs to send a
                  message and will wait for the modem to raise CTS before sending it. Otherwise, set this
                  value to match your device requirements.

Offset 10   Option Bits - Bit 16 is used to determine whether the Modem Control Feature is used
                  (See *Chapter 8. 2573 Modem Control Feature*).
                  Bit 16 = 0 means Modem Control Feature is not used
                  Bit 16 = 1 means Modem Control Feature is used (See also offset 11).

Offset 11   *V Memory Address of Modem Configuration Table* - Used with 2573 Modem Control
                  Feature. See offset 10.

Offset 12-15   *Unused* - Reserved for future use. Set to 0.

5.4. Close Connection Command

This command is used to close a connection that was previously created by a CREATE CONNECTION
command. This command is common to both the RTU and ASCII Protocol Managers and is used to close
either type of connection. Once a connection has been closed, you can initiate another CREATE
CONNECTION command which reuses the physical port and/or connection number.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Close Connection)	**0002**	**2**
2	Connection Number (19221 - 19299)		
3 - 15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error
                  code is cleared. If a processing error is encountered, the protocol manager will write an
                  error code into this word.

Offset 1   *Command Code* - The Command for Close Connection is 2.

Offset 2   *Connection Number* - Enter the connection number you wish to close.

Offset 3-15   *Unused* - Set to 0.

5.5. Send Modbus Query

Introduction

The SEND MODBUS QUERY command is a basic means for sending messages over Modbus to other devices. The query is handled by three blocks of V memory. These blocks of V memory are labeled; SEND MODBUS QUERY command block, Send Query Data Segment, and Response Block (see figure below). The SEND MODBUS QUERY command block contains all the information needed by the protocol manager to perform the command. This includes: what slave device to address, what function to send, where to get the query data and where to put any responses, and how to handle the connection. The Send Query Data Segment contains the actual data that will be placed in the Data Field of the Modbus message. The Response Block is V memory reserved for placement of the response data, if any. Only one Modbus Query can be sent at a time by executing this command.

Send Query Data Segment

Beginning of data

Send Modbus Query command block

Offset 0
Offset 1
Offset 2
Offset 3
Offset 4
Offset 5
Offset 6
Offset 7
Offset 8
Offset 9
Offset 10
Offset 11
Offset 12
Offset 13
Offset 14
Offset 15
Offset 16

Response Block

Beginning of data
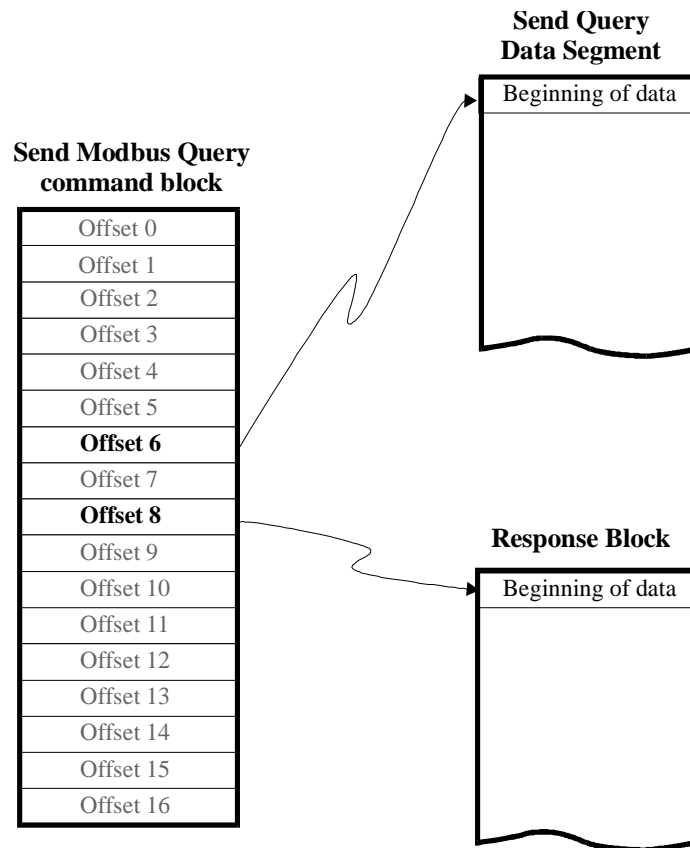
Send Modbus Query command block

Once a CREATE CONNECTION command has been completed for the appropriate Modbus Master protocol manager, the following SEND MODBUS QUERY command block can be used to send a Modbus query. Ensure that the Command value entered into offset 1 corresponds to protocol manager used in the associated CREATE CONNECTION command.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command: Send Modbus RTU Query	2801	10241
	Command: Send Modbus ASCII Query	2901	10497
2	Connection Number (19221 - 19299)		
3	Command Flag	**0000**	**00**
4	Modbus Slave Address		
5	Modbus Function Code		
6	Send Query Data Segment - V Memory Location		
7	Send Query Data Segment - Size in Bytes		
8	Response Block - V Memory Location.		
9	Response Block - Amount of V memory to reserve.		
10	Command Time-out In Seconds (0 = Default = 9 seconds; >999 = infinite time-out).		
11-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Send Modbus RTU Query is 2801hex (10241 decimal). The Command for Send Modbus ASCII Query is 2901hex (10497 decimal).

Offset 2   *Connection Number* - This must match the connection number you used in the previous Create Connection command.

Offset 3   *Command Flag* - Reserved for future use. Use a value of 0 for this release.

Offset 4   *Modbus Slave Address* - Set to the address of the slave to which you are sending the query. Valid Values are 0 - 247 (0x00-0xF7). A value of 0 is a broadcast to all slave units on the line.

Offset 5   *Modbus Function Code* - Valid Values are 1 - 127. See the device documentation. The official Modbus Specification identifies 0x01-0x18 as valid command codes.

Offset 6   *Send Query Data Segment: V Memory Location* - Starting V memory location of the Send Query Data Segment, which contains the data portion of the Modbus query. Some Modbus queries do not contain a data portion. In this case set this to 0.

Offset 7   *Send Query Data Segment - Size in Bytes* - Number of **bytes** of V memory needed for the Send Query Data Segment (1 word = 2 bytes). Some Modbus queries do not contain any data. In this case, set this to 0. This value must exactly match the query data length.

Offset 8   *Response Block: V Memory Location* - Starting V memory location of the Response Block. This is where the response will be stored.

Offset 9   *Response Block: Amount of V Memory to Reserve* - Number of **bytes** of V memory to *reserve* for storing the Modbus response (1 word = 2 bytes). The size of the stored response will be equal to the size of the response data **plus 4 bytes** (2 words) to store the function code and the byte count. You may set this value to any number greater than or equal to the size of the response as defined above. It is a reasonable practice to reserve slightly more than you think you will need; if you do not reserve enough V memory, the protocol manager will return an error code. See the following section, *Send Query Data Segment*.

Offset 10   *Command Timeout* - This value specifies the total time allowed to process the command. If the command has not been completed within the specified time, the protocol manager will return a timeout error code (0x289D). If you set this word to 0, the protocol manager will use the default timeout value (approximately 9 seconds). If you set this word to a value greater than 999, the protocol manager will never time out. For broadcast queries (to Slave address 0) which do not have a response, the timeout is ignored.

Offset 11-15   *Unused* - Reserved for future use. Set to 0.

Send Query Data Segment

You should place the entire data portion of the Modbus query message in the Send Query Data Segment. Start at the V memory address you specified in offset 6 of the Command Block. Modbus message data is byte oriented. You should place the first byte of data in the high byte of the first word, the second byte of the data in the low byte of the first word, the third byte of the data in the high byte of the second word, etc. See the following example.

For function code 03 (Read Holding Registers), assume that you want the data portion to be:

    Starting Address Hi   00
    Starting Address Lo   6B
    No. of Registers Hi   00
    No. of Registers Lo   02

Then V memory should contain:

Offset	Value (hex)
0	006B
1	0002

For most commands, the above technique will align word oriented data with V memory boundaries. However, two commonly used Modbus functions (function code 15 - Force Multiple Coils and function code 16 - Preset Multiple Registers) contain a byte count field which would cause the data following the byte count to be non-aligned with V memory boundaries. The protocol manager solves this problem by allowing you to use a *full word* for the byte count field. When the protocol manager creates the output message, it will not include the high byte in the message. See the following example:

Example - Function Code 16 Query

For function code 16 (Preset Holding registers)
assume that you want the data portion to be:

Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data1 Hi	00
Data1 Lo	0A
Data2 Hi	01
Data2 Lo	02

Then V memory should contain:

Offset	Value	Comments
0	0001	Address Hi/Lo
1	0002	No. of Registers Hi/Lo
2	0004	Byte Count
3	000A	Data1 Hi/Lo
4	0102	Data2 Hi/Lo

Modbus Message Response Block

Modbus Response message data is stored in a block of V Memory called the Response Block, starting in the V memory address that you specified in offset 8. For additional detail see *Modbus Message Response Block* section on page 50.

Modbus Exception Codes

The SEND QUERY command will return an error if response function code does not match the query function code or the corresponding exception code. The corresponding exception code is the function code with the high bit set. The SEND QUERY command does not treat a valid exception code as a command error. Your logic should examine the exception code and determine what action should be taken.

5.6. Modbus Initiate Poll List Command

Introduction

Many times situations occur in the nature of process controls that require a group of Modbus queries to routinely be performed sequentially. The INITIATE POLL LIST command is a Modbus Master command which allows multiple Modbus Queries to be initiated sequentially from a single command block. The INITIATE POLL LIST command block contains all the information needed for controlling the execution of the query poll list on a global level.

Multiple queries are handled by the utilization of Query Blocks. Each Query Block places all the information needed by an individual query into adjacent V memory locations.

Space must be provided for the query response data. The Response Block is where this data is stored. The Response Block location in V memory for each query is pointed to by the associated Query Block.

The INITIATE POLL LIST command executes queries by including in the command a list of addresses. These addresses are pointers to the starting locations in V memory of all the queries to be done. The poll list can be as short as one query or as long as 60 queries (Figure below shows how two queries are linked to the command). Therefore, this command is *not* a fixed block length of 16 words.
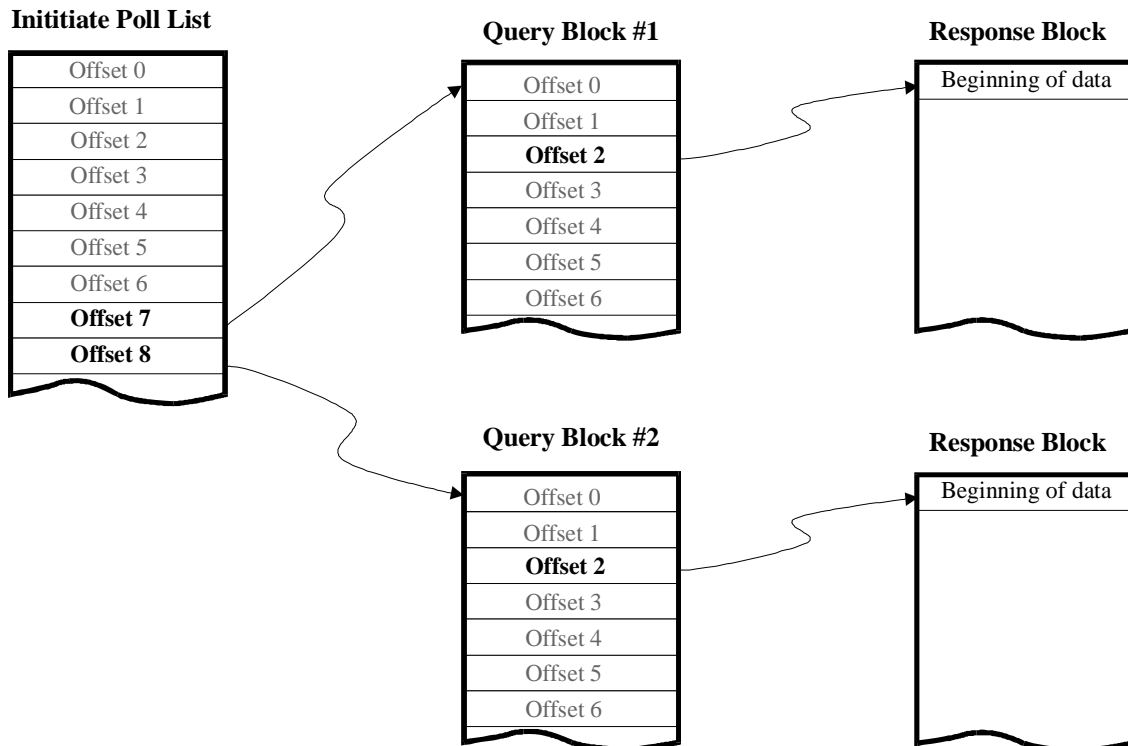
Implementation Considerations

Performance Issues

When obtaining data from multiple devices on a single network, the INITIATE POLL LIST command will be faster overall than using a separate SEND QUERY Command for each device. However, when the INITIATE POLL LIST command is executed, *all* active queries in the poll list must be competed before the COMMAND BUSY bit is lowered. The more queries you include in the list, the longer this will take. Assuming you employ the typical approach of using the COMMAND BUSY bit to determine when the data is valid, this will require that you wait for all queries to complete before reading data from any one. If response time for a particular device is important, you may choose to split the poll list among multiple commands. Alternately you may choose to use other programming techniques to determine when the data is valid. Contact CTI for application examples.

Error Processing

For many practical applications, you will want to continue polling even though a particular device on the Modbus network returned an error. Unless you explicitly set the Command Flags to stop the INITIATE POLL LIST command on error, when a error is encountered with a query, execution will continue with the next Query Block in the poll list. The applicable error code will be written to the error word of the Query Block in error; but the CMD ERR bit will not be set and nothing will be written to the Command Error Word in the INITIATE POLL LIST Command Block. Therefore, if you choose to continue polling on when an error occurs, your logic should examine the error word in the Query Block before assuming the data in the corresponding Response Block is valid.

Setting Query Timeout

The INITIATE POLL LIST Command Block allows you to set a query timeout. This represents the maximum amount of time allotted to each query. If the query cannot be competed within the specified timeout period, then the protocol manager will either proceed to the next query or stop execution of the poll list, depending upon the setting of control flags defined below. Most timeouts are due to the device failing to respond. This could happen occasionally, since the Modbus protocol specifies that the slave device should not respond if it detects a communications error such as bad parity.

Setting the query timeout to 0 will cause the protocol manager to use the default value of 9 seconds. You may wish to explicitly set the query timeout to a smaller value so that polling is not delayed excessively when a timeout occurs. Make sure that the timeout value you enter is greater than the longest time it takes to complete a query, else you could timeout before the device response is received and processed. If you set the value too small, you will encounter network problems.

Initiate Poll List Command Block

Once a CREATE CONNECTION command has been completed for the Modbus Master protocol Manager, the following command block is used to send a sequential list of Modbus queries.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command: Initiate RTU Poll List	2804	10244
	Command: Initiate ASCII Poll List	2904	10500
2	Connection Number (19221 - 19299)		
3	Command Flags - (See definitions below)		
4	Query Timeout (resets at beginning of each new query)		
5	Query Block with error (set by protocol manager when an error stops the Poll List execution)	**0000**	**0**
6	Query Block Polling List Length (1 - 60)		
7	V Memory Address of Query Block #1		
8	V Memory Address of Query Block #2		
9 - 66	V Memory Addresses of additional Query Blocks		

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered that stops command execution, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The command code for INITIATE POLL LIST is:
2804hex (10244 decimal) for RTU; 2904hex (10500 decimal) for ASCII.

Offset 2   *Connection Number* - This must match the connection number you used in the previous Create Connection command.

Offset 3   *Command Flags* - Used to conditionally stop poll list execution. When a bit is set and the corresponding error is encountered, execution of the INITIATE POLL LIST command will stop, the CMD ERR bit will be set, and the error code will be written to the Command Error word. When set, this flag overrides the settings of the Query Block Flags.

Bit Description	Bit(s)	Definitions
Unused - Reserved	1 - 13	These are the most significant bits in the word. They are unused and are reserved for future use. Set bits to 0.
Global Query Timeout Flag	14	Determines whether to stop poll list execution if any Command Timeout is encountered during poll list processing: 0 = Use flag settings in Query Blocks 1 = Stop on timeout error in any Query Block
Global Exception Error Flag	15	Determines whether to stop poll list execution if an Exception Response is received during poll list processing: 0 = Use flag settings in Query Blocks 1 = Stop on exception error in any Query Block
Global Protocol Error Flag	16	Determines whether to stop poll list execution if a protocol error* is encountered during poll list processing: 0 = Use flag settings in Query Blocks 1 = Stop on protocol error in any Query Block

* Protocol errors include: port hardware error, checksum error, message length error, wrong slave responded, and wrong function code in response.

Offset 4   *Query Timeout* - This value specifies the approximate time (in seconds) allowed for each query within the poll list. The timer is reset at the beginning of each new query in the poll list. If you set the timeout to 0, the protocol manager will use the default timeout value (approximately 9 seconds). See *Setting Query Timeout* on the previous page for considerations in setting this value. The timeout is ignored for broadcast queries (to Slave Address 0), which do not have a response.

Offset 5   *Query Block with Error* - If an error occurs that stops the processing of the poll list, the V memory address of the query block that encountered the error will be placed in this word. Your PLC logic should set this word to 0 so that any previous address is cleared.

Offset 6   *Query Block Polling List Length* - Indicates to the protocol manager the length of the polling list to follow. Valid list lengths are from 1 to 60 inclusive.

Offset 7   *V Memory Address of Query Block 1* - Contains the V memory address of the first query block in the list.

Offset 8   *V Memory Address of Query Block 2* - Contains the V memory address of the second query block in the list (If more than one query is desired).

Offset 9-66   *V memory Address of Additional Query Blocks* - Contains the V memory address of any additional query blocks in the list. The number of entries on the list cannot exceed 60.

> *NOTE:*
> *For offsets 7 and greater: Query Block address values of 0 are skipped; command processing continues with the next query in the list. Your logic may use this feature to dynamically change the poll list each time you execute the INITIATE POLL LIST Command*

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Query Blocks

A Query Block, when used with an INITIATE POLL LIST command, contains all the information needed to perform one complete query. The structure of a Query Block is shown below:

Offset	Description	Hex Value	Decimal Value
0	Error Word	**0000**	**0**
1	Query Block Control Flags - (See definitions below)		
2	Response - V memory Address		
3	Response - Amount of V memory to reserve (1 - 256)		
4	Modbus Slave Address (1 - 247)		
5	Modbus Function Code		
6	Number of Bytes of V Memory for Query Data		
7 - ?	Query Data		
? + 1	Query Block Validation Code (ASCII "QB")	**5142**	**20802**

Offset 0　　　　*Error Word* - If a processing error is encountered during execution of this query, the protocol manager will write an error code into this word. Your PLC logic should set this to 0 so that any previous error code is cleared.

Offset 1　　　　*Query Block Control Flags* - Used to control poll list execution. The flags are similar in function to the Command Control Flags defined previously except that they apply only to a specific query.

Bit Description	Bit(s)	Definitions
Unused - Reserved	1 - 12	These are the most significant bits in the word. They are unused and are reserved for future use. Set bits to 0.
Query Block Disable	13	When set, the INITIATE POLL LIST command skips the processing of this query in the poll list. Your logic may use this bit to conditionally enable/disable a query.
Command Timeout Flag	14	Determines whether to stop poll list execution if any Command Timeout is encountered the processing of this query: 0 = Do not stop poll list execution 1 = Stop processing the poll list on Command Timeout
Exception Error Flag	15	Determines whether to stop poll list execution if an Exception Response is received during the processing of this query: 0 = Do not stop poll list execution 1 = Stop processing the poll list on Exception Response
Protocol Error Flag	16	Determines whether to stop poll list execution if a protocol error* is encountered during the processing of this query: 0 = Do not stop poll list execution 1 = Stop processing the poll list on Protocol Error.

* Protocol errors include: port hardware error, checksum error, message length error, wrong slave responded, and wrong function code in response.

Offset 2	*Response: V Memory Address* - Starting V memory address where the Modbus response to this query will be stored.
Offset 3	*Response: Amount of V memory to Reserve* - Number of **bytes** of V memory to *reserve* for storing the Modbus response (1 word = 2 bytes). The size of the stored response will be equal to the size of the response data **plus 4 bytes** (2 words) to store the function code and the byte count. You may set this value to any number greater than or equal to the size of the response as defined above. It is a reasonable practice to reserve slightly more than you think you will need; if you do not reserve enough V memory, the protocol manager will return an error code.
Offset 4	*Slave Address* - Set to the address of the slave to which you are sending the query. Valid Values are 1 - 247 (0x01-0xF7). A value of 0 is a broadcast to all slave units on the line. If the slave address is 0 and no errors are encountered, this query is considered complete.
Offset 5	*Modbus Function Code* - Valid Values are 1 - 127. See the device documentation. The official Modbus Specification identifies 0x01 - 0x18 as valid command codes. However, it is possible that your device may implement additional command codes.
Offset 6	*Number of Bytes of V Memory for Query Data* - Indicates the number of V memory **bytes** (1 word = 2 bytes) used to store the data portion of the query block. Some Modbus queries do not contain a data portion. In this case, set this value to 0.
Offset 7-?	*Query Data* - Modbus Query Data
Offset ?+1	*Query Block End Validation Code* - Validation Code used to check that the Query Data length matches the value given in Offset 6.

Query Block Data Length

You should place the entire data portion of the Modbus query message in V memory, starting at offset 7 of the Query Block. Modbus message data is byte oriented. You should place the first byte of data in the high byte of the first word (offset 7), the second byte of the data in the low byte of the first word (of offset 7), the third byte of the data in the high byte of the second word (offset 8), etc. See the following example.

For function code 03 (Read Holding Registers), assume that you want the data portion to be:

Starting Address Hi	00
Starting Address Lo	6B
No. of Registers Hi	00
No. of Registers Lo	02

Then V memory should contain:

Offset	Value (hex)
7	006B
8	0002

> *NOTE:*
> *Modbus Function Codes 15 and 16 are treated as a special case by the protocol manager. See*
> *explanation and example below.*

For most commands, the above technique will align word oriented data with V memory boundaries. However, two commonly used Modbus functions (function code 15 - Force Multiple Coils and function code 16 - Preset Multiple Registers) contain a byte count field which would cause the data following the byte count to be non-aligned with V memory boundaries. The protocol manager solves this problem by allowing you to use a *full word* for the byte count field. When the protocol manager creates the output message, it will not include the high byte in the message. See the following example:

Example - Function Code 16 Query

For function code 16 (Preset Holding registers)
assume that you want the data portion to be:

Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data1 Hi	00
Data1 Lo	0A
Data2 Hi	01
Data2 Lo	02

Then V memory should contain:

Offset	Value	Comments
7	0001	Address Hi/Lo
8	0002	No. of Registers Hi/Lo
9	0004	Byte Count
10	000A	Data1 Hi/Lo
11	0102	Data2 Hi/Lo

Modbus Message Response Block

Modbus Response message data is stored in a block of V Memory called the Response Block, starting in the V memory address that you specified in offset 8. Offset 0 will contain the function code from the response message. This is provided so that you can evaluate whether you received a message containing an exception code. If the response to the function contained a byte count, Offset 1 will contain the byte count from the message data area. Otherwise the byte count will be set to 0. This approach is used to force the alignment of numeric data on V memory word boundaries for most response formats.

NOTE:
See the Modbus Documentation for response formats containing a byte count.

The following table illustrates how the Modbus response is stored in V memory.

Offset	Contents
0	Function Code from Response message
1	Byte Count from Response message (or 0)
2	Message data
..	Message Data
n	Message Data

For example, the response to a function code 03 query which contained the following data:

Byte Count	04
Data1 High	02
Data1 Low	2B
Data2 High	00
Data2 Low	64

would be stored in V memory as:

Offset	Value (hex)	Comments
0	0003	Function code from response
1	0004	Byte Count
2	022B	Data 1
3	0064	Data 2

The response to a function code 16 (hex10) which contained the following data:

       Starting Address Hi   00
       Starting Address Lo   01
       No. Registers Hi   00
       No. Registers Lo   02

would be stored in PLC V memory as:

Offset	Value (hex)	Comments
0	0010	Function Code from response
1	0000	Byte Count (0 = not used)
2	0001	Starting Address
3	0002	No. of Registers

CHAPTER 6. TASK CODE & CAMP PROTOCOL MANAGERS

6.1. Overview

This chapter describes the Command Blocks used to control 2573 module functions. These include Command Blocks to control the Task Code Master, and CAMP Master protocol managers.

The command block illustrations in this chapter show their contents in both Hexadecimal and Decimal (integer) format.   You can enter and display values in either format using your PLC programming software. The hexadecimal presentation format is usually used to display the error word.

NOTE:
In Command Block descriptions, a boldface entry in the value column designates a required value.
Other entries are recommended values.

6.2. Task Code Master Commands

General Information

Using Task Code Master commands, you can transfer a block of V memory between a SIMATIC® 505 PLC with a 2573 module installed and another SIMATIC® 505 or TI500 PLC. When command block is triggered by the PLC, the 2573 module generates the applicable sequence of task codes required to read or write V memory in the remote PLC. The task codes are encoded in the Non-Intelligent Terminal Protocol (NITP) message format, which is supported by all SIMATIC® 505 and TI500 Series PLC program ports.

The Command Blocks allow you to transfer up to 256 words with a single command trigger; however, the NITP message format supports only 15 words per message.   The module firmware accomplishes the transfer of more than 15 words by generating multiple NITP messages. Although this operation is transparent to the local PLC logic, transferring a large number of words may require a multiple scans. At baud rates of 9600 or above, the primary determination of the command completion time is the transfer performance of the attached PLCs . This performance is determined by the PLC scan time and other task code activity.

> *NOTE:*
> *When designing applications that transfer large amounts of V memory, you should carefully evaluate the response time to ensure that it meets your requirements. If you need to improve performance, you may wish to install a CTI 2573 module in the remote PLC and use the Common ASCII Message Protocol (CAMP) protocol manager instead of the Task Code Master.*

Create Connection Command

Normally, you will use the module DIP switches to configure a serial port for the Task Code Master protocol. However, if you want to create a connection for the Task Code Master protocol manager using PLC logic, you can use this command block. To use this command, the serial port must have been configured as *PLC Select.*

This command starts a copy (instance) of the protocol manager and associates this copy with a physical port. After you have successfully completed this command, you then use the connection number to refer to this instance in all subsequent commands.

Assuming you have not created any previous connections or automatically started any protocol managers via dipswitch settings, you can create up to four connections (one for each physical port). Ensure that the physical port you select in offset 4 has been enabled for *PLC Select*. See *Chapter 2. Installation*, for information regarding switch settings.

The following table illustrates the Create Connection command for the Task Code Master protocol manager.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 - 19299)		
3	Protocol Manager Number (Task Code Master)	**0020**	**32**
4	Physical Port Number (1,2,3,4)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (7 or 8)	**0007**	**7**
7	Parity (0=None, 1=Odd, 2= Even)	**0001**	**1**
8	Stop Bits (1 or 2)	**0001**	**1**
9	Handshake (0=None, 1=Software, 2=Hardware, 6=Modem RTS/CTS)	0000	0
10	Option Bits (0x0000 = None, 0x0001 = Modem)		
11	V memory Address of Modem Configuration Table		
12-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 19221 for port 1, 19222 for port 2).

Offset 3   *Protocol Manager Number* - Set to 32 (Hex 0020) to select the Task Code Master protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1-4) you wish to use.

Offset 5-8   *Communications Parameters* - This must match the characteristics of your device. The values entered will override any switch settings on the module. NITP requires the following settings: Bits per Character = 7, Parity = Odd, Stop Bits = 1.

Offset 9   *Handshake*- Set this to 6 to enable RTS/CTS modem flow control (*requires firmware Version 7.0 and above*). When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this value to match your device requirements.

Offset 10   *Option Bits* - These are used to select special protocol manager options. Bit 16 is used to select the 2573 Modem Control Feature (see *Chapter 8. 2573 Modem Control Feature*). Other options bits are currently unused and should be set to 0.

Offset 11   *V Memory Address of Modem Configuration Table* - Used with the 2573 Modem Control Feature. See offset 10 above.

Offset 12-15 *Unused* - Reserved for future use. Set to 0.

NOTE:
If you have used the module dipswitches to start the Task Code Master protocol manager, you should not execute this command.

Close Connection Command

This command is used to close a connection that was previously created by a Create Connection command. Once a connection has been closed, you can initiate another Create Connection command which reuses the physical port and/or connection number.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Close Connection)	**0002**	**02**
2	Connection Number (19221 - 19299)		
3 - 15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Close Connection is 2.

Offset 2   *Connection Number* - Enter the connection number you wish to close.

Offset 3-15   *Unused* - Set to 0.

Write Remote PLC Memory

This function reads the contents of a block of V memory in the local PLC and writes the values in a block of V memory in the other PLC. You can specify the starting V memory addresses on both the local and remote PLCs and the number of words to be transferred (up to 256 words). Based on the specification, the protocol manager reads the local V memory, generates NITP messages containing the applicable task code and data, and sends the message via the designated port. When the response is returned from the remote PLC, the protocol manager verifies that the task code was properly processed. If not, the protocol manager sets the applicable error bit and writes an error code to the Command Error Word.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Write Remote Memory)	**0201**	**513**
2	Connection Number (19201 - 19299)		
3	Source: Starting V Memory Address in this PLC		
4	Number of Words to Transfer (1-256)		
5	Destination: Starting V Memory Address in Remote PLC		
6	Command Timeout in seconds (0=Use Default)	0000	0
7-15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command Code for Write Remote Memory is 513 decimal (0x0201).

Offset 2   *Connection Number* - If the Task Code Master protocol was initiated using a Create Connection command, use the connection number that was established in that Command Block (19221-19299).

If the Task Code Master protocol was *automatically started* using the applicable module dipswitch setting, the connection number entry in offset 2 should correspond to the physical port number as follows:

Connection Number 19201 specifies the protocol manager started on serial port 1,
Connection Number 19202 specifies the protocol manager started on serial port 2,
Connection Number 19203 specifies the protocol manager started on serial port 3,
Connection Number 19204 specifies the protocol manager started on serial port 4.

Offset 3   *Source: Starting V Memory* - Enter the address of V memory in the local PLC which contains the first data value you want to transfer.

Offset 4   *Number of Words to Transfer* - Enter the number of words you want to transfer. For
                large transfers see the caution on performance on page 53.

Offset 5   *Destination: Starting V Memory* - Enter the beginning address of the V memory block in
                the remote PLC you want to update.

Offset 6   *Command Timeout in seconds* - You should normally set this value to 0, indicating the
                default. The Task Code Master protocol manager default is approximately 9 seconds. A
                value greater than 999 specifies an infinite timeout.

Read Remote PLC Memory

This function reads the contents of a block of V memory in the remote PLC and writes the values in a
block of V memory in the local PLC. You can specify the starting V memory addresses on both the local
and remote PLCs and the number of words to be transferred (up to 256 words). Based on the
specification, the protocol manager generates NITP messages containing the applicable task code, verifies
the response, and writes the data in local V memory. If an error is encountered, the protocol manager will
set the applicable error bit and write an error code in the Command Error Word.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Read Remote Memory)	**0202**	**514**
2	Connection Number (19201 - 19299)		
3	Source: Starting V Memory Address in Remote PLC		
4	Number of Words to Transfer (1-256)		
5	Destination: Starting V Memory Address in this PLC		
6	Command Timeout in seconds (0=Use Default)	0000	0
7-15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error
                code is cleared. If a processing error is encountered, the protocol manager will write an
                error code into this word.

Offset 1   *Command Code* - The Command Code for Read Remote Memory is 514 (hex 0202).

Offset 2   *Connection Number* - If the Task Code Master protocol was initiated using a Create
            Connection command, use the connection number that was established in that Command
            Block.

            If the task code master protocol was *automatically started* using the applicable module
            DIP switch setting, the connection number entry in offset 2 should correspond to the
            physical port number as follows:

            Connection Number 19201 specifies the protocol manager started on serial port 1,
            Connection Number 19202 specifies the protocol manager started on serial port 2,
            Connection Number 19203 specifies the protocol manager started on serial port 3,
            Connection Number 19204 specifies the protocol manager started on serial port 4.

Offset 3   *Source: Starting V Memory* - Enter the first V memory location in the remote PLC which
            contains the values which you want to transfer.

Offset 4   *Number of Words to Transfer* - Enter the number of words you want to transfer. For
            large transfers see caution on performance on page 53.

Offset 5   *Destination: Starting V Memory* - Enter the beginning address of the V memory block in
            the local PLC you want to update.

Offset 6   *Command Timeout in seconds* - You should normally set this value to 0, indicating the
            default. The default is approximately 9 seconds

6.3. CAMP Master Commands

The CAMP Master protocol manager is used to transfer data between CTI communications modules.
You may wish to use the CAMP Master rather than the Task Code Master protocol when:
- You are transferring data between SIMATIC® Series 505 PLCs,
- Each PLC rack contains a 2573-MOD or a 2573-TCM2 module,
- You are transferring larger amounts of data,
- Throughput is an important consideration.

Note that the memory transfer commands for CAMP are exactly the same as the Task Code Master. The
difference is in the underlying protocol. The protocol which is determined by which protocol manager is
selected in the Create Connection command.

NOTE:
The use of CAMP to perform memory transfer requires that both PLCs have a 2573 communications
module installed.

Create Connection Command

This command starts a copy (instance) of the CAMP protocol manager and associates this copy with a physical port. After you have successfully completed this command, you then refer to this instance by the logical connection number in all subsequent commands.

Assuming you have not created any previous connections or automatically started any protocol managers via dipswitch settings, you can create up to four connections (one for each physical port). Ensure that the physical port you select in offset 4 has been enabled for *PLC Select*. See *Chapter 2. Installation* for information regarding switch settings.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 - 19299)		
3	Protocol Manager Number (CAMP Master)	**0024**	**36**
4	Physical Port Number (1,2,3,4)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200		
6	Bits Per Character (7 or 8)	**0007**	**7**
7	Parity (0=None, 1=Odd, 2= Even)	**0001**	**1**
8	Stop Bits (1 or 2)	**0001**	**1**
9	Handshake (0=None, 1=Software, 2=Hardware, 6=Modem RTS/CTS)		
10	Option Bits (0x0000 = None, 0x0001 = Modem)		
11	V memory Address of Modem Configuration Table		
12-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**2** for port 2).

Offset 3   *Protocol Manager Number* - Set to 36 (Hex 0024) to select the CAMP protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1-4) you wish to use.

Offset 5-8   *Communications Parameters* - This must match the characteristics of your device. The values entered will override any switch settings on the module. CAMP requires the following settings: Bits per Character =7, Parity = Odd, Stop Bits =1.

             CTI 2573-MOD2/TCM2 Installation and Operation Guide

Offset 9   *Handshake*- Set this to 6 to enable RTS/CTS modem flow control (*requires firmware Version 7.0 and above*). When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this value to match your device requirements.

Offset 10   *Option Bits* - These are used to select special protocol manager options. Bit 16 is used to select the 2573 Modem Control Feature (See *Chapter 8. 2573 Modem Control Feature*). Other options bits are currently unused and should be set to 0.

Offset 11   *V Memory Address of Modem Configuration Table* - Used with the 2573 Modem Control Feature.

Offset 12-15   *Unused* - Reserved for future use. Set to 0.

Close Connection Command

This command is used to close a connection that was previously created by a Create Connection command. Once a connection has been closed, you can initiate another Create Connection command which reuses the physical port and/or connection number.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Close Connection)	**0002**	**2**
2	Connection Number (19221 - 19299)		
3 - 15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Close Connection is 2.

Offset 2   *Connection Number* - Enter the connection number you wish to close.

Offset 3-15   *Unused* - Set to 0.

Write Remote PLC Memory

The command block for Write Remote PLC Memory using CAMP is identical to the command block for the same function using the Task Code Master protocol manager. See page 57 for a description of the command block.

Read Remote PLC Memory

The command block for Read Remote PLC Memory using CAMP is identical to the command block for the same function using the Task Code Master protocol manager. See page 58 for a description of the command block.

Word Transfer - Memory Exchange (Firmware V4.8 and above)

This function is a combination of the Write Remote and Read Remote commands. As the name implies, it is intended to be used in applications that exchange values between to PLCs. The command writes the contents of a block of local PLC V memory to a block of memory in the remote PLC, then reads a block of V memory in the remote PLC and writes the values to a block of V memory in the local PLC. Since this is accomplished in one transaction, it is much faster than executing separate read and write commands.

You can specify the locations of the V memory blocks in both PLCs, the number of words to be written to the remote PLC, and the number of words to be read from the remote PLC. When the command is triggered, the CAMP protocol manager sends a message containing the data to be written to the remote PLC. When the response message is returned, the CAMP protocol manager verifies the response and, if there is no error, writes the data in local V memory. If an error is encountered, it will set the applicable error bit and write the error code into the Command Error Word.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command Code (Memory Exchange)	**0203**	**515**
2	Connection Number (19221 - 19299)		
3	Write Source: Starting V Memory Address in Local PLC		
4	Number of Words to Write (1-254)		
5	Write Destination: Starting V Memory Address in Remote PLC		
6	Command Timeout in seconds (0=Use Default)	0000	0
7	Read Source: Starting V Memory Address in Remote PLC		
8	Number of Words to Read (1-256)		
9	Read Destination: Starting V Memory Address in Local PLC.		
10 - 15	Unused (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command Code for READ REMOTE MEMORY is 515 (hex 0202).

Offset 2   *Connection Number* - Use the connection number that was established in the CREATE CONNECTION command block.

Offset 3   *Write Source: Starting V Memory* - Enter the first V memory location in the *local* PLC which contains the values which you wish to write to the remote PLC.

Offset 4   *Number of Words to Transfer* - Enter the number of words you want to write.

Offset 5   *Write Destination: Starting V Memory* - Enter the beginning address of the V memory block in the *remote* PLC to which you want to write this data.

Offset 6   *Command Timeout in seconds* - You should normally set this value to 0, indicating the default. The CAMP protocol manager default is approximately 9 seconds. A value of 999 specifies an infinite timeout.

Offset 7   *Read Source: Starting V Memory* - Enter the first V memory location in the *remote* PLC from which you wish to read data.

Offset 8   *Number of Words to Transfer* - Enter the number of words you want to read.

Offset 9   *Write Destination: Starting V Memory* - Enter the beginning address of the V memory block in the *local* PLC to which you want to write this data.

6.4. CAMP/NITP Slave Commands

Create Connection Command

Normally, you will use the module DIP switches to configure a serial port for the CAMP/NITP Slave protocol. However, if you want use the modem handshaking feature, you must to create a connection for the using PLC logic using this command block. To use this command, the serial port must have been configured as *PLC Select. T*his command starts a copy (instance) of the protocol manager and associates this copy with a physical port.

Assuming you have not created any previous connections or automatically started any protocol managers via dipswitch settings, you can create up to four connections (one for each physical port). Ensure that the physical port you select in offset 4 has been enabled for *PLC Select*. See *Chapter 2. Installation*, for information regarding switch settings.

The following table illustrates the Create Connection command for the Task Code Master protocol manager.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	**1**
2	Connection Number (19221 - 19299)		
3	Protocol Manager Number (CAMP/NITP Slave)	**0023**	**35**
4	Physical Port Number (1,2,3,4)		

5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (7 or 8)	**0007**	**7**
7	Parity (0=None, 1=Odd, 2= Even)	**0001**	**1**
8	Stop Bits (1 or 2)	**0001**	**1**
9	Handshake (0=None, 1=Software, 2=Hardware, 6=Modem RTS/CTS)	0000	0
10-15	Unused - reserved for future use (Set to 0)	**0000**	**0**

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Create Connection is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 19221 for port 1, 19222 for port 2).

Offset 3   *Protocol Manager Number* - Set to 35 (Hex 0023) to select the CAMP/NITP Slave protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1-4) you wish to use.

Offset 5-8   *Communications Parameters* - This must match the characteristics of your device. The values entered will override any switch settings on the module. CAMP/NITP requires the following settings: Bits per Character = 7, Parity = Odd, Stop Bits = 1.

Offset 9   *Handshake-* Set this to 6 to enable RTS/CTS modem flow control (*requires firmware Version 7.0 and above*). When this option is selected, the module will raise RTS when it needs to send a message and will wait for the modem to raise CTS before sending it. Otherwise, set this value to match your device requirements.

Offset 10-15   *Unused* - Reserved for future use. Set to 0.

CHAPTER 7. APPLICATION EXAMPLES

7.1. General Comments

The following examples illustrate how you can use the 2573 modules in a variety of typical applications. The first example, NITP/CAMP Slave, does not require PLC logic. The second example, Task Code Master, illustrates how to use a protocol manager that is automatically started but requires PLC logic to initiate messages. The third example, CAMP Master, shows how to use a protocol manager that must be initialized using PLC logic and requires PLC logic to initiate messages. The fourth example, illustrates using a 2573-MOD to communicate with a Modbus device.

These examples assume that you have the 2573 logged in starting at WX1. Therefore, the module will use words 1 - 8 as follows:

- WX1   Module Status Word
- WX2   Command Status Word
- WY3   Module Control Word
- WY4   Command Control Word
- WY5   Command Slot 1
- WY6   Command Slot 2
- WY7   Command Slot 3
- WY8   Command Slot 4.

See the *WX/WY Quick Reference* in on page 158 of this manual for an illustration of word use. Note that the four groups of Command Status bits and Command Control bits correspond to the four command slots. In these examples:

- WX2.1 - WX2.4 and WY4.1 - WY4.4 are used with Command Slot 1 (WY5),

- WX2.5 - WX2.8 and WY4.5 - WY4.8 are used with Command Slot 2 (WY6),

- WX2.9 - WX2.12 and WY4.9 - WY4.12 are used with Command Slot 3 (WY7),

- WX2.13- WX2.16 and WY4.13 - WY4.16 are used with Command Slot 4 (WY8).

For complete details, refer to Appendix D.

These examples use the command slots so that the slot number corresponds to the physical port number. For example, Command Slot 2 is used for commands to the protocol manager running on physical port 2 and Command Slot 3 is used for commands to the protocol manager running on physical port 3. Although any command slot can be used for any port, this is a useful convention that helps you keep track of your application logic.

Although these examples show ladder logic, you can trigger the command block execution manually. See *Chapter 9. Troubleshooting* for manual triggering techniques. To observe the action of the commands, you will probably want to create a chart using your programming software. See the example in Chapter 9.

7.2. NITP/CAMP Slave

This application configures a serial port to act as a slave for devices that use either the NITP (task code) or CAMP protocol. This allows you to connect operator interfaces such as the CTI 5250-TI5.
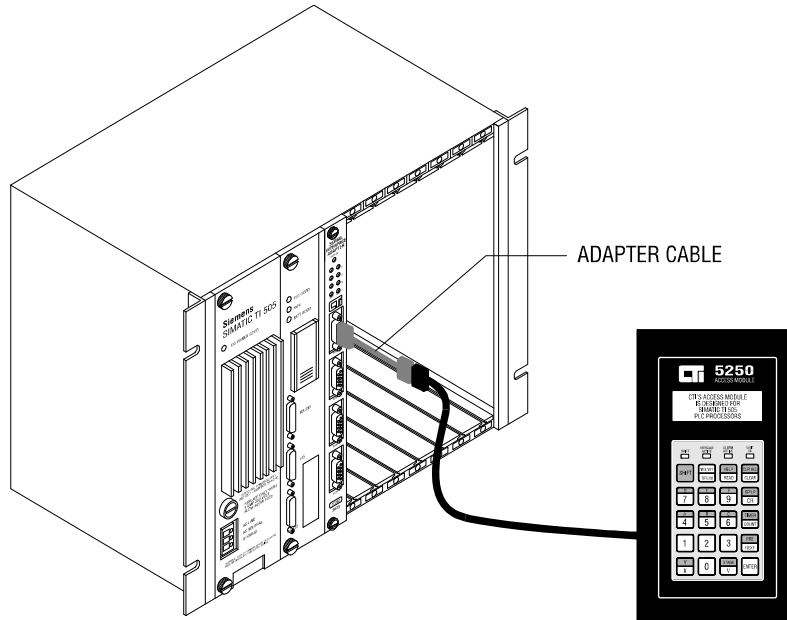
Figure 10. TCAM to 2573 Module

In slave mode, the protocol manager for the serial port will accept and validate incoming NITP or CAMP messages. For NITP, the module will then transfer the task code commands and data directly from the message to the PLC via the backplane. For CAMP, the module will construct the applicable PLC command(s) and transfer the command and data to the PLC via the backplane. When the PLC responds, the module will build the applicable response message and send it via the serial port.

To configure port 1 for a NITP/CAMP Slave with a baud rate of 9600 and no hardware handshaking, you would set all dipswitches for port 1 (on switchblock SW1) to the *off* position. See information in Chapter 2, beginning on page 6 for details. With this switch setting, when the 2573 module is powered on, Serial Port 1 will function as a NITP/CAMP slave. No PLC logic is required.

If you use the standard TCAM cable, you will need to construct an adapter cable. See Appendix A for further information on building cables.

7.3. Task Code Master

In this example, you will use the module to communicate with a remote SIMATIC® 505 Series PLC via the serial port. Although the example illustrates use of the RS-232 serial port, you can use RS-422 communications instead. See *Appendix A* for information on construction of the appropriate cable. Your local PLC logic will cause the module to generate task codes which will read data from the remote PLC V memory and place the results in local PLC V memory.
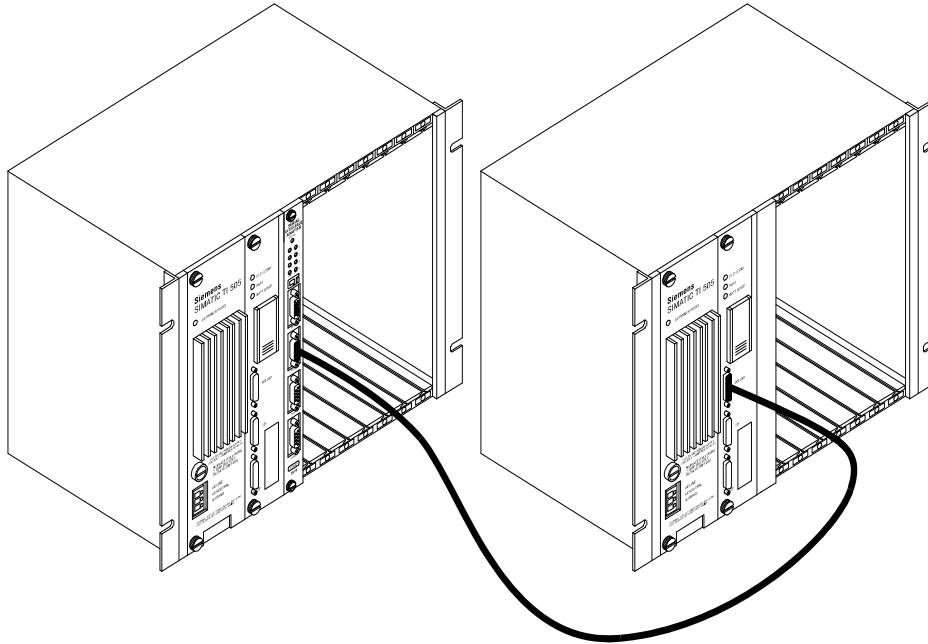
Figure 11. 2573 Module to PLC RS-232 Program Port

NOTE:
You can simulate the remote PLC by cabling to the programming port of the local PLC or to a serial port of the 2573 module configured as a NITP/CAMP slave.

This example will use the same baud rate and hardware handshake settings as the previous one. However, it will use a different port (port 2) and select a different protocol manager (the Task Code Master) using the module dipswitches.

Set the switches on switchblock 2 to select the port protocol and communications parameters (See *Chapter 2. Installation*, page 6). Figure 3 on page 8 illustrates how to select the protocol manager. When the module is powered on, serial port 2 will be configured for the Task Code Master protocol manager using the baud rate determined by the switch setting.

Command Block- Task Code Master Example

To use the Task Code Master, you will need to create some PLC logic. First, use your PLC programming software to store the command block in V memory. This example assumes that you will store the command block in V memory, starting at V100.

Location	Description	Decimal Value
V100	Command Error Word	**0**
V101	Command Code (Read Remote Memory)	**514**
V102	Connection Number (19202)	19202
V103	Source: Starting V Memory Address in Remote PLC	400
V104	Number of Words to Transfer (1-256)	25
V105	Destination: Starting V Memory Address in this PLC	500
V106	Command Timeout in seconds (0=Use Default)	0
V107-V115	Unused (Set to 0)	**0**

The command block above, when triggered by the PLC, will cause the 2573 module to read a block of 25 words from the remote PLC V memory (V400-V424) and to place the results in V500-V524 in the local PLC. Since you used the dipswitch to start the protocol manager on port 2, the Connection Number is 192*02*. See *Chapter 6. Task Code & CAMP Protocol Managers*, page 58, for details on the command block.

PLC Logic - Task Code Master Example

Once you have stored the command block in V memory, you will need to add some PLC logic to cause the 2573 module to actually execute the command. The following ladder logic example will load the address of the command block in WY6 (Command Slot 2) and then trigger the command. It is typical of how you might use an uncoupled mode trigger.

In this example, the module will execute the command as fast as possible as long as the enabling control relay (C19) is on. You might use this mode when you want to continuously update the PLC. The example logic simply acknowledges any errors, without implementing any other error processing logic. Acknowledging the error, by raising WY4.5 when the module raises WX2.5, will cause the 2573 module to lower the Command Error bit (WX2.5) and the Command Busy bit (WX2.7). If the trigger bit (WY4.7) is set, the module will attempt to re-execute the command.

This rung loads the location of the command block in WY6 (Command Slot 2). Control relay C1 is used to seal the rung, so it is executed only once.
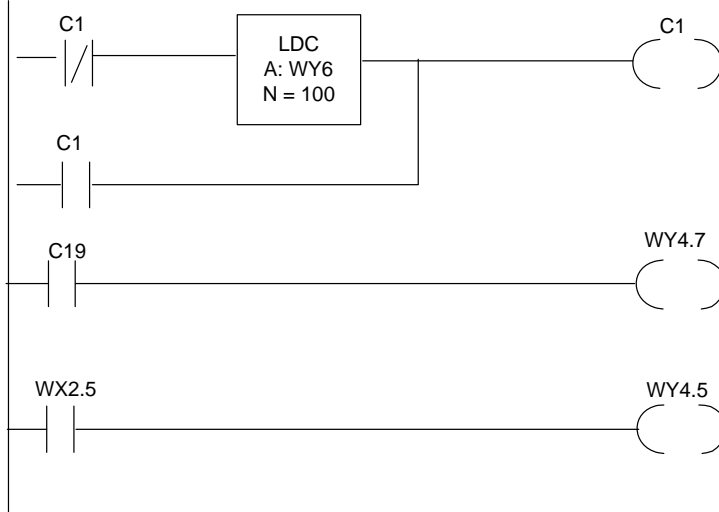
This rung sets the trigger bit on as long as C19 is on. The coupled mode bit (WY4.6) is not set, so the command will execute continuously as long as the trigger is high.

This rung acknowledges the error if the 2573 raises the error bit.

```
      C1                                          C1
 ─────┤/├────┌─────────────┐────────────────────( )──
            │    LDC      │
            │   A: WY6    │
            │   N = 100   │
 ─────┬─────└─────────────┘───┐
      C1                      │
 ─────┤ ├─────────────────────┘

      C19                                       WY4.7
 ─────┤ ├─────────────────────────────────────( )──

      WX2.5                                     WY4.5
 ─────┤ ├─────────────────────────────────────( )──
 │
```

The example above will cause the module to execute the command continuously. However, there are many applications where you want the command to execute only once per trigger. This requires the use of coupled mode. In coupled mode, the module will execute the command once when you raise the trigger. After you raise the trigger, you must lower it and then raise it before the command will execute again. See *Appendix D* for additional information.

The following ladder example uses coupled mode. The second rung is typical of how you should structure all coupled mode trigger rungs. The example also shows how you could keep the last command error word encountered in a special V memory location.
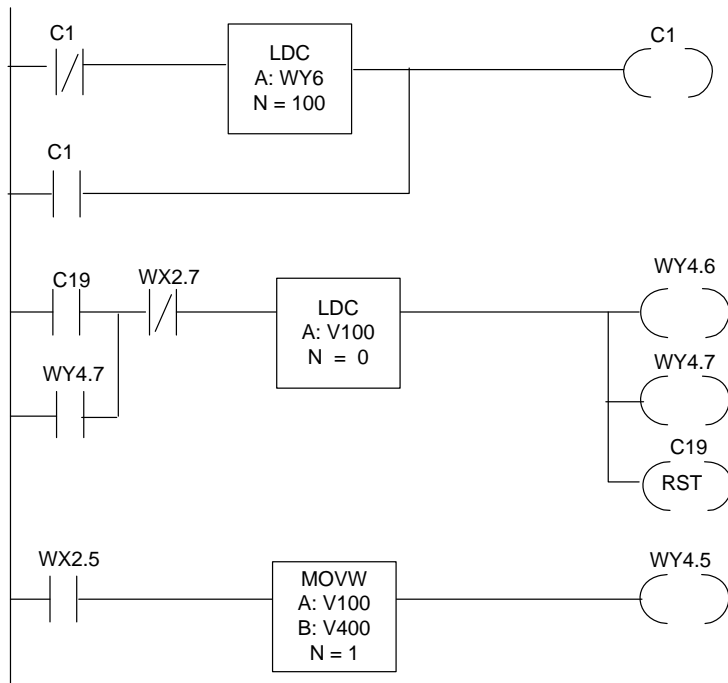
This rung loads the address of the command block in WY6.

This rung illustrates a coupled mode trigger. Note that the mode bit (WY4.6) is set as well as the trigger bit (WY4.7). These bits will be set when C19 is on and Command Busy (WX2.7) is off. After the trigger is set, WY4.7 will hold the bits high on subsequent scans until the Command Busy bit comes on. When the trigger is turned on, C19 is reset, preventing re-triggering for the same event. This rung also clears the error word.

When the command error bit is set, this rung stores the command error word and acknowledges the error.

7.4. Modbus Master

This example illustrates how the 2573-MOD, using the Modbus Master RTU protocol manager, can be used to communicate with devices which use the Modbus RTU protocol. See the specific device documentation and information in *Appendix A. Connectors And Cabling* for construction of a suitable cable.

This example reads a set of 4x holding registers from two Modbus slave flow meters on an RS-485 multi-
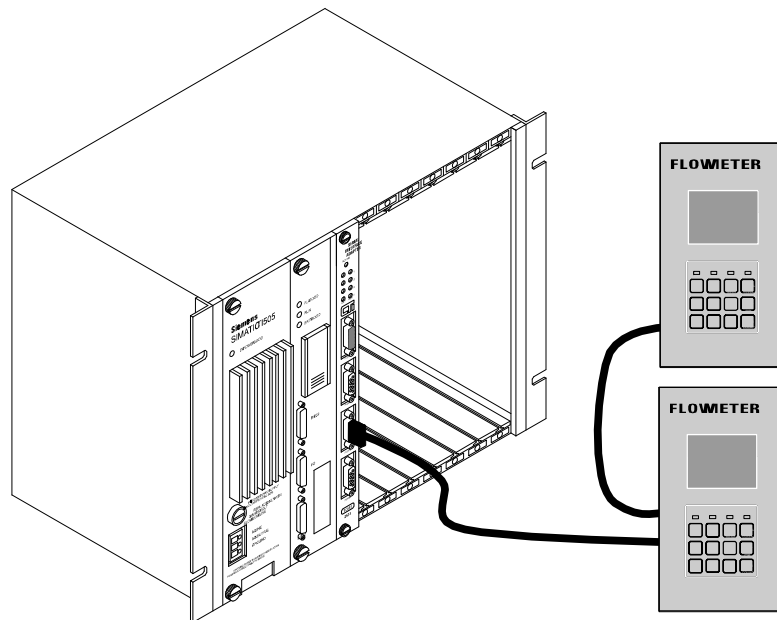
Figure 12. 2573-MOD to Modbus Compatible Flowmeter

drop network. Three holding registers are to be read from the first device, starting at 40108. Five holding registers are to be read from the second device, starting at 40208. The two devices have slave addresses of 1 and 2 respectively and are assumed to be connected to port 3.

The user has decided that any Modbus protocol errors encountered during the processing of the queries in the poll list will stop poll list execution and flag an error. This requirement will be implemented by setting the applicable Command Control flag in the INITIATE POLL LIST command block (bit 16 in V223).

The user also wishes to stop poll list execution and flag an error if the flow meter with Modbus address 2 returns a Modbus exception error. This requirement will be implemented by setting the appropriate Query Control Flag in Query Block #2 (bit 15 in V401).

The data portions of the two Modbus query messages to perform these two Read functions are:

For holding registers at 40108
Starting Address Hi/Lo: 0x006B
No. of Registers Hi/Lo: 0x0003

For holding registers at 40208
Starting Address Hi/Lo: 0x00CF
No. of Registers Hi/Lo: 0x0005

Note:
Modbus addresses are 0 relative and the 40 is implied. Therefore address 40108 is represented by hex 6B (decimal 107) and 40208 is represented by hex CF (decimal 207).

Command Blocks

Create Connection - Modbus RTU

The following command block is used to create a connection to Port 3, using the Modbus Master RTU protocol manager (Number 0x28).

Location	Description	Hex Value	Decimal Value	Comments
V200	Command Error Word	0000	0	
V201	Command (Create Connection)	0001	1	
V202	Connection Number = 19223	4B17	19223	
V203	Protocol Manager Number (Modbus Master RTU)	0028	40	PM number for Modbus Master RTU
V204	Physical Port Number = 3	0003	3	
V205	Port Baud Rate = 9600	2580	9600	Must match device setting
V206	Reserved (set to 0)	0000	0	This word is not used
V207	Parity (0= None, 1 = Odd, 2=Even)	0001	1	Must match device Setting
V208	Stop Bits	0000	0	Use default settings
V209	Handshake (0 = None, 4 = RS-485)	0004	4	RS-485 communication selected
V210	Option Bits	0000	0	Standard RTU timing; PLC does not initiate calls over modem
V211	Modem Configuration Table address V memory	0000	0	Modem not used; no table to reference
V212	RTU End-of-Message Interval	0000	0	feature not being used
V213-215	Unused - reserved for future use (Set to 0)	0000	0	

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Initiate Poll List

The following INITIATE POLL LIST command block is used to send the two queries.

Location	Description	Hex Value	Decimal Value	Comments
V220	Error Word	0000	0	
V221	Command Code (Initiate Poll List - RTU)	2804	10244	
V222	Connection Number = 19223	4B17	19223	Matches Create Connection command block
V223	Command Flags	0001	1	Bit 16 is set, which means to stop Poll List on any protocol error.
V224	Query Timeout	0003	3	The query timeout is specified at 3 seconds. The user has determined that this value is greater the largest query processing time.
V225	Query Block With Error	0000	0	If a Query Block encounters an error that stops poll list execution, the beginning address of the offending Query Block will be written here.
V226	Query Block Polling List Length	0002	0002	In this example there are two queries to perform. So the query list length is 2.
V227	V Memory Address of 1st Query Block	012C	300	Arbitrarily chosen to be at V memory address 300
V228	V Memory Address of 2nd Query Block	0190	400	Arbitrarily chosen to be at V memory address 400
V229-V232	Reserved Space for future expansion of Poll List	0000	0	Reserved for future use. This is optional but recommended.

The INITIATE POLL LIST command shown above points to two Query Blocks that contain the information about the two required Modbus queries. Note that the error flag (bit 16) has been set. This will cause the poll list to stop execution and flag an error to the PLC if *any* query in the poll list encounters a Modbus protocol error. The two Query Blocks referenced above (V memory locations 300 and 400) are shown next.

Query Block #1

The first query will read 3 holding registers from slave address 1.

To send this query data:

> Starting Address Hi: 0x00
> Starting Address Lo: 0x6B
> No. of Registers Hi: 0x00
> No. of Registers Lo: 0x03

V memory 307 should contain 0x006B; V308 should contain 0x0003.

The V memory space required to contain the response for reading 3 holding registers will be 10 bytes. Therefore, V303 should contain 0x000A (decimal 10).

Offset	Description	Hex Value	Decimal Value	Comments
V300	Command Error Word	**0000**	**0**	
V301	Query Block Control Flags	0000	0	No flags set
V302	Response - V memory Address	0FAA	4010	Write response from this query into V memory starting at address 4010
V303	Response - Amount of V memory to reserve	000A	10	Amount of V memory (bytes) to reserve for response
V304	Modbus Slave Address (1 - 247)	0001	1	Slave address 1
V305	Modbus Function Code	0003	3	3 = Read Holding Registers
V306	Number of Bytes of V Memory for Query Data	0004	4	In this example 4 bytes are needed. This equals 2 words of V memory (V307-V308)
V307	Query Data (Bytes 1 and 2)	006B	107	Starting register address to read in slave 1.
V308	Query Data (Bytes 3 and 4)	0003	3	Number of registers to read.
V309	Query Block Validation Code (ASCII "QB")	**5142**	**20802**	Query Block Validation Code

Query Block #2

The second query will read 5 holding registers from slave address 2. The user has decided that if a Modbus Exception Response occurs from this query, poll list execution should stop and flag an error.

To send this query data:

Starting Address Hi:	0x00
Starting Address Lo:	0xCF
No. of Registers Hi:	0x00
No. of Registers Lo:	0x05

V memory 407 should contain 0x00CF; V408 should contain 0x0005.

Since the user in this example wanted to stop poll list execution on Modbus Exception Responses, the flag in V401 bit 15 is set. Since no other flags are going to be set, V401 should contain 0x0002.

The V memory space required to contain the response for reading 5 holding registers will be 14 bytes (10 bytes for data and 4 bytes for function code and byte count). Therefore, V403 should be set to 0x000E (decimal 14) or greater

Offset	Description	Hex Value	Decimal Value	Comments
V400	Command Error Word	**0000**	**0**	
V401	Query Block Control Flags Bit 15 set to 1.	0002	2	Stop poll list execution on Modbus Exception Response to this query.
V402	Response - V memory Address	100E	4110	Write response from this query starting in V 4110
V403	Response - Amount of V memory to reserve	000E	14	Amount of V memory (bytes) to reserve for response
V404	Modbus Slave Address (1 - 247)	0002	2	Slave address 2
V405	Modbus Function Code	0003	3	3 = Read Holding Registers function
V406	Number of Bytes of V Memory for Query Data	0004	4	In this example 4 bytes are needed. This equals 2 words of V memory (V407-V408)
V407	Query Data (Bytes 1 and 2)	00CF	207	Starting register address to read in slave 1.
V408	Query Data (Bytes 3 and 4)	0005	5	Number of registers to read.
V409	Query Block Validation Code (ASCII "QB")	**5142**	**20802**	Query Block Validation Code

Below is an illustration of what the data might look like coming back from the two queries and how they would be placed in V memory.

Assuming the Modbus response data from Query Block #1 contained the following hex data:

Byte Count	0x06
Data Hi (Reg 40108)	0x02
Data Lo (Reg 40108)	0x2B
Data Hi (Reg 40109)	0x01
Data Lo (Reg 40109)	0x1A
Data Hi (Reg 40110)	0x00
Data Lo (Reg 40110)	0x64

After the response has been returned and the protocol manager has checked the response (checksum, response address, and response function code), V memory would contain the following after the command successfully completed:

Location	Hex	Decimal	Comments
V4010	0x0003	3	Response Function Code
V4011	0x0006	6	Response Byte Count
V4012	0x022B	555	Data Register 40108 Value
V4013	0x011A	282	Data Register 40109 Value
V4014	0x0064	100	Data Register 40110 Value

Assuming the Modbus response data from Query Block #2 contained the following hex data:

Byte Count	0x0A
Data Hi (Reg 40208)	0x01
Data Lo (Reg 40208)	0xC9
Data Hi (Reg 40209)	0x02
Data Lo (Reg 40209)	0x82
Data Hi (Reg 40210)	0x00
Data Lo (Reg 40210)	0x80
Data Hi (Reg 40211)	0x01
Data Lo (Reg 40211)	0x00
Data Hi (Reg 40212)	0x04
Data Lo (Reg 40212)	0x00

After the response has been returned and the protocol manager has checked the response (checksum, response address, and response function code), V memory would contain the following after the command successfully completed:

Location	Hex	Decimal	Comments
V4110	0x0003	3	Response Function Code
V4111	0x000A	10	Response Byte Count
V4112	0x01C9	457	Data Register 40208 Value
V4113	0x0282	642	Data Register 40209 Value
V4114	0x0080	128	Data Register 40210 Value
V4115	0x0100	256	Data Register 40211 Value
V4116	0x0400	1024	Data Register 40212 Value

PLC Logic - Modbus RTU Master Example

The following example shows typical PLC logic used when the *PLC Select* dipswitch option is selected. You must execute a CREATE CONNECTION command to start the protocol manager as well as a INITIATE POLL LIST command to communicate the data. Command Slot 3 (WY7) is used for both commands. The particular command to be executed is determined by the value loaded into WY7. If 200 is loaded, then the CREATE CONNECTION command block at V200 will be selected; if 220 is loaded, then the INITIATE POLL LIST command block at V220 will be selected.

The logic must first execute the CREATE CONNECTION command before it can execute any INITIATE POLL LIST commands. To determine when to load the address of the Read Remote Memory command, you must recognize that a command cycle completed while WY7 contained the address of the CREATE CONNECTION command. A command cycle is completed when Command Busy (WX2.11) goes high and *then* goes low. C14 and C15 track this event. C1 is set when WY7 points to V200, C12 is set when WY7 points to V220. See *Appendix D. PLC Command Interface* for related timing diagrams.

NOTE:
Since the example does not stop the Poll List on all errors, additional logic, not shown in this example, should be used to check the Query Block error code before using the data.

```
   WX1.2    C2          ┌─────────┐                              C1
────┤ ├──────┤/├────────┤  LDC    ├────────────────────────────( SET )
    │        │          │  A: WY7 │                    │
    │        │          │  N = 200│                         C12
    │        │          └─────────┘                    ├───( RST )
    │        │                                         │
    │                                                       C14
    │                                                  ├───( RST )
    │                                                 │
    │                                                      C15
    │                                                  ├───( RST )
    │                                                 │
    │                                                      C19
    │                                                  └───( SET )
    │
   WX1.2                                                   C2
────┤ ├───────────────────────────────────────────────────(   )
    │
   WX2.11                                                  C14
────┤ ├────────────────────────────────────────────┬───( SET )
    │                                                │
    │                                                    C15
    │                                                └───( RST )
    │
   C14     WX2.11                                         C15
────┤ ├──────┤/├────────────────────────────────────┬───( SET )
    │                                                │
    │                                                    C14
    │                                                └───( RST )
```
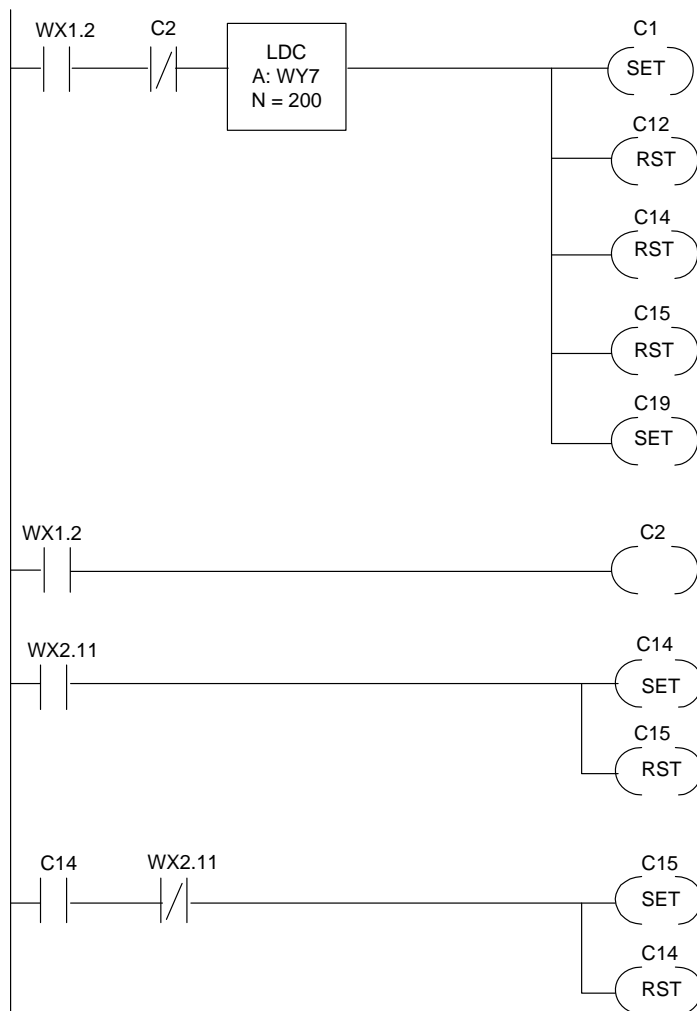
This rung selects the Create Connection command block by loading the V memory address of 200 in WY7. The 2573-MOD will turn the WX1.2 (SER CFG) bit on when the module is reset; it will remain on until all ports have been configured (either by dipswitch or by a Create Connection). Thus this rung is executed which WX1.2 first goes high.

C1 is set high to indicate that WY7 points to the Create Connection command. C14 and C15 are reset to indicate no command cycle is in progress. The trigger enabling relay (C19) is turned on so the command trigger will fire.

When WX1.2 is on, C2 is turned on. This prevents the previous rung from re-executing on subsequent scans.

C14 is used as a flag which indicates that the command cycle was started. The cycle starts when Command Busy (WX2.11) comes on. C15 is reset at the beginning of a command cycle.

Once the command cycle is started (C14 is on), when command busy goes off, the command cycle is complete. The Cycle Complete flag (C15) is set and C14 is reset.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

PLC Logic Example (Continued)
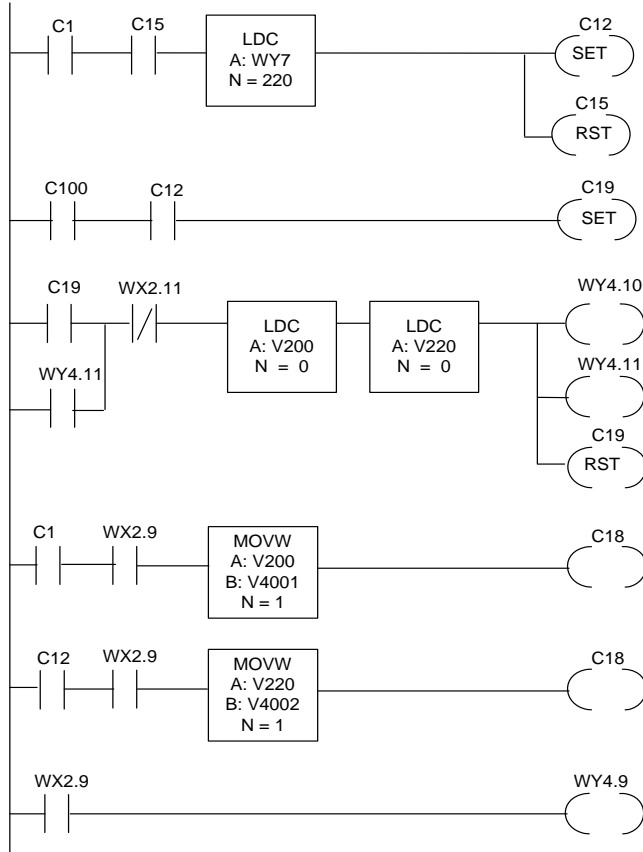
If C1 is on, then WY7 contains the address of the Create Connection command block. When C15 is on, the module has completed a command cycle. When both events are true, you have successfully created a connection and are ready for the Send Query command. This rung load WY7 with the address of this command block (V220), sets C12 to indicate this, and reset the Cycle Complete flag (C15).

This rung is used to enable the trigger for the memory read command, assuming that other logic sets C100 when a particular event happens.

This rung is the command trigger for coupled mode. It works exactly like the one in the previous example except that the Command Error words in both Command Blocks are cleared.

If an error occurs (WX2.9 is on) and the Create Connection command is executing (C1 is on), this copies the Command Error Word to another V memory location.

This rung performs a function similar to the rung above for errors that occur when executing the Send Query command (C12 is on).

This rung raises ERR ACK (WY4.9) when the command error (WY2.9) is raised. When the error is acknowledged, the 2573 will lower command busy and command error. You can then trigger a new command.

CHAPTER 8. 2573 MODEM CONTROL FEATURE

8.1. Overview

> *NOTE:*
> *The 2573 Modem Control Feature has been extensively tested with Hayes brand modems. If you plan on using another brand, you should test your particular application thoroughly. Contact CTI for additional information.*

Modems provide a means for transmitting data across voice grade telephone lines. They operate by modulating digital signals into a series of tones at the transmitting end and de-modulating the tones back into digital signals at the receiving end. When a modem is in *data mode*, characters sent to the modem serial port are transmitted via the telephone line.

Most modern modems can also operate in *command mode*. In this mode, characters sent to the serial port are interpreted as commands to the modem rather than data to be sent out the wire. Some commands may configure the modem while others may cause the modem to dial a number or to hang up a connection.

When your application requires a modem and the PLC is the answering location, you can simply use an auto-answer modem attached to the serial port of CTI communications module or to the program port of the PLC; no unique PLC support is required. However, when the PLC must *initiate* the dial-up connection, special support is required to enable the PLC to control the modem. The CTI 2573 module provides this support via a facility known as the 2573 Modem Control Feature. This feature is available in firmware versions 4.3 and above.

Using the 2573 Modem Control Feature you may:
1. Use PLC logic to establish a dial-up modem connection for sending and receiving data;
2. Use PLC logic to establish a dial-up modem connection and send DTMF* signals;
3. Use PLC logic to hang up the modem connection;
4. Configure the 2573 to automatically establish a modem connection before sending data;
5. Configure the 2573 to automatically hang up a modem connection after a specified period with no data traffic.

> *NOTE:*
> ** DTMF stands for Dial Tone Multi-Frequency. DTMF is commonly referred to as tone dialing. Button phones use tone dialing. Rotary phones typically use pulse dialing.*

The 2573 Modem Control Feature may be used with the NITP (Task Code) Master, CAMP Master, Modbus Master and GAS (General ASCII Support) protocol managers. The 2573 Modem Control Feature is selected by entries in offset 10 and 11 of the CREATE CONNECTION command block.

Once the modem connection is established, the protocol managers use their standard commands to send and receive messages using the protocol manager message format. For example, the CAMP Master protocol manager would send messages using the CAMP protocol. The GAS protocol manager may send messages as configured by the user.

The 2573 Modem Control Feature uses the "AT" modem commands to control the modem. Originally used on modems manufactured by the Hayes corporation, the "AT" command set has become the de-facto standard for modem control. These commands allow you to control the modem operation, including establishing and terminating switched line connections. For example, when dialing a number you may specify a fully qualified number, including access code, area code, and local number as well as characters which cause the modem to wait for dial tones or pause for switching equipment delays.

> *NOTE:*
> *The following two modems have been fully tested and are supported by CTI: Hayes Accura 144 and Hayes Optima 288. Other models and brands, even ones termed "Hayes compatible", may not function correctly with the Modem Control Feature. Contact CTI concerning the use of any other modems.*

Manual Mode

The 2573 Modem Control Feature allows PLC logic to initiate the following functions:

1. Cause the modem to go off hook and to dial a sequence based on a string of ASCII characters stored in PLC V memory. If the modem cannot make the connection because there is no local dial tone, the line is busy, or there is no answer after a specified time, the user logic will be notified via the standard 257x error reporting structure with unique error codes. Switching between Pulse Dial and DTMF can be done via a character in the dial string.

2. Hang up the modem and place the modem in the default state as defined by the Modem Configuration Table and the 2573 Modem Control Feature defaults.

The modem connection is established by using a DIAL command. A HANG-UP command is used to terminate the modem connection. Should the connection be lost, the protocol manager will return an error to the PLC logic when an attempt is made to send a message. The PLC logic is responsible for re-establishing the connection.

Auto-Dial Mode

In the AutoDial mode, the PLC logic is not required to trigger a DIAL command. The 2573 Modem Control Feature will determine whether to initiate a dial connection when the PLC logic attempts to send a message.

If the modem carrier is present, the message will be sent immediately. If there is no modem connection, the 2573 Modem Control Feature will attempt to make the connection based on a default dial string stored in V memory. If the connection is successful, the message will be sent, else an error code will be returned to the PLC logic in the command block.

If no data has been sent or received via the modem link after a user specified time period, the protocol manager will hang up the connection (Auto Hang-Up). Alternately, PLC logic may explicitly issue a hang up command.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

8.2. Modem Setup

When the 2573 Modem Control Feature is initiated, it will attempt to set the modem to the following:

> Result Codes in Verbose Mode
> Ignore DTR
> Always Assert DSR.
> Suppress Echo

If possible, you should use the modem's configuration program to set the modem defaults to this configuration. In addition, you should set the modem default configuration to force a fixed baud rate between the modem and the 2573.

NOTE:
Some modems change the baud rate "on the fly" when data line communications rates change. The 2573 firmware does not support variable baud rates. Therefore, you must set the modem default to fixed baud rate or you may experience intermittent failures and data errors.

The cable between the modem and the 2573 serial port should not connect DTR and DSR signal lines. See Figure 13 in the module Installation and Operation Guide for an example of a cable that connects only transmit, receive, and ground.

Modem Configuration Table

The 2573 Modem Control Feature uses a block of V memory for a Modem Configuration Table. This table contains certain default parameters which will be used by the 2573 Modem Control Feature. It must be stored in V memory before the 2573 Modem Control Feature can be initiated. The Modem Configuration Table is read when the CREATE CONNECTION command is executed.

The Modem Configuration Table has the following structure:

Offset	Description
0	Signature Value
1	Reserved for Future Use. Set this value to 0.
2	Auto Answer Rings (0 - 9) (0 = No Auto Answer)
3	Auto-Dial Selection (0 = Manual Dial, 1 = Auto Dial)
4	Modem Option Flag Bits 1 - 13 (Unused) Bit 14: Busy Detection (0 = Enabled, 1 = Disabled) Bit 15: Dial Tone Detection (0 = Enabled, 1 = Disabled) Bit 16 - Default Dial Method (0 = DTMF, 1 = Pulse Dial).
5	Auto Hang-Up Timeout in Seconds (0 = No auto hang-up)
6	V Memory Address of Default Dial String
7	AutoDial Timeout in Seconds
8 - 15	Reserved for Future Use (Set these values to 0)

Offset 0   *Signature Word*. The signature word is a unique number which identifies the start of a Modem Configuration Table. It is used to help ensure that the correct memory block is referenced. It is the ASCII equivalent of the letters AT (0x4154)

Offset 1   *Reserved*. You should set this value to 0.

Offset 2   *Auto Answer Rings*. This word specifies the number of rings before the modem will auto answer. If this value is non-zero, the modem will be placed in auto answer mode after a Hang-Up command has been executed. ***This value should be set to 0 unless you explicitly need to support both dial-up and auto answer on this line.***

NOTE:
Auto-answer is not supported by the Task Code Master or the CAMP Master protocol managers. When using the auto-answer mode on the same line as dial-out, it is possible to place some modems in a state from which the 2573 cannot command a recovery. To minimize the chance of this occurrence, you must set the attention guard band value (S register 12) to a value of 0. See the documentation for your modem.

Offset 3   *Auto-Dial Option*. Selects the automatic dial option. If the value is set to 0, PLC logic must explicitly establish the dial-up connection. If the value is set to 1, then the 2573 will automatically attempt to make the connection using information stored in the Modem Configuration Table.

Offset 4   *Modem Option Bits*. This word contains bits that specify certain modem defaults. The lower bit (bit 16) determines the default dial mode (tone or pulse). In manual dial mode, you may override the defaults in the Dial command block.

     CTI 2573-MOD2/TCM2 Installation and Operation Guide

Offset 5   *Auto Hang-Up Timeout*. This word used to specify that the 2573 should issue a HANG-UP command if no messages have been sent or received within the timeout interval. A value of 0 specifies the default value of no Auto Hang-Up. The maximum value is 3600 seconds; the minimum non-zero value is 60 seconds. Auto Hang-Up timeout applies to both Auto-Dial and Manual Dial modes.

Offset 6   *Dial String V Memory Address*. This value is the address of the V memory word that contains the first character of the dial string. It must be greater than 0 if AUTO-DIAL is selected in offset 3 in the Modem Configuration Table. The dial string must contain only valid digits and the special characters listed in the following section of this manual. Since the dial string is variable length, the dial string must be enclosed in ASCII double quote marks.

NOTE
Other required modem control characters, including the AT header, progress monitoring modifier, and command state modifier, will be inserted by the 2573 Modem Control Feature.

NOTE:
If you use the standard ASCII entry format of TISOFT, the characters will be stored in this sequence.

The maximum length of the dial string is 64 characters. Some older modems allow only 40 characters. You must ensure that you avoid exceeding the modem limit.

Offset 7   *AutoDial Timeout*. This sets the timeout for the AUTODIAL function in seconds. Minimum value is 30 seconds. 0 = 2573 default of 30 seconds.

NOTE:
In Auto-Dial mode, this timeout value is added to the timeout value specified for the protocol manager command.

Offset 8 - 15   *Reserved*. Reserved for future use (ignored by this release). You should set these words to 0.

Dial Strings

Dial strings are a sequence of characters which are sent to the modem. The dial strings indicate the sequence of numbers to be dialed and control how the numbers are dialed. The dial string may consist of the ASCII characters 0 - 9 (0x30 - 0x39), * (0x2A), # (0x23), and other characters as described below. Dial strings are stored in PLC V memory and must be enclosed in double quote marks (0x22).

Dial Modifier Characters

Dial modifier characters control the transmission of the dial characters. The following dial modifier characters are supported.

ASCII Char	Hex Value	Description	Use
,	(0x2C)	Calibrated Pause	Pauses two seconds before continuing with the dial string. For longer pauses you may use multiple consecutive commas.
R	(0x52)	Reverse Mode of Operation	This allows you to originate a call, then go into answer mode. Used when communicating with an "originate only" modem, a vanishing breed. This modifier should be used with caution.
!	(0x21)	Switch Hook Flash	Simulates a quick cycling of the switchhook. Some switchboard systems react to the flash to transfer to a specific extension.
@	(0x40)	Wait for Silence	This waits for one or more ringbacks followed by a 5 second period of silence. It is used for accessing systems that do not provide a dial tone.
P	(0x50)	Pulse Dialing	Sets pulse dialing mode. Pulse dialing is a method used by rotary-dial telephones. Dial characters following the P will be sent using pulse dial.
T	(0x54)	Tone Dialing	Sets the tone dialing mode. Tone dialing is a method used by push-button (touch tone) phones. This is also called DTMF (dual tone multi frequency). Dial characters following the D will be sent using tone dial.
W	(0x57)	Wait for Additional Dial Tone	The modem will automatically wait for the first dial tone before proceeding. This modifier will cause the modem to wait for a second dial tone before proceeding. This is commonly used when you must route through a local switchboard. This modifier will not work if dial tone detection is disabled.
$	(0x24)	Wait for Credit Card Tone (Supported by some modems).	This will cause the modem to wait for the AT&T "calling card" tone before proceeding. Check your modem documentation to determine support.

Miscellaneous Control Characters

The following characters may be embedded into the dial string to control modem functions.

ASCII Char	Hex Value	Description	Use
L	(0x4C)	Speaker Volume Control L0 = off or very low L1 = Low L2 = Medium L3 = High	Sets the volume of the modem speaker. You may wish to set this parameter default manually (using the modem setup software) rather than controlling it in the dial string.
M	(0x4D)	Speaker Control M0 = Always Off M1 = On until carrier link is established. M2 = Always On M3 = On except during dialing, then on until carrier is established.	Controls when the speaker is enabled. The audio signal from the speaker can be used to monitor call progress. The 2573 Modem Control Feature will default to M1 if no value is included in the dial string.
&J	(0x264A)	Auxiliary Relay Control	This is used to turn on/off an auxiliary relay. Some radio modems may require this relay to turn on the transmitter before sending.

Alphabetic characters may be entered in upper or lower case. You may insert the following characters in the dial string for readability.

Character	Description	Hex Equivalent
(	Left Parenthesis	0x28
)	Right Parentheses	0x29
-	Dash	0x2D
.	Period	0X2E
	Space	0x20

If a character other than those indicated in the sections above appears in the dial string, the protocol manager will return an error code (invalid dial string).

Example Dial Strings

To dial the area code 423 and the number 584-0440, you could use one of the following: "4235840440" or "423-584-0440" or "(423)-584-0440".

"9,,1-800-537-8398" will dial a 9 (access to the switchboard), wait 4 seconds, then continue the dial. You could also use the following dial string: "9W1-800-537-8398" to accomplish the same thing, where the W waits for a second dial tone.

8.3. Command Blocks

Create Connection

The CREATE CONNECTION command block is used to start a protocol manager on a specific serial port. For the protocol managers which implement the 2573 Modem Control Feature, the standard command block has been extended to make use of offsets 10 and 11. If the low bit (bit 16) of Offset 10 is 1, the Modem Configuration Table pointed to by Offset 11 is read. See the appropriate chapter for additional information on using the CREATE CONNECTION command with a specific protocol manager.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Create Connection)	**0001**	
2	Connection Number (19221 -- 19299)		
3	Protocol Manager Number		
4	Physical Port Number (1,2,3, or 4)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Bits Per Character (7 or 8)		
7	Parity (0 = none, 1 = Odd, 2 = Even)		
8	Stop Bits (1 or 2)		
9	Handshake (0=none, 1= Software, 2 = Hardware). *If you choose handshaking, you must manually configure the modem to use it.*		
10	Special Option Bits (0x0000 = None, 0x0001 = Modem) Bit 16 must be set to indicate modem use. Table referenced in offset 11 will be read if bit is set.		
11	V Memory Address of Modem Configuration Table		
12 - 15	Reserved		

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for CREATE CONNECTION is 1.

Offset 2   *Connection Number* - Any valid number within the range 19221 to 19299 may be assigned as long as it is has not been used previously. For clarity, you may wish to set the lower digit to match the physical port number (e.g. 1922**1** for port 1, 1922**2** for port 2).

Offset 3   *Protocol Manager Number* - Set to 36 (Hex 0024) to select the CAMP Master protocol manager. Set to 38 (Hex 0026) to select the GAS protocol manager. Set to 32 (Hex 0020) to select the NITP (Task Code) Master protocol manager.

Offset 4   *Physical Port Number* - Set to the physical port (1-4) you wish to use.

Offset 5-8   *Communications Parameters* - This must match the characteristics of the remote device. CAMP and Task Code Master protocols require the following settings: Bits per Character = 7, Parity = Odd, Stop Bits = 1. GAS may be set to any value which matches the attached device.

Offset 9   *Handshake-* This allows you to select the method of flow control. Unless you have a special requirement, you should set this value to 0, no handshake.

NOTE:
The use of handshaking with a modem can be complex, since several implementations of handshaking may be available, including "passing through" software handshake to the remote device. For this reason, if you choose to use handshaking, you must manually configure the modem to default to the handshake option you wish to use.

Offset 10   *Option Bits* - Set low order bit (bit 16) to 1 to select the 2573 Modem Control Feature. When this bit is set, offset 11 must contain a valid address for the Modem Configuration Table.

Offset 11   *V Memory Address of the Modem Configuration Table* - This value must contain the V memory address of the first word (signature word) of the Modem Configuration Table.

Dial Command

The following command block will be used to initiate a series of modem commands which will take the modem off hook and dial up the remote user. *If you have selected Auto-Dial, you are not required to initiate this command.*

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Dial Up))	**0401**	
2	Connection Number (19221 -- 19299)		
3	Command Flag - (see below)		
4	V Memory address of the dial string (0 is invalid)		
5	Busy Signal Detection (0 = Enable, 1 = Disable)		
6	Dial Tone Detection (0 = Enable, 1 = Disable)		
7	Command Complete Mode (0= Data Mode, 1 = Command Mode)		
8	Command Timeout in Seconds		
9 - 15	Reserved - Set to 0	0000	0

Offset 0   *Command Error Word* - Your PLC logic should set this to 0 so that any previous error code is cleared. If a processing error is encountered, the protocol manager will write an error code into this word.

Offset 1   *Command Code* - The Command for Dial is 0x0401

Offset 2   *Connection Number* - The connection number must match the one used in the CREATE CONNECTION command.

Offset 3 *Command Flag*

Bits 1 - 15	Bit 16
Unused by this release and should be set to 0.	0 = Reset Modem before dial command and load CTI defaults. 1 = Do not reset modem or load CTI defaults before dial command.

NOTE:
Except for special cases, you should set offset 3 to a value of 0. Disabling automatic reset could result in the modem being placed in a state unknown to the protocol manager.

Offset 4   *Dial String Location.* V memory Address which contains the dial string. See additional information regarding the dial string in the following section.

     CTI 2573-MOD2/TCM2 Installation and Operation Guide

Offset 5   *Busy Signal Detection*. Determines whether the modem detects a busy signal. This should be set to 0 (Enable) unless there is a specific reason to do otherwise.

Offset 6   *Dial Tone Detection*. Determines whether the modem will detect a dial tone signal. This should be set to 0 (Enable) unless there is a specific reason to do otherwise. You might set this to 1 (disable) if you are using the dial command to send additional tone commands to a DTMF device. If this function is disabled, dial modifiers which specify waiting on a dial tone (such as the W modifier) will not work; you must use a timed delay instead.

Offset 7   *Command Complete Mode*.

                   0 = Place in data mode after completing command. This is the normal mode to send data to a modem.

                  1= Leave in command mode after completing command. *This bit is used when sending DTMF commands to devices which do not present a modem "carrier" (for example, pagers).*

NOTE:
When you leave the modem in command mode, some modems do not return an error message if the dialed number was busy or did not answer. If you use this option, you should test this condition using your modems.

The command will be considered complete when the modem signals that the dial command is completed. An error code will be returned if the connection could not be made (See Error Code listing).

Hang-Up Command

This command will terminate the connection and place the modem on hook. If specified in the Modem Configuration Table, the modem will be placed in auto answer mode.

Offset	Description	Hex Value	Decimal Value
0	Command Error Word	**0000**	**0**
1	Command (Hang Up)	**0402**	**1026**
2	Connection Number (19221 -- 19299)		
3-15	Reserved - set to 0	0000	0

In the event a hang-up command is issued and the modem is already on-hook, no error will be returned to the PLC logic.

NOTE:
In manual mode, if a dial sequence is aborted for any reason (including modem errors), you should issue the hang-up command to reset the modem to a known state.

8.4. Application Example

The following illustrates how the 2573 Modem Control Feature can be used. The example assumes the following:
1. You have acquired compatible modems configured as described in the previous section.
2. You have a "local" PLC configuration which includes a 2573 with the modem connected to port 3.
3. You have a "remote" location which also has a 2573/PLC combination.
4. On a certain event, you wish to connect to the remote site and transfer 150 words to the remote PLC.
5. You want to use the auto-dial mode of 2573 Modem Control Feature, to avoid having to provide additional PLC logic.
6. You want to automatically hang up the connection if no data has been sent or received within 60 seconds.

Since both sites have a 2573 installed, we will use the CAMP protocol to transfer the data. CAMP will be the fastest and most reliable choice when both sites have a CTI 2573 Serial Interface Adapter module installed.

In your particular application, you might substitute a remote computer for the remote PLC and use the GAS protocol manager rather than CAMP, since GAS allows you to design your own protocol.

Modem Configuration Table

This table is entered in to V memory starting at V1000

Location	Value	Description
V1000	0x4154	Signature Value = ASCII characters AT
V1001	0	Reserved for Future Use. Set this value to 0.
V1002	0	Auto Answer Rings (0 - 9) **(0 = No Auto Answer)**
V1003	1	Auto-Dial Selection (0 = Manual Dial, **1 = Auto Dial**)
V1004	0x0000	Modem Option Flag Bits 1 - 13 (Unused) Bit 14: Busy Detection (**0 = Enabled**, 1 = Disabled) Bit 15: Dial Tone Detection (**0 = Enabled**, 1 = Disabled) Bit 16 - Default Dial Method **(0 = DTMF**, 1 = Pulse Dial).
V 1005	120	Auto Hang-Up Timeout in Seconds (0 = No auto hang-up)
V1006	1016	V Memory Address of Default Dial String
V1007	0	AutoDial Timeout in Seconds (**0= default of 30 seconds**)
V1008- V1015		Reserved for Future Use (Set these values to 0)

Dial String

The following dial string is entered into V memory, starting a V1016 (referenced in V1006 above).
```
"1,4235551216"
```
This will dial the long distance access code, pause for two seconds, then dial the rest of the number. The string is stored in V memory as shown below.

Location	V1016		V1017		V1018		V1019		V1020		V1021		V1022	
Hex Value	22	31	2C	34	32	33	35	35	35	31	32	31	36	22
ASCII	"	1	,	4	2	3	5	5	5	1	2	1	6	"

Create Connection Command Block

The following Command Block can be used to start up the CAMP Master protocol manager on the 2573 port 3 using a baud rate of 9600.

Offset	Description	Decimal Value
V200	Command Error Word	0
V201	Command (Create Connection)	1
V202	Connection Number 19223	19223
V203	CAMP Master Protocol Manager Number	36
V204	Physical Port Number 3	3
V205	Port Baud Rate = 9600	9600
V206	Bits Per Character = 7 (required setting for CAMP)	7
V207	Parity = 1 (required setting for CAMP)	1
V208	Stop Bits = 1 (required setting for CAMP)	1
V209	Handshake = 0 = none,	0
V210	Special Option Bits 0x0001 = Modem	1
V211	V Memory Address of Modem Configuration Table	1000
V212 215	Reserved	0

Memory Transfer Command Block

The following command block will read 150 words of data from V 2000 in the local PLC and write them to V3000 in the remote PLC.

Offset	Description	Decimal Value
V220	Command Error Word	0
V221	Command (Write Remote Memory)	513
V222	Connection Number (19223)	19223
V223	Source: Starting V Memory in this PLC	2000
V224	Number of Words	150
V225	Destination: Starting V Memory in Remote PLC	3000
V226	Command Timeout (0 = default = 9 seconds)	0
V227- 235	Reserved, Set to 0	0

Since the 2573 Modem Control Feature has been configured for auto-dial and auto-hang-up, the PLC logic to use the dial up modem is *exactly the same* as the logic to initiate data transfer using dedicated lines. See the PLC logic examples in the previous chapter. However, you may wish to add special error identification and recovery logic.

No PLC logic is required to initiate the dial-up connection or to hang up after the data has been sent. If the modem connection is not established when you trigger the command block, the 2573 will automatically initiate the dial sequence using the values stored in the Modem Configuration Table. When the 2573 Modem Control Feature must dial the connection, it calculates the timeout by adding the Auto-Dial timeout of 60 seconds (see offset 7 of the Modem Configuration Table) to the Command Timeout of 9 seconds specified in V226 above. If the connection cannot be made, the 2573 will return an error code and set the error bit. After your logic acknowledges the error, the 2573 will retry the dial sequence when you trigger the command block again.

Hang-Up

After a period of no activity specified in offset 5 of the Modem Configuration Table, the 2573 will automatically hang up the modem. If you wish, you can use PLC logic to explicitly hang up the connection even though the 2573 is configured for auto-hang up. To explicitly hang up the modem, you would use the following command block. You would also need to add the applicable PLC logic to trigger the hang-up command.

Offset	Description	Decimal Value
0	Command Error Word	0
1	Command (Hang Up)	1026
2	Connection Number = 19223	19223
3-15	Reserved - set to 0	0

8.5. Implementation Tips

If you are developing a complex application using the 2573 Modem Control Feature, you may wish to isolate the implementation and testing modem part of the application from the implementation and testing of the serial data transfer portion. A typical method for accomplishing this is as follows:

1. Develop and test the PLC data transfer using direct connected serial lines. If you are using the auto-dial and auto-hang-up features, you need only to set the modem Option Bit in the CREATE CONNECTION command to 0. If your are using the manual dial feature, you will need to disable that portion of your PLC logic.

2. Check out the modems using a standard PC application (such as Microsoft Terminal) to establish the dial connection and to transfer data.

3. Once both portions of the application are operating successfully, then integrate the two by connecting the modem to the PLC and enabling the 2573 Modem Control Feature.

Please ensure that you have set up the modems as explained in Section *8.2. Modem Setup* on page 83.

CHAPTER 9. TROUBLESHOOTING

9.1. General Module Problems

When the module is first powered up, the red Active LED at the top of the module should light after a few seconds. If the Active LED remains on, the module hardware is operational. If the LED blinks, the module diagnostic software has detected a hardware problem. If the LED fails to illuminate and power has been applied to the module, there is a problem with the hardware; the module should be replaced.

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
Active LED not Lit	No Power to Module.	Ensure that power is supplied to the rack and that the module is securely inserted.
	Defective Module.	Cycle the power. If the Active LED does not remain on, replace the module.
Active LED cycles off and on about once every two seconds.	The 2573 diagnostic program has detected a defective module component.	Cycle the power. If the problem persists, replace the module.
Active LED cycles off and on about once a second.	The 2573 has been put in Listen Only mode while acting as a Modbus Slave.	None. The Active LED is just indicating a non-standard mode of operation.

9.2. Serial Communications Problems

Common sources of communications problems are bad or improperly wired cables and improperly set communications parameters. The receive (RCV) and transmit (XMT) indicators can be a valuable aid in troubleshooting these errors. The RCV LED illuminates when a signal is received at the serial port receive line. The XMT LED illuminates when the 2573 sends a signal to the serial port.

In addition, you can use the Loopback setting for a port in conjunction with a loopback connector to verify the serial port operation and cable wiring. See *Chapter 2. Installation* for the dipswitch settings and *Appendix A* for instructions on making a loopback connector.

A valuable tool in locating and correcting communications problems is a serial line analyzer. A serial line analyzer displays the actual messages transmitted via the serial line. Several vendors offer software products which work with a PC to perform serial line analyzer functions. Typically, these products may be purchased for about $300.00 or less.

See the following table for a list of symptoms and possible solutions.

SYMPTOM	PROBABLE CAUSE	CORRECTIVE ACTION
RCV indicator glows steadily when you attach the communications cable.	You are using an RS-232 communications cable which straps pin 6 high. Many cables designed to work with the 545 PLC program port strap pins 1 & 6. In the standard PC/AT 9 pins serial port, pin 6 is Data Set Ready. On the 2573, pin 6 is RS-422a Rx-.	Replace the cable with one properly configured or construct an adapter cable which does not connect pins used for RS-422. See Appendix A for cabling instructions.
RCV Indicator does not flash when you attempt to send data to the 2573 from a device attached to the serial port.	No signal is being received by the 2573. This is probably due to a defective cable. It may be due to a faulty device or, if you are using a PC, a faulty communications card.	Check the communications cable wiring. Replace with a known good cable and retry. If failure still occurs, try another device or communications card.
	If you are communicating using RS-422/485 on Rev. D boards and above, the port's Disconnect 422/485 switch may be inadvertently set to the 'OFF' (disconnected) position.	Set the port's RS-422/485 Disconnect switch to 'ON' (connected) position. See page 10 for settings switches.
When using the NITP/CAMP Slave protocol: a signal is received from the attached device but the 2573 does not respond (XMT LED does not flash).	The 2573 is receiving a message which it cannot interpret. Check the following:	
	Communications Parameter Mismatch.	Check communications settings for the device and module.
	Incorrect Protocol. The attached device is not using NITP or CAMP message protocols.	Refer to the device documentation. Try a device known to use NITP or CAMP such as a CTI 5250-TI5.
	Hardware Handshaking Mismatch. The 2573 is waiting on the attached device to raise the handshake line before sending.	Enable hardware handshaking on the attached device or DISABLE hardware handshaking on the 2573.
	Cable is not wired for hardware handshaking and hardware handshake is enabled in the 2573	Check cable for proper wiring. Temporarily DISABLE hardware handshake on the 2573 to test results.
When using the NITP/CAMP Slave protocol manager: No signal is received by the attached device AND the 2573 XMT light flashes.	This is probably due to a defective cable. It may be a defective device.	Replace cable with one known to be good. Try an alternate device.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

When you attempt to send a message, no signal is received by the remote device **and** the XMT LED **does not** flash.	You are not sending a message.	Check the PLC logic used to Create Connection and to send the message. Inspect the Command Error Word for an error code. If one is present see Appendix B for an explanation. Ensure that hardware handshake is not erroneously enabled.
When you attempt to send a message, the XMT LED **does flash** but no signal is received by the remote device.	The cable is defective or improperly wired.	Check the communications cable wiring. Replace with a known good cable and retry. If failure still occurs, try another device or communications card.
When you attempt to send a message, the XMT LED **does flash** and a signal is received By the remote device but the device does not respond.	Protocol Problem The message you are sending is not understood by the device.	Ensure that the remote device supports NITP or CAMP.
	Communications Parameter Mismatch.	Ensure that the settings for data bits, stop bits, parity and baud rate match.
Intermittent Communications Problem.	Communications "noise."	Reduce the baud rate on the 2573 and the attached device and retry. Re-route or replace cabling.
	Ground Loop This may occur when more than one device is connected via separate cables to the module and the devices have different ground potentials.	Remove cables to other ports. If the problem subsides, install isolation. The 2573 module does not provide channel to channel isolation.
RS-485 does not function but other electrical interfaces work properly.	The Create Connection command for the protocol manager did not select the RS-485 handshake option in offset 9.	Correct the entry in the Command Block. Note that only Modbus and GAS protocol managers support RS-485.

Communications Cables

Ensure that the cables are properly wired. Refer to the documentation for the plant floor device and the wiring diagrams shown in Appendix A.

Communications Parameters Mismatch

First, check to ensure that the baud rate settings on both the 2573 module and the plant floor device are set to the same data rate. If you note a problem, set both to the same baud rates and retry. Also check the data bits, parity, and stop bit settings.

Hardware Handshaking Error - RS-232

If your device is NOT configured for hardware handshaking, ensure that hardware handshaking is DISABLED. If your device IS configured for hardware handshaking, ensure that the handshaking wires are properly connected (see the documentation for the connected device). In some cases you can check out a handshake wiring problem by temporarily disabling hardware handshaking on the module and the plant floor device. If the 2573 responds to the attached device (XMT LED flashes), then you probably have a wiring problem.

Hardware handshaking is not used with RS-422-A or RS-485 electrical interfaces.

Protocol Problems

When the port is set for NITP mode, the attached device must issue and respond to task code packets as defined in the Non-Intelligent Terminal Protocol (NITP). If you are unsure of this fact, you can attach the device directly to the programming port to determine if it works properly. If it does not work and cabling/configuration items discussed in other sections above have been checked, then you probably have a protocol problem.

If it does work, there is still a chance that the device is using Transparent Byte Protocol (TBP), which is not supported in the 2573 module. For example, TISOFT will attempt to use TBP before switching to NITP. Most application programs and operator interface devices that can use TBP allow you to force the use of Non-Intelligent Terminal Protocol (NITP), which is supported.

Intermittent Communications Problems

If you experience intermittent communications failures, you may be experiencing noise on the communications cable which interferes with data transmission. If you are communicating at rates higher than 1200 baud, you can reduce the baud rate to check for improvement. If reducing the baud rate does correct the problem, you most likely have a noise problem. You should first ensure that you have not exceeded the maximum distances for cabling lengths (50 feet for RS-232 and 4000 feet for RS-422). Also check for proper grounding of the cable shields; we recommend connecting the shield at one end only. Longer cable runs may be subject to ground loops; you may require the addition of an electrical isolation device.

9.3. Development and Debugging Tips

Manual Triggering

You can test your command blocks and format specifications separately from the PLC logic that actually executes them by manipulating the command control bits manually. First, place the PLC in program mode so that the PLC logic will not be executing. Then you can manipulate the bits as shown below. **If you are not familiar with the 2573 PLC Command Interface, you should read Appendix D**. Also see the WX/WY Quick Reference in Appendix C.

The command control bits are located in the 4th module word. For example, if you logged the module in starting at Word 1, then the command control bits will be located in WY4. See the WX/WY Quick Reference at the end of this manual. There are 4 sets of control bits, one set for each command slot.

WY4	Bits 1 -4	Bits 5 - 7	Bits 8 - 11	Bits 12 -16
	Command Control Bits - Slot 1	Command Control Bits - Slot 2	Command Control Bits - Slot 3	Command Control Bits - Slot 4
	Hex 0-F	Hex 0 - F	Hex 0 - F	Hex 0 - F

Within each set of four Command Control bits:
- The first bit is the Error Acknowledge,
- The second bit is the Command Mode bit
- The third bit is the Command Trigger
- The fourth bit is the Abort Trigger.

So a bit pattern of 0110 will set the command mode bit and the command trigger. Similarly, the bit pattern 1000 will set the Error Acknowledge bit. Using hexadecimal notation is a convenient way to observe and manipulate these bits, because each hexadecimal digit represents 4 bits. Thus, bit pattern 0110 is represented by hexadecimal 6 (0+4+2+0). When the error acknowledge bit on is represented by hexadecimal 8 (1000).

Similarly, the command status bits written by the module are located at the second module word (WX2 in this example) in a bit grouping that matches the command control word.

WX2	Bits 1 -4	Bits 5 - 7	Bits 8 - 11	Bits 12 -16
	Command Status Bits - Slot 1	Command Status Bits - Slot 2	Command Status Bits - Slot 3	Command Status Bits - Slot 4
	Hex 0-F	Hex 0 - F	Hex 0 - F	Hex 0 - F

Within each set of four Command Status bits:
- The first bit is the Command Error bit,
- The second bit is the PLC Error bit,
- The third bit is the Command Busy bit,
- The fourth bit is the Abort Busy bit,

Therefore, to set a command trigger only (uncoupled mode) for the first command slot, set WY4 to hex 2000. To set the command trigger for the second slot, set the WY4 to hex 0200. To set the error acknowledge for the third Command Slot, set WY4 to hex 0080.

When you are using PLC logic to control the PLC, it is often useful to display the module WX/WY words, parts of the command blocks, and sections of the format specifications as a chart. Following is an example chart.

```
LOCATION      STATUS        LOCATION      STATUS        LOCATION      STATUS
WX1    =      HEX           V200   =      HEX           V500   =      INTEGER
WX2    =      HEX           V201   =      INTEGER       V501   =      INTEGER
WY3    =      HEX           V202   =      INTEGER       V502   =      INTEGER
WY4    =      HEX           V203   =      INTEGER       V503   =      INTEGER
WY5    =      INTEGER       V204   =      INTEGER       V504   =      INTEGER
WY6    =      INTEGER                                   V505   =      INTEGER
WY7    =      INTEGER       V220   =      HEX           V600   =      INTEGER
WY8    =      INTEGER       V221   =      INTEGER       V601   =      INTEGER
                           V222   =      INTEGER       V602   =      INTEGER
                           V223   =      INTEGER       V603   =      INTEGER
                           V224   =      INTEGER       V604   =      INTEGER
                                                       V605   =      INTEGER
```

The above chart allows you to examine at the value of the module status word (WX1), the command status word (WX2), the module control word (WY3), and the command control word (WY4) in hexadecimal format. The command slots (WY5-WY8) are displayed as integers. V200-V215 and V220-V235 are assumed to contain command blocks you are using. This display shows the first 4 words of each the command block with the error words in hex. V500-V505 and V600-V605 are assumed to contain data you want to observe. You will want to change the display to match your setup.

To trigger a coupled mode command for command slot 1 (Assuming the module is logged at Word WX1):

1. Put the address of the Command Block you want to use in WY5.

2. Change the value of WY4 to hex 6000. WX2 should change to hex 2000, indicating that the module is executing the command.

3. After WX2 changes to hex 2000, set the value of WY4 to hex 0000 to clear the trigger. If WX2 transitions from hex 2000 to hex 0000, the command was executed successfully.

4. If WX2 changes to hex A000, a command error was encountered. Inspect the Command Error Word (in the command block - offset 0) and clear the error by changing WY4 to hex 8000. Then lower error acknowledge bit by setting WY4 to hex 0000. Perform the necessary error recovery action based on the error code in the Command Error Word.

Slowing the PLC Scan

Because of the communications latency between you PLC programming software and the PLC, you may not be able to observe all bit transitions at the highest PLC scan rate. One way to improve this situation is to change the PLC to the slowest fixed scan rate available. Generally, this value is 255 milliseconds.

You can further slow things down by setting the task codes per scan to the minimum value, usually 1 task code per scan.

> *NOTE:*
> *If you use these techniques for debugging, be sure to set the parameters back before running the actual application.*

APPENDIX A. CONNECTORS AND CABLING

2573 Serial Port Cabling

2573 Module Serial Port Pinout

The serial ports on the 2573 module use a DB9 male connector which provides multiple electrical
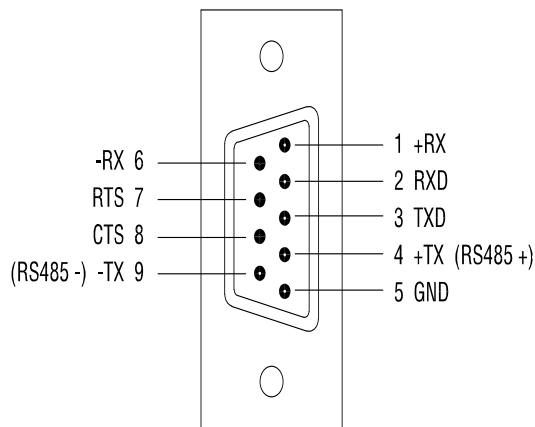
Figure 13. 2573 Serial Port Pinout

interfaces. The electrical interface in use is determined by which pins are connected to the serial communications cable. The illustration above shows the pinouts for one of the ports; all ports are identical.

RS232 Cabling

To use the RS232 electrical interface, connect to the pins as shown below. Do not connect pins 1,4,6, or 9.

Pin #	RS-232 Signal
2	RxD: Received Data (input)
3	TxD: Transmitted Data (output)
5	SG: Signal Ground
7	RTS: Request to Send (output)
8	CTS: Clear to Send (input)

The following figure illustrates how to construct an RS-232 cable typically used for connecting plant floor equipment to the module serial port. This "rolls" the transmit/receive and RTS/CTS lines so that the output line on one connector is tied to the input line on the other connector. This cable can be used to connect a typical PC to the 2573 serial port. This cable is similar to a null modem cable, except only RTS and CTS handshake lines are connected.
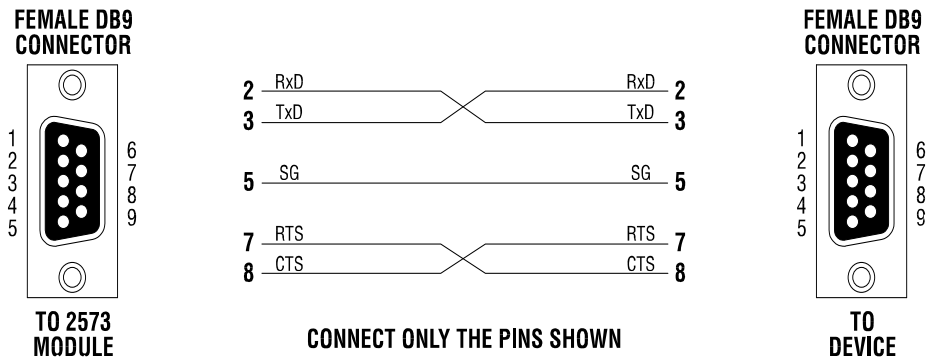
Figure 14. 2573 Module to RS-232 Device Cable (PC Style)

NOTE:
The 2573 module supports only the RTS and CTS hardware handshake lines. Other pins typically used for other handshake lines are dedicated to RS-422-A and RS-485. The cable shield should be connected to the chassis ground at one end only to minimize the potential for ground loops. To connect to chassis ground, attach the shield to the female DB9 connector base.

At typical baud rates and electrical environments, you should limit the length of RS-232 cables to 50 feet or less.

Most cables constructed for connecting to the serial port of a 545 PLC *will not work* with the 2573 modules below Rev D because they strap lines that are used for RS-422 on the 2573. 2573 Modules Rev D and above provide RS-422 Disconnect switches for this purpose. If you have a module below Rev D and you want to use the *existing* cable for both the PLC program port and the 2573 module, you can build this sample adapter shown below. This adapter simply eliminates connections from any hardware handshake lines. Plug this adapter into the module serial port, then plug the existing cable into the adapter.
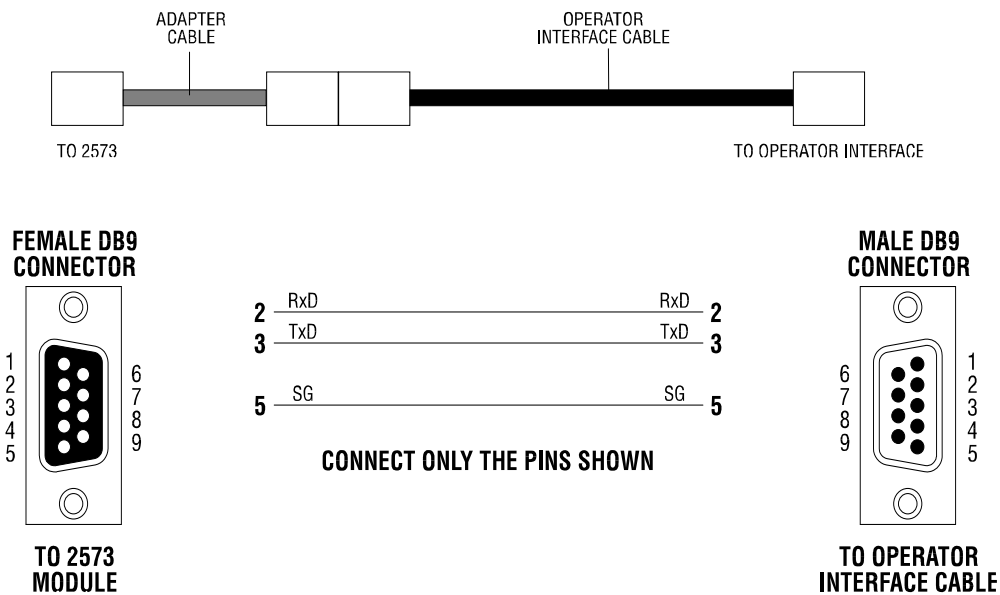
Figure 15. TCAM to 2573 Module RS-232 Adapter

If you want to construct a *new* cable to use between the 2573 and the RS-232 serial port of a SIMATIC® 545 PLC, see the diagram below.
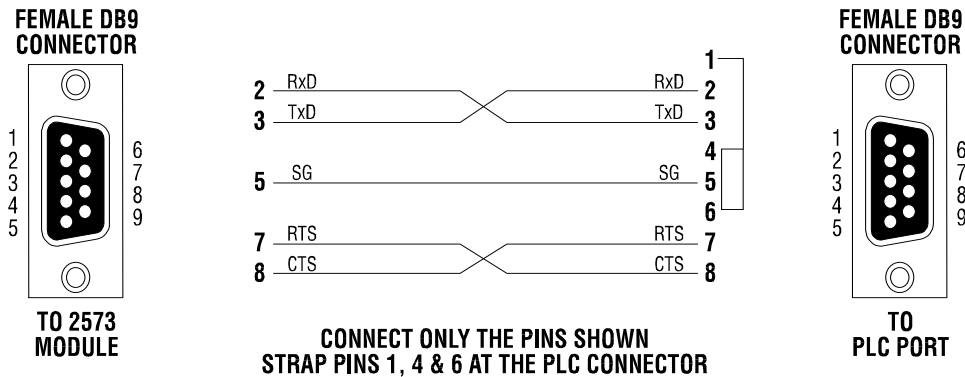
Figure 16. 2573 Module to PLC Cable (RS-232)

RS-422-A Cabling

The RS-422-A electrical interface is typically used for point to point serial connections where extended distance or improved noise immunity is required. An RS-422-A cable consists of two pairs of twisted pair wires enclosed by a shield. One pair of wires is used for transmitting data, the other pair for receiving data. The pair employs a balanced voltage circuit which tends to cancel out externally generated interference. To use the RS-422-A electrical interface, connect the pins as follows:

Pin #	RS-422-A Signal
1	RX+: Received Data - Positive polarity (input)
6	RX-: Received Data - Negative Polarity (input)
4	TX+: Transmitted Data - Positive Polarity (output)
9	TX-: Transmitted Data - Negative Polarity (output)

> *NOTE:*
> *The cable shield should be connected to the chassis ground at one end only to minimize the potential for ground loops. To connect to chassis ground at the 2573 and the PLC, attach the shield to the female DB9 connector base.*

Although there are no standards for pinout configuration for RS-422, you may already have cables used with CTI operator interface products which connect to the 545 programming port. If you want to use these cables with both the 2573 module and the PLC serial port, you can construct this simple adapter cable.
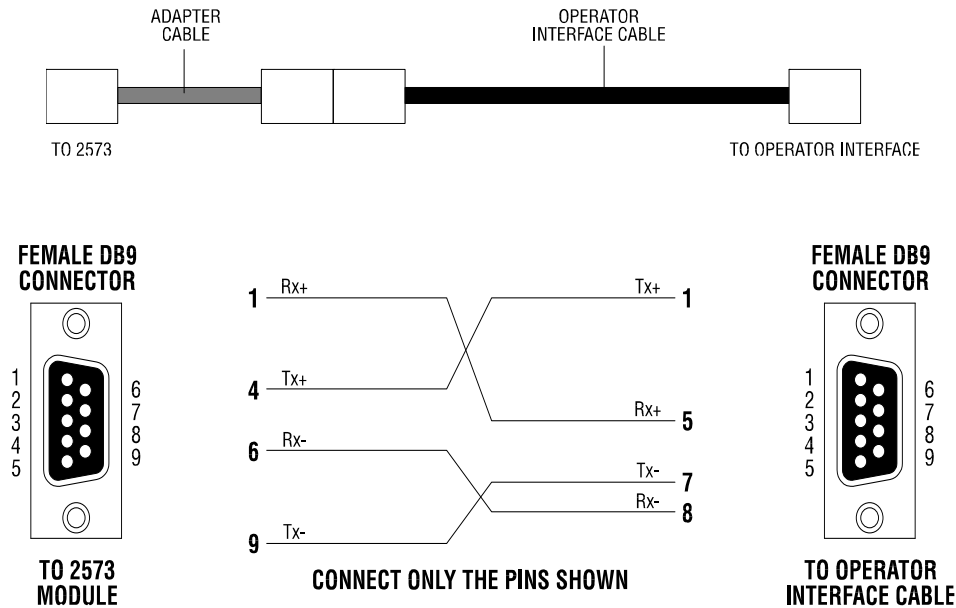
Figure 17. TCAM to RS-422A Adapter Cable

If you wish to build an RS-422 cable that will connect between a 2573 module serial port and a SIMATIC® 545 or 555 PLC serial port, see the following examples.

NOTE:
TheRS-422 serial port pin out varies between SIMATIC® PLC models.

The pin out for the 545-1101 PLC is:
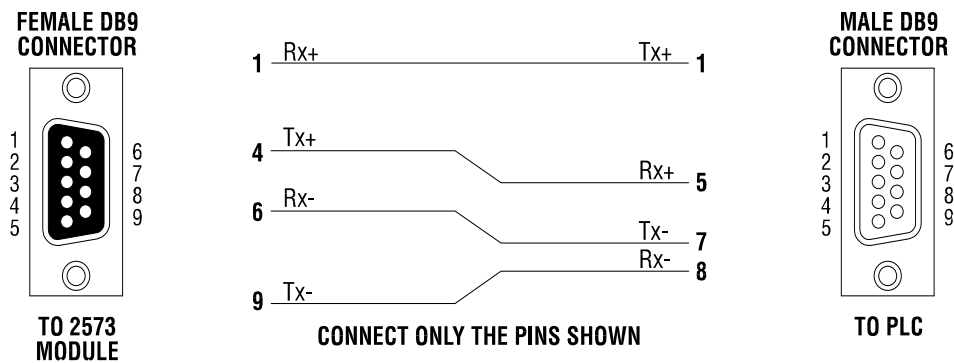
Figure 18. 2573 Module to PLC Cable (RS-422) Old Configuration

For the 545-1102 and above, and for the 555-1101 and above, use the pin out below:
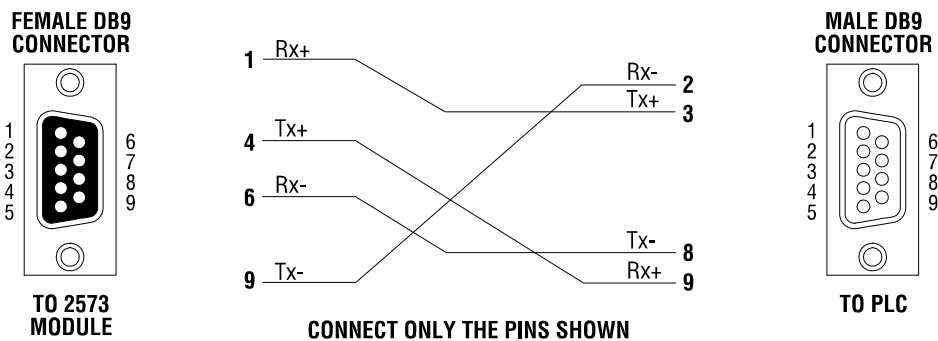
Figure 19. 2573 Module to PLC Cable (RS-422) New Configuration

RS-485 Cabling

The RS-485 electrical interface is typically used for multi-point serial connections. This electrical interface uses one pair of wires for both transmit and receive. Like RS-422-A, RS-485 uses a balanced circuit, with one wire positive polarity and the other negative polarity. To use the RS-485 electrical interface, connect the pins as shown below. Do not connect to any other pins at the module serial port.

Pin #	RS-485 Signal	
4	Tx+/Rx+:	Transmitted and Received Data Positive Polarity
9	Tx-/Rx-:	Transmitted and Received Data Negative Polarity

NOTE:
The cable shield should be connected to the chassis ground at one end only to minimize the potential for ground loops. To connect to chassis ground, attach the shield to the female DB9 connector base.

NOTE:
Ensure that that offset 9 ((Handshake) in the Create Connection command block is set to the value of 4 to select RS-485 port communications. Only Modbus and GAS protocol managers support RS-485.

Loopback Connector Construction

The following shows how to construct an RS-232 loopback connector. You can attach this directly to the module serial port to verify operation for RS-232. The 2573 module has a switch setting to support loopback testing. You may also be able to use this connector to check out certain cable configurations (such as the 2573 Module to RS- 232 cable shown earlier in this chapter.

The following figure illustrates the construction of an RS-422-A loopback connector. You can attach this directly to the module serial port to verify operation for RS-422.
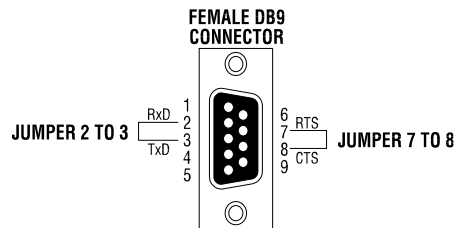
Figure 20. RS-232 Loopback Connector
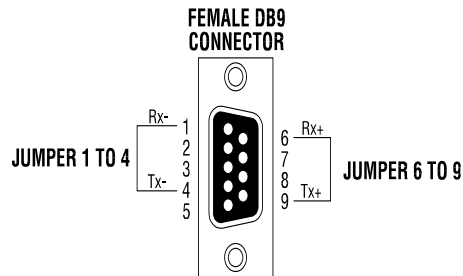
Figure 21. RS-422 Loopback Connector

> *NOTE:*
> *There is no RS-485 loopback connector, since RS-485 uses a single pair for both transmit and receive.*

Termination Resistance Jumpers
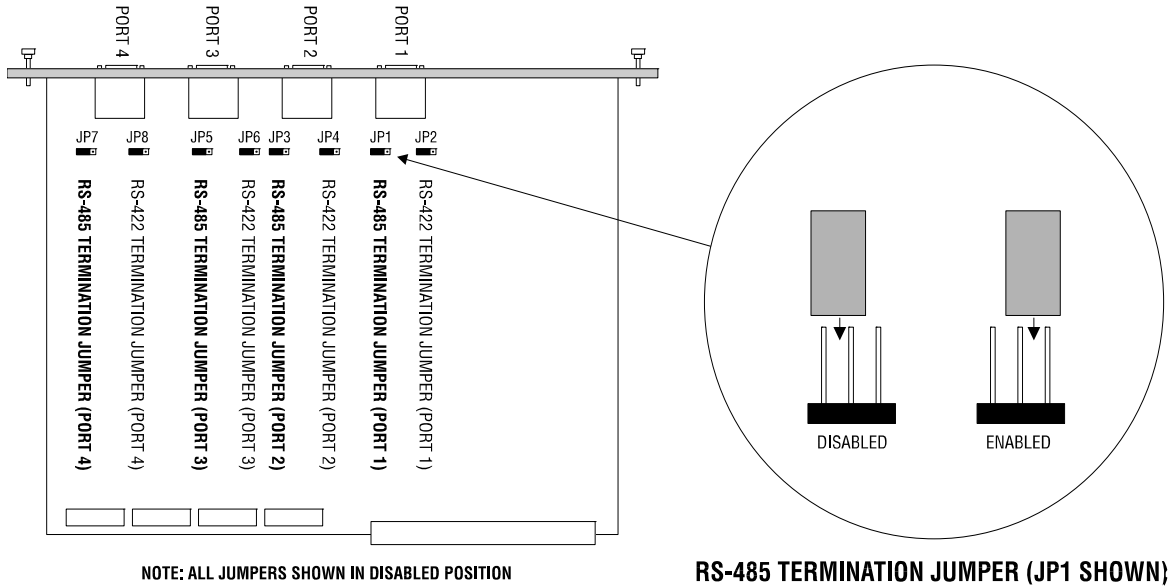
Figure 22. Termination Resistor Jumper Locations

RS-485 Termination Resistance

RS-485 circuits require that 120 ohm termination resistors be installed at *each end* of the transmission line to suppress signal reflection. In a typical installation, the 2573 module will be at one end and will require a terminating resistance. You should refer to the documentation for the plant floor device you are connecting for specific instructions about your installation.

The 2573 module circuit board contains a 120 ohm termination resistor which can be placed in the circuit (enabled) by repositioning a jumper on the board. See the illustration above for the location of the RS-485 termination jumpers (JMP1, JMP3, JMP5, JMP7). If your particular installation requires the resistor at the module connection, you can enable it by moving the appropriate jumper.

RS-422-A Termination Resistance

Most RS-422 installations **do not** require a termination resistance; however, a few RS-422 installations may specify a termination resistance. If your particular installation is one of these, you can reposition the jumpers labeled *RS-422 Termination Jumper* to insert the resistance into the module RS-422 receive circuit. Do not enable termination resistance unless specified by your device vendor.

APPENDIX B. ERROR CODES

Error Processing Overview

This Appendix provides a description of the error codes that may be returned to the PLC by the 2573 module except for errors returned by the GAS Protocol Manager. The GAS Protocol Manager Errors are documented in the General ASCII Support Protocol Manager Reference Manual.

Potential solutions to the error conditions are also presented. When an error is detected by the 2573 module, it sets the PLC ERR bit in the command status word, places a command error code in offset 0 of the command block, and suspends processing on the particular port. See Appendix D for a complete description of error processing.

The error codes are shown in both Hexadecimal and Integer format. When viewed in hexadecimal format, the high byte indicates the software/protocol manager that is responding to the error; it will usually contain the number of the protocol manager who generated to error code. If the 2573 operating environment generated the error code, the high byte will be 00. For memory transfer commands, the high byte of the error code will be the 0x02, regardless of the underlying protocol. The lower byte will contain the specific error code. The integer format is useful when you are using PLC logic to respond to certain error codes.

> *NOTE:*
> *When you are using your PLC programming software to view the error codes, you should set the display to view the error codes in hexadecimal format. This will improve troubleshooting by allowing you to determine the protocol manager that generated the error.*

> *NOTE:*
> *The 2573 module protocol managers do not reset the Command Error Word to 0 after the error is cleared. You must do this in your PLC logic.*

Error codes are listed in the following sections in numerical order.

System Errors Codes

System errors are detected by the module operating environment. System error codes have hex 00 in the high byte of the command error word.

HEX	DEC	DESCRIPTION	SOLUTION
0097	151	INVALID COMMAND While attempting to execute a system command, you specified an invalid command number.	Correct the Command Number in the Command Block.
009E	158	ALREADY BUSY You attempted to send a command to a function that was already busy executing a previous command.	You can cause this error by using two command slots to send commands to the same connection number. Don't do this.
009F	159	INVALID CONNECTION NUMBER The command block contains an invalid connection number. Valid connection numbers range from 19221-19229 when using a Create Connection command.	Correct the parameter in the connection number in the command block.
00A5	165	INVALID SYSTEM COMMAND Connection number 19200 (system) was specified in the command block but the command is not a valid system command.	Change the command block parameter to a valid connection number or the command number to a valid system command.
00A6	166	CONNECTION NOT ACTIVE An attempt was made to send a command to a connection number that has not been created. Note: You can receive this error if you incorrectly enter the connection number.	Check the following for the Create Connection Command: 1) Make sure that the command block contains the correct data, 2) Ensure that the Command Slot points to the proper command block, 3) Ensure that the Command Trigger has been set. 4) Check for reported errors for the Create Connection Command Ensure that you have entered the correct connection number in the command block.

HEX	DEC	DESCRIPTION	SOLUTION
00A7	167	DUPLICATE CONNECTION NUMBER An attempt was made to start two protocol managers with the same connection number and the same physical port.	Check the Command Blocks for the respective Create Connection commands. Ensure that the Create Connection commands use different connection numbers and different ports. Make sure that another protocol manager was not started via dipswitch with the same connection number. See the product IOG.
00A8	168	DUPLICATE CONNECTION NUMBER An attempt was made to start two protocol managers with the same connection number.	See error 00A7 above.
00A9	169	INVALID PROTOCOL MGR NUMBER An attempt was made to start a protocol manager which does not exist in the list of valid protocol manager numbers.	This error will occur if you are using PLC logic to create a connection to a port and you specify an invalid protocol manager number in the command block. Review the documentation and correct the command block.
00AA	170	INVALID PHYSICAL PORT NUMBER An attempt was made to start a protocol manager with an invalid physical port number. Valid physical port numbers for the 2573 are 1,2,3, or 4.	This error will occur if you enter an invalid physical port number in the command block for the Create Connection command. Correct the command block and retry the command.
00AB	171	PORT ALREADY CONNECTED An attempt was made to connect two protocol managers to the same physical serial port. Only one protocol manager can be connected to one physical serial port.	This error is typically caused by one of three circumstances. 1) The PLC logic triggers the same Create Connection command more than once. Examine and correct the PLC logic. 2) You have inadvertently used the same physical port number more than once in separate Create Connection commands. Change the physical port number to eliminate the conflict. 3) You have automatically started the protocol manager on this port using the module dip switch settings (See page 7). You should either set the port dipswitch to PLC Select mode or you should not execute a Create Connection command for this port.

HEX	DEC	DESCRIPTION	SOLUTION
00B2	178	INVALID V MEMORY ADDRESS The Command Block V memory address of 0.	Correct the Command Block data.
00B3	179	INSUFFICIENT V MEMORY RESERVED You failed to reserve enough V memory to hold the data returned by the command.	Correct the Command Block data.

General Memory Transfer Error Codes (NITP)

The following error codes (0201-0261) are generated by the Siemens® PLC, only when using the Task Code Master. The Task Code Master protocol manager will simply pass on these errors to the PLC.

HEX	DEC	DESCRIPTION
0201	513	Reset Current Transaction
0202	514	Address out of Range (Other than Ladder Logic)
0203	515	Requested Data not Found
0204	516	Illegal Task Code Request (e.g. Task Code not Supported)
0205	517	Request Exceeds Program Memory Size (Ladder Logic)
0206	518	Diagnostic Fail upon Power Up
0207	519	Fatal Error Detected
0208	520	Keylock Protect Error
0209	521	Incorrect amount of Data sent with Request
020A	522	Illegal Request in Current Operational Mode
020B	523	Network was not Deleted
020C	524	Attempted Write Operation Did Not Verify
020D	525	Illegal Number of ASCII Characters Received
020E	526	Illegal Write to Program Memory (Non Volatile)
020F	527	Data not Inserted
0210	528	Data not Written
0211	529	Invalid Data sent with the Command
0212	530	Invalid Operation with NIM (Obsolete)
0213	531	The store and forward buffer is busy
0214	532	No response from the Special Function Module
0215	533	Illegal Instruction found in program memory on a Program to Run transition
0216	534	Attempted Write to a Protected Variable (e.g. TCC, TCP)
0217	535	No response from PLC (e.g. Single Scan not performed)
0218	536	Requested memory size exceeds total available memory
0219	537	Requested Memory size is not a multiple of block allocation size
021A	538	Requested memory size is less than minimum defined value.
021B	539	Requested memory size is larger than maximum defined value
021C	540	PLC Busy - Cannot complete the requested operation
021D	541	Comm error in HOLD mode - Transition to Run not allowed
021E	542	Port Lockout is Active (ref task Code 48)
0221	545	I/O Configuration Error - too many points
0222	546	I/O Configuration Conflict
023F	575	Bus Error Detected

HEX	DEC	DESCRIPTION
0240	576	Operating System Error Detected
0241	577	Invalid Control Block Type
0242	578	Control Block Number out of range
0243	579	Control Block does not exist
0244	580	Control Block already exists
0246	582	Offset out of range.
0247	583	Arithmetic error detected while writing loop or analog alarm parameters
0248	584	Invalid SF Program type
0249	585	Instruction number or RAMP/SOAK Step number out of range
024A	586	Attempt to access an integer only variable as a real.
024B	587	Attempt to access a real -only value as an integer.
024C	588	Task Code buffer overflow -- too much data requested
024D	589	Control Block size error. (Maximum = 32767 bytes)
024E	590	Attempt to write a read only variable.
024F	591	Invalid variable type for this operation
0250	592	Task code request buffer too large (PLC internal error)
0251	593	Invalid SF Statement size specified
0252	594	Invalid return value
0253	595	Attempt to execute a Cyclic statement in a non-cyclic SF program
0254	596	Control Block is disabled
0255	597	Control Block is not disabled
0256	598	Attempt to perform a FTSR-OUT Statement on an empty FIFO
0257	599	Attempt to perform a FTSR-IN Statement on a full FIFO
0258	600	Stack overflow while evaluating a MAATH, IF-THEN, or IMATH expression
0259	601	Maximum SF Subroutine nesting level exceeded (Max depth = 4)
025A	602	Arithmetic Overflow
025B	603	Invalid Operator in an IF, MATH, or IMATH expression
025C	604	S Memory Overflow
025D	605	Attempt to divide by 0 (IMATH Statement)
025E	606	FIFO is incompatible with FTSR-IN or FTSR-OUT statement
025F	607	FIFO is invalid
0260	608	Invalid Data Type code (Usually MATH, IMATH, or IF-THEN statement)
0261	609	RAMP/SOAK step type mismatch. Error is logged by Task Code 6 (Write RAMP/SOAK step) if the fields are being updated and the step type indicated but the task code does not match the step type of the step being modified.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

General Memory Transfer Error Codes (NITP and CAMP)

The following error codes are generated by protocol managers which support memory transfer commands. Either the Task Code Master protocol manager or the CAMP Master protocol manager may generate these errors.

HEX	DEC	DESCRIPTION	SOLUTION
026E	622	NITP PROTOCOL ERROR The 2573 received an invalid NITP message. NITP protocol errors include: - Odd number of characters received, - Invalid characters received, - Checksum error, - Incorrect character count field.	This usually results from "noise" on the communications line. Retry the command. This may occur if you are using a computer system to emulate a task code slave and the system has been improperly programmed. Try using communicating with another 257x module (2571, 2572, 2573). If the problem goes away, then the computer system program is in error.
026F	623	REMOTE MEMORY READ: WORD COUNT ERROR The remote PLC responded with fewer words than requested by the Remote Memory Read command block.	This error usually occurs when the number of words requested in the command block causes the read request to exceed the V memory boundary in the remote PLC. In other words, you are trying to read V memory that isn't there. You should reduce the number of words requested, change starting V memory address to a smaller value, or change the memory configuration for the remote PLC.
0271	625	MEMORY TRANSFER WORDS OUT OF RANGE - LOW Offset 4 in the command block for a Remote Memory Read, Remote Memory Write, or Memory Exchange contained 0 or a negative value. Offset 8 in the command block for a Memory Exchange contained 0 or a negative number.	Change the value to set the number of words to transfer within the valid range of values. Check that the value in the Command Slot points to the Command Block
0272	626	MEMORY TRANSFER WORDS OUT OF RANGE - HIGH Offset 4 in the command block for a Remote Memory Read, Remote Memory Write, or Memory Exchange contained a value greater than maximum allowed. Offset 8 in the command block for a Memory Exchange contained a value greater than the maximum allowed.	Change the value to set the number of words to transfer within the valid range of values. Check that the value in the Command Slot points to the Command Block

HEX	DEC	DESCRIPTION	SOLUTION
0273	627	BAD OR MISSING DELIMITER The receiving device detected a bad or missing delimiter in the command message	This error usually indicates a transient error in the data link. You should retry the command. If the receiving node is a computer simulating a slave, ensure that the software is correctly processing the message. If the error persists contact CTI.
0274	628	BAD BLOCK CHECK CHARACTER The receiving device found the block check character to be bad.	See error code 0273 above.
0275	629	INVALID TYPE The receiving device detected an invalid TYPE character in the message.	See error code 0273 above.
0276	630	INVALID DATA CHARACTER The receiving device detected an invalid data character. Only characters 0-9 and A-F are allowed.	See error code 0273 above.
0277	631	ODD NUMBER OF CHARACTERS The receiving device detected an odd number of characters in the message. All CAMP messages have an even number of characters.	See error code 0273 above.
0278	632	INVALID DEVICE CODE The *receiving* device reported an invalid device code in the message.	See error code 0273.
027A	634	INVALID DEVICE CODE The 2573 detected an invalid device code in the response from the remote device	This error may result from a transient error in the data link. You should retry the command. If the problem persists, check the quality of the data link. If the remote node is a computer simulating a 2573 PLC server, ensure that the software is correctly creating the response the response message
027B	635	BAD OR MISSING DELIMITER The 2573 detected a missing delimiter in the response message	See error code 027A.
027C	636	BAD BLOCK CHECK CHARACTER The 2573 found a bad block check character in the response message.	See error code 027A.

HEX	DEC	DESCRIPTION	SOLUTION
027D	637	INVALID TYPE The 2573 detected an invalid TYPE character in the message sent from the remote device.	See error code 027A.
027E	638	INVALID DATA CHARACTER The 2573 detected an invalid data character in the message sent from the remote device. Only characters 0-9 and A-F are allowed.	See error code 027A.
027F	639	ODD NUMBER OF CHARACTERS The 2573 detected an odd number of characters in the message sent from the remote device. All CAMP messages have an even number of characters.	See error code 027A.
0280	640	UNKNOWN ERROR CHARACTER The response contained an unknown error character.	This error is probably caused by an improperly programmed slave node. If the slave is a PLC emulating a slave, review the code and correct as required. If the slave is another 2573 module and the error persists, contact CTI.
0284	644	INCOMPLETE WRITE The actual number of words written does not match the requested number of words. The actual number of words written is contained in the response message from the remote device.	This error is probably caused by an improperly programmed slave. If the slave is a computer system emulating a slave, review the code and correct as required. If the slave is another 2573 module, contact CTI.
0290	656	INVALID ADDRESS CLASS The address class is not supported by the server	See error 0284 above.
0292	658	MESSAGE ID MISMATCH The response message contained an ID field which did not match the ID field in the command.	This error could be caused by a timeout value which is too short followed by an attempt to re-send. Therefore, by the time the response to the first command arrives, you have already sent a second command. You may correct this condition by increasing the timeout value. This may also be caused by an improperly programmed server. If the server is a PLC emulating a 2573 server, review the code and correct as required.

HEX	DEC	DESCRIPTION	SOLUTION
029A	666	ERROR WRITING SERIAL PORT The protocol manager encountered an error while writing to a serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
029D	669	CMD TIMEOUT ERROR The time to process the requested command exceeded the Command Timeout value. Command processing has been suspended.	The device may be slow in responding, increase the timeout value to determine if the problem will go away. This error may indicate a bad connection (faulty cable, loose connector, etc.). Check all cabling carefully.
029E	670	ALREADY BUSY ERROR The protocol manager received a new command prior to the completing the previous command.	This error can also be produced by simultaneously triggering two command slots which reference the same command block. If more than one command slot is used, make sure that they reference different command blocks and that the command blocks that they reference have different connection numbers.
02AC	684	MEMORY READ ERROR The protocol manager was unable to read the requested V memory. If this occurs on a WRITE REMOTE command block, it indicates that the 2573 could not obtain the data from the local PLC V memory. If this occurs on a READ REMOTE command block, it indicates that the memory in the remote device could not be read.	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value for Source in Offset 3 and make corrections if necessary. When this error is returned in response to a READ REMOTE command block, it is also possible that there is a problem in the remote PLC which prevents the 2573 from reading V memory. Retry the command. If the problem persists, check the remote device.
02AD	685	MEMORY WRITE ERROR The was unable to write the requested memory. If this occurs on a WRITE REMOTE command block, it indicates that the remote device could not write the data into remote memory. If this occurs on a READ REMOTE command block, it indicates that the local PLC could not write the result from the remote device into local V memory	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value for Destination in Offset 5 and make corrections if necessary. When this error is returned in response to a WRITE REMOTE command block, it is possible that there is a problem in the remote PLC which prevents the 2573 from writing V memory. Retry the command, if the problem persists, check the remote device

Modem Control Feature Error Codes

The following error codes are unique to the modem control feature.

HEX	DEC	DESCRIPTION	SOLUTION
0499	1177	ERROR READING PORT The protocol manager detected an error during a read on a 2573 serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
049D	1181	AUTO DIAL TIMEOUT The auto dial command did not complete within the allowable time.	Ensure that the modem is connected and has power.
04BA	1210	MODEM HARDWARE ERROR The modem reported a general error.	Check the modem, modem setup, and cabling. Correct as required.
04BB	1211	NO OR LOST CARRIER This indicates that there is no connection to the remote modem. This error may be encountered during dial up or when attempting to send data. Some modems may return a "No Carrier" message when the remote connection does not answer.	Manual Mode: Acknowledge the error, issue a Hang-Up Command, then re-trigger the Dial Command. Auto-Dial Mode: Re-trigger the command to transmit a data message. If the condition persists you should check the remote modem, the signal quality of the phone line, and ensure that the modem is configured for a fixed baud rate.
04BC	1212	NO DIAL TONE No dial tone was detected by the modem This probably results from the modem not connected to the phone line.	Check the phone cable and connections and make necessary corrections.
04BD	1213	BUSY TONE DETECTED The number you are dialing is busy.	Manual Mode: Issue a hang-up command and retry at a later time. Auto-Dial Mode: Retry the command to which sends a data message.
04BE	1214	NO ANSWER The number you are dialing does not answer.	See error 04BD above.
04BF	1215	INVALID V MEMORY ADDRESS FOR MODEM CONFIGURATION TABLE You have specified a V memory location in the Create Connection command block, offset 11, which is not valid for this PLC memory configuration.	Check the value in the Command block. If the modem flag bit in Offset 10 is set to 1, then offset 11 must contain a valid V memory address. *Note that 0 is not a valid V memory address.*
04C0	1216	INVALID SIGNATURE IN CONFIG TABLE The first word in the V memory block referenced in offset 11 does not contain the String "AT" (0x4154). This word is used to help ensure that the referenced V memory is really a Modem Configuration Table.	Ensure that the first word in the Modem Configuration Table contains the valid signature value and correct if required. Check that the value in offset 11 points to the first word in the Modem Configuration Table. Correct if required.

HEX	DEC	DESCRIPTION	SOLUTION
04C1	1217	INVALID V MEMORY ADDRESS FOR DIAL STRING In the dial command block, you have specified a value in offset 4 which is not a valid memory address for this PLC memory configuration.	Correct the entry in offset 4
04C2	1218	INVALID DIAL STRING The high byte of the first word in the dial string does not contain a double quote mark ("), 0x22. OR The ending double quote mark delimiter for the string could not be found.	For manual dial, ensure that the value in Offset 4 of the dial command block points to the first word of the dial string. For auto dial, ensure that the value in offset 6 of the Modem Configuration Table points to the first word of the dial string. Make sure the high byte of the first word contains the double quote mark and that. you have entered a double quote mark after the last character in the dial string.
04C3	1219	MODEM SUPPORT IS NOT ACTIVE You have attempted to issue a modem command (dial or hang-up) to a protocol manager which has not been configured for modem support in the Create Connection command block.	Make sure that the modem flag bit has been set for the Create Connection command.
04C4	1220	MODEM STATE MISMATCH You attempted to dial a number while you are already connected and in data mode.	Issue a Hang-Up command before dialing.
04C5	1221	UNKNOWN RESPONSE The modem responded with an unsupported, non-standard reply.	Ensure that you have configured the modem to use the Hayes compatible AT command/reply set (verbose mode).
04C6	1222	INVALID CHARACTER in DIAL STRING An unsupported character has been detected in the dial string	Examine the character string and correct erroneous character entries. Ensure that a double quote delimiter immediately follows the last character in the dial string.
04EA	1258	MODEM BAD CONTROL STATE A modem DIAL command terminated before the call was established	Ensure that the command time-out is not too short and is timing out while the command is actually in progress. Make sure the modem is configured to make responses to our commands. Depending upon why the command failed, a subsequent HANGUP command may free the port from this state.

HEX	DEC	DESCRIPTION	SOLUTION
04EB	1259	MODEM PORT READ FAILURE The module detected an error during a read on a 2573 serial port.	Ensure that the modem is using the same fixed baud rate that the port has been configure to.
04EC	1260	MODEM PORT WRITE FAILURE The module detected an error during a write on a 2573 serial port.	Ensure that the modem is using the same fixed baud rate that the port has been configure to.

Task Code Master Error Codes

The following error codes may be returned by the Task Code Master protocol manager.

HEX	DEC	DESCRIPTION	SOLUTION
206E	8302	NITP PROTOCOL ERROR The 2573 received an invalid NITP message. NITP protocol errors include: - Odd number of characters received, - Invalid characters received, - Checksum error, - Incorrect character count field.	This usually results from "noise" on the communications line. Retry the command. This may occur if you are using a computer system to emulate a task code slave and the system has been improperly programmed. Try using communicating with another 257x module (2571, 2572, 2573). If the problem goes away, then the computer system program is in error.
2097	9343	UNKNOWN COMMAND The protocol manager received a command number which is not valid.	Ensure that Offset 1 in the Command block contains a valid Command Code. Check to be sure that the Command Slot points to the correct V memory Address for the Command Block. Make sure that you have started the correct protocol manager and that you are using the correct connection number.
2099	8345	ERROR READING PORT The protocol manager detected an error during a read on a 2573 serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
209A	8346	ERROR WRITING PORT The protocol manager detected an error during a write on a 2573 serial port.	See error code 0299 above.
209D	8349	CMD TIMEOUT ERROR The time to process the requested command exceeded the Command Timeout value. Command processing has been suspended.	The device may be slow in responding, increase the timeout value to determine if the problem will go away. This error may indicate a bad connection (faulty cable, loose connector, etc.). Check all cabling carefully.

HEX	DEC	DESCRIPTION	SOLUTION
209E	8350	**ALREADY BUSY ERROR** The protocol manager received a new command prior to the completing the previous command.	This error can also be produced by simultaneously triggering two command slots which reference the same command block. If more than one command slot is used, make sure that they reference different command blocks and that the command blocks that they reference have different connection numbers.
20AD		**MEMORY WRITE ERROR** The 2573 was unable to write the requested memory. If this occurs on a WRITE REMOTE command block, it indicates that the remote device could not write the data into remote memory. If this occurs on a READ REMOTE command block, it indicates that the local PLC could not write the result from the remote device into local V memory	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value for Destination in Offset 5 and make corrections if necessary. When this error is returned in response to a WRITE REMOTE command block, it is possible that there is a problem in the remote PLC which prevents the 2573 from writing V memory. Retry the command, if the problem persists, check the remote device

CAMP Master Error Codes

The following error codes may be returned by the CAMP Master protocol manager.

HEX	DEC	DESCRIPTION	SOLUTION
2465	9317	NITP - CHARACTER PARITY ERROR The receiving device detected a parity error in the NITP message.	This error may occur because of "noise" on the communications line. Retry the command. You may also observe this error if you have set the communications parameters incorrectly.
2466	9318	NITP - INVALID CHARACTER The receiving device detected an invalid character in the NITP message.	See Error Code 2465.
2467	9319	NITP - INVALID CHARACTER COUNT FIELD The number of characters detected by the receiving device did not match the character count in the NITP message.	See Error Code 2465.
2468	9320	NITP - ODD NUMBER OF CHARACTERS The receiving device detected an odd number of characters in the received NITP message.	See Error Code 2465.
2369	9321	NITP - CHECKSUM ERROR The receiving device detected a checksum error. Device detected a parity error in the message.	See Error Code 2465.
246E	9326	NITP - PROTOCOL ERROR When the 2573 sent an NITP message, the receiving device replied with an invalid NITP message. NITP protocol errors include: - Odd number of characters received, - Invalid characters received, - Checksum error, - Incorrect character count field.	This error may occur due to communications error (line noise, etc.) on serial lines. It may also result from an improperly programmed remote device or a bad cable. Retry the operation. If the error persists, check out the remote device.
2473	9331	BAD OR MISSING DELIMITER The *receiving* device detected a bad or missing delimiter in the command message	This error usually indicates a transient error in the data link. You should retry the command. If the receiving node is a computer simulating a 2573 PLC server, ensure that the software is correctly processing the message. If the error persists contact CTI.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

HEX	DEC	DESCRIPTION	SOLUTION
2474	9332	BAD BLOCK CHECK CHARACTER The *receiving* device detected an block check character error in the command message.	See error code 2473
2475	9333	INVALID TYPE CODE The *receiving* device detected a bad command type in the message.	See error code 2473
2476	9334	INVALID DATA CHARACTER The *receiving* device reported an invalid data character. Only characters 0-9 and A-F are allowed.	See error code 2473.
2477	9335	ODD NUMBER OF CHARACTERS The *receiving* device reported an odd number of characters in the message. All CAMP messages have an even number of characters.	See error code 2473.
2478	9336	INVALID DEVICE CODE The *receiving* device reported an invalid device code in the message.	See error code 2473.
247A	9338	INVALID DEVICE CODE The 2573 detected an invalid device code in the response from the remote device	This error may result from a transient error in the data link. You should retry the command. If the problem persists, check the quality of the data link. If the remote node is a computer simulating a 2573 PLC server, ensure that the software is correctly creating the response the response message
247B	9339	BAD OR MISSING DELIMITER The 2573 detected a missing delimiter in the response message	See error code 247A.
247C	9340	BAD BLOCK CHECK CHARACTER The 2573 found a bad block check character in the response message.	See error code 247A.
247D	9341	INVALID TYPE The 2573 detected an invalid TYPE character in the message sent from the remote device.	See error code 247A.

HEX	DEC	DESCRIPTION	SOLUTION
247E	9342	INVALID DATA CHARACTER The 2573 detected an invalid data character in the message sent from the remote device. Only characters 0-9 and A-F are allowed.	See error code 247A.
247F	9343	ODD NUMBER OF CHARACTERS The 2573 detected an odd number of characters in the message sent from the remote device. All CAMP messages have an even number of characters.	See error code 247A.
2480	9344	UNKNOWN ERROR CHARACTER The response contained an unknown error character.	This error is probably caused by an improperly programmed server node. If the server is a PLC emulating a 2573 server, review the code and correct as required. If the server is another 2573, contact CTI.
2482	9346	INVALID WORD COUNT The number of words in the response message does not match the actual number of words in the message.	This error is probably caused by an improperly programmed remote device. If the device is a PC emulating a 2573 server, review the code and correct as required. If the device is another 2573, contact CTI.
2490	9360	INVALID ADDRESS CLASS The address class is not supported by the server	See Error Code 2482
2492	9362	CAMP ID MISMATCH The response did not contain a message ID which matched the command. This can be caused by an improperly programmed remote device. It can also be caused by a response delays which exceed the command timeout value. Thus it may be a response to a previous command.	Check the remote device for problems and correct as required. Determine whether you have response delays that exceed the command timeout. You may be able to increase the timeout value so that the command waits longer for a response.
2497	9367	UNKNOWN COMMAND The protocol manager received a command number which is not valid.	Ensure that Offset 1 in the Command block contains a valid Command Code. Check to be sure that the Command Slot points to the correct V memory Address for the Command Block. Make sure that you have specified the correct protocol manager number in the Create Connection command and that you are using the right connection number.

HEX	DEC	DESCRIPTION	SOLUTION
2498	9368	COULD NOT OPEN PORT The protocol manager could not open the 2573 serial port. This may occur during a Create Connection command.	This could result from a hardware error, which may be transient. Retry the command. If the error persists, reset the module. If the error re-occurs after reset, replace the module.
2499	9369	ERROR READING PORT The protocol manager detected an error during a read on a 2573 serial port.	See error code 2498 above.
249A	9370	ERROR WRITING PORT The protocol manager detected an error during a write on a 2573 serial port.	See error code 2498 above.
249D	9373	CMD TIMEOUT ERROR The time to process the requested command exceeded the Command Timeout value. Command processing has been halted. Note that the timeout value may have been explicitly set in the Command Block or the default may have been used (timeout = 0).	In some circumstances, you may occasionally get a timeout error, especially with heavy network traffic or remote modem data links. In these circumstances, your logic should retry the command. This error may indicate a bad connection (faulty cable, loose connector, etc.). Check all cabling carefully. If the error persists and you have explicitly set the timeout value, it is possible that you have set the timeout value too small. Try increasing the value or using the default value.
249E	9374	PROTOCOL MANAGER BUSY The protocol manager received a new command prior to completing the previous command.	You can cause this error by using two command slots to send concurrent commands to the same connection number. Don't do this.
249F	9375	INVALID CONNECTION NO. The connection number used during Create Connection was out of the valid range. The valid range of connection numbers is 19221 - 19299.	Correct the command block data.
24A0	9376	BAUD RATE SELECTION ERROR The protocol manager was passed an invalid baud rate. Valid baud rates are 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200.	This error usually results from an incorrect value in the command block for the Create Connection command. Check the value and correct any entry errors. Ensure that the Command Slot points to the correct command block.

HEX	DEC	DESCRIPTION	SOLUTION
24A1	9377	DATA BITS SELECTION ERROR The protocol manager was passed an invalid value for data bits. Valid data bit values are 5, 6, 7, 8.	See error code 24A0 above.
24A2	9378	PARITY SELECTION ERROR The protocol manager was passed an invalid parity value. Valid parity values are 0-None, 1-Odd, 2-Even.	See error code 24A0 above.
24A3	9379	STOP BIT SELECTION ERROR The protocol manager was passed an invalid number of stop bits. Valid stop bit values are 1 and 2.	See error code 24A0 above.
24A4	9380	HANDSHAKE SELECTION ERROR The protocol manager was passed an invalid handshake value. Valid handshake values are: 0-No Handshake, 1-Software Handshake, 2-Hardware Handshake.	See error code 24A0 above.
24AC	9388	MEMORY READ ERROR The 2573 was unable to read the requested V memory.	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value for Source in Offset 3 and make corrections if necessary. Check to ensure that the PLC you are addressing is configured for the address you are trying to access.
24AD	9389	MEMORY WRITE ERROR The 2573 was unable to write the requested V memory.	See Error Code 24AC above

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Modbus Master RTU Error Codes

The following error codes may be returned by the Modbus Master protocol manager.

HEX	DEC	DESCRIPTION	SOLUTION
286E	10350	INVALID V MEMORY ADDRESS A value of 0 was specified for the V memory address of the query data and the value for data length was not zero. OR A value of 0 was specified for the V memory address of the response data.	Correct the entry in the command block. If your query has a data length greater than 0, you must specify a starting V memory location for the query data. Because the protocol manager stores the response address and function code, you must specify a V memory location for the response, even though the response may contain no data.
286F	10351	QUERY – INVALID DATA LENGTH A value larger than 252 was specified for the query data length.	Correct the entry in the command block, offset 6.
2870	10352	RESPONSE - V MEMORY RESERVATION ERROR You did not reserve enough V memory to hold the response.	Correct the entry in command block, offset 9.
2871	10353	RESPONSE - CHECKSUM ERROR The response checksum did not match the checksum calculation.	This error is usually caused by a "noisy" communications line. Retry the command. If this error persists, you may need to reduce the baud rate or improve the signal quality. Also, you should check the quality of the cable and replace if required.
2872	10354	RESPONSE - WRONG SLAVE ID The response message contained a slave ID which did not match the query slave ID.	This error is probably caused by a slave device which responds with the wrong ID or a slave device which improperly responds to a query addressed to another slave ID. If the error persists you must troubleshoot and correct the slave device(s). If the device is very slow to respond, it is possible to cause this error by timing out on one query, sending another query to a different slave, and then receiving the response from the previous query. Try increasing the timeout value.

HEX	DEC	DESCRIPTION	SOLUTION
2873	10355	RESPONSE - WRONG FUNCTION CODE The response message contained a function code which did not match the query function code or was not a corresponding exception code.	This error results from a slave device which does not return the correct function code or corresponding exception code in the response. If the error persists you must troubleshoot and correct the slave device(s). If the device is very slow to respond, it is possible to cause this error by timing out on one query, sending another query, and then receiving the response from the previous query. Try increasing the timeout value.
2874	10356	RESPONSE CHARACTER ERROR A character parity or framing error was detected in the response.	This error is usually caused by a "noisy" communications line. Retry the command. If this error persists, you may need to reduce the baud rate or improve the signal quality. Also, you should check the quality of the cable and replace if required.
2875	10357	LATE CHARACTER RECEIVED A character was received after the specified inter-character time for the RTU protocol.	This error is usually caused by a device which cannot maintain inter-character timing required by the Modbus RTU specification. Retry the command. If the error persists, you must reduce the baud rate.
2877	10359	UART OVERRUN	PLC could not remove data in the input buffer fast enough to prevent overrun. This may indicate a need to lower the baud rate of the serial communications line.
2879	10361	QUERY BLOCK TERMINATOR NOT PRESENT The expected query block terminator, in a query block that is pointed to by a Poll List Query, was not found.	The length of the data portion of the query block was incorrect for the value entered by the programmer.
287A	10362	INVALID QUERY BLOCK PARAMETER	This error results from a query block that either contained a slave address >247, a data field length for the query or response that will be >255 bytes, or a function code >255.
287B	10363	PARTIAL MESSAGE RECEIVED	The response message returned was too short to be a legitimate Modbus packet.
287C	10364	MODBUS FUNCTION RESPONSE INDICATES AN ERROR	This error indicates that a slave device returned an error code as a result of a command sent as part of a Poll List Command. This will cause a halt in the executing the Poll List Commands that occur in the list after the incurred error.
287D	10365	MODBUS POLL LIST LENGTH INVALID	The length specified was either 0 or >60.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

HEX	DEC	DESCRIPTION	SOLUTION
2897	10391	UNKNOWN COMMAND The protocol manager received a command number which is not valid.	Ensure that Offset 1 in the Command block contains a valid Command Code. Check to be sure that the Command Slot points to the correct V memory Address for the Command Block. Make sure that you have specified the correct protocol manager number in the Create Connection command and that you are using the right connection number.
2898	10392	COULD NOT OPEN PORT The protocol manager could not open the 2573 serial port. This may occur during a Create Connection command.	This could result from a hardware error, which may be transient. Retry the command. If the error persists, reset the module. If the error re-occurs after reset, replace the module.
2899	10393	ERROR READING PORT The protocol manager detected an error during a read on a 2573 serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
289A	10394	ERROR WRITING PORT The protocol manager encountered an error while writing to a serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
28C8	10440	INVALID END OFMESSAGE INTERVAL	Value in create connection block for inter-message gap time is <5 or >100.

HEX	DEC	DESCRIPTION	SOLUTION
289D	10397	CMD TIMEOUT ERROR The time to process the requested command exceeded the Command Timeout value. Command processing has been halted. Note that the timeout value may have been explicitly set in the Command Block or the default may have been used (timeout = 0).	In some circumstances, you may occasionally get a timeout error, especially with heavy network traffic or remote modem data links. In these circumstances, your logic should retry the command. This error may indicate a bad connection (faulty cable, loose connector, etc.). Check all cabling carefully. If the error persists and you have explicitly set the timeout value, it is possible that you have set the timeout value too small. Try increasing the value or using the default value.
289F	10399	INVALID CONNECTION NO. The connection number used during Create Connection was out of the valid range. The valid range of connection numbers is 19221 - 19299.	Correct the command block data.
28A0	10400	BAUD RATE SELECTION ERROR The protocol manager was passed an invalid baud rate. Valid baud rates are 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200.	This error usually results from an incorrect value in the command block for the Create Connection command. Check the value and correct any entry errors. Ensure that the command slot value points to the correct command block.
28A2	10402	PARITY SELECTION ERROR The protocol manager was passed an invalid parity value. Valid parity values are 0-None, 1-Odd, 2-Even.	See error code 28A0 above.
28AC	10412	MEMORY READ ERROR The 2573 was unable to read the requested V memory	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value and make corrections if necessary.
28AD	10413	MEMORY WRITE ERROR The 2573 was unable to write the requested V memory.	See error code 28AC above.

Modbus Master ASCII Error Codes

The following error codes may be returned by the Modbus Master protocol manager.

HEX	DEC	DESCRIPTION	SOLUTION
296E	10606	INVALID V MEMORY ADDRESS A value of 0 was specified for the V memory address of the query data and the value for data length was not zero. OR A value of 0 was specified for the V memory address of the response data.	Correct the entry in the command block. If your query has a data length greater than 0, you must specify a starting V memory location for the query data. Because the protocol manager stores the response address and function code, you must specify a V memory location for the response, even though the response may contain no data.
296F	10607	QUERY - INVALID DATA LENGTH A value larger than 252 was specified for the query data length.	Correct the entry in the command block, offset 6.
2970	10608	RESPONSE - V MEMORY RESERVATION ERROR You did not reserve enough V memory to hold the response.	Correct the entry in command block, offset 9.
2971	10609	RESPONSE - CHECKSUM ERROR The response checksum did not match the checksum calculation.	This error is usually caused by a "noisy" communications line. Retry the command. If this error persists, you may need to reduce the baud rate or improve the signal quality. Also, you should check the quality of the cable and replace if required.
2972	10610	RESPONSE - WRONG SLAVE ID The response message contained a slave ID which did not match the query slave ID.	This error is probably caused by a slave device which responds with the wrong ID or a slave device which improperly responds to a query addressed to another slave ID. If the error persists you must troubleshoot and correct the slave device(s). If the device is very slow to respond, it is possible to cause this error by timing out on one query, sending another query to a different slave, and then receiving the response from the previous query. Try increasing the timeout value.

HEX	DEC	DESCRIPTION	SOLUTION
2973	10611	RESPONSE - WRONG FUNCTION CODE The response message contained a function code which did not match the query function code or was not a corresponding exception code.	This error results from a slave device which does not return the correct function code or corresponding exception code in the response. If the error persists you must troubleshoot and correct the slave device(s). If the device is very slow to respond, it is possible to cause this error by timing out on one query, sending another query, and then receiving the response from the previous query. Try increasing the timeout value.
2974	10612	RESPONSE CHARACTER ERROR A character parity or framing error was detected in the response.	This error is usually caused by a "noisy" communications line. Retry the command. If this error persists, you may need to reduce the baud rate or improve the signal quality. Also, you should check the quality of the cable and replace if required.
2976	10614	INVALID DATA CHARACTER RECEIVED IN ASCII MODE	A data character was received while in ASCII mode that was not a valid ASCII character.
2977	10615	UART OVERRUN	PLC could not remove data in the input buffer fast enough to prevent overrun. This may indicate a need to lower the baud rate of the serial communications line.
2978	10616	ODD CHARACTER COUNT IN ASCII MODE	An odd number of ASCII characters were counted while in ASCII mode.
2979	10617	QUERY BLOCK TERMINATOR NOT PRESENT The expected query block terminator, in a query block that is pointed to by a Poll List Query, was not found.	The length of the data portion of the query block was incorrect for the value entered by the programmer.
297A	10618	INVALID QUERY BLOCK PARAMETER	This error results from a query block that either contained a slave address >247, a data field length for the query or response that will be >255 bytes, or a function code >255.
297B	10619	PARTIAL MESSAGE RECEIVED	The response message returned was too short to be a legitimate Modbus packet.
297C	10620	MODBUS FUNCTION RESPONSE INDICATES AN ERROR	This error indicates that a slave device returned an error code as a result of a command sent as part of a Poll List Command. This will cause a halt in the executing the Poll List Commands that occur in the list after the incurred error.
297D	10621	MODBUS POLL LIST LENGTH INVALID	The length specified was either 0 or >60.

HEX	DEC	DESCRIPTION	SOLUTION
2997	10647	UNKNOWN COMMAND The protocol manager received a command number which is not valid.	Ensure that Offset 1 in the Command block contains a valid Command Code. Check to be sure that the Command Slot points to the correct V memory Address for the Command Block. Make sure that you have specified the correct protocol manager number in the Create Connection command and that you are using the right connection number.
2998	10648	COULD NOT OPEN PORT The protocol manager could not open the 2573 serial port. This may occur during a Create Connection command.	This could result from a hardware error, which may be transient. Retry the command. If the error persists, reset the module. If the error re-occurs after reset, replace the module.
2999	10649	ERROR READING PORT The protocol manager detected an error during a read on a 2573 serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
299A	10650	ERROR WRITING PORT The protocol manager encountered an error while writing to a serial port.	Retry the command. If the error persists, this condition may indicate a module hardware problem. Contact CTI.
299D	10653	CMD TIMEOUT ERROR The time to process the requested command exceeded the Command Timeout value. Command processing has been halted. Note that the timeout value may have been explicitly set in the Command Block or the default may have been used (timeout = 0).	In some circumstances, you may occasionally get a timeout error, especially with heavy network traffic or remote modem data links. In these circumstances, your logic should retry the command. This error may indicate a bad connection (faulty cable, loose connector, etc.). Check all cabling carefully. If the error persists and you have explicitly set the timeout value, it is possible that you have set the timeout value too small. Try increasing the value or using the default value.
299F	10655	INVALID CONNECTION NO. The connection number used during Create Connection was out of the valid range. The valid range of connection numbers is 19221 - 19299.	Correct the command block data.

HEX	DEC	DESCRIPTION	SOLUTION
29A0	10656	BAUD RATE SELECTION ERROR The protocol manager was passed an invalid baud rate. Valid baud rates are 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200.	This error usually results from an incorrect value in the command block for the Create Connection command. Check the value and correct any entry errors. Ensure that the command slot value points to the correct command block.
29A2	10658	PARITY SELECTION ERROR The protocol manager was passed an invalid parity value. Valid parity values are 0-None, 1-Odd, 2-Even.	See error code 29A0 above.
29AC	10668	MEMORY READ ERROR The 2573 was unable to read the requested V memory	When this error occurs, it is usually due to an invalid memory address specification in the command block. Check the value and make corrections if necessary.
29AD	10669	MEMORY WRITE ERROR The 2573 was unable to write the requested V memory.	See error code 29AC above.
29C8	10696	INVALID END OFMESSAGE INTERVAL	Value in create connection block for inter-message gap time is <5 or >100.

Modbus Slave Error Codes

HEX	DEC	DESCRIPTION	SOLUTION
2A98	10904	COULD NOT OPEN PORT The protocol manager could not open the 2573 serial port. This may occur during a Create Connection command.	This could result from a hardware error, which may be transient. Retry the command. If the error persists, reset the module. If the error re-occurs after reset, replace the module.
2AC8	10952	INVALID INTER-MESSAGE TIMEOUT	Value in create connection block for inter-message gap time is <5 or >100.
2B98	11160	COULD NOT OPEN PORT The protocol manager could not open the 2573 serial port. This may occur during a Create Connection command.	This could result from a hardware error, which may be transient. Retry the command. If the error persists, reset the module. If the error re-occurs after reset, replace the module.

General ASCII Support (GAS) Protocol Manager Error Codes

The error codes generated by the GAS protocol manager can be found in the *CTI General ASCII Protocol Manager Reference Manual.*

2573 Internal Error Codes

Error codes which have a value greater than hex C8 in the *low* byte are internal code errors. If you should experience one of these errors, contact CTI for assistance.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

APPENDIX C. OPERATIONAL REFERENCE

Hardware Handshake Operation

If hardware handshaking is enabled, the Request to Send (RTS) and Clear to Send (CTS) hardware handshake line are controlled as follows:

- *Input Flow Control* (RTS). When hardware handshake is enabled, the 2573 module controls the RTS control line. RTS is normally high unless the input buffer is almost full.   If the number of characters in the input buffer exceeds a preset value (about 2/3 of the buffer size), the 2573 module will pull the RTS line low.   When the number of characters in the buffer goes below this value, the 2573 module will drive RTS high.

- *Output Flow Control (CTS)*. When hardware handshake is enabled, the 2573 module senses the state of the CTS control line.   When the state of the line is high, the 2573 module will transmit if there are characters to be transmitted.   If the state of the line is low, no output occurs.

NOTE:
Unless the device you are attaching explicitly requires hardware handshaking, you should leave hardware handshaking disabled.

Modem Handshake (RTS/CTS) Operation

When the Modem RTS/CTS option is selected the 2573 module controls the RTS and CTS lines as follows:
- The 2573 module asserts the RTS signal when it has data to send
- When the modem asserts the CTS signal the 2573 module will begin sending data
- The 2573 module will continue to send data as long as CTS is asserted (and there is data to send)
- If the 2573 module detects that CTS is de-asserted, it will stop sending data until CTS is re-asserted.
- When the 2573 module has no more data to send, it will de-assert RTS.

NOTE:
Because the 2573 will not transmit unless CTS is high, output data will be stored in the serial port output buffer until the modem asserts CTS. If the modem never detects CTS high (e.g. the cable is disconnected), then the 2573 will drop RTS only when the command times out.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

APPENDIX D. PLC COMMAND INTERFACE

General Description

The module logs into the PLC as 2 WX inputs and 6 WY outputs.   The PLC controls the module by placing values in specific WY words. The PLC obtains status information from the module via the WX words. Areas of V memory called *Command Blocks* are used to store command information.

Figure 23 illustrates the 2573 PLC Command Interface. The control structures consist of the module
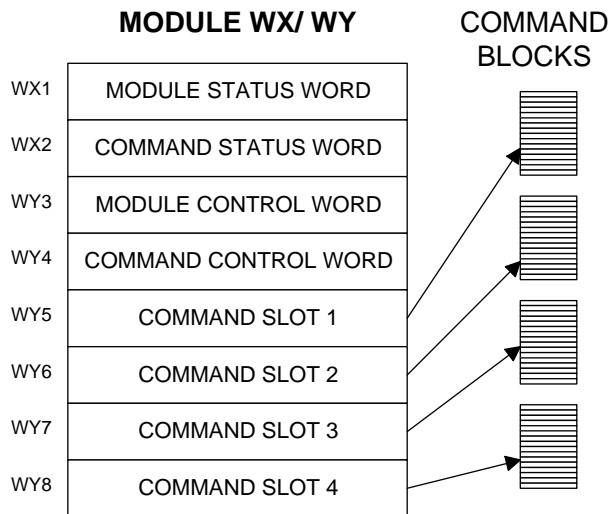
Figure 23. PLC Command Interface

WX/WY words and the Command Blocks. Words WY5-WY8 contain the V memory addresses of the command blocks. Using this command interface, the PLC can initiate up to four commands at one time. The status and control bits for the four commands are located in WX2 and WY4, respectively.

Command Processing Modes

You can specify how the module command execution is synchronized with the PLC logic by setting the Command Mode bit in WY4. The Command Mode bit is described in detail later in this appendix. You can select either *coupled* mode or *uncoupled* mode.

Coupled Mode Processing

Coupled Mode provides for tight "handshaking" between the module and the PLC. When the PLC triggers the command, the module will execute the command *only once.* The module will then wait for acknowledgment from the PLC before allowing the command to be triggered again. Coupled mode is useful for initiating transfers that must be done exactly one time per PLC event (such as Starting the Network Server).

Uncoupled Mode Processing

Uncoupled Mode does not require a handshaking sequence from the PLC. When Uncoupled Mode is selected, the module will execute the command continuously as long as the command trigger is set. Uncoupled mode is useful for providing continuous updates of PLC data.

Command Processing Procedure

Following is the typical procedure used to initiate a module command:

- Set up the Command Block(s) in V memory using you PLC programming software.

- Load the command slot(s) with the address of the Command Block(s).

- Set the applicable command trigger and mode bits in WY4.

- Monitor the status of the command processing using WX2.

WX/WY Description

Figure 24 shows the layout of the WX and WY words assigned to the module. WX1 and WY3 are used for global module control and status. The remaining words are used for command processing. Note there are four command slots, which allows the PLC to trigger up to four module commands in one scan. The four sets of bits in WX2 and WY4 correspond to the four command slots.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Module Status Word																WX1
Command Status Word																WX2
CMD 1 Status Bits				CMD 2 Status Bits				CMD 3 Status Bits				CMD 4 Status Bits				
Module Control Word																WY3
Command Control Word																WY4
CMD 1 Control Bits				CMD 2 Control Bits				CMD 3 Control Bits				CMD 4 Control Bits				
Command Slot 1: V Memory Address of Command Block for CMD1																WY5
Command Slot 2: V Memory Address of Command Block for CMD2																WY6
Command Slot 3: V Memory Address of Command Block for CMD3																WY7
Command Slot 4: V Memory Address of Command Block for CMD4																WY8

Figure 24. WX/WY Map

Module Status Word

The module Status Word is located in the module WY1. This word is used to communicate the overall status of the module. The module uses the following bit structure:

1	2	3	4	5	6	7	8	9 - 15
MOD FAIL	SER CFG	NET CFG	DIAG ERR	CFG ERR	AUI ACT	RSVD	RSVD	Timer E Code

MOD FAIL - *Severe Module Failure*

If the diagnostics discovers a severe fault, the module sets the MOD FAIL bit and (usually) one of the of the error bits to indicate the problem source. After a delay of about 1 second, the module will perform an internal reset and attempt to perform normal processing. If the fault recurs, the wait and reset cycle will be repeated indefinitely. If the fault is cleared, the module will resume normal operations.

SER CFG - *Serial Configuration Required*

This bit will remain set until *all* serial ports have been configured. When all ports have been configured, this bit will be cleared.

NET CFG - *Network Configuration Required*

Not used by the 2573 module.

DIAG ERR - *Diagnostic Detected Error*

If the module diagnostics detect a faulty RAM or ROM, the DIAG ERR bit will be set. When this type of error occurs, the problem is severe. The module will also set the module fail bit and perform a reset after a short delay.

CFG ERR - *Configuration Error*

This bit will be set if the module detects invalid hardware switch settings or invalid EEPROM data at startup / reset time.

AUI ACT - *AUI Active*

Not used by the 2573 module.

RSVD - *Reserved*

These bit positions are reserved for future use.

TIMER/ ECODE - *Timer or Error Code*

When the module processor is operating and there are no significant errors, the lower 8 bits of WX1 will display the value of a module timer. If a hardware error occurs, an error code will be written to this field, if possible.

NOTE:
The changing timer bits are an indication that the module processor is operating.

Module Control Word

Module Control Word (WY3)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mod Rst	Reserved Bits (Set to 0)														

Module Control Bits are contained in WY3. Bit 1 is a module reset bit. **When the bit is set *and* all four abort triggers are set, the module will perform a hardware reset.** This operation resets the processor and starts all hardware diagnostics and startup functions. It is available for extreme error cases where simply acknowledging an error or aborting a command does not clear the error.

NOTE:
Once the module has reset your PLC logic must re-execute any commands used to configure the module or to start the network server.

Bits other than bit 1 are reserved for future use and should be set to 0.

Command Control and Command Status Words

The command status word and the command control word contains bits that are used to synchronize and control the execution of module commands. The Command Control Word is located in module word WY4 and contains bits that are set by the PLC logic. The Command Status Word is located in WX2 and contains bits that are set by the module.

Since the PLC can send up to four commands to the module per scan, both the Command Control Word and the Command Status Word contain four sets of bits, one set for each possible concurrent command. For example, the first four bits (bits 1-4) in both words relate to command Slot 1, the next four bits (bits 5-8) relate to command Slot 2, etc.

The four command slots execute commands independently. Thus, an error on one command does not affect processing on the other commands than happen to be executing at the same time. The command slots can be triggered at different times or some (or all) may be triggered concurrently.

Command Status Word

The Command Status Word contains bits that are set by the module. The following illustration indicates how the bits within the word are used.

COMMAND STATUS WORD (WX2)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CMD Err	PLC Err	CMD Busy	Abrt Busy	CMD Err	PLC Err	CMD Busy	Abrt Busy	CMD Err	PLC Err	CMD Busy	Abrt Busy	CMD Err	PLC Err	CMD Busy	Abrt Busy
COMMAND 1				COMMAND 2				COMMAND 3				COMMAND 4			

CMD Err   *Command Error.* The 2573 sets this bit when it encounters an error while attempting to execute the command. The 2573 will write an error code into Offset 0 of the command block unless PLC ERR is also set (see below).

PLC Err   *PLC Read/Write Error.* The 2573 sets this bit to indicate that it could not read a valid command block from V memory. Therefore, there is no place to write an error word. This may be caused by loading a value in the Command Slot which is not a valid V memory location (such as 0) or which points to V memory that does not contain a Command Block.

CMD Busy   Indicates the module is in the process of executing a command.

ABORT Busy   Indicates the module is attempting to abort a command.

Command Control Word

The Command Control Word contains bits that are set by the PLC logic. The following illustration indicates how the bits within the word are used.

COMMAND CONTROL WORD (WY4)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ERR Ack	CMD Mode	CMD Trig	Abort Trig	ERR Ack	CMD Mode	CMD Trig	Abort Trig	ERR Ack	CMD Mode	CMD Trig	Abort Trig	ERR Ack	CMD Mode	CMD Trig	Abort Trig
COMMAND 1				COMMAND 2				COMMAND 3				COMMAND 4			

CTI 2573-MOD2/TCM2 Installation and Operation Guide

ERROR Ack   *Error Acknowledge.* The PLC sets this bit to acknowledge the error bit which was set by the 2573.

CMD Mode   *Command Mode.* The PLC sets this bit to control how the module command execution is synchronized with the PLC logic:

         0 = Uncoupled mode - The 2573 command processing is not synchronized with the PLC logic.

         1 = Coupled mode - The 2573 command processing is synchronized with the PLC logic.

CMD Trig   *Command Trigger.* The PLC sets this bit to initiate a command.

ABORT Trig   *Abort Trigger.* The PLC sets this bit to abort a command in progress.

WX2 and WY4 Relationships

The following illustrates the relationship between the first four bits in WX2 and WY4. This relationship applies to the other three sets of bits.

Word	Bit 1	Bit 2	Bit 3	Bit 4
WX2	CMD Err	PLC Err	CMD Busy	Abort Busy
WY4	ERR Ack	CMD Mode	CMD Trigger	Abort Trigger

Error Processing Bits

The module will set CMD ERR any time an error occurs in processing a command. The module will also set PLC ERR if a valid Command Block cannot be read from V memory. After setting the error bits, the module will halt processing on this command. The PLC acknowledges the error by setting ERR ACK. When the module sees ERR ACK go high, it aborts the command, clears the error bits, and is ready to accept another command trigger for this slot.

Command Processing bits

The PLC sets the CMD MODE bit to select whether Coupled Mode or Uncoupled Mode is used. The PLC sets the CMD TRIGGER to initiate the command. When the module begins command processing, it sets the CMD BUSY bit. The CMD BUSY bit will remain on until the module has competed processing the command.

Abort Processing Bits

The PLC can set the ABORT TRIGGER to request the module to abort a command that is in process. When the module starts the process of aborting a command, it raises the ABORT BUSY bit. When the process is complete, it lowers ABORT BUSY (and also CMD BUSY).

The timing diagrams in this appendix describe the interaction of the various Command Control and Command Status bits.

Command Slots (WY5-WY8)

Module words WY5, WY6, WY7, and WY8 are called Command Slots. There are four command slots, one for each possible concurrent command. The value in the command slot points to the V memory address of a command block as illustrated below. Addresses of 0 or less are invalid. Addresses greater than 64k cannot be referenced.

COMMAND SLOTS

WY5	Command Slot 1: V Memory Address of Command Block for CMD1
WY6	Command Slot 2: V Memory Address of Command Block for CMD2
WY7	Command Slot 3: V Memory Address of Command Block for CMD3
WY8	Command Slot 4: V Memory Address of Command Block for CMD4

Note that the command slots match up with the command control and command status bits in WY4 and WX2. For example, the command block whose address in Command Slot 1 will be triggered by setting bit 3 in WY4 (Bit 3 is the Command Trigger for Command 1).

Command Blocks

Command Blocks contain information which specifies how the command is to be executed. The standard Command Block consists of 16 contiguous words located in the first 64k of PLC V memory and has the following general structure.

Offset	Contents
0	Error Word
1	Command Code
2	Connection Number
3	Command Parameter 1
4 - 15	Additional parameters

Error Word   Assuming the Command Block is valid, the module will write an error code into this location when it encounters an error in executing the command. The error code can be used during development as an aid to de-bugging logic. It may also be used by the PLC logic as an indication to execute specific error processing logic.

Command Code   The Command Code is a number which uniquely defines the command you wish to execute.

Connection Number   The connection number identifies the instance of the protocol manager with which you are communicating. A connection number is similar to a file handle used in other operating environments. The connection number is established when a CREATE CONNECTION or CREATE SOCKET command is executed. During this process, a specific protocol manager is loaded and associated with a particular physical port or a network socket. If the connection is created using PLC logic, the programmer can assign a connection number. If the connection is automatically created by the module, the module will assign a connection number.

The high byte of the Connection number should always contains hex 4B. The module uses this byte as a Command Block "signature". The use of the signature byte provides a way for the module to verify that the address you loaded in the command slot indeed points to a valid command block and not some arbitrary area of V memory. When the module detects that you have referenced an area of V memory where the high byte of offset 3 does not contain hex 4B, it asserts the CMD Err and PLC Err. Since the module assumes that a block without a signature byte is not a command block but a reference error, it does not attempt to process the block of V memory and does not write any error information back into offset 0 of the block .

Command Parameters   The command parameters provide command execution information. For example, for a Read Command they may specify where to obtain the data, how many words to read, and where to store the results. The number of command parameters required will vary with different commands.

Command Timing Diagrams

Coupled Mode Timing (CMD Mode = 1)

Coupled Mode should be selected when you want to ensure that the module will execute a command only once per trigger. In Coupled Mode the PLC and the module use the CMD TRIG (Command Trigger) and the CMD BUSY (Command Busy) bits in a "handshaking" sequence. The following figure illustrates the timing relationships between CMD TRIG and CMD BUSY for a Coupled Mode command.

The PLC has cleared CMD TRIG (Time A) and the module has cleared CMD BUSY (Time B) indicating that the processing of the previous command is complete.

At Time C, the PLC asserts CMD TRIG to initiate a command. When the module recognizes that CMD TRIG is high, it asserts CMD BUSY (Time D) and samples the CMD MODE bit to determine the command mode. It then retrieves the command block and starts processing the command. Any time after the PLC sees CMD BUSY go high, it may lower CMD TRIG (Time E).
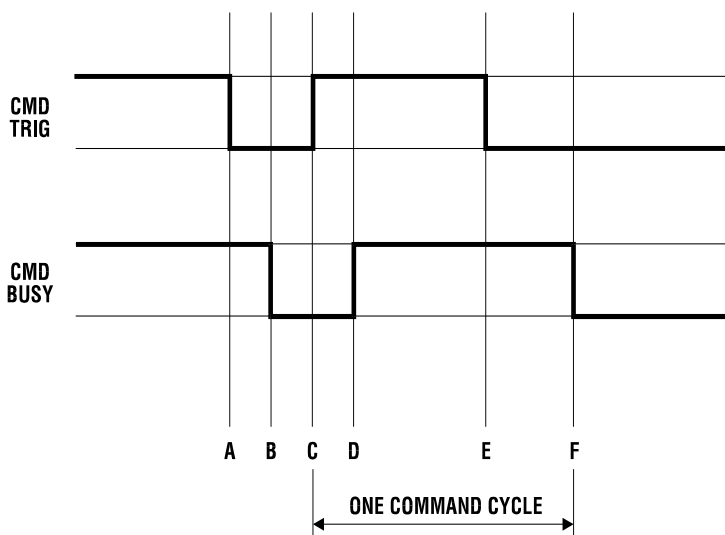
Figure 25. Coupled Mode Timing

Once the module completes processing the command, it samples CMD TRIG. If CMD TRIG is off, the Module will lower CMD BUSY (Time F) and wait for the next trigger. If CMD TRIG is on, the module will hold busy high and wait until CMD TRIG is turned off. Time F will be delayed as required until CMD TRIG is off. At time F (equivalent to time B), both CMD TRIG and CMD BUSY are low and a new cycle can begin whenever the PLC asserts CMD TRIG.

> *NOTE:*
> *The Command Block is subject to access and change by the module anytime after CMD TRIG is asserted (time C) until CMD BUSY is cleared (time F), Thus, the Command Block should not be changed by the PLC at any time between time C and F.*

CTI 2573-MOD2/TCM2 Installation and Operation Guide

Uncoupled Mode Timing (CMD Mode =0)

Uncoupled Mode may be selected when close synchronization between the PLC and the module is not required. In this mode, a single trigger can cause the module to execute a command many times. Uncoupled mode is useful in applications where you want to continuously update the remote PLC. Uncoupled Mode usually takes less PLC logic to implement than Coupled Mode. The following figure illustrates the timing relationships between CMD TRIG and CMD BUSY for an Uncoupled Mode command. The shaded area in the diagram indicates that it does not matter what the status of the bit happens to be (*don't care* condition).

At time A, the module has cleared CMD BUSY, indicating that Command processing is complete. When CMD BUSY is low and CMD TRIG is high the module will assert CMD BUSY (Time B) and begin command processing. CMD BUSY will remain on during the processing of the Command Block. When the processing is complete, the module will lower CMD BUSY (Time C). *Note that in Coupled Mode, the module would wait for the PLC to lower the trigger before clearing CMD BUSY.*

At the next end of the PLC I/O scan, the module will again sample CMD TRIG. If CMD TRIG is high, the module will raise CMD BUSY and processing of the Command Block will start again.
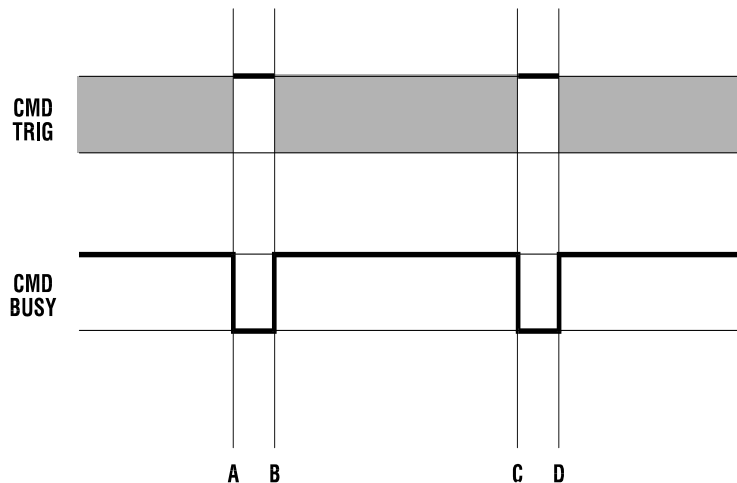
Figure 26. Uncoupled Mode Timing

Note that the PLC logic is not required to cycle CMD TRIG and that the PLC may change the state of CMD TRIG at any time. As soon as the module completes a command and lowers CMD BUSY, it will sample CMD TRIG. If CMD TRIG is set, it will execute the command.

If constant operation is desired, the PLC can simply leave the CMD TRIG on. The Module will then repeat the command as fast as possible.

NOTE:
The contents of the Command Block are subject to access and change by the module at any time from the assertion of CMD TRIG (Time A) until CMD BUSY is cleared (Time C). If CMD TRIG is held high, the CMD BUSY transition may never be detected by the PLC. Therefore, if you wish to change the command to the module, the PLC logic should clear CMD TRIG and ensure CMD BUSY is cleared before changing the contents of the Command Block or the associated Command Slot.

Command Error Timing

The module provides a mechanism for command execution communicating errors to the PLC. This is accomplished via the CMD ERR bit and the ERROR WORD in the Command Block. The following figure illustrates Command Error timing. Note that the shaded area indicates a "don't care" condition for the particular bit.

At time A, the module has set CMD BUSY and is attempting to retrieve the command block and to process the command.

Some time later, a command processing error occurs. The module writes error information into the error word of the Command Block and asserts CMD ERR at time B. Processing of this command is halted.

If the module cannot write the error word to the Command Block, it also asserts the PLC ERR bit (see the following section). *The error word is valid only when CMD error is asserted and PLC ERR is clear.*
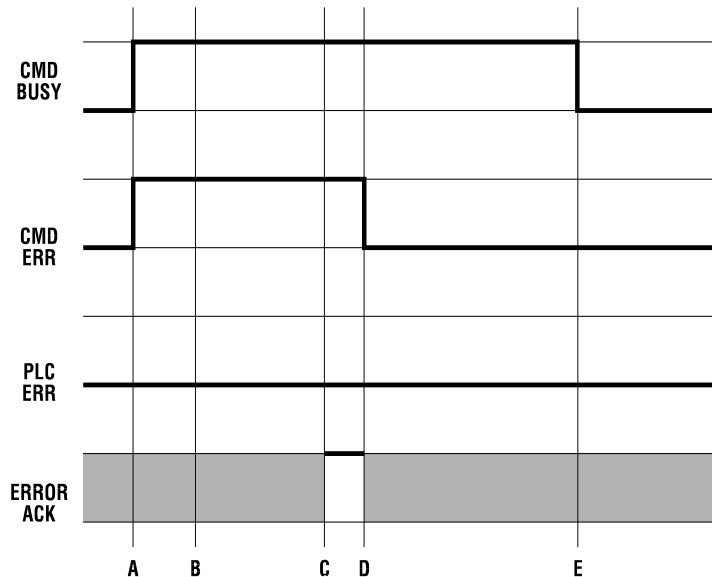
Figure 27. Command Error Timing

At the end of each PLC I/O scan, the module samples the state of ERROR ACK (Error Acknowledge). As soon as the module detects that ERROR ACK is set (Time C), it will turn off CMD ERR (and PLC ERR if on) at Time D and proceed with the error recovery.

After error recovery is competed, the module will clear CMD BUSY (Time E). This may or may not be the in the same PLC I/O scan. CMD ERR will never be asserted unless CMD BUSY is asserted. CMD ERR is always cleared prior to or at the same time as the release of the CMD BUSY signal (Time D is always before or the same as time E).

NOTE:
The contents of the error word in the Command Block is valid only when CMD ERR is asserted and PLC ERR is clear.

The ERROR ACK signal is sampled only when CMD ERR or PLC ERR is asserted; transitions are not part of the timing cycle. Thus, if you wish to ignore all errors in your PLC logic, the PLC can set the ERROR ACK bit to 1 and leave it on.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

PLC Error Timing

A PLC error occurs when the module cannot read a valid Command Block from the PLC V memory or it cannot write back to this location. The following diagram illustrates timing for the PLC ERR bit. Note that the shaded area indicates a "don't care" condition for the particular bit.

At time A, the module has set CMD BUSY and attempts to retrieve the command block from V memory. At time B, the module determines it cannot retrieve a valid command block and asserts the CMD ERR and the PLC ERR bit. Since a valid command cannot be retrieved, the module halts processing of this command.

At time C the module samples the ERROR ACK bit and determines that it is asserted, indicating that the PLC has acknowledged the error. Some time later the module clears the CMD ERR and PLC ERR bits (Time D) and the CMD BUSY bit (Time E). The error bits will always be cleared prior to or at the same time as CMD BUSY.
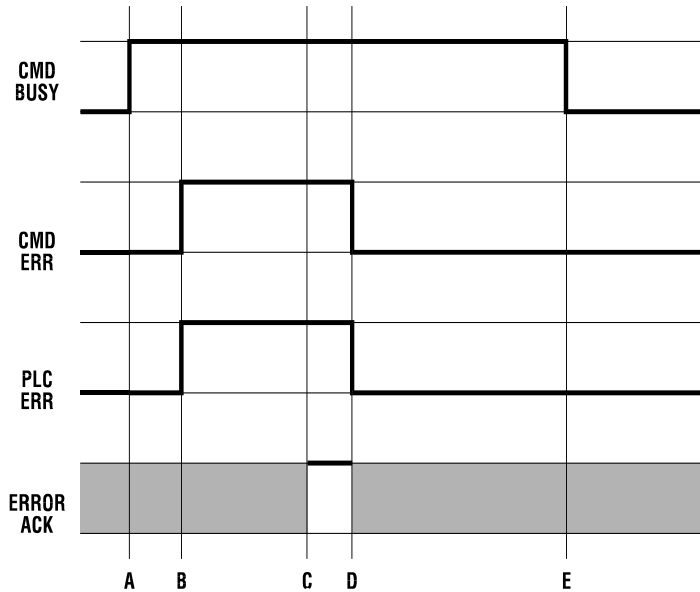
Figure 28. PLC Error Timing

Note that, in this example, no error code is written to the command block. Since a Command Block could not be read, the module does not attempt to write an error code into the V memory location indicated in the command slot.

NOTE:
When the PLC ERR bit is set, it is almost always the result of :
1) an invalid address in the Command Slot (WY5, WY6, WY7, or WY8) or
2) a command block connection number that does not have hex 4B in the high byte
If the PLC ERR bit is on, you should check your PLC logic carefully!

Command Abort Timing

There may be rare occasions when you need to abort a command that is in progress. For example, if you triggered a READ REMOTE command with an infinite timeout and the data was never returned, the protocol manager would wait forever, blocking further execution in the command slot. Figure 29 illustrates abort timing.

Assume the module is busy processing a command (Time A). At time B, the PLC asserts ABORT TRIG to stop the ongoing command processing. When the module recognizes that ABORT TRIG is high, the module will assert ABORT BUSY (Time C).

After detecting ABORT BUSY, the PLC should clear ABORT TRIG (Time D). When the Module has completed abort processing, it clears ABORT BUSY (Time E).

After ABORT BUSY is cleared, the module will clear CMD BUSY (Time F). Time F may be on the same scan or on a later scan than time D. ABORT BUSY will never be asserted unless CMD BUSY is also asserted.
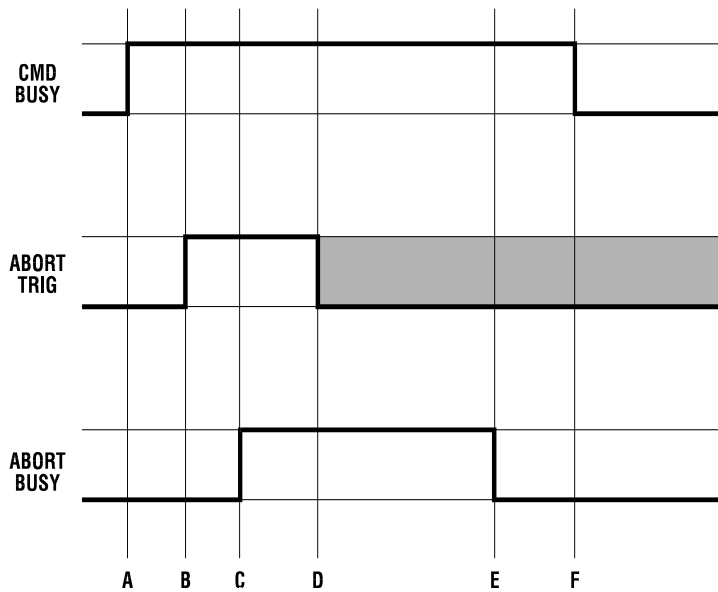
Figure 29. Command Abort Timing

NOTE:
ABORT BUSY will be cleared even if the ABORT TRIG is still active. If the PLC does not clear ABORT TRIG, no further commands will be processed for the Command slot.

Depending upon the nature of the command and circumstances of execution, some commands cannot be aborted.

NOTE:
Setting all four abort triggers and asserting the module reset bit in WY3 will cause the module to perform a complete reset.

CTI 2573-MOD2/TCM2 Installation and Operation Guide

General Command Considerations

When designing your PLC logic you should be aware of the following:

- The Command Trigger is ignored for a command if the associated Abort Trigger is set.

- In Uncoupled Mode, the PLC may not see CMD BUSY go low in between command cycles. After you lower the trigger, the module may execute the command additional times.

- If ERR ACK is set and left high, the associated CMD ERR bit may be set and cleared without the PLC detecting it.

With the notable exception of Module Reset (all four abort triggers are set along with the module reset bit) the command slots are totally independent. Unless the commands are directed to a common resource, the commands should have no effect on each other.

WX / WY Quick Reference

1	2	3	4	5	6	7	8
MOD FAIL	SER CFG	NET CFG	DIAG ERR	CFG ERR			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Module Status Bits								Timer or Error Value								WX1
Cmd 1 Status Bits				Cmd 2 Status Bits				Cmd 3 Status Bits				Cmd 4 Status Bits				WX2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy
cmd 1 status				cmd 2 status				cmd 3 status				cmd 4 status			

Bit 1 = Reset Module Control Bits	WY3
Cmd 1 Control Bits Cmd 2 Control Bits Cmd 3 Control Bits Cmd 4 Control Bits	WY4
Cmd 1 V Memory Address of Command Block	WY5
Cmd 2 V Memory Address of Command Block	WY6
Cmd 3 V Memory Address of Command Block	WY7
Cmd 4 V Memory Address of Command Block	WY8

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig
command 1				command 2				command 3				command 4			

APPENDIX E. MODBUS REFERENCE

General Information

Many plant floor devices use a subset of the Modicon Modbus Protocol for serial communications. Modbus devices communicate using a master-slave technique, in which only one device (master) can initiate transactions (called queries). The other devices (slaves) respond by supplying the requested data to the master or by performing the action requested in the query. The 2573-MOD module provides the protocol managers necessary for the user to use the module as either a master or as a slave in their application.

This protocol is fully specified in the Modicon Modbus Protocol Reference Guide (PI-MBUS-300 Rev E) published by:

> MODICON Inc. Industrial Automation Systems
> One High Street
> North Andover
> MA 01485

This chapter assumes that you have access to this document or the equivalent information.

Serial Transmission Modes

The Modbus protocol specification includes two serial transmission modes, ASCII and RTU. The selection of ASCII or RTU will determine the format of messages transmitted serially on the network. ASCII encodes data into characters using specific bit patterns defined by the American Standard Code for Information Interchange. RTU, on the other hand, is a binary protocol.

ASCII is the preferred mode when using modems because it is not timing sensitive. Delays of up to one second between characters will not cause an error. RTU typically provides better performance because it takes fewer bits to transmit a given amount of data.

ASCII Mode

When devices communicate on a network using ASCII, each 8-bit byte in a message is sent as two ASCII characters representing the hexadecimal equivalent of the byte. ASCII messages start with a 'colon' (:) character (ASCII 3A hex), and end with a 'carriage return - line feed' (CRLF) character pair (ASCII 0D and 0A hex). The allowable characters transmitted for all other fields are in hexadecimal (0-9, A-F).

The framing of a Modbus ASCII message is shown below:

Start	Address	Function	Data	LRC Check	End
: (0x3A)	2 Chars	2 Chars	n Chars	2 Chars	<CR> <LF>

Start:	The message begins with an ASCII Colon (hex 3A).
Address:	The address field designates the slave address on a multidrop line.
Function:	The function code specifies the task to be performed (e.g. Read Input Register).
Data:	This field contains the data to/from the device. It may also contain a byte count.
LRC Check:	This field is a Longitudinal Redundancy Check used for error checking.
End:	The message ends with an ASCII Carriage Return (0x0D) and Line Feed (0x0A).

RTU Mode

RTU is a timing dependent, binary format. As opposed to ASCII, where unique characters delimit the beginning and ending of messages, RTU uses a period of silence of 3.5 character times to determine when a message has ended. This manual refers to this time as the End of Message Timeout. A character received after that time is considered to belong to the next message.

The framing of a Modbus RTU message is shown below:

Start	Address	Function	Data	CRC Check	End
> 3.5 Ch Tim	8 Bits	8 Bits	n x 8 Bits	16 Bits	> 3.5 Ch Tim

Start:	Messages start with the first character after a silent interval of at least 3.5 character times.
Address:	The address field designates the address of slave device.
Function:	The function code specifies the task to be performed (e.g. Read Input Register).
Data:	This field contains the data to/from the device. It may also contain a byte count.
CRC Check:	This field is a Cyclical Redundancy Check used for error checking.
End:	The message ends after a silent interval of more than 3.5 character times has passed.

Note:
Since RTU is timing dependent, it may not work with all modems. In general, to work with Modbus RTU, the modems must be configured to send a constant rate bit stream. When using RTU, you should plan on testing your specific brand of modems to ensure compatibility.

Exception Responses

If the Modbus Slave protocol manager receives a query which cannot be processed (for example, a request to read a non-existent coil or register), it will return an exception response informing the master of the nature of the error. The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the slave echoes the function code of the original query in the function code of the response. All function codes have a Most Significant Bit (MSB) of 0. In an exception response, the slave set the MSB of the function code to 1. This makes the function code value in the exception response exactly 80 hexadecimal higher than the value would be for a normal response.

Data Field: In a normal response, the slave returns data or statistics in the data field. In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

Enron Extensions

Enron Corporation has specified extensions to the Modbus protocol for use with Electronic Flow Equipment. These specifications have become a de-facto standard for applications in the oil and gas pipeline industry. The Modbus Slave protocol manager will support Enron Modbus extensions in both ASCII and RTU mode when the appropriate option bit is set.

Function Codes

When the Enron Extensions option bit is set, the Modbus Slave Protocol manager supports the following Enron Modbus Function Codes.

FC (Dec)	FC (Hex)	Function Codes	505 Variable	Comments
01	01	Read Coil Status	C	Max number of coils = 2000
03	03	Read Holding Register	V	Max = 125 16 bit words or 62 32 bit words
05	05	Force Single Coil	C	
06	06	Preset Single Register	V	
15	0F	Force Multiple Coils	C	Max = 800
16	10	Preset Multiple Registers	V	Max = 100 16 bit words or 50 32 bit words

Enron Registers

Enron Modbus defines the format of the registers by the Modbus Address Range. These formats include discrete bits, 16 Bit Integers, 32 Bit Integers, or 32 Bit IEEE Floating Point values as shown in the table below:

Modbus Address	Data Format
1000 - 2999	Status Bits & Commands On/Off
3000 - 4999	16 Bit Integers
5000 - 6999	32 Bit Integers
7000 +	32 Bit IEEE Floating Point

When the Enron Extensions mode is selected, the 2573 Modbus Slave protocol managers will interpret Modbus commands as follows:
- Queries addressing Modbus registers 5000 – 6999 will be interpreted as long (32 bit) words.
- Queries addressing Modbus registers 7000 – 9999 will be interpreted as 32 bit floating-point values.

Register Mapping

To provide flexibility in implementing Enron Modbus, the Modbus Slave protocol manager allows you to define the actual starting address in C and V memory in the Modbus Slave CREATE CONNECTION command block. For example, you can specify that Modbus Coil address 1000 starts at C1 and that Modbus Register address 7000 (Floating Point), starts at V8000. Once you define the location of the floating point registers, all the other word register addresses are translated relative to this value. See the following example.

Assuming the "V MEMORY ADDRESS OF MODBUS REGISTER 7000" parameter is set to 10,000, the Modbus register ranges will be mapped as shown below. Note that 32 bit values consume 2 V memory words. Floating point registers start at V10,000 and extend upward, consuming two V memory addresses for each Modbus register value. Since there are 2000 Long Word registers, consuming two V memory words each , the Long Word registers start at V6000 (4000 words below V10,000). The Enron short words, which consume one V memory word each, start at V4000 (2000 words below V6000).

Modbus Register Address Range	Register Type	V Memory Address Range
1- 2999 (2999 Registers)	16 Bit Integer	1001 – 3999 (2999 Words)
3000 – 4999 (2000 Registers)	Enron Short Word (16 bit integer)	4000 – 5999 (2000 Words)
5000 – 6999 (2000 Registers)	Enron Long Word (32 bit integer)	6000 – 9999 (4000 Words)
7000 – 9999 (3000 Registers)	Enron Floating Point	10,000 – 15,999 (6000 Words)
10,000 +	16 bit Integer	16, 000 +

- Grey area represents Enron Modbus Defined area.

All Enron Modbus 32 bit values consume two V memory words. The following illustrates how floating point values are stored in V memory, assuming the "V MEMORY ADDRESS OF MODBUS REGISTER 7000" parameter is set to 10,000.

Modbus Register	V Memory Address

Address	
7000	10000 10001
7001	10002 10003
7002	10004 10005
7003	10006 10007
7004	10008 10009
7005	10010 10011
7006	10012 10013
7007	10014 10015
7008	10016 10017
etc	etc. etc.

Example 1 – Reading a single floating point register

The following example, which reads one 32 bit floating point value from register 7021 of device 4, illustrates how the special ranges are processed. The value returned 0x476C4A00, represents floating point value 60490.0

Query

SLAVE ADDR	FC	ADDR HI	ADDR LO	# of Registers HI	# of Registers LO
04	03	1B	6D	00	01

Response

SLAVE ADDR	FC	BYTE COUNT	MSW HI	MSW LO	LSW HI	LSW LO
04	03	04	47	6C	4A	00

Assuming the "Modbus Register 7000 V Memory Address" was specified as 7000, the MSW (Most Significant Word) for Modbus Register 7021 (0x476C) is retrieved from V7042 and the LSW (Least Significant Word) is located in V7043. Note that the byte count in the response reflects the number of bytes actually in the message.

Example 2 – Writing a single floating point register

This example writes a floating-point value of 2000.0 decimal (0x44FA000) to Modbus register 7001 (0x1B59) in Modbus Slave 1.

Query

SLAVE ADDR	FC	ADDR HI	ADDR LO	MSW HI	MSW LO	LSW HI	LSW LO
01	06	1B	59	44	FA	00	00

Response

SLAVE ADDR	FC	ADDR HI	ADDR LO	MSW HI	MSW LO	LSW HI	LSW LO
01	06	1B	59	44	FA	00	00

The normal response to this Function code is to echo the query after the register contents have been written. Assuming the "Modbus Register 7000 V Memory Address" is 8000, the MSW will be written to V8002 and the LSW to V8003.

Example 3 – Writing multiple floating point registers

This example writes two long integer values 2046 (0x07FE) and 75984 (0x128D0) to two Modbus registers starting at Modbus register 5020 (0x139C) in Modbus Slave 5.

Query

SLAVE ADDR	FC	ADDRESS		# of Reg		Byte Cnt	R1 MSW		R1 LSW		R2 MSW		R2 LSW	
		HI	LO	HI	LO		HI	LO	HI	LO	HI	LO	HI	LO
05	10	13	9C	00	02	08	00	00	07	FE	00	01	28	D0

Response

SLAVE ADDR	FC	ADDRESS		# of Reg	
		HI	LO	HI	LO
05	10	13	9C	00	02

Restrictions and Limitations

- Reading of variable length Enron Event/Alarm record is not supported.
- Depending upon the "Modbus Register 7000 V Memory Address" value entered, Modbus addresses less than 7000 may generate Modbus exception errors because the request will reference an invalid V memory address.

HARDWARE SPECIFICATIONS

Serial Communications Ports:

4ea. - combined RS-232, RS422, and RS-485
Hardware Handshake lines - RTS, CTS
1500 VDC Isolation - channel to PLC

Switch Configurable Serial Port Communications Parameters:

Data Rate:	1200,2400,**9600,**19200 Baud
Data Bits:	7
Parity:	Odd
Stop Bits:	1
Hardware Handshaking:	**Disabled**, Enabled (switch selectable, RS-232 port)

Values in bold represent the factory switch settings.

Software Configurable Serial Port Communications Parameters

Data Rate:	300, 600, 1200, 2400, 4800, 9600, 19200 Baud
Data Bits:	7 or 8
Parity:	Even, Odd, None
Stop Bits:	1,2
Handshaking:	Hardware, Software, None

Diagnostic LEDs:

1 ea. - Module Status (ACTIVE)
4 ea. - Transmit Data (Serial Port 1-4)
4 ea. - Receive Data (Serial Port 1-4)

Module Size:

Single Wide 505 I/O

Backplane Power Consumption:

6.0 watts @ 5 VDC

Operating Temperature:

0º to 60º C (32º to 185º F)

Storage Temperature:

-40º to 85º C (-40º to 185º F)

Humidity:

0% to 95%, non-condensing

Shipping Weight:

2 Lb.

LIMITED PRODUCT WARRANTY

CTI warrants that this CTI Industrial Product shall be free from defects in material and workmanship for a period of one (1) year after purchase from CTI or from an authorized CTI Industrial Distributor. This CTI Industrial Product will be newly manufactured from new and/or serviceable used parts which are equal to new in the Product.

Should this CTI Industrial Product fail to be free from defects in material and workmanship at any time during this (1) year warranty period, CTI will repair or replace (at its option) parts or Products found to be defective and shipped prepaid by the customer to a designated CTI service location along with proof of purchase date and associated serial number. Repair parts and replacement Product furnished under this warranty will be on an exchange basis and will be either reconditioned or new. All exchanged parts or Products become the property of CTI. Should any Product or part returned to CTI hereunder be found by CTI to be without defect, CTI will return such Product or part to the customer.

This warranty does not include repair of damage to a part or Product resulting from: failure to provide a suitable environment as specified in applicable Product specifications, or damage caused by an accident, disaster, acts of God, neglect, abuse, misuse, transportation, alterations, attachments, accessories, supplies, non-CTI parts, non-CTI repairs or activities, or to any damage whose proximate cause was utilities or utility like services, or faulty installation or maintenance done by someone other than CTI.

Control Technology Inc. reserves the right to make changes to the Product in order to improve reliability, function, or design in the pursuit of providing the best possible Product. CTI assumes no responsibility for indirect or consequential damages resulting from the use or application of this equipment.

THE WARRANTY SET FORTH ABOVE IN THIS ARTICLE IS THE ONLY WARRANTY CTI GRANTS AND IT IS IN LIEU OF ANY OTHER IMPLIED OR EXPRESSED GUARANTY OR WARRANTY ON CTI PRODUCTS, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE AND IS IN LIEU OF ALL OBLIGATIONS OR LIABILITY OF CTI FOR DAMAGES IN CONNECTION WITH LOSS, DELIVERY, USE OR PERFORMANCE OF CTI PRODUCTS OR INTERRUPTION OF BUSINESS, LOSS OF USE, REVENUE OR PROFIT. IN NO EVENT WILL CTI BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH MAY VARY FROM STATE TO STATE.

REPAIR POLICY

In the event that the Product should fail during or after the warranty period, a Return Material Authorization (RMA) number can be requested orally or in writing from CTI main offices. Whether this equipment is in or out of warranty, a Purchase Order number provided to CTI when requesting the RMA number will aid in expediting the repair process. The RMA number that is issued and your Purchase Order number should be referenced on the returning equipment's shipping documentation. Additionally, if the product is under warranty, proof of purchase date and serial number must accompany the returned equipment. The current repair and/or exchange rates can be obtained by contacting CTI's main office at 1-800-537-8398.

When returning any module to CTI, follow proper static control precautions. Keep the module away from polyethylene products, polystyrene products and all other static producing materials. Packing the module in its original conductive bag is the preferred way to control static problems during shipment. Failure to observe static control precautions may void the warranty. For additional information on static control precautions, contact CTI's main office at 1-800-537-8398.

WX / WY Quick Reference

1	2	3	4	5	6	7	8
MOD FAIL	SER CFG	NET CFG	DIAG ERR	CFG ERR			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Module Status Bits								Timer or Error Value								WX1
Cmd 1 Status Bits				Cmd 2 Status Bits				Cmd 3 Status Bits				Cmd 4 Status Bits				WX2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy	CMD Err	PLC Err	CMD Busy	ABORT Busy
cmd 1 status				cmd 2 status				cmd 3 status				cmd 4 status			

Bit 1 = Reset Module Control Bits																WY3
Cmd 1 Control Bits				Cmd 2 Control Bits				Cmd 3 Control Bits				Cmd 4 Control Bits				WY4
Cmd 1 V Memory Address of Command Block																WY5
Cmd 2 V Memory Address of Command Block																WY6
Cmd 3 V Memory Address of Command Block																WY7
Cmd 4 V Memory Address of Command Block																WY8

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig	ERR Ack	CMD Mode	CMD Trig	ABORT Trig
command 1				command 2				command 3				command 4			