2573-MOD Application Note

Upgrading from a Siemens® Simatic® 505-5184 Module

This manual is published by Control Technology Inc., 5734 Middlebrook Pike, Knoxville, TN 37921.

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.

1.0 2573-MOD Differences from 505-5184 Modbus NIM

We at CTI would like to thank you for purchasing our 2573-MOD. You have purchased a communication module that offers a significant amount of more functionality than was available in the Siemens ® Simatic ® 505-5184. The 2573-MOD will provide additional communication capabilities and extend the communication applications required by current systems and additions or changes in the future. Although this application note is primarily structured as a guide for replacing a Siemens ® Simatic ® 505-5184 with a CTI 2573-MOD, we have included reference material out of the 2573-MOD Installation and Operation Guide and some information on the Siemens ® Simatic ®505-5184 as well. For those that are not already familiar with the operation of the 2573-MOD, the next several pages provide a brief tutorial. Those reading this who are already familiar with the 2573-MOD may go on to the actual application example.

An additional significant difference between the Siemens ® Simatic ®505-5184 is found in the serial port electrical communication interface. The Siemens ®Simatic ®505-5184 only supported RS-232 where the 2573-MOD supports RS-232, 422 and 485. If the original application of the Siemens ® Simatic® 505-5184 was connected to a RS-232 to a RS-485 converter, then, with the 2573-MOD the converter is no longer required due to RS-485 support built into the 2573-MOD.

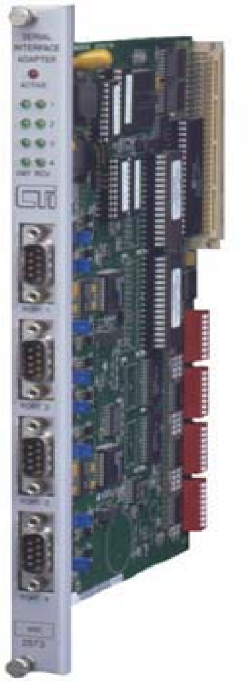
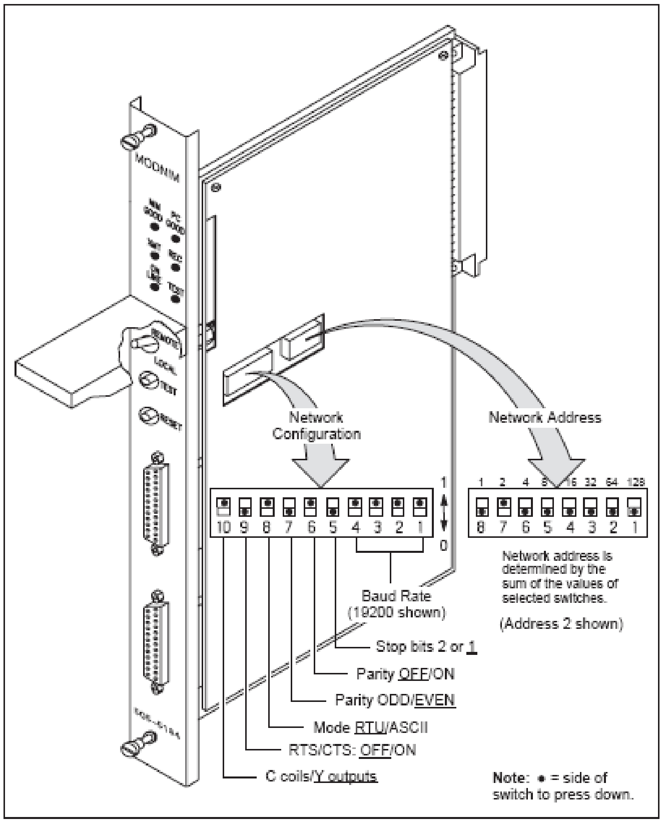
Another difference is in how the 2573-MOD logs into the IO configuration. The Siemens ® Simatic ®505-5184 module logs into the IO as an 8 Y Special Function module. The 2573-MOD logs into the IO as 2WX, 6WY, Special Function Module.

1.1 Introduction

The CTI 2573-MOD and 2573-TCM2 Serial Interface Adapters are a microprocessor-based communications module for Siemens ®Simatic ® 505 controllers and CTI 2500 Series 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. The 505-5184 provided 2 Serial Ports which had the same configuration and basic function while the CTI 2573 module provides 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.

1.2 General Module Front Panel Configuration and PCB

In the diagram below you will see the appearance for both the 2573-MOD and the Siemens ®Simatic ® 505-5184 MODNIM.

2573-MOD Front Panel and PCB	Siemens ®Simatic ® 505 MODNIM Module
	 MODNIM NIM GOOD PC GOOD XMT REC LOC LINE TEST REMOTE LOCAL TEST RESET 505-5184 Network Configuration Network Address 1 0 10 9 8 7 6 5 4 3 2 1 1 2 4 8 16 32 64 128 8 7 6 5 4 3 2 1 Baud Rate (19200 shown) Stop bits 2 or 1 Parity OFF/ON Parity ODD/EVEN Mode RTU/ASCII RTS/CTS: OFF/ON C coils/Y outputs Network address is determined by the sum of the values of selected switches. (Address 2 shown) Note: ● = side of switch to press down. Figure 1-8 Series 505 MODNIM Example Dipswitch Settings

1.3 2573 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 commands 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.

> Note: Complete details and application examples can be found in CTI's 2573 Installation and Operation Guide. CTI Part #062-00187-024

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 specified V memory locations.

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. This is the configuration required when replacing the MODNIM module.

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.

The NITP support emulates the program port of a SIMATIC® 505 series PLC CPU. 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 Siemens ®Simatic ® 505 and TI500 series PLCs. The common application for this protocol is for one PLC to request data or send data to another PLC providing a Peer to Peer communication strategy.

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

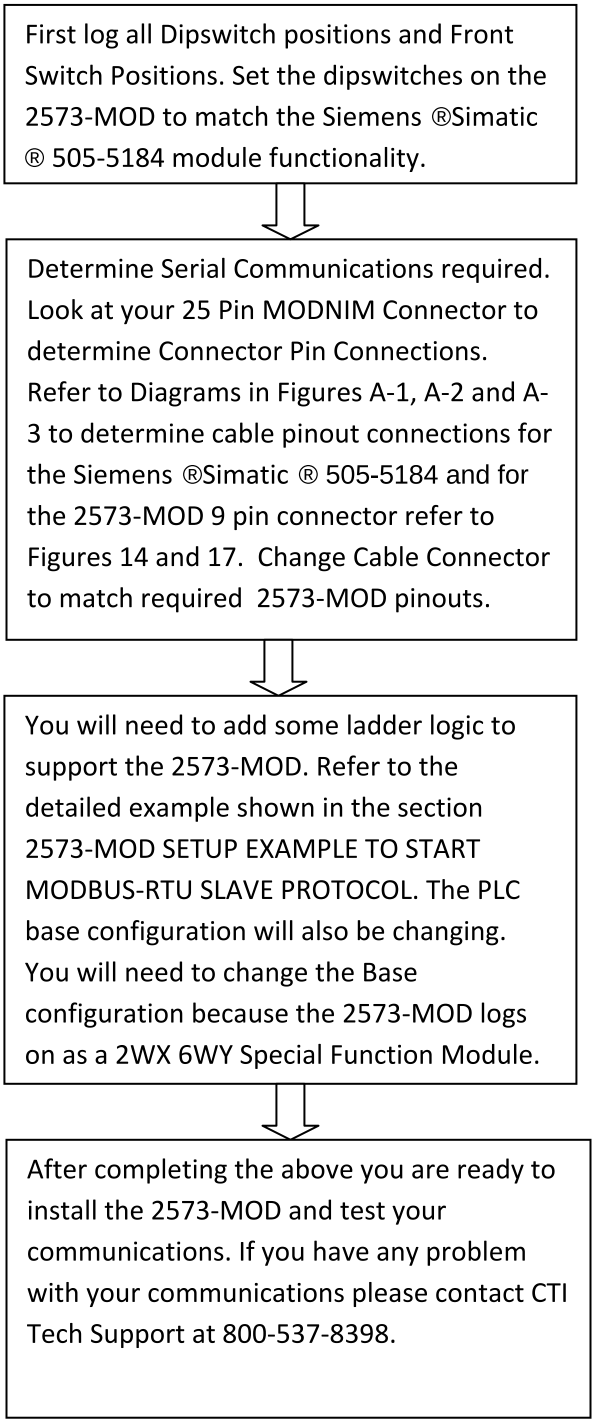
When a 2573-MOD or 2573-TCM2 serial port is configured to use the CAMP Master protocol manager, the PLC can *initiate* messages using the Common ASCII Messaging Protocol format. Because CAMP allows up to 256 words of data to be sent in one message, it offers improved performance over NITP. CAMP is typically 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 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.

Note: For all the protocol listed above please refer to CTI's 2573 Installation and Operation Guide. CTI Part #062-00187-024 for complete details and application examples

1.4 Installation Steps for converting from the 505-6684 MODNIM to the 2573-MOD



First log all Dipswitch positions and Front Switch Positions. Set the dipswitches on the 2573-MOD to match the Siemens ®Simatic ® 505-5184 module functionality.

Determine Serial Communications required. Look at your 25 Pin MODNIM Connector to determine Connector Pin Connections. Refer to Diagrams in Figures A-1, A-2 and A-3 to determine cable pinout connections for the Siemens ®Simatic ® 505-5184 and for the 2573-MOD 9 pin connector refer to Figures 14 and 17. Change Cable Connector to match required 2573-MOD pinouts.

You will need to add some ladder logic to support the 2573-MOD. Refer to the detailed example shown in the section 2573-MOD SETUP EXAMPLE TO START MODBUS-RTU SLAVE PROTOCOL. The PLC base configuration will also be changing. You will need to change the Base configuration because the 2573-MOD logs on as a 2WX 6WY Special Function Module.

After completing the above you are ready to install the 2573-MOD and test your communications. If you have any problem with your communications please contact CTI Tech Support at 800-537-8398.

1.5 First log all Dip Switch Settings and Communication Port Cable Connections on the Siemens ® Simatic ® 505-5184

Log your 25 Pin Out Connections or Enter your Siemens ®Simatic ® 505-5184 Serial Communication Electrical Connection.

Chart 1 Siemens ® Simatic ® 505-5184 DIP Switch Log					
Block of 8 Dipswithes			Block of 10 Dipswitches		
Position	On	Off		On	Off
Switch 1			Switch 1		
Switch 2			Switch 2		
Switch 3			Switch 3		
Switch 4			Switch 4		
Switch 5			Switch 5		
Switch 6			Switch 6		
Switch 7			Switch 7		
Switch 8			Switch 8		
			Switch 9		
			Switch 10		

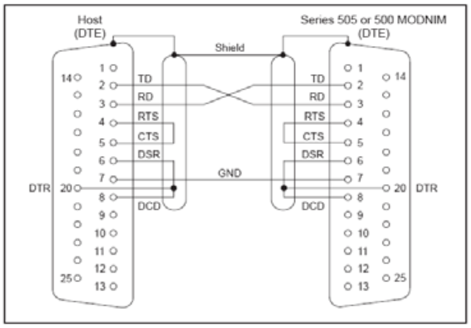


Figure A-1 MODNIM to Host (without Handshaking)

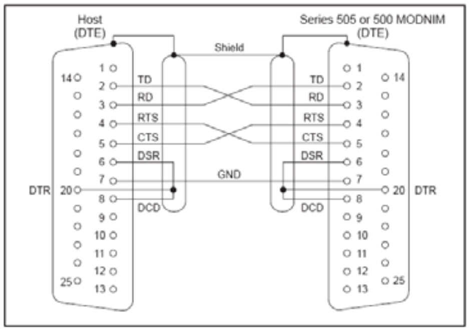


Figure A-2 MODNIM to Host (with Handshaking)

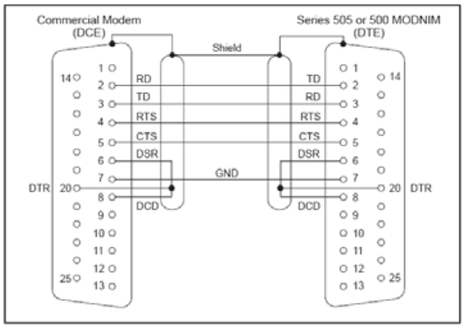


Figure A-3 MODNIM to Commercial Modem (with Handshaking)

Your connector pin out should look like one of these. Identify the one that is yours and refer to our Serial communications -port pinouts that corresponds.

2573-MOD Serial Port Cable Configurations

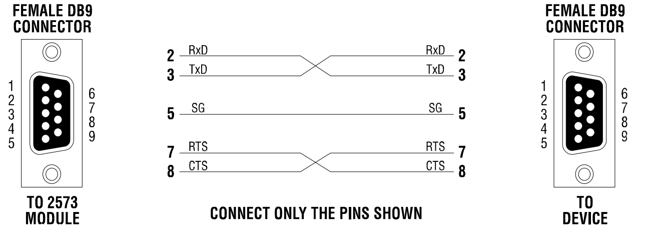


Figure 4 2573-MOD to RS-232 Device Cable

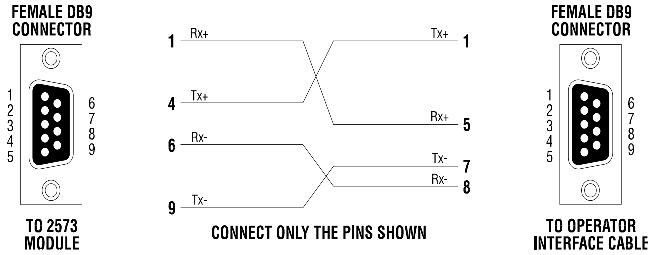


Figure 5 TCAM to RS-422A Adapter Cable

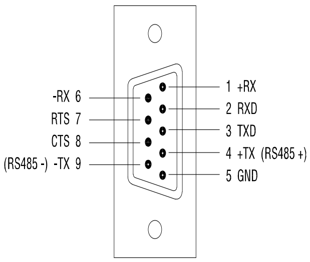


Figure 6 RS232 Cabling

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)

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

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.

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)

1.6 2573-MOD Serial Ports

Unlike the 505-5184 the 2573-MOD has 4 ports which are individually configurable and all four ports can operate concurrently. Thus, you can have up to four differnet types of devices communicating with the PLC on the same 2573-MOD at the same time.

The serial ports use a male DB9 unlike the Siemens ®Simatic ® 505-5184 which is a DB25. The 2573-MOD serial ports provide the most common serial electrical interfaces such as RS-232-C, RS-422 and RS-485-A. You select the electrical interface by connecting the appropraite connector pins to the communications cable as shown in Figures 14 and 17 above. Note: Remember to check your communication cable connections to insure your new 2573-MOD 9 pin connector is wired properly. Please refer to **Appendix A** which includes a copy of Section **1.2. Serial Ports** contained in the 2573-MOD Installation and Operations Guide This section will provide complete details on the 2573-MOD Serial Port cabling. Refer to Chart 2 below for a 505-5184 conversion chart.

Chart 2 - 505-5184 to 2573-MOD Serial Conversion Chart		
505-5184 Serial Port Configuration shown in Figure	Description	2573 Pin Out Figure
Figure A-1	MODNIM to HOST(without Handshaking)	Figure 4
Figure A-2	MODNIM to Host(with Handshaking)	Figure 4
Figure A-3	MODNIM to Commercial Modem(with handshaking)	Figure 4

1.7 2573-MOD LED Indicators

One of the other big differences between the 2573-MOD and the Siemens Simatic 505-5184 is the LED Indicators on the module front panel and the missing Switches. The 2573-MOD has an ACTIVE LED and a Transmit and Receive LED for each port. Refer to Appendix B for the505-5184 MODNIM Switches and LEDs Diagram.

1.7.1 Active LED

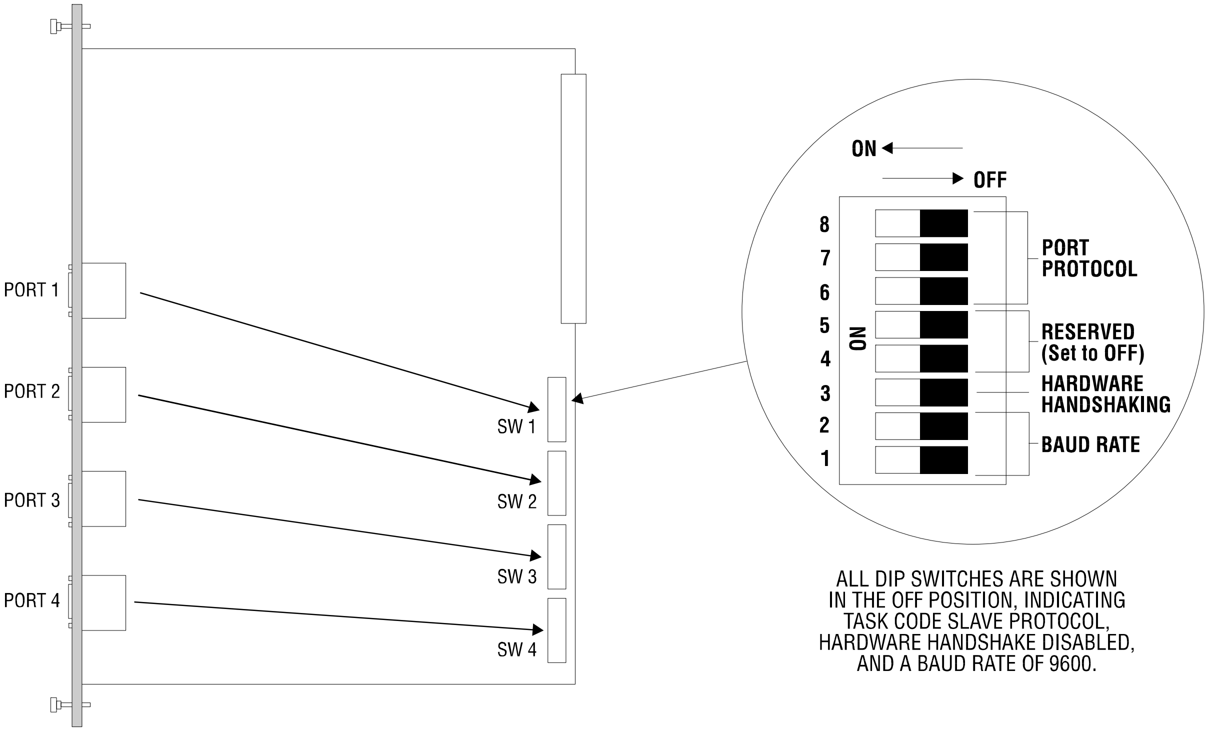
The ACTIVE LED will illuminate steady when the module hardware is performing properly. Should a hardware problem be detected by the module diagnostic software, the LED will cycle OFF to ON about once every two seconds. The LED will also flash about once a second when a Listen Only Modbus command is sent to the 2573-MOD 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.

1.7.2 Transmit / Receive LED’s

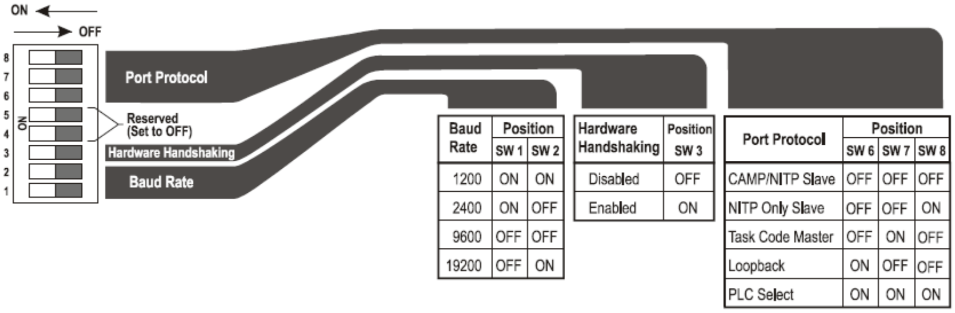
The module provides a pair of Transmit and LEDs for each serial port which flash when a signal is present. The Transmit LED will illuminate when an output signal is present; the Receive LED will illuminate when a signal is being received. During normal data communications activity the LEDs will appear 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.

1.8 Setting the 2573-MOD 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. The figure below shows the 2573-MOD switch locations and group selections.



ALL DIP SWITCHES ARE SHOWN IN THE OFF POSITION, INDICATING TASK CODE SLAVE PROTOCOL, HARDWARE HANDSHAKE DISABLED, AND A BAUD RATE OF 9600.



Baud Rate	Position	
	SW 1	SW 2
1200	ON	ON
2400	ON	OFF
9600	OFF	OFF
19200	OFF	ON

Hardware Handshaking	Position SW 3
Disabled	OFF
Enabled	ON

Port Protocol	Position		
	SW 6	SW 7	SW 8
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

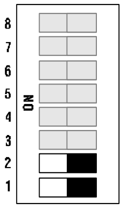
Figure 2 Switchblock Setting Definition Tables

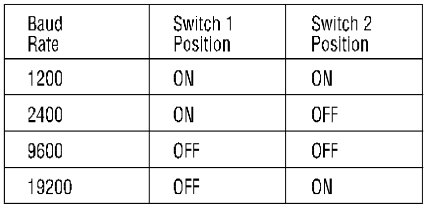
1.9 Serial Port Baud Rate Selection

Refer to the Baud Rate selected on your 505-5184 MODNIM which is found on the Block of 10 Dip Switches switch 1 through 4. Appendix B will provide a diagram of the switch location and settings for the 505-5184.

1.9.1 2573-MOD Serial Port Baud Rate

You can individually set the default baud rate settings for each of the four serial ports. The diagram below applies to all switchblocks (SW1-SW4) for all 4 ports.





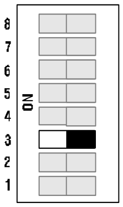
Baud Rate	Switch 1 Position	Switch 2 Position
1200	ON	ON
2400	ON	OFF
9600	OFF	OFF
19200	OFF	ON

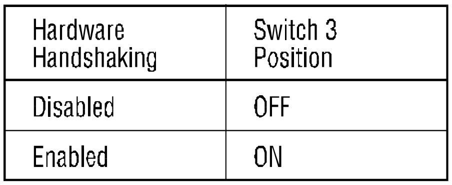
NOTE:
If switches 6, 7, and 8 are set to PLC Select, the baud rate parameters are obtained from the PLC command block. The switch settings for baud rate are ignored.

1.10 Hardware Handshaking

The figure below shows the switch settings for hardware handshake.

NOTE:
If switches 6, 7, and 8 are set to PLC Select, the handshake parameter is obtained from the PLC command block. The switch settings for hardware handshake are ignored.





Hardware Handshaking	Switch 3 Position
Disabled	OFF
Enabled	ON

1.11 RS-422/485 Disconnect Switches

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.

1.11.1 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.

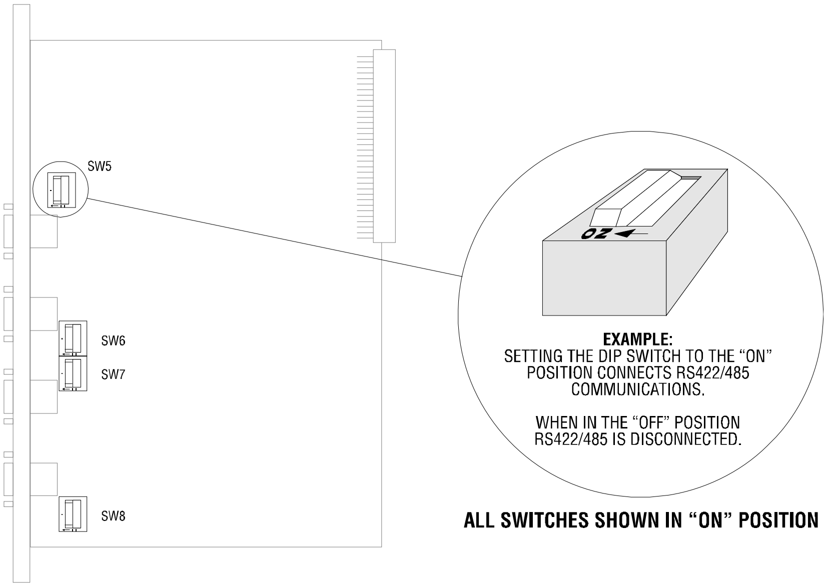

SW5
SW6
SW7
SW8
ON
EXAMPLE:
SETTING THE DIP SWITCH TO THE “ON” POSITION CONNECTS RS422/485 COMMUNICATIONS.
WHEN IN THE “OFF” POSITION RS422/485 IS DISCONNECTED.
ALL SWITCHES SHOWN IN “ON” POSITION

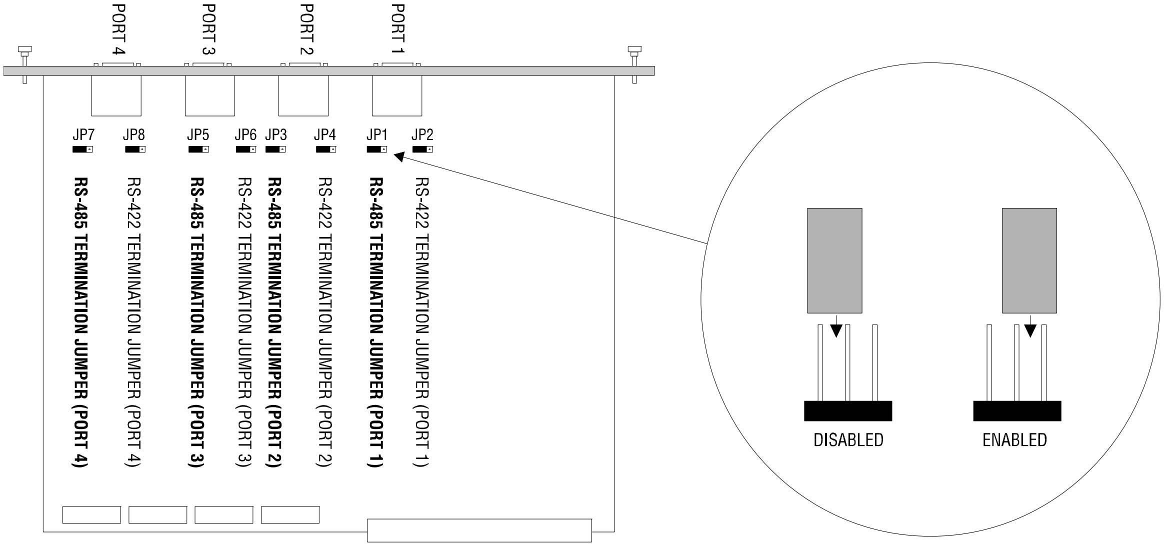
Figure 1. 2573 RS-422/485 Disconnect Switch Locations

1.12 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 disabled (not installed in the circuit).

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.


PORT 4
PORT 3
PORT 2
PORT 1
JP7
JP8
JP5
JP6
JP3
JP4
JP1
JP2
RS-485 TERMINATION JUMPER (PORT 4)
RS-422 TERMINATION JUMPER (PORT 4)
RS-485 TERMINATION JUMPER (PORT 3)
RS-422 TERMINATION JUMPER (PORT 3)
RS-485 TERMINATION JUMPER (PORT 2)
RS-422 TERMINATION JUMPER (PORT 2)
RS-485 TERMINATION JUMPER (PORT 1)
RS-422 TERMINATION JUMPER (PORT 1)
DISABLED
ENABLED

NOTE: ALL JUMPERS SHOWN IN DISABLED POSITION

RS-485 TERMINATION JUMPER (JP1 SHOWN}

2.0 The Modbus Protocol

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.

2.1 INTRODUCTION TO MODBUS SLAVE

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.

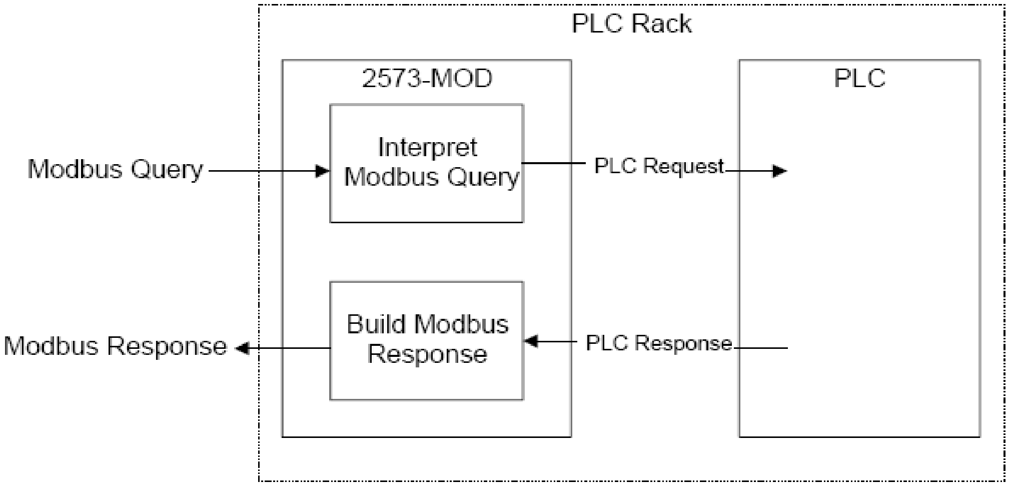


Figure 8. Modbus Slave Command Flow

2.2 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.

2.3 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.

2.3.1 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).

2.3.2 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).

2.3.3 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:

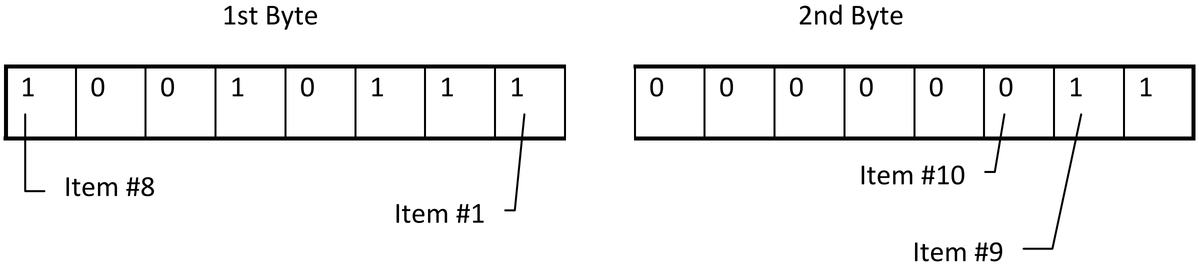
* If the other devices (masters and/or slaves) use the Modbus convention, you will want to choose Modbus Compatibility.
* If the module is on a network with MODNIM modules or a master devices expects you to emulate MODNIM slave, you should choose MODNIM compatibility.

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

2.4 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

1	0	0	1	0	1	1	1

2nd Byte

0	0	0	0	0	0	1	1

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).

2.5 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

2.6 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

2.7 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

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.

2.8 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.

2.9 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 Section 2.3 Relative Addressing Section 2.8 Exception Codes.. 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.
Bit Description	Bit(s)	Definitions

NOTE:
Selecting non-standard end of message timeout in multidrop configurations can cause unpredictable results. For example, multiple broadcast messages which meet the standard Modbus timing specification may be erroneously merged into one message and lost.

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.

NOTE:
When character timing is a problem, the preferable solution is to use Modbus ASCII. Using ASCII may actually result in better performance than using RTU with an long End of Message Interval because the RTU protocol manager must wait until the interval passes before processing the message. 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 *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 in the 2573-MOD and 2573-TCM2 Installation and Operation Guide.

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. .

2.10 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 **Error! Reference source not found.** on page 17 and **Error! Reference source not found.** on page **Error! Bookmark not defined.**. 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.

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 in the 2573-MOD and 2573-TCM2 Installation and Operation Guide..

3.0 2573-MOD SETUP EXAMPLE TO START MODBUS-RTU SLAVE PROTOCOL

As previously discussed in this application note, configuring the 2573-MOD is different from the Siemens ® Simatic ®505-5184 modules. All the configuration of the Modnim is done via dipswitches. Configuring a 2573-MOD Slave port requires setting up a 16 word contiguous “command block” in V memory. This command block contains information for communication parameters, options, module port number, Modbus slave address, along with other information. A minimal amount of ladder logic is also required to execute command processing. The command processing uses the PLC Command Interface, described previously in this document.

The 2573-MOD supports MODBUS RTU protocol Master or Slave as well as MODBUS ASCII protocol Master or Slave. ***MODBUS RTU Master or Slave can only be used on Port #1 or Port #3***. ***MODBUS ASCII Master or Slave can be used on any of the 4 ports.***

STEP 1

SET UP THE 2573-MOD PORT TO BE USED AS MODBUS SLAVE

Whichever port you decide to use for Modbus Slave you must set the dipswitches for that port to “PLC SELECT”. Switch 6, 7, and 8 must be set to the “closed” position. In this example port 1 has been chosen as the Modbus Slave port.

NOTE: Switch 6, 7, and 8 for any ports that are NOT being started by PLC logic should be set for the default state of OFF.

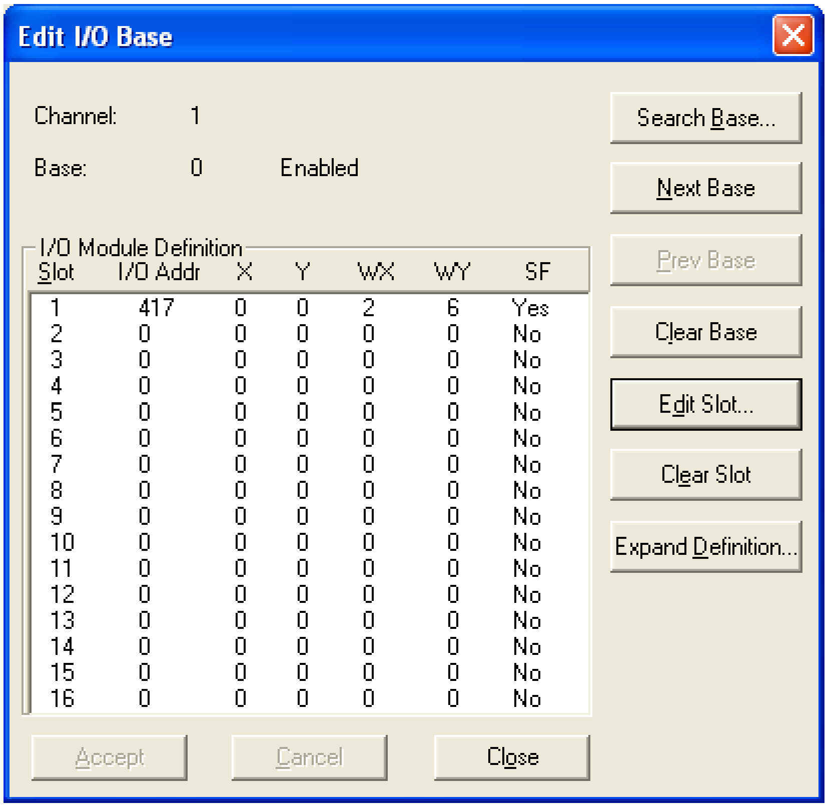
STEP 2

LOG THE 2573-MOD INTO THE PLC IO CONFIGURATION

The 2573-MOD must be logged in to the IO configuration. The 2573-MOD uses 8 contiguous locations of word IO. The module logs in as a 2WX 6WY Special Function Module. The high byte of the first WX word contains module status bits that can be read by ladder logic. The low byte reflects the status of the modules internal counter (counts from 0 to 255) and can be used as and indication that the module is alive and well. The second WX word contain command status bits for each of the four “command slots”. The first WY word contains module control bits that are written to by ladder logic. The second WY word contain command control bits for each of the four “command slots” that can be written to by ladder logic. The starting address assignment may be any WX point on any 8 point boundary that is unused. For this example we have chosen 417 as the starting address. The word assignments for the 2573-MOD will be WX417, WX418, WY419, WY420, WY421, WY422, WY423, and WY424.

3.1 EXAMPLE OF IO ASSIGNMENT FOR THE 2573-MOD IN FasTrak's 505 WORKSHOP

NOTE: Although for this example the 2573-MOD resides in Slot #1 of the local base, the module may be placed in any slot in the local base or any slot in a RS485 Remote Base. ***The 2573-MOD will not work in a Profibus Remote Base due to the fact that Profibus does not support SF Modules.***



Channel: 1

Base: 0 Enabled

I/O Module Definition

Slot	I/O Addr	X	Y	WX	WY	SF
1	417	0	0	2	6	Yes
2	0	0	0	0	0	No
3	0	0	0	0	0	No
4	0	0	0	0	0	No
5	0	0	0	0	0	No
6	0	0	0	0	0	No
7	0	0	0	0	0	No
8	0	0	0	0	0	No
9	0	0	0	0	0	No
10	0	0	0	0	0	No
11	0	0	0	0	0	No
12	0	0	0	0	0	No
13	0	0	0	0	0	No
14	0	0	0	0	0	No
15	0	0	0	0	0	No
16	0	0	0	0	0	No

STEP 3

CONFIGURE "CREATE MODBUS SLAVE COMMAND BLOCK" TO MATCH SETTINGS OF THE 505-5184 THAT YOU ARE REPLACING.

As discussed previously, the 505 MODNIM module options and communication parameters are set up via dipswitches. All the 2573-MOD Modbus Slave options are set up in a V memory table called a "Command Block". You must choose a 16 word block of V memory that is not used by anything else in the program. Some method must be used to load the parameters to the V memory Command block in such a way as to either be loaded by ladder execution or saved as part of the program. In this example a SFSUB program is being used to set the V memory table contents. You will need to examine the 505 MODNIM module for options enabled, communication parameters, and slave address and modify the SFSUB program and/or SFSUB parameters to ensure that it will be configured the same.

NOTE: The parameters from V memory are read only at module start up. The 2573-MOD module does not have a reset button like the 505 Modnim module so if parameter changes are made to the command block, the module must either be reset from ladder logic or power cycled in order for the new parameters to take effect. An example for resetting the module from ladder logic is provided in this example.

STEP 4

SET UP LADDER LOGIC TO PROCESS COMMAND TO START PORT AS A MODBUS SLAVE

NOTE: Refer to *Appendix D of the 2573-MOD Installation and Operation Gu*ide for a detailed description of the PLC Command Interface.

There are many different ways of programming ladder logic which will achieve the same results. The example described here is only one of many methods. The advantage of this method is that it is easy to “cut and paste” this application into existing programs while only having to make minimal changes to apply it to a particular program. This method employs a Special Function Subroutine to enter values into the Create Connection command block required to set up the Modbus Slave for a particular port. This SFSUB makes use of indexing to assign V memory locations. The starting V memory address of the command block is one of the parameters passed to the SFSUB from ladder. This makes is very easy for the user to assign an unused block of V memory to be used for the command block and to change parameters. For an alternate programming example, refer to ***Section 4.11 of the 2573-MOD Installation and Operation Guide.***

The ladder rung shown here provides the logic necessary to start the Modbus Slave for the appropriate port. Some elements of the logic will change based on the starting login address assigned to the 2573-MOD. WX471.2 is the “Serial Config” bit for the 2573-MOD. On startup this bit will be turned ON by the module until all four ports are configured, either by PLC logic or a default configuration by dip switch. When all ports are configured, the module will turn this bit OFF. When WX471.2 is ON, the SFSUB executes that loads the configuration parameters to V memory and loads WY421 with the V memory address pointed to in P1 of the SFPGM into WY421 which is Command Slot 1. The Command Mode bit (WY420.2) and Command Trigger bit (WY420.3) which are associated with Command Slot 1 are turned ON. Once these bits are turned ON, the 2573-MOD executes the command to start the port as a Modbus Slave. Once the port has been configured properly, WX417.2 is turned OFF. This rung is only executed once on module start up.

The parameters passed to SFPGM are as follows:

P1 = Starting V memory address where the parameters are to be assigned.

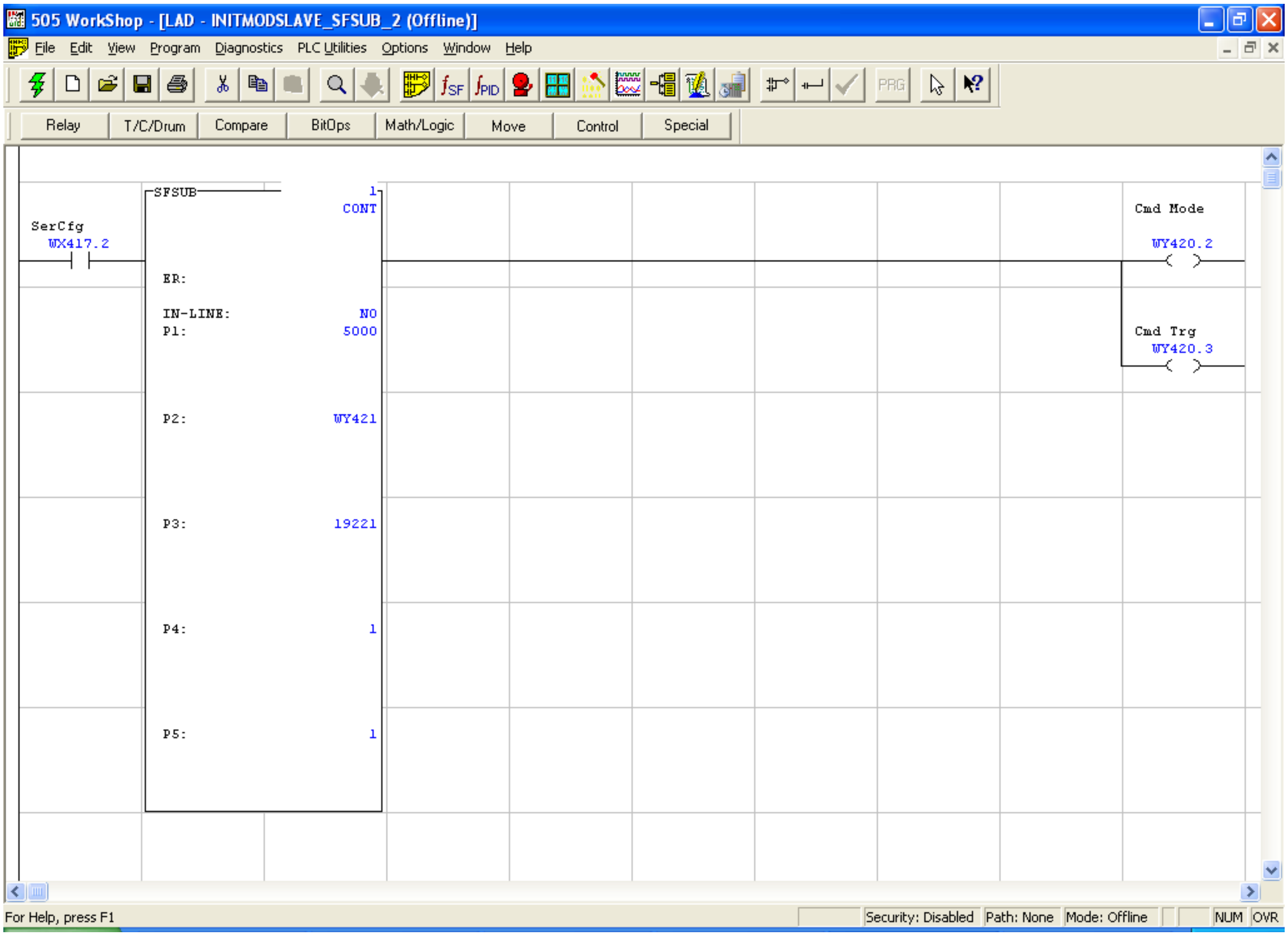
P2 = WY command slot to use for this command.

P3 = Connection number assigned to this port connection.

Valid values are 19221 to 19228. If more than one port is being started from PLC logic, a good rule of thumb is to assign 19221 to port 1, 19222 to port 2, 19223 to port 3, and 19224 to port 4. This makes it easier to keep track of which command is associated with each port.

P4 = Physical 2573-MOD port number to use for this connection 1, 2, 3, or 4).

P5 = Modbus Slave address assigned to this slave port connection (1 to 254).



505 WorkShop - [LAD - INITMODSLAVE_SFSUB_2 (Offline)]

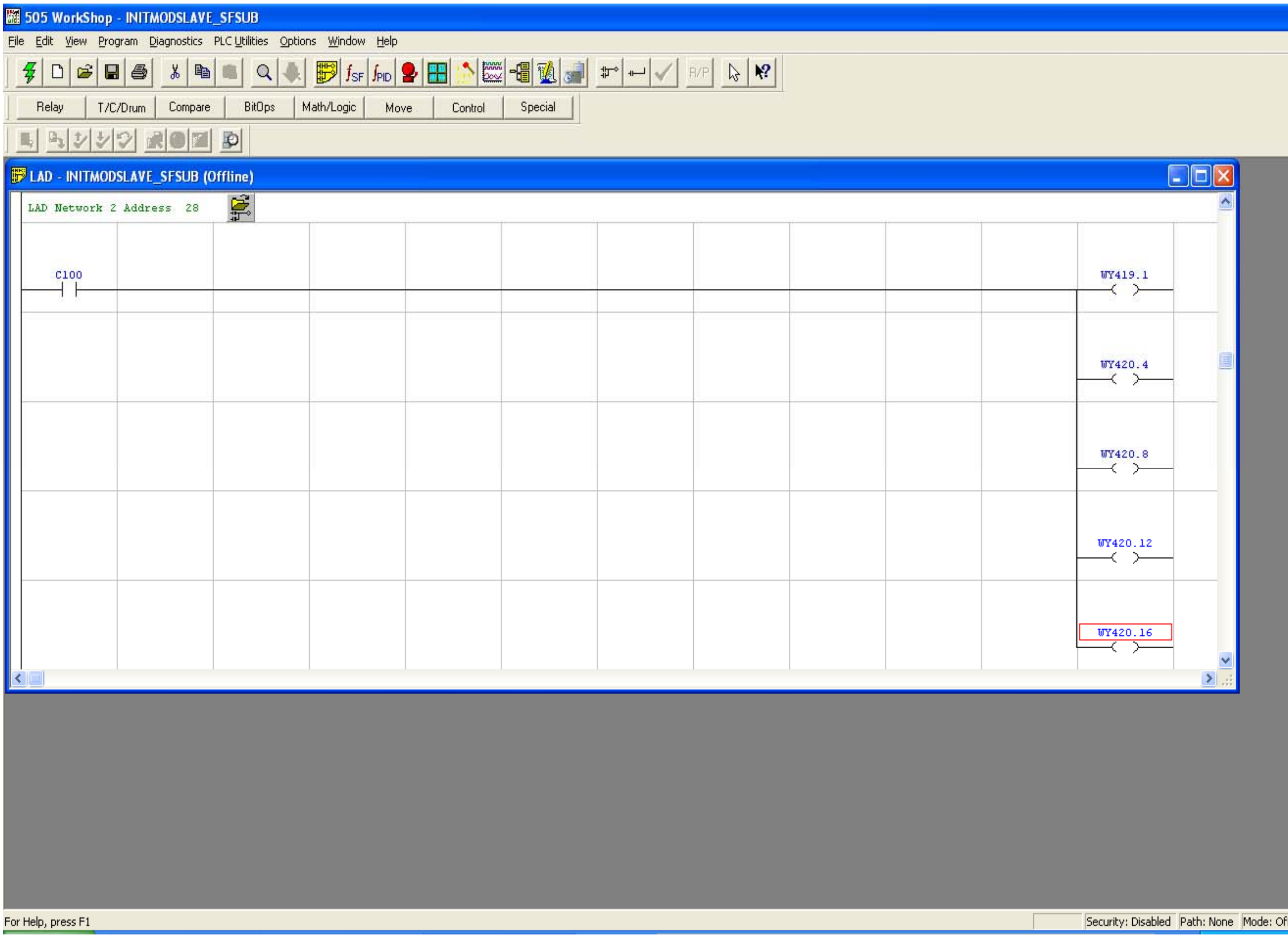
```
SerCfg        SFSUB              1          Cmd Mode
WX417.2                          CONT       WY420.2

              ER:
              IN-LINE:           NO         Cmd Trg
              P1:                5000       WY420.3
              P2:                WY421
              P3:                19221
              P4:                1
              P5:                1
```

For Help, press F1 | Security: Disabled | Path: None | Mode: Offline | NUM | OVR

RESETTING THE 2573-MOD MODULE FROM LADDER LOGIC

If changes are made to the Create Connection Command Block, the 2573-MOD must be reset before the changes will take effect. The ladder rung below causes the 2573-MOD module to go through a reset on a transition of the input from OFF to ON. As long as the input in true, the 2573-MOD module will be in a constant state of reset so the input must be transitioned from OFF to ON and then back OFF. Refer to ***Appendix D of the 2573-MOD Installation and OperationGuide*** for a complete description of this procedure.


505 WorkShop - INITMODSLAVE_SFSUB
File Edit View Program Diagnostics PLC Utilities Options Window Help
Relay T/C/Drum Compare BitOps Math/Logic Move Control Special
LAD - INITMODSLAVE_SFSUB (Offline)
For Help, press F1
Security: Disabled Path: None Mode: Of

LAD Network 2 Address 28

C100

WY419.1

WY420.4

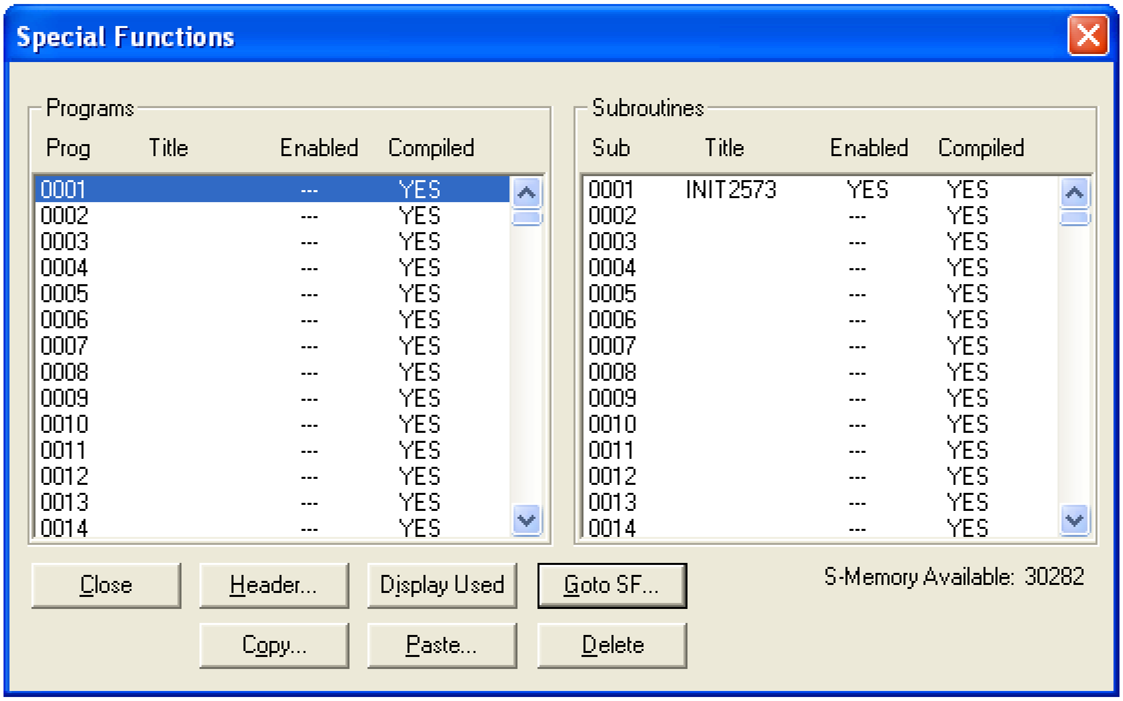
WY420.8

WY420.12

WY420.16

3.2 Special Function Subroutine Program

In this example, the Special Function Subroutine, INIT2573, is SFSUB# 1. This subroutine may be copied into any empty SFSUB slot, but note that the SFSUB number in the ladder program must be changed to match the SFSUB slot number that it is copied into.



Programs

Prog	Title	Enabled	Compiled
0001		---	YES
0002		---	YES
0003		---	YES
0004		---	YES
0005		---	YES
0006		---	YES
0007		---	YES
0008		---	YES
0009		---	YES
0010		---	YES
0011		---	YES
0012		---	YES
0013		---	YES
0014		---	YES

Subroutines

Sub	Title	Enabled	Compiled
0001	INIT2573	YES	YES
0002		---	YES
0003		---	YES
0004		---	YES
0005		---	YES
0006		---	YES
0007		---	YES
0008		---	YES
0009		---	YES
0010		---	YES
0011		---	YES
0012		---	YES
0013		---	YES
0014		---	YES

S-Memory Available: 30282

Note that the parameters are indexed off of the P1 parameter passed from the SFSUB instruction. The P1 parameter passed from the SFSUB instruction is 5000, therefore these parameters will be written into V5000 through V5015.

```
00001  *    THE FOLLOWING PARAMETERS ARE PASSED FROM THE SFSUB BOX:

            P1 = STARTING V MEMORY LOCATION WHERE COMMAND BLOCK IS TO BE

               WRITTEN

            P2 = WY COMMAND SLOT TO USE FOR THIS COMMAND

            P3 = CONNECTION NUMBER FOR THIS COMMAND

            P4 = PHYSICAL 2573-MOD PORT NUMBER ASSOCIATED WITH THIS COMMAND

            P5 = MOBUS SLAVE ADDRESS ASSIGNED TO THIS CONNECTION

00002  *

00003  *        LOAD NETWORK PARAMETERS INTO V MEMORY

00004  *        OFFSET 0 - COMMAN ERROR WORD - INITIALLY SHOULD BE SET TO 0
```

```
00005  IMATH       V1 ( P1 ) := 0

00006  *           OFFSET 1 - LOAD CODE FOR "CREATE CONNECTION"

00007  IMATH       V1 ( P1 + 1 ) := 1

00008  *           OFFSET 2 - LOAD CONNECTION NUMBER (19221 - 19224)

                   THIS PARAMETER IS PASSED FROM P3 IN THE SFSUB BOX INSTRUCTION

00009  IMATH       V1 ( P1 + 2 ) := P3

00010  *           OFFSET 3  - PROTOCOL MANAGER NUMBER FOR "MODBUS RTU SLAVE"

00011  IMATH       V1 ( P1 + 3 ) := 42

00012  *           OFFSET 4 - PHYSICAL 2573 MODULE PORT TO USE

                   THIS PARAMETER IS PASSED FROM P4 IN THE SFSUB BOX INSTRUCTION

00013  IMATH       V1 ( P1 + 4 ) := P4

00014  *           OFFSET 5 - PORT BAUD RATE

00015  IMATH       V1 ( P1 + 5 ) := 19200

00016  *           OFFSET 6 - BITS PER CHARACTER (RTU REQUIRES 8)

00017  IMATH       V1 ( P1 + 6 ) := 8

00018  *           OFFSET 7 - PARITY (0 = NONE, 1 = ODD, 2 = EVEN)

00019  IMATH       V1 ( P1 + 7 ) := 0

00020  *           OFFSET 8 - NUMBER OF STOP BITS (0 OR 1)

00021  IMATH       V1 ( P1 + 8 ) := 1

00022  *           OFFSET 9 - HANDSHAKE (0 = NONE, 4 = RS485, 6 = MODEM RTS/CTS)

00023  IMATH       V1 ( P1 + 9 ) := 0

00024  *           OFFSET 10 - STARTUP OPTION BITS (FOR MODNIM ADDRESSING

                   COMPATIBILITY SET BIT 15)

00025  IMATH       V1 ( P1 + 10 ) := 2

00026  *           OFFSET 11 - MODBUS SLAVE ADDRESS TO ASSIGN TO THIS PORT

                   THIS PARAMETER IS PASSED FROM P5 IN THE SFSUB BOX INSTRUCTION

00027  IMATH       V1 ( P1 + 11 ) := P5

00028  *           OFFSET 12 - RTU END OF MESSAGE INTERVAL IN MILLISECONDS (5 -
                    100). ONLY USED IF BIT 12 IS SET IN START OPTIONS IN OFFSET 10.
00029   IMATH       V1 ( P1 + 12 ) := 0
00030   *           OFFSET 13 - ENRON EXTENSIONS: CONTROL RELAY ADDRESS OF MODBUS
                    COIL 1000. (ONLY USED OF BIT 11 IS SET IN STARTUP OPTION WORD IN
                    OFFSET 10.)
00031   IMATH       V1 ( P1 + 13 ) := 0
00032   *         OFFSET 14 - ENRON EXTENSIONS: V MEMORY ADDRESS OF MODBUS REGISTER
                  7000 (MOST SIGNIFICANT WORD). ONLY USED IF BIT 11 OF STARTUP
                  OPTION WORD IS SET IN OFFSET 10.
00033   IMATH       V1 ( P1 + 14 ) := 0
00034   *         OFFSET 15 - ENRON EXTENSIONS: V MEMORY ADDRESS OF MODBUS REGISTER
                  7000 (LEAST SIGNIFICANT WORD.) ONLY USED IF BIT 11 OF START
                OPTION WORD IS SET IN OFFSET 10.)
00035   IMATH       V1 ( P1 + 15 ) := 0
00036   *         THIS INSTRUCTION LOADS THE V MEMORY POINTER WORD (P1)PASSED FROM
                  THE SFSUB BOX INTO THE WY COMMAND SLOT (P2)ALSO PASSED FROM THE
                SFSUB BOX.
00037   IMATH       P2 := P1


                          PROGRAM END
```

The SFSUB program will load the following parameters into the Create Modbus RTU Slave Command Block.

V Memory Location	Hex Value	Integer Value	Comments
V5000	**0000**	**00000**	Command Error Word - Initially Set to 0
V5001	**0001**	**00001**	Command Code for Create Connection
V5002	**4B15**	**19221**	Connection Number
V5003	**002A**	**00042**	Protocol Manager Number for RTU Slave
V5004	**0001**	**00001**	Selects Physical Port # 1
V5005	**4B00**	**19200**	Selects 19200 as the baud rate
V5006	**0008**	**00008**	Bits per character (RTU requires 8 bits/character)
V5007	**0000**	**00000**	Selects Parity (0=None, 1=Odd, 2=Even)
V5008	**0001**	**00001**	Stop bits (0 selects Modbus default for parity selected)
V5009	**0000**	**00000**	Handshake option (0=None, 4=RS-485, 6=RTS/CTS)
V5010	**0002**	**00002**	Option bits – 2 = MODNIM ADDRESSING COMPATIBILITY
V5011	**0001**	**00001**	Modbus Slave Address
V5012	**0000**	**00000**	RTU End of Message feature not being used - Set to 0
V5013-V5015	**0000**	**00000**	Unused - Set to 0

3.3 Troubleshooting Tips

The only indication you will have that the Modbus Slave port was created successfully is if the NetCfg bit (WX1.2), in this case, WX417.2, goes OFF. This bit will go OFF when all four ports of the 2573-MOD are configured, either by PLC start or dipswitch selection. This is why it is important that any ports not being configured by the PLC should be placed in a default configuration (all dipswitches OFF).

In the event that this bit does not go OFF, ensure that the dipswitches 6, 7, and 8 for the port you are trying to start as a Modbus Slave are set correctly (ON) for PLC Select and that dipswitches 6, 7, and 8 for all unused ports are set to OFF. If the dip switchs are set correctly, refer to the trouble shooting guide below.

3.4 STATUS WORDS

WX1 is the Module Status Word. 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 -15	16
MOD FAIL	SER CFG	N/A	DIAG ERR	CFG ERR	N/A	RSVD	RSVD	TIMER E CODE

Bit 1 – MOD FAIL – *Severe Module Failure*

This bit is set to 1 if a severe module failure is detected.

Bit 2 – SER CONFIG – *Serial Configuration Required*

This bit is reserved for future use.

Bit 3 – *Not Used by 2573-MOD*

Bit 4 – 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.

Bit 5 – 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.

Bit 6 – Not used by 2573-MOD

Bits 7 and 8 – RSVD – *Reserved*

These bits are reserved for future use.

Bits 9 - 16 – 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, and error code will be written to this field, if possible.
MODULE STATUS WORD TROUBLESHOOTING MATRIX

WX1

WX1.1 MOD FAIL	WX1.2 SER CFG	WX1.9 - WX1.16 TIMER	POSSIBLE PROBLEM
0	**1**	**Counting (0 to FF Hex)**	**Module is running and waiting for serial configuration by PLC or a command error occurred during execution of the Create Connection Command (see note 1)**
1	**0**	**Not counting or all 0**	**A module failure has occurred and will soon reset**
0	**0**	**Counting (0 to FF Hex)**	**All serial ports are configured and the module is ready to use**
0	**0**	**All 0**	**The PLC is not reading the module. Check that the module is logged into the PLC IO configuration properly.**

Note 1

If the ladder logic has run and the serial configuration bit is still ON, check the following things:

1. **Look at the Command Error Word of the Create Connection command. If there is an error code written there, look up the code in the error code listing in the *2573-MOD Installation and Operation Guide.* Generally, the problem is an incorrect data entry in the command block. If so, correct the command block and reset the 2573-MOD module.**
2. **If no error code has been written to the Command Error Word, see the Command Status Word troubleshooting matrix.**

WX2 is the Command Status Word. The module sets bits in this word to communicate command status to the PLC. The status bits in this word are directly associated with the corresponding command control bits in the Command Control Word (WY4) and the Command Slots (WY5 – WY8). The bit structures for these words are as follows:

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
Command 1				**Command 2**				**Command 3**				**Command 4**			

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 - WY5				**Command 2 - WY6**				**Command 3 -WY7**				**Command 4 - WY8**			

These words are divided into 4 sets of 4 bits each. Bits **1 to 4** of the Command Status Word (WX2) reflect command status of command slot 1, **5 to 8** – command slot 2, **9 to 12** – command slot 3, and **13 to 16** – command slot 4. Likewise, bits **1 to 4** of the Command Control Word are associated with command slot 1 (WY5), **5 to 8** – command slot 2 (WY6), **9 to 12** –command slot 3 (WY7), and **13 to 16** – command slot 4 (WY8).

COMMAND STATUS BIT DEFINITIONS (WX2)

CMD Err – *Command Error*

The 2572-A sets this bit to 1 when it encounters an error while attempting to execute the command specified in its corresponding command slot. The 2573-MOD will write an error code into offset 0 of the specified command block unless the PLC Err bit is also set.

PLC Err – *PLC READ/WRITE ERROR*

The 2573-MOD sets this bit to 1 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 into the command slot which is not a valid V memory location (such as 0) or which points to a V location that does not contain a command block.

CMD Busy – *Command Busy*

The 2573-MOD sets this bit 1 while it is processing a command. The module then sets this bit to 0 when command processing is finished.

ABORT Busy

This indicates that the module is attempting to abort a command after an “abort trigger” has been received in WY4.

COMMAND CONTROL BIT DEFINITIONS (WY4)

ERR Ack – *Error Acknowledge*

The PLC sets this bit to acknowledge the error bit which was set by the 2573-MOD.

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-MOD command processing is not synchronized with the PLC logic.
1 = Coupled mode – The 2573-MOD 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. In most applications, use of this bit is not required.

3.5 COMMAND STATUS WORD TROUBLESHOOTING MATRIX

Only the status bits for command slot 1 (bits 1 – 4) are shown here since they operate the same for all command slots. Monitoring command slots 2, 3, and 4 would be done in the same way.

WX2

WX2.1 CMD Err	WX2.2 PLC Err	WX2.3 CMD Busy	WX2.4 Abort Busy	POSSIBLE PROBLEM
0	0	0	0	No command is in process. This could mean that a command has been triggered and successfully completed, or that no command has been triggered. Check ladder logic to ensure that the correct status bits are being monitored and the correct control bits are being exercised.
0	0	1	0	A command is currently in process. If uncoupled triggering mode is used, this bit will remain ON as long as the associated trigger bit in WY4 is ON. If coupled mode triggering is used, this bit will toggle ON when the command is triggered from plc logic and toggle OFF when the command has finished processing.
1	0	1	0	A command error has occurred. Examine the command error word of the command being processed for an error code. Find the listing for the error code in the IO Guide for possible causes. Correct the problem and then reset the 2573-MOD module.
1	1	1	0	The module could not read a valid command block from V memory. Ensure that the correct pointer value is being loaded into WY5 and that the command block being pointed to contain the correct entries. Correct and then reset the 2573-MOD module.

If you have been through all the trouble shooting steps and still cannot communicate with the Modbus Master, here are some more things to check.

Monitor the Receive LED of the 2573-MOD port that you are connected to. This LED should blink when it is being polled by the Master. If the LED does not blink, check the following:

1. Ensure that the Master is actually polling. The 2573-MOD Receive LED will only blink when it receiving data on the port.
2. If you confirm that the Master is actually polling, check your cable connections carefully.
3. Ensure that the RS-422/485 disconnect switch is in the correct position for the communication you are using.

If the Receive LED is blinking when polled by the Master, check the following:

1. Ensure that the 2573-MOD port communication parameters (baud rate, bits per character, stop bits, etc.) match the communication parameters of the Master.
2. Ensure that the Modbus Slave Address configured in the 2573-MOD slave port is the correct Slave address that the Master is polling.

If you have been through all the above trouble shooting steps and still cannot communicate with the Master, there is a good software tool available from Win-Tech Software. This is a Modbus Master that is very useful in validating that the 2573-MOD Slave port is correctly configured. This software is called MODSCAN. An evaluation copy of this software may be downloaded from the Win-Tech website at the link below.http://www.win-tech.com/html/modbus1.htm.

Appendix A – "Section 1.2. Serial Ports" from the 2573-MOD IOG

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* 1.11.1 Setting the RS-422/485 Disconnect *Switches on page 13.*

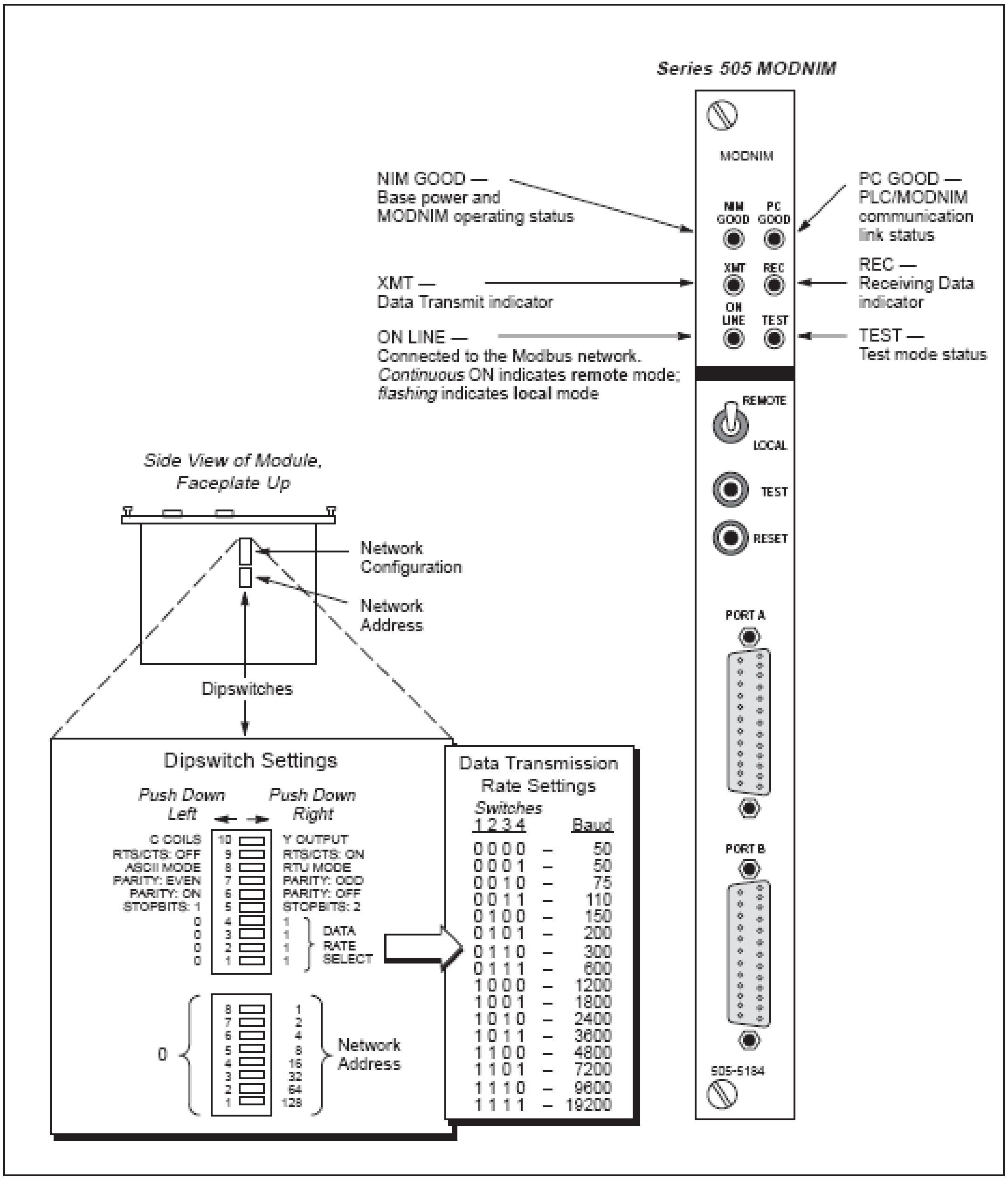
The RS-422 interface is typically used to improve noise immunity and/or to extend the length of cable that can be reliably used. RS-422 uses two sets of twisted pair wires, one for transmit and one for receive. Within each set of twisted pair, one wire carries the signal in positive polarity and the 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.

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 **Error! Reference source not found.**, page 11. Otherwise the baud rate and other communication parameters are set by PLC logic.

Appendix B MODNIM Switches and LEDs Diagram


Series 505 MODNIM
MODNIM
NIM GOOD
PC GOOD
XMT
REC
ON LINE
TEST
NIM GOOD — Base power and MODNIM operating status
PC GOOD — PLC/MODNIM communication link status
XMT — Data Transmit indicator
REC — Receiving Data indicator
ON LINE — Connected to the Modbus network. *Continuous* ON indicates **remote** mode; *flashing* indicates **local** mode
TEST — Test mode status
REMOTE
LOCAL
TEST
RESET
PORT A
PORT B
505-5184
Side View of Module, Faceplate Up
Network Configuration
Network Address
Dipswitches

Dipswitch Settings

Push Down Left ←		→ *Push Down Right*	
C COILS	10	Y OUTPUT	
RTS/CTS: OFF	9	RTS/CTS: ON	
ASCII MODE	8	RTU MODE	
PARITY: EVEN	7	PARITY: ODD	
PARITY: ON	6	PARITY: OFF	
STOPBITS: 1	5	STOPBITS: 2	
0	4	1	DATA RATE SELECT
0	3	1	
0	2	1	
0	1	1	

0	8	1	Network Address
	7	2	
	6	4	
	5	8	
	4	16	
	3	32	
	2	64	
	1	128	

Data Transmission Rate Settings

Switches 1 2 3 4		Baud
0 0 0 0	–	50
0 0 0 1	–	50
0 0 1 0	–	75
0 0 1 1	–	110
0 1 0 0	–	150
0 1 0 1	–	200
0 1 1 0	–	300
0 1 1 1	–	600
1 0 0 0	–	1200
1 0 0 1	–	1800
1 0 1 0	–	2400
1 0 1 1	–	3600
1 1 0 0	–	4800
1 1 0 1	–	7200
1 1 1 0	–	9600
1 1 1 1	–	19200

Figure 1-5 Series 505 MODNIM Switches and LEDs

Appendix C – Communications Port Loop Back Connector Construction

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.

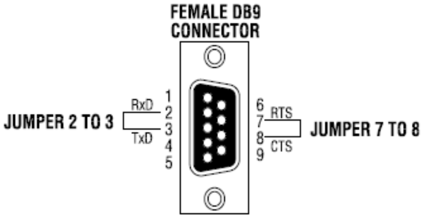


Figure 20. RS-232 Loopback Connector

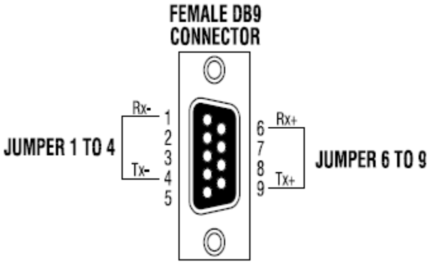


Figure 21. RS-422 Loopback Connector

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