Application User Manual

Distributed PID Temperature Control From a PLC®

USING

Eurotherm Series 2000™
PID Temperature Controllers
with Modbus® RTU Communications

TO

Control Technology Inc.
2573-MOD
Serial Interface Adapter Module

Fitted In Simatic® 505

Family of Programmable Controllers

Prepared By:
Control Technology Inc.
5734 Middlebrook Pike
Knoxville, TN 37921

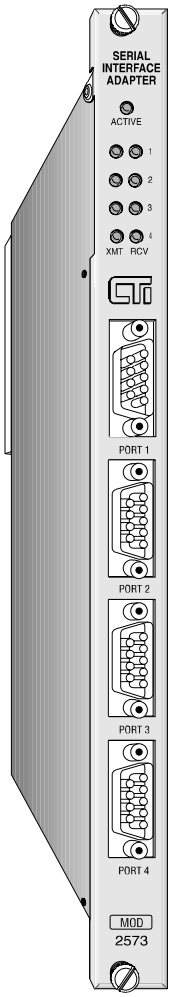
January 1998

Revision 1.0

This application note describes the proven steps required to implement a Modbus RTU communications interface between a CTI 2573-MOD module acting as communications *Master* in a Simatic PLC chassis and 1 or more Eurotherm Series 2000 temperature controllers acting as communication *Slaves*.

With this type of communications interface, all Modbus addressable parameters in a Eurotherm PID temperature controller are accessible as read or write parameters in the PLC's V memory. This includes but is not limited to, measured value, setpoint, output power, PID and cutback, alarm and setpoint limits.

The Eurotherm 2000 series controllers that support Modbus RTU communications are the 2216, 2208, 2204, 2416, 2408, and 2404. Any other Eurotherm PID temperature controller that supports Modbus RTU communications may be used with a CTI 2573-MOD although this application note focuses on the Eurotherm 2000 series.



Control Technology Inc. offers the 2573-MOD for simple integration into the Siemens® SIMATIC®505 PLC chassis.

Both the Eurotherm PID temperature controllers and CTI 2573-MOD modules support Modbus RTU communications through standard 2-wire RS-485 serial ports for a *direct* connection. No other hardware - serial converter or gateway - is necessary.

Maximum baud rate is 19K2. The CTI 2573-MOD supports 2 Modbus Master ports (ports 1 and 3) for a maximum of 64 multi-dropped Eurotherm temperature controllers (32 per port) per 2573-MOD module. Multiple 2573-MOD modules may be fitted per Siemens® SIMATIC® 505 PLC chassis.

Left: CTI 2573-MOD

The purpose in connecting the Eurotherm temperature controllers to a SIMATIC® 505 PLC is multi-fold:

1) To decrease installation, configuration, and debug time of PID loops.

2) To offload the PID loops from the PLC's CPU enabling utilization of a less expensive processor, faster or more deterministic scan times.

3) Provide advanced Eurotherm 4th generation PID and Auto-Tune algorithms to a PLC system. This permits Auto-Tuning PID, Adaptive Tuning PID, copyrighted overshoot inhibition, and rapid independent deterministic loop execution times.

4) Provide distributed specialized PID loop application capability at considerably less expense. For example: heat-cool, cascade, ratio, pressure, carbon control, and pH control.

5) Provide built-in process diagnostic capability in the form of temperature and load alarms.

6) Provide remote, self contained PID temperature controller modules that have their own local display and power supply. These modules remain operational even should the PLC fail.

7) The Eurotherm PID temperature controller can monitor the health of the heater, heater amperage, heater fuse status, SCR health, and wiring health via patent pending PDSIO method. This diagnostic information is transmitted to the PLC for monitoring and/or action.

8) The Eurotherm PID temperature controller supports direct thermocouple inputs, linear inputs, high impedance infrared sensor inputs, or custom linearization analog inputs negating the requirement for extra programming at the PLC.

Figure 1. below shows a typical multi-dropped configuration utilizing a SIMATIC® 505 PLC, the CTI 2573-MOD, and Eurotherm distributed PID temperature controllers. *Serial* communications is Modbus RTU in a Master-Slave (query-response) relationship to continuously poll temperature controller data. Modbus function codes supported by the controllers are:

- *Read: 1, 2, 3, 4*
- *Write: 5, 6, 15, 16*

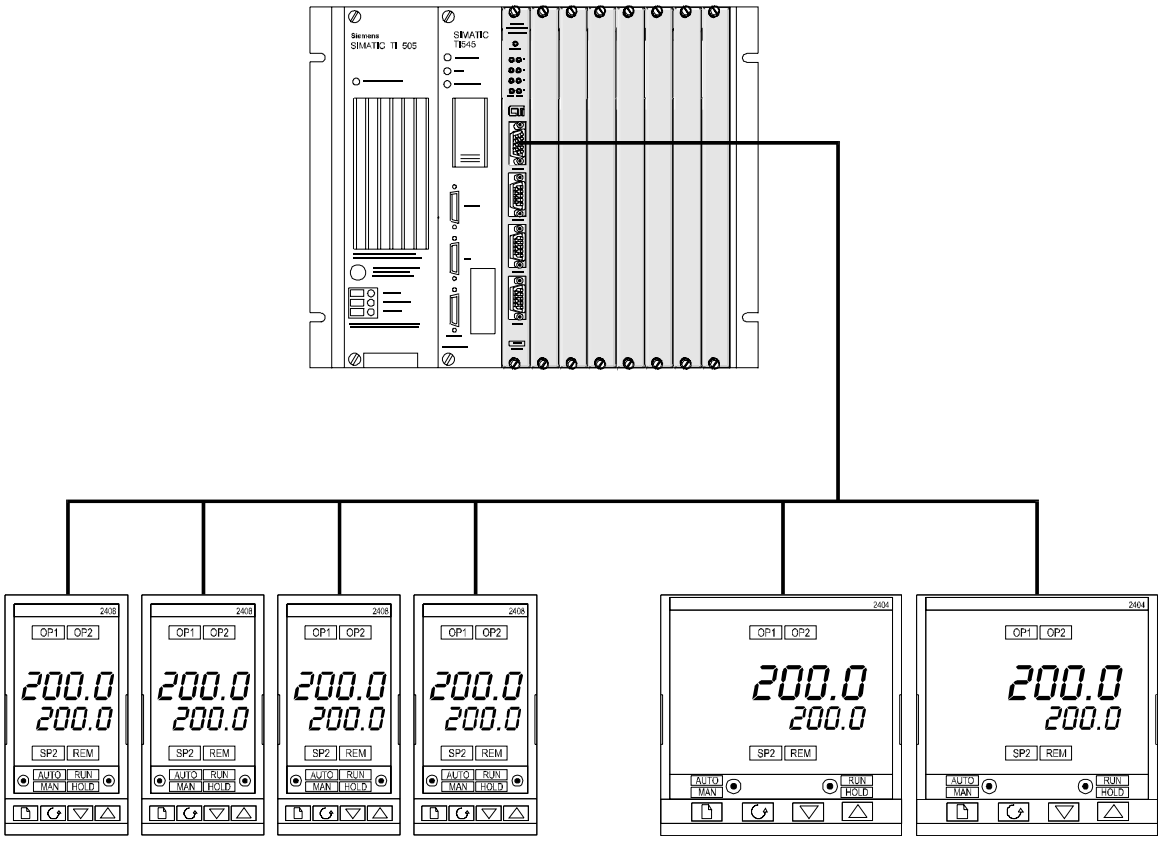


Figure 1.
PLC with Distributed PID Controllers

General 2573-MOD Module Operation

It is useful to understand the basic means and mapping of data as it is transferred from the controller back and forth to the PLC through the CTI 2573-MOD module before starting an application design.

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 the ***CTI 2573-MOD Serial Device Interface Adapter Installation and Operation Guide, Chapter 9. Troubleshooting*** for more information.

PLC Command Interface

The 2573-MOD provides a standard PLC logic interface for sending messages and processing 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. These structures are summarized below. Please refer to the ***CTI 2573-MOD Serial Device Interface Adapter Installation and Operation Guide, Appendix D. PLC Command Interface*** for a more complete description of the command block, **WX/WY** usage, and command timing diagrams.

2573-MOD WX and WY Words

Word	Function	Description
WX1	Module Status	The high 8 bits contain module status. The lower 8 bits are a counter that increments approximately once per second.
WX2	Command Status	The 2573-MOD will set these bits to indicate the status of command processing. There are 4 sets of 4 bits. The 4 sets correspond to the 4 command slots and to the 4 sets of command control bits in WY4.
WY3	Module Control	Contains bits which your PLC logic can use to control general module functions, e.g., reset the module.
WY4	Command Control	Your PLC logic will set these bits to control command processing by the 2573-MOD. There are 4 sets of 4 bits. The 4 sets correspond to the 4 command slots and to the 4 sets of Command Status bits in WX2.
WY5	Command Slot 1	Contains the V memory address of the Command Block used with the first set of command and status bits.
WY6	Command Slot 2	Contains the V memory address of the Command Block used with the second set of command and status bits.
WY7	Command Slot 3	Contains the V memory address of the Command Block used with the third set of command and status bits.
WY8	Command Slot 4	Contains the V memory address of the Command Block

used with the fourth set of command and status bits.

Command Block

A Command Block is a contiguous group of V memory words used to store values which specify which module command will be executed and how the command will be processed. The exact content of the Command Block will vary with the command being issued. The Command Block is typically created in programming software such as TISOFT, and stored permanently in the PLC V memory.

Some of the Command Blocks allow you to set a timeout value for executing the commands. The timeout represents the number of seconds that can elapse from starting the command until the command is completed. If the command has not been completed within the elapsed time, the 2573-MOD will suspend command processing and will set the CMD ERR bit in WX2. The protocol managers used with the 2573-MOD have a default timeout of approximately 9 seconds. For most applications, you should use the default value.

All Command Blocks must contain a hex 4B in the high byte of offset 2 of the command block. This signature byte is used to ensure that the referenced V memory is actually a Command Block. This is why you will use Connection Numbers such as 19221 decimal, which is 4B15 in hexadecimal notation.

Later in this application note, we will show how to build a **Create Connection** command block to start Modbus RTU Master protocol on Port #1, an **Initiate Poll List** command block to control the polling of multi-dropped devices, and **Query** command blocks which will contain the information needed to query individual devices on the network, such as; slave address, Modbus function code, starting Modbus address, and number of points to read or write.

Appendix C of this document illustrates how the module WX/WY words and the command blocks are used together. The 2573-MOD controls the value in the WX words. WX1 contains bits which indicate module status. WX2 contains bits which indicate the status of command execution. For example, WX2 bits indicate the module is busy executing a command or that a command error has occurred.

To use the module command interface, your PLC logic typically loads a Command Slot with the address of the desired command block. It then sets a corresponding trigger bit in WY4 to cause the 2573-MOD to execute the command. Your logic then monitors the condition of the command status bits to determine whether the command has completed successfully. You should refer to **Appendix D** of the **2573-MOD Serial Device Interface Adapter Installation and Operation Guide** for additional information; also see ***Appendix B*** of this document for a ladder logic example.

Command blocks may start at any V memory location, as specified by the user, where there is sufficient free memory that is not used by any other application. The command blocks must use contiguous V memory locations. In the following Command Block Tables, Offset 0 refers to the starting V memory location of the Command Block. Offset 1 refers to starting address +1, Offset 2, starting address +2, etc. For example, if a Command Block starts at V100 then Offset 0 = V100, Offset 1 = V101, Offset 2 = V102, etc.

2573-MOD Modbus RTU Master Operation

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.

The command flow for a Modbus Query is as follows:

1. PLC logic triggers a command sequence in the 2573-MOD
2. The 2573-MOD reads the specified command block from the PLC V memory
3. The 2573-MOD builds the Modbus Query and sends it out the serial port
4. The 2573-MOD receives and interprets the response from the slave device
5. The 2573-MOD writes the response data to the specified V memory location

Create Connection Command

Before the 2573-MOD 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.

NOTE:

The Eurotherm temperature controllers support only the Modbus RTU protocol.

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*** of the ***2573-MOD Serial Device Interface Adapter Installation and Operation Guide*** for information regarding switch settings. The following command block is used to create a connection for the Modbus RTU Master.

V Memory Offset	Description	Hex Value	Integer Value
0	Command Error Word	0000	00000
1	Command (Create Connection)	0001	00001
2	Connection Number (19221 - 19299)		
3	Protocol Manager Number (Modbus RTU Master)	0028	00040
4	Physical Port Number (1 or 3)		
5	Port Baud Rate (300, 600, 1200, 2400, 4800, 9600, 19200)		
6	Reserved (Unused)	0000	00000
7	Parity (0 = None, 1 = Odd, 2 = Even)		
8	Number of Stop Bits (0 = default, 1 = 1 stop bit)	0000	00000
9	Handshake (0 = None, 4 = RS-485)		
10	Option Bits (see definitions)		
11	V Memory Address of Modem Configuration Table		
12	RTU End-of-message Interval - in Milliseconds (5 - 100)	0000	00000
13 - 15	Unused - reserved for future use (Set to 0)	0000	00000

Definitions

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 of 19221 to 19299 may be assigned as long as it 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 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.

Offset 8 *Stop Bits* - Use 0 for the Modbus default. This will automatically set the number stop bits according to the parity setting.

Offset 9 *Handshake* - If you are using RS-485 (two wire) electrical interface, set this value to 4. 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 - 11	**Bit 12**	**Bit 13**	**Bit 14**	**Bit 15**	**Bit 16**

Bit Description	**Bit(s)**	**Definitions**
Unused - Reserved They	1 - 11	These are the most significant bits in the word. are unused and are reserved for future use. Set to 0.
End of Message	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.
Unused - Reserved	13 - 15	These bits are reserved and should be set to 0.
Modem Control Feature	16	This bit determines whether the Modem Control Feature is used. 0 = Modem Control Feature not used. 1 = Modem Control Feature is used.

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

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.

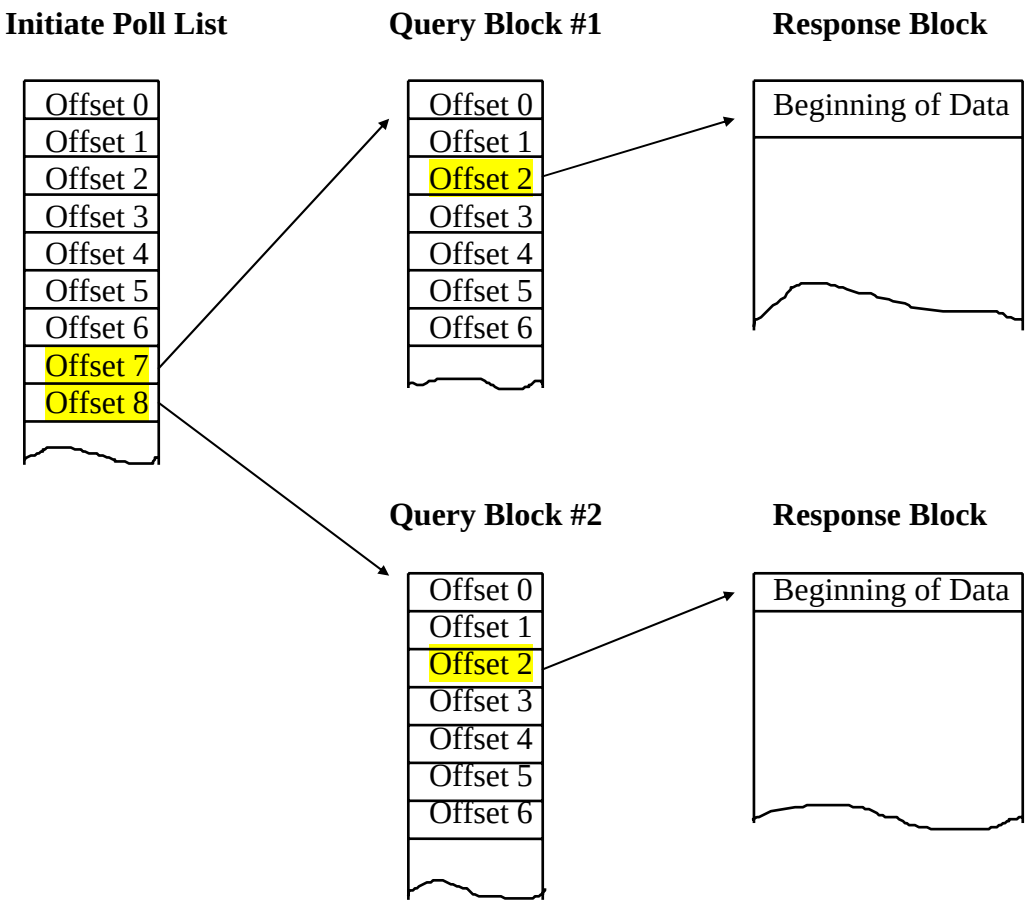
2573-MOD Initiate Poll List Command

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. The figure below shows how two queries are linked to the command. This command is *not* a fixed 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 completed 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.

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 an 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 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 completed 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 communication 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.

V Memory Offset	Description	Hex Value	Integer Value
0	Command Error Word	0000	00000
1	Command (2804 hex = Initiate RTU Poll List 2904 hex = Initiate ASCII Poll List)		
2	Connection Number (19221 - 19299)		
3	Command Flags - conditionally stop poll list on error (See definition 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	00000
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 Address of additional Query Blocks		

Definitions

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:
2804 hex (10244 decimal); 2904 hex (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 Time-out 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 Block 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 page 11 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 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.

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 in 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 the Initiate Poll List command is executed.

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:

V Memory Offset	Description	Hex Value	Integer Value
0	Error Word	**0000**	**00000**
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	**5142**	**20802**

Definitions

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 Time-out Flag	14	Determines whether to stop Poll List execution if any Command Timeout is encountered during 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.
Query 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 a 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 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 (01 - F7 hex). 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.

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 (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 High 00
Starting Address Low 6B
No. of Registers High 00
No. of Registers Low 02

Then V memory should contain:

Offset	Hex Value	Integer Value
7	006B	00107
8	0002	00002

For most commands, the above technique will align word oriented data with V memory boundaries. However, two commonly used Modbus function (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
Data 1 Hi 00
Data 1 Lo 0A
Data 2 Hi 01
Data 2 Lo 02

Then V memory should contain:

Offset	Hex Value	Integer Value	Comments
7	0001	00001	Address Hi/Lo
8	0002	00002	No. of Registers Hi/Lo
9	0004	00004	Byte Count
10	000A	00010	Data 1 Hi/Lo
11	0102	00258	Data 2 Hi/Lo

Modbus 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 2 of the Query Block. Offset 0 of the Response Block 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 of the Response Block 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.

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

V Memory Offset	Contents
0	Function Code from Response Message
1	Byte Count from Response Message or 0 if function does not return a byte count
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
Data 1 High 02
Data 1 Low 2B
Data 2 High 00
Data 2 Low 64

would be stored in V memory as:

V Memory Offset	Hex Value	Integer Value	Comments
0	0003	00003	Function Code from Response
1	0004	00004	Byte Count
2	022B	00555	Data 1
3	0064	00100	Data 2

The response to a function code 16 (hex 10) which contained the following data:

Starting Address High 00
Starting Address Low 01
No. Registers High 00
No. Registers Low 02

would be stored in V memory as:

V Memory Offset	Hex Value	Integer Value	Comments
0	0010	00016	Function Code from Response
1	0000	00000	Byte Count (0 = not used)
2	0001	00001	Starting Address
3	0002	00002	No. of Registers

1. Determine Controller Parameters

As the purpose of the CTI 2573-MOD is to transfer data from 1 or more Modbus compatible external devices - here Eurotherm Series 2000 PID temperature controllers - the first step is to determine what parameters are required to be accessed.

Appendix A is a list of the 2000 series Operating mode parameters with their corresponding Modbus addresses. While there may seem to be an overwhelming number of parameters, most communication interfaces are primarily concerned with a small number of parameters, much like the parameters normally displayed on the front of the controller.

Furthermore, this interface is not designed to be a configuration interface between a PLC and the temperature controller, but rather an interface where the CTI 2573-MOD continuously polls primary temperature controller parameters as rapidly as possible and infrequently writes parameters to the temperature controllers.

Table 2 below lists typical primary PID temperature controller parameters. They provide an indication of the temperature and setpoint, the controller output to the load, and an indication of the mode and alarm conditions as mapped into the Fast Status byte.

Parameter	Modbus Address	Type
Process Value	1	Read
Target Setpoint	2	Read / Write
Output Power	3	Read
Fast Status Byte	74	Read

Table 2. Primary Controller Parameters

For this application note, the parameters listed in Table 2 for (4) 2208 temperature controllers will be used. The Modbus Slave Addresses of the 4 Eurotherm temperature controllers are assumed to be 1, 2, 3, and 4.

Additional attributes to point out are:

- For the 4 controllers, the first 3 parameters have contiguous addresses. This attribute enables those parameters to be retrieved with a single read request. The 4th parameter (Fast Status byte) requires a second read request.

  The 2200 series supports 32 integer (64 bytes) maximum block transfers while the 2400 series supports up to 125 integer (250 bytes) maximum block transfers.

- The parameter values are all signed 16 bit integer (+/-32768) values with implicit decimal point positions. Negative numbers are in two's complement form.

- Not all of the parameters listed in Appendix A are available in all controllers and depends on the model type and options ordered.

  For example, the Run List parameters would not be valid for a controller not having program capability.

- The number of parameters does not necessarily need to be the same from controller to controller nor do the controllers have to be the same model number.

- It is possible to read float values from the temperature controllers. The floats are stored in standard IEEE 4 byte format. They must be read as 2 word integers. These values may be stored in the SIMATIC® 505 CPU in adjacent V memory locations and read in *Real Number*
  format.

Now that we have determined the parameters we wish to read and write, we can move onto the next step - implementing the application.

2. Plan PLC V Memory Usage

At this point, we know what is required to implement a Modbus RTU Master application using the 2573-MOD. Now we need to decide how many devices you will have on the network, what parameters you will be reading/writing in the temperature controllers, and the total number of queries we will have to build in V memory. We should calculate the total amount of V memory required to store all the command blocks and responses from the slave devices and choose an area of V memory which is unused and large enough to accommodate the application.

For this application example, we will assume we have 4 Eurotherm Temperature Controllers on the network and will be reading the parameters listed in **Table 2.** The first 3 parameters are contiguous and will require 1 query per controller. The fourth parameter will require 1 query per controller and 1 query per controller will be required to write to the **Target Setpoint** parameter. This is a total of 12 query blocks and response blocks that will have to be built and stored in V memory along with 1 Create Connection command block and 1 Initiate Poll List command block. It would be good to have at least 1000 words of V memory reserved in the PLC for this application.

In this example, we are using V memory in the range of V100 to V700. Command Blocks and Query Blocks can be placed anywhere in the PLC's V memory range and don't necessarily have to be in contiguous areas of V memory, e.g., a query block could start at V128 and the response for the query could begin at V1462, although the query **blocks** themselves must be contiguous V memory locations.

Following is a list of the Command Blocks and Query blocks we will need to build:

Create Connection Command Block - To start the Modbus RTU Master Protocol on Port #1 and set up communication parameters.

Initiate Poll List Command Block - To provide Poll List control parameters and list of Queries.

Query Block #1 - Query to **read** PV, TS, and OP parameters from Slave Address 1.
Query Block #2 - Query to **read** PV, TS, and OP parameters from Slave Address 2.
Query Block #3 - Query to **read** PV, TS, and OP parameters from Slave Address 3.
Query Block #4 - Query to **read** PV, TS, and OP parameters from Slave Address 4.

Query Block #5 - Query to **read** FS parameter from Slave Address 1.
Query Block #6 - Query to **read** FS parameter from Slave Address 2.
Query Block #7 - Query to **read** FS parameter from Slave Address 3.
Query Block #8 - Query to **read** FS parameter from Slave Address 4.

Query Block #9 - Query to **write** TS parameter to Slave Address 1.
Query Block #10 - Query to **write** TS parameter to Slave Address 2.
Query Block #11 - Query to **write** TS parameter to Slave Address 3.
Query Block #12 - Query to **write** TS parameter to Slave Address 4.

3. Build Command and Query Blocks

Next we will build the Command blocks and Query blocks using the command block structure described previously. Values are provided in Hexadecimal and Decimal Integer. Either format may be used to enter data into PLC V memory.

Create Connection Command Block

V Memory Location	Hex Value	Integer Value	Comments
V100	**0000**	**00000**	Command Error Word - Initially Set to 0
V101	**0001**	**00001**	Command Code for Create Connection
V102	**4B15**	**19221**	Connection Number
V103	**0028**	**00040**	Protocol Manager Number for RTU Master
V104	**0001**	**00001**	Selects Physical Port #1
V105	**4B00**	**19200**	Selects 19K2 baud rate
V106	**0000**	**00000**	Reserved - Set to 0
V107	**0002**	**00002**	Selects Even Parity
V108	**0000**	**00000**	Use default settings for stop bits
V109	**0004**	**00004**	Select RS-485 communication
V110	**0000**	**00000**	Option bits - None used
V111	**0000**	**00000**	Modem not used - no table to reference
V112	**0000**	**00000**	RTU End of Message feature not being used - Set to 0
V113-115	**0000**	**00000**	Unused - Set to 0

Initiate Poll List Command Block

V Memory Location	Hex Value	Integer Value	Comments
V120	**0000**	**00000**	Command Error Word - Initially Set to 0
V121	**2804**	**10244**	Command Code - Initiate RTU Poll List
V122	**4B15**	**19221**	Connection Number - Matches Create Connection number
V123	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to *globally* stop the Poll List on any error
V124	**0002**	**00002**	Specifies a query timeout of 2 seconds
V125	**0000**	**00000**	Query Block with Error - Initially set to 0
V126	**000C**	**00012**	Specifies a polling list length of 12
V127	**00C8**	**00200**	V Memory Address of Query #1
V128	**00F0**	**00240**	V Memory Address of Query #2
V129	**0118**	**00280**	V Memory Address of Query #3
V130	**0140**	**00320**	V Memory Address of Query #4
V131	**0168**	**00360**	V Memory Address of Query #5
V132	**0190**	**00400**	V Memory Address of Query #6
V133	**01B8**	**00440**	V Memory Address of Query #7
V134	**01E0**	**00480**	V Memory Address of Query #8
V135	**0208**	**00520**	V Memory Address of Query #9
V136	**0230**	**00560**	V Memory Address of Query #10
V137	**0258**	**00600**	V Memory Address of Query #11
V138	**0280**	**00640**	V Memory Address of Query #12

NOTE:

The Queries are arranged in the Poll List so that the 3 parameter reads are executed to addresses 1 - 4 consecutively, then the 1 word read, and finally the writes. These queries could be executed in any order, although it is a good practice when polling a number of devices on a network, to try not to poll the same device twice in a row. When a response is received from a query, the next query in the poll list is executed immediately. If two queries to the same device address are adjacent in the poll list, the second query may be issued before the device is ready to accept another query and may result in occasional timeouts which can slow the data throughput of the poll list.

The order of Query execution may be changed by rearranging the order of the V memory addresses of the Query Blocks in the poll list. For example, if you wanted to do the writes first, simply move the Function 16 query blocks to the beginning of the poll list.

Query Block #1

This Query Block reads PV, TS, and OP parameters from Slave Address 1.

V Memory Location	Hex Value	Integer Value	Comments
V200	**0000**	**00000**	Command Error Word - Initially Set to 0
V201	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V202	**00DC**	**00220**	V Memory Address where the response from this Query is to be written
V203	**000A**	**00010**	Reserve 10 *bytes* of V Memory (5 words) for response
V204	**0001**	**00001**	Modbus Slave Address of Controller #1
V205	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V206	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V207	**0001**	**00001**	Starting Modbus Register Address in Slave
V208	**0003**	**00003**	Number of Registers to Read
V209	**5142**	**20802**	Query Block Validation Code

Response Block for Query #1

When the Query is completed successfully, the response from Query #1 will be written to the V memory response block in the following format:

V Memory Location	Contents
V220	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V221	Response Byte Count - the value returned here would be **6** because we are reading 3 words (1 word = 2 bytes)
V222	**PV** Value read from Controller #1 *or* a Modbus Exception Error Code if V220 = **0083** hex
V223	**TS** Value read from Controller #1
V224	**OP** Value read from Controller #1

Reading Parameters as Floating Point Variables

Some parameters, such as the Process Variable and Target Setpoint, may be read as floating point numbers. The *float address* for these variables is 2 times the standard Modbus address plus 8000h. So the **PV** float address is (2 X 1) + 8000h = 8002h = 32770 decimal and the **TS** float address is (2 X 2) + 8000h = 8004h = 32772 decimal. If we wanted to read **PV** and **TS** as floats using the Query Block on preceding page we would change the following values in the Query Block:

V203 =	**00012** (Integer)	Number of bytes for response
V207 =	**32770** (Integer)	Starting Register Address
V208 =	**00004** (Integer)	Number of Registers to Read

All other values would remain the same.

The Response Block would change as follows:

V221 =	**00008** (Integer)	Response Byte Count (Now reading 4 registers)
V222 =	**PV Data**	First 16 bits of Float Value for **PV**
V223 =	**PV Data**	Second 16 bits of Float Value for **PV**
V224 =	**TS Data**	First 16 bits of Float Value for **TS**
V225 =	**TS Data**	Second 16 bits of Float Value for **TS**

All other values would remain the same.

NOTE:

The PV floating point value can be read directly in the SIMATIC® 505 PLC by reading V222 as a Real Number and likewise the TS floating point value can be read at V224 as a Real Number.

Example:	**V222**	**REAL**	**+0.00000**	**PV Float Value**
	V224	**REAL**	**+0.00000**	**TS Float Value**

Query Block #2

This Query Block reads PV, TS, and OP parameters from Slave Address 2.

V Memory Location	Hex Value	Integer Value	Comments
V240	**0000**	**00000**	Command Error Word - Initially Set to 0
V241	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V242	**0104**	**00260**	V Memory Address where the response from this Query is to be written
V243	**000A**	**00010**	Reserve 10 *bytes* of V Memory (5 words) for response
V244	**0002**	**00002**	Modbus Slave Address of Controller #2
V245	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V246	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V247	**0001**	**00001**	Starting Modbus Register Address in Slave
V248	**0003**	**00003**	Number of Registers to Read
V249	**5142**	**20802**	Query Block Validation Code

Response Block for Query #2

When the Query is completed successfully, the response from Query #2 will be written to the V memory response block in the following format:

V Memory Location	Contents
V260	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V261	Response Byte Count - the value returned here would be **6** because we are reading 3 words (1 word = 2 bytes)
V262	**PV** Value read from Controller #2 *or* a Modbus Exception Error Code if V220 = **0083** hex
V263	**TS** Value read from Controller #2
V264	**OP** Value read from Controller #2

Query Block #3

This Query Block reads PV, TS, and OP parameters from Slave Address 3.

V Memory Location	Hex Value	Integer Value	Comments
V280	**0000**	**00000**	Command Error Word - Initially Set to 0
V281	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V282	**012C**	**00300**	V Memory Address where the response from this Query is to be written
V283	**000A**	**00010**	Reserve 10 *bytes* of V Memory (5 words) for response
V284	**0003**	**00003**	Modbus Slave Address of Controller #3
V285	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V286	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V287	**0001**	**00001**	Starting Modbus Register Address in Slave
V288	**0003**	**00003**	Number of Registers to Read
V289	**5142**	**20802**	Query Block Validation Code

Response Block for Query #3

When the Query is completed successfully, the response from Query #3 will be written to the V memory response block in the following format:

V Memory Location	Contents
V300	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V301	Response Byte Count - the value returned here would be **6** because we are reading 3 words (1 word = 2 bytes)
V302	**PV** Value read from Controller #3 *or* a Modbus Exception Error Code if V220 = **0083** hex
V303	**TS** Value read from Controller #3
V304	**OP** Value read from Controller #3

Query Block #4

This Query Block reads PV, TS, and OP parameters from Slave Address 4.

V Memory Location	Hex Value	Integer Value	Comments
V320	**0000**	**00000**	Command Error Word - Initially Set to 0
V321	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V322	**0154**	**00340**	V Memory Address where the response from this Query is to be written
V323	**000A**	**00010**	Reserve 10 *bytes* of V Memory (5 words) for response
V324	**0004**	**00004**	Modbus Slave Address of Controller #4
V325	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V326	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V327	**0001**	**00001**	Starting Modbus Register Address in Slave
V328	**0003**	**00003**	Number of Registers to Read
V329	**5142**	**20802**	Query Block Validation Code

Response Block for Query #4

When the Query is completed successfully, the response from Query #4 will be written to the V memory response block in the following format:

V Memory Location	Contents
V340	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V341	Response Byte Count - the value returned here would be **6** because we are reading 3 words (1 word = 2 bytes)
V342	**PV** Value read from Controller #4 *or* a Modbus Exception Error Code if V220 = **0083** hex
V343	**TS** Value read from Controller #4
V344	**OP** Value read from Controller #4

Query Block #5

This Query Block reads the FS parameter from Slave Address 1.

V Memory Location	Hex Value	Integer Value	Comments
V360	**0000**	**00000**	Command Error Word - Initially Set to 0
V361	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V362	**017C**	**00380**	V Memory Address where the response from this Query is to be written
V363	**0006**	**00006**	Reserve 6 *bytes* of V Memory (3 words) for response
V364	**0001**	**00001**	Modbus Slave Address of Controller #1
V365	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V366	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V367	**004A**	**00074**	Starting Modbus Register Address in Slave
V368	**0001**	**00001**	Number of Registers to Read
V369	**5142**	**20802**	Query Block Validation Code

Response Block for Query #5

When the Query is completed successfully, the response from Query #5 will be written to the V memory response block in the following format:

V Memory Location	Contents
V380	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V381	Response Byte Count - the value returned here would be **2** because we are reading 1 word (1 word = 2 bytes)
V382	**FS** Value read from Controller #1 *or* a Modbus Exception Error Code if V380 = **0083** hex

Query Block #6

This Query Block reads the FS parameter from Slave Address 2.

V Memory Location	Hex Value	Integer Value	Comments
V400	**0000**	**00000**	Command Error Word - Initially Set to 0
V401	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V402	**01A4**	**00420**	V Memory Address where the response from this Query is to be written
V403	**0006**	**00006**	Reserve 6 *bytes* of V Memory (3 words) for response
V404	**0002**	**00002**	Modbus Slave Address of Controller #2
V405	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V406	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V407	**004A**	**00074**	Starting Modbus Register Address in Slave
V408	**0001**	**00001**	Number of Registers to Read
V409	**5142**	**20802**	Query Block Validation Code

Response Block for Query #6

When the Query is completed successfully, the response from Query #6 will be written to the V memory response block in the following format:

V Memory Location	Contents
V420	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V421	Response Byte Count - the value returned here would be **2** because we are reading 1 word (1 word = 2 bytes)
V422	**FS** Value read from Controller #2 *or* a Modbus Exception Error Code if V420 = **0083** hex

Query Block #7

This Query Block reads the FS parameter from Slave Address 3.

V Memory Location	Hex Value	Integer Value	Comments
V440	**0000**	**00000**	Command Error Word - Initially Set to 0
V441	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V442	**01CC**	**00460**	V Memory Address where the response from this Query is to be written
V443	**0006**	**00006**	Reserve 6 *bytes* of V Memory (3 words) for response
V444	**0003**	**00003**	Modbus Slave Address of Controller #3
V445	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V446	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V447	**004A**	**00074**	Starting Modbus Register Address in Slave
V448	**0001**	**00001**	Number of Registers to Read
V449	**5142**	**20802**	Query Block Validation Code

Response Block for Query #7

When the Query is completed successfully, the response from Query #7 will be written to the V memory response block in the following format:

V Memory Location	Contents
V460	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V461	Response Byte Count - the value returned here would be **2** because we are reading 1 word (1 word = 2 bytes)
V462	**FS** Value read from Controller #3 *or* a Modbus Exception Error Code if V460 = **0083** hex

Query Block #8

This Query Block reads the FS parameter from Slave Address 4.

V Memory Location	Hex Value	Integer Value	Comments
V480	**0000**	**00000**	Command Error Word - Initially Set to 0
V481	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V482	**01F4**	**00500**	V Memory Address where the response from this Query is to be written
V483	**0006**	**00006**	Reserve 6 *bytes* of V Memory (3 words) for response
V484	**0004**	**00004**	Modbus Slave Address of Controller #4
V485	**0003**	**00003**	Modbus Function Code for *Read Holding Registers*
V486	**0004**	**00004**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V487	**004A**	**00074**	Starting Modbus Register Address in Slave
V488	**0001**	**00001**	Number of Registers to Read
V489	**5142**	**20802**	Query Block Validation Code

Response Block for Query #8

When the Query is completed successfully, the response from Query #8 will be written to the V memory response block in the following format:

V Memory Location	Contents
V500	Response Function Code - Value will be **0003** hex if good response, **0083** hex if slave device returns a Modbus Exception Code
V501	Response Byte Count - the value returned here would be **2** because we are reading 1 word (1 word = 2 bytes)
V502	**FS** Value read from Controller #4 *or* a Modbus Exception Error Code if V500 = **0083** hex

Query Block #9

This Query Block writes a setpoint value to the TS parameter in Slave Address 1.

V Memory Location	Hex Value	Integer Value	Comments
V520	**0000**	**00000**	Command Error Word - Initially Set to 0
V521	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V522	**021C**	**00540**	V Memory Address where the response from this Query is to be written
V523	**0008**	**00008**	Reserve 8 *bytes* of V Memory (4 words) for response
V524	**0001**	**00001**	Modbus Slave Address of Controller #1
V525	**0010**	**00016**	Modbus Function Code for *Preset Registers*
V526	**0008**	**00008**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V527	**0002**	**00002**	Starting Modbus Register Address in Slave
V528	**0001**	**00001**	Number of Registers
V529	**0002**	**00002**	Byte Count
V530	**nnnn**	**nnnnn**	Setpoint value to write to Controller #1
V531	**5142**	**20802**	Query Block Validation Code

Response Block for Query #9

When the Query is completed successfully, the response from Query #9 will be written to the V memory response block in the following format:

V Memory Location	Contents
V540	Response Function Code - Value will be **0010** hex if good response, **0090** hex if slave device returns a Modbus Exception Code
V541	Response Byte Count - the value returned here would be **0.** The byte count field is not used with Function 16 however, the 2573-MOD protocol manager reserves this field in the response for consistency.
V542	Will contain a value of **2** (dec.) - Starting Address in slave
V543	Will contain a value of **1** - Number of Registers written to.

Query Block #10

This Query Block writes a setpoint value to the TS parameter in Slave Address 2.

V Memory Location	Hex Value	Integer Value	Comments
V560	**0000**	**00000**	Command Error Word - Initially Set to 0
V561	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V562	**0244**	**00580**	V Memory Address where the response from this Query is to be written
V563	**0008**	**00008**	Reserve 8 *bytes* of V Memory (4 words) for response
V564	**0002**	**00002**	Modbus Slave Address of Controller #2
V565	**0010**	**00016**	Modbus Function Code for *Preset Registers*
V566	**0008**	**00008**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V567	**0002**	**00002**	Starting Modbus Register Address in Slave
V568	**0001**	**00001**	Number of Registers
V569	**0002**	**00002**	Byte Count
V570	**nnnn**	**nnnnn**	Setpoint value to write to Controller #2
V571	**5142**	**20802**	Query Block Validation Code

Response Block for Query #10

When the Query is completed successfully, the response from Query #10 will be written to the V memory response block in the following format:

V Memory Location	Contents
V580	Response Function Code - Value will be **0010** hex if good response, **0090** hex if slave device returns a Modbus Exception Code
V581	Response Byte Count - the value returned here would be **0.** The byte count field is not used with Function 16 however, the 2573-MOD protocol manager reserves this field in the response for consistency.
V582	Will contain a value of **2** (dec.) - Starting Address in slave
V583	Will contain a value of **1** - Number of Registers written to.

Query Block #11

This Query Block writes a setpoint value to the TS parameter in Slave Address 3.

V Memory Location	Hex Value	Integer Value	Comments
V600	**0000**	**00000**	Command Error Word - Initially Set to 0
V601	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V602	**026C**	**00620**	V Memory Address where the response from this Query is to be written
V603	**0008**	**00008**	Reserve 8 *bytes* of V Memory (4 words) for response
V604	**0003**	**00003**	Modbus Slave Address of Controller #3
V605	**0010**	**00016**	Modbus Function Code for *Preset Registers*
V606	**0008**	**00008**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V607	**0002**	**00002**	Starting Modbus Register Address in Slave
V608	**0001**	**00001**	Number of Registers
V609	**0002**	**00002**	Byte Count
V610	**nnnn**	**nnnnn**	Setpoint value to write to Controller #3
V611	**5142**	**20802**	Query Block Validation Code

Response Block for Query #11

When the Query is completed successfully, the response from Query #11 will be written to the V memory response block in the following format:

V Memory Location	Contents
V620	Response Function Code - Value will be **0010** hex if good response, **0090** hex if slave device returns a Modbus Exception Code
V621	Response Byte Count - the value returned here would be **0.** The byte count field is not used with Function 16 however, the 2573-MOD protocol manager reserves this field in the response for consistency.
V622	Will contain a value of **2** (dec.) - Starting Address in slave
V623	Will contain a value of **1** - Number of Registers written to.

Query Block #12

This Query Block writes a setpoint value to the TS parameter in Slave Address 4.

V Memory Location	Hex Value	Integer Value	Comments
V640	**0000**	**00000**	Command Error Word - Initially Set to 0
V641	**0000**	**00000**	Command Flags - In this example, we are choosing *not* to stop the Poll List on any error of this Query
V642	**0294**	**00660**	V Memory Address where the response from this Query is to be written
V643	**0008**	**00008**	Reserve 8 *bytes* of V Memory (4 words) for response
V644	**0003**	**00004**	Modbus Slave Address of Controller #4
V645	**0010**	**00016**	Modbus Function Code for *Preset Registers*
V646	**0008**	**00008**	Number of *bytes* of Query Data to follow (1 word = 2 bytes)
V647	**0002**	**00002**	Starting Modbus Register Address in Slave
V648	**0001**	**00001**	Number of Registers
V649	**0002**	**00002**	Byte Count
V650	**nnnn**	**nnnnn**	Setpoint value to write to Controller #4
V651	**5142**	**20802**	Query Block Validation Code

Response Block for Query #12

When the Query is completed successfully, the response from Query #12 will be written to the V memory response block in the following format:

V Memory Location	Contents
V660	Response Function Code - Value will be **0010** hex if good response, **0090** hex if slave device returns a Modbus Exception Code
V661	Response Byte Count - the value returned here would be **0.** The byte count field is not used with Function 16 however, the 2573-MOD protocol manager reserves this field in the response for consistency.
V662	Will contain a value of **2** (dec.) - Starting Address in slave
V663	Will contain a value of **1** - Number of Registers written to.

4. Develop Ladder Logic

Appendix B has a SIMATIC® 505 CPU ladder logic example for controlling 2573-MOD command execution. The example follows the application note.

The ladder logic shown is available for downloading via BBS or Internet - either Control Technology Inc. or Eurotherm Controls Inc.

Configure the CTI 2573-MOD and install in the SIMATIC® 505 I/O chassis

The 2573-MOD is shipped as a complete unit for placement in the PLC chassis. The module has 4 dipswitch blocks toward the rear of the module for configuring the ports (1 switch block per port). The default setting for these switches are all OFF which configures the ports for CAMP/NITP SLAVE. In order to start a protocol manager, such as MODBUS RTU MASTER, from PLC ladder logic, the dipswitches for the corresponding port you are going to use must be set for PLC SELECT. For this application note the dipswitches for Port #1 must be set as follows:

Switch Block SW1 - Switches 1 - 5 **OFF**, 6 - 8 **ON**.

Also, on hardware Revision D boards and higher, there are 4 switches toward the front of the module which enables/disables RS-422/485. **Ensure that the switch which corresponds to the port you are using for this application is in the ON position which connects RS-422/485 communications.**

Termination Resistor Jumpers

RS-422C and RS-485 electrical interfaces may use termination resistors to reduce circuit reflections and to improve signal quality. The 2573-MOD uses a set of jumpers to insert or remove the termination resistors from the circuit. The 2573-MOD is shipped from the factory with the jumpers positioned so that the termination resistance is disabled. If you need to change the factory setting refer to ***Appendix A*** of the document referenced below.

Refer to the CTI 2573-MOD SERIAL DEVICE INTERFACE ADAPTER INSTALLATION AND OPERATION GUIDE, CHAPTER 1 for more details on dipswitch and jumper settings.

Logging the 2573-MOD into the PLC's I/O Configuration Table

Once the 2573-MOD hardware has been configured and placed into the PLC rack, it must be logged into the PLC's word I/O. The 2573-MOD logs in as a 2WX 6WY Special Function Module. If the module were placed in the first I/O slot in Base 0 and given a starting WX address of 1, the module would be configured in the I/O configuration table as follows:

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

Configure Eurotherm Controllers

The communication parameters defined for the CTI 2573-MOD must match those in the controllers. For the Eurotherm 2200 series the following configuration parameters are recommended in bold in Table 9 below.

HA	Comms Module Config	Functions	Meaning
id	Identity of the option installed	PDS.i **cmS**	PDSIO setpoint input EIA 485 comms module
Func	Function	**mod** nonE	Modbus protocol None
bAud	Baud Rate	1200 2400 4800 9600 **19.2**(19200)	
Prty	Comms Parity	nonE **EvEn** Odd	No parity Even parity Odd parity
rESn	Comms Resolution	**FuLL** Int	Full Resolution Integer Resolution

Table 9.

Reference the 2200 series "Installation and operation handbook" for details on accessing and setting these parameters.

For the Eurotherm 2400 series the following configuration parameters are recommended in bold in Table 10 below.

HA	Comms 1 Module Config	Functions	Meaning
id	Identity of the option installed	**cmS** PDS PDS.i	EIA 485 comms module PDSIO retransmission PDSIO setpoint input
Func	Function	**mod** EI.bi	Modbus protocol Eurotherm Bisync protocol
bAud	Baud Rate	1200 2400 4800 9600 **19.2**(19200)	
Prty	Comms Parity	nonE **EvEn** Odd	No parity Even parity Odd parity
rESn	Comms Resolution	**FuLL** Int	Full Resolution Integer Resolution
dELY	Delay - quiet period, required by some comms adaptors	no **YES**	No delay Delay active - 10ms

Table 10.

Reference the 2400 series “Installation and operation handbook” for details on accessing and setting these parameters.

For either series, the *Addr* parameter must be set to uniquely identify each controller on the communications link. Up to 32 controllers may be multi-dropped from each 2573-MOD serial port. Valid controller slave addresses are from 1 to 254.

Wire Eurotherm Controllers to 2573-MOD Module

The wiring diagram for the Eurotherm 2200 and 2400 series controllers to the 2573-MOD is shown in Figure 3 below. The communications is RS-485 two wire. Up to 32 controllers may be multi-dropped per CTI 2573-MOD port.

Port 1 or 3 Cable Diagram

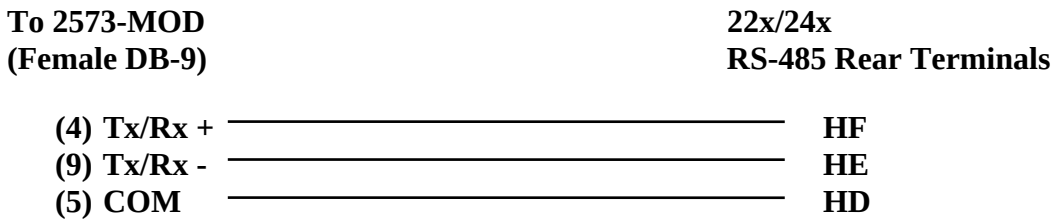


Figure 3. Communications Wiring

Note that the 2573-MOD has 4 Male DB-9 connectors. Therefore, the cable required for Modbus communications to Ports 1 or 3 must have the sex as listed in the table above. The controllers utilize screw terminals.

Additional Information

Ordering 2573-MOD

CTI 2573-MOD SERIAL INTERFACE ADAPTER

Ordering Eurotherm Controllers

The Eurotherm discrete controllers are the 2200 (2216, 2208, and 2204) or 2400 (2416, 2408, and 2404) series. Modbus communications is optional and requires the "**MB"** option in the Comms order code field.

Other Eurotherm controllers that support Modbus RTU are candidates for this communications interface. This includes the Eurotherm models 808, 900EPC, and 90.

Web Sites

Both Eurotherm and Control Technology Inc. maintain Internet web sites which provide additional information. The URL's (Universal Resource Locators) are:

Eurotherm: http://www.eurotherm.com
CTI: http://www.controltechnology.com

Eurotherm Contact Numbers

Telephone: 703.471.4870
FAX: 703.787.3436
BBS: 703.787.3444

CTI Contact Numbers

Telephone: 423.584.0440
FAX: 423.584.5720

Appendix A.

This information has been reproduced from the Series 2000 Communications manual. Please refer to that manual for additional information.

Modbus Addresses.

This section of the manual provides a list of all parameters in Series 2000 instruments that are available over the communications link. As far as possible, it follows the same organization as the instrument user interface itself. Definitions of parameters and status information not available via the instrument display are also provided.

Series 2000 instruments may be configured for a wide variety of functions and some parameters will only be available if the related function is configured. Further, the 2200 series parameters are a subset of the 2400 series, therefore, two columns are shown for the 2200 and 2400 series Modbus addresses. Modbus addresses that are not supported have no parameter assigned or are shown as N/A.

In normal operating mode all configuration parameters are read only. To be able to write to these parameters, the instrument must be in configuration mode. If the Modbus protocol is used to read a parameter that is not configured, an undefined value will be returned. Modbus function 6 single parameter write operations to unconfigured or read only parameters will be rejected with a Modbus 'data error' return code.

These Modbus addresses apply to Series 220x having software V1.3 or greater, and Series 240x having software V2.01 or greater.

NB: Blocks of data written using Modbus function 16 containing values in positions corresponding to the addresses of unconfigured parameters are not generally rejected, although the values of any unconfigured parameters are discarded. This allows relatively large blocks of parameter data to be written in a single operation, even if the block contains a little 'empty' space. This is particularly useful for operations such as ramp/dwell program downloading, recipes, or instrument cloning. However this also leads to a potential pitfall: if the block of data contains only a single parameter, and the destination address refers to an unconfigured or unused Modbus address, the write operation will appear to be successful, although the instrument will have discarded the value.

Attempts to write to read only parameters over Modbus, even when they are embedded within a block of data, will be rejected with a Modbus 'data error'. Any subsequent values in the block will also be discarded.

Operating Mode Parameters

Home List parameters.

Parameter	***Modbus 220x***	***Modbus 240x***	***Display***
Process Variable	1	1	
Target Setpoint	2	2	SP
Auto/Manual Mode 0: Auto 1: Manual	273	273	m-A
Output Power (PID Controller). Not writeable unless the instrument is in 'manual' mode.	3	3	OP
Working Setpoint. Read only: use Target Setpoint or currently selected setpoint (1 to 16) to change the setpoint value.	5	5	w.SP
Load Current returned over PDSIO.	80	80	Lcur
Control Output (On Off Controller). Not writeable unless the instrument is in 'manual' mode 0: -100% 1: 0% 2: 100%	N/A	85	OP
VP Manual Output	N/A	60	-
Valve Position	53	53	-

Run List

Parameter	***Modbus 220x***	***Modbus 240x***	***Display***
Current Program Running	N/A	22	Prg
Current program logic 1 output 0: Off 1: On	N/A	464	out.1
Current program logic 2 output	N/A	465	out.2
Current program logic 3 output	N/A	466	out.3
Current program logic 4 output	N/A	467	out.4
Current program logic 5 output	N/A	468	out.5
Current program logic 6 output	N/A	469	out.6
Current program logic 7 output	N/A	470	out.7
Current program logic 8 output	N/A	471	out.8
Current Sync Status 0: Off 1: On	N/A	488	SYnc

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Program Status 1: Reset 2: Run 4: Hold 8: Holdback 16: Complete	23	23	StAt
Programmer setpoint	N/A	163	PSP
Program Cycles Remaining	N/A	59	CYC
Current Segment Running	N/A	56	SEG
Current Segment Type 0: End 1: Ramp (Rate) 2: Ramp (Time to Target) 3: Dwell 4: Step 5: Call	N/A	29	StyP
Segment Time Remaining	N/A	36	SEG.t
Current segment target SP	N/A	160	tgt
Current segment ramp rate	N/A	161	rAtE
Program Time Remaining	N/A	58	PrG.t
Fast run option 0: No 1: Yes	57	57	FASt

Alarm List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Alarm 1 Setpoint	13	13	1 AL
Alarm 2 Setpoint	14	14	2 AL
Alarm 3 Setpoint	81	81	3 AL
Alarm 4 Setpoint	82	82	4 AL
Alarm 1 Hysteresis	N/A	47	HY 1
Alarm 2 Hysteresis	N/A	68	HY 2
Alarm 3 Hysteresis	N/A	69	HY 3
Alarm 4 Hysteresis	N/A	71	HY 4
Loop Break Time	83	83	Lb t
Enable diagnostic messages 0: No Diagnostics 1: Diagnostics	N/A	282	diAg

Autotune List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Autotune Enable 0: No Tune 1: Tune	270	270	tunE
Adaptive Tune enable 0: No Adaptive Tune 1: Tune	N/A	271	drA
DRA Trigger value	N/A	100	drA.t
Automatic Manual Reset Calculation (Auto droop) 0: Manual Reset 1: Calculated	272	272	Adc

PID List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Gain Scheduler Setpoint	N/A	153	G.SP
Current PID set (Read Only if gain scheduling is selected) 0: Set 1 1: Set 2	N/A	72	SET
Proportional Band PID1	6	6	PB
Integral Time PID1 0: Off	8	8	Ti
Derivative Time PID1 0: Off	9	9	Td
Manual Reset	28	28	rES
Cutback High 0: Auto	18	18	Hcb
Cutback Low PID1 0: Auto	17	17	Lcb
Relative Cool Gain PID1	19	19	rEL.C

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Proportional Band PID2	N/A	48	Pb2
Integral Time PID2 0: Off	N/A	49	Ti2
Derivative Time PID2 0: Off	N/A	51	Td2
Manual Reset PID 2	N/A	50	rES.2
Cutback High PID 2 0: Auto	N/A	118	Hcb2
Cutback Low PID2 0: Auto	N/A	117	Lcb2
Relative Cool Gain PID2	N/A	52	rEL.2
Cool (Reserved) Proportional Band	N/A	90	Pb.c
Cool (Reserved) Deadband	N/A	91	db.c
FeedForward Proportional Band	N/A	97	FF.Pb
FeedForward Offset Value	N/A	98	FF.tr
Feedforward Trim Limit	N/A	99	FF.dv

ON/OFF List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Heat Hysteresis	86	86	hYS.H
Cool Hysteresis	88	88	hYS.C
Heat Cool Deadband	16	16	HC.db
On Off Sensor break output power 0: -100% 1: 0% 2: 100%	N/A	40	Sb.OP

MTR List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
VP cycle time	N/A	132	cyc.t
VP Raise Inertia 0: Off	N/A	123	In u
VP Lower Inertia 0: Off	N/A	130	In d
VP raise backlash 0: Off	N/A	124	bAc.u
VP Lower backlash 0: Off	N/A	129	bAc.d
VP raise channel Velocity limit	N/A	125	VEL.u
VP Lower channel velocity limit	N/A	126	VEL.d
VP position low limit	N/A	42	Pot.L
VP position high limit	N/A	43	Pot.H
Boundless Sensor break OP 0: Rest 1: Up 2: Down	N/A	128	Sb.OP

SP List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Internal Setpoint Select 0: SP 1 1: SP 2 2: SP 3 3: SP 4 4: SP 5 5: SP 6 6: SP 7 7: SP 8 8: SP 9 9: SP 10 10: SP 11 11: SP 12 12: SP 13 13: SP 14 14: SP 15 15: SP 16	15	15	SSEL
Remote Setpoint Enable 0: Local SP 1: Remote SP	276	276	L-r
Setpoint One	24	24	SP 1
Setpoint Two	25	25	SP 2
Setpoint Three	N/A	164	SP 3
Setpoint Four	N/A	165	SP 4
Setpoint Five	N/A	166	SP 5
Setpoint Six	N/A	167	SP 6
Setpoint Seven	N/A	168	SP 7
Setpoint Eight	N/A	169	SP 8
Setpoint Nine	N/A	170	SP 9
Setpoint Ten	N/A	171	SP10
Setpoint Eleven	N/A	172	SP11
Setpoint Twelve	N/A	173	SP12
Setpoint Thirteen	N/A	174	SP13
Setpoint Fourteen	N/A	175	SP14
Setpoint Fifteen	N/A	176	SP15
Setpoint Sixteen	N/A	177	SP16
Remote setpoint	N/A	485	rm.SP
Remote Setpoint Trim	N/A	486	rmt.t
Ratio Setpoint	N/A	61	rAt
Local Setpoint Trim	27	27	Loc.t
Setpoint 1 Low Limit	112	112	SP L
Setpoint 1 High Limit	111	111	SP H
Setpoint 2 Low Limit	114	114	SP2.L
Setpoint 2 High Limit	113	113	SP2.H
Local Setpoint Trim Low Limit	67	67	Loc.L

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Local Setpoint Trim High Limit	66	66	Loc.H
Setpoint Rate Limit 0: Off	35	35	SPrr
Setpoint Rate Limit Holdback 0: None 1: Low 2: High 3: Band	N/A	70	Hb.ty
Setpoint Rate Limit Holdback Value	N/A	65	Hb

IP List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
PV Time Constant	101	101	FiLt
Pyrometer Emmisivity	N/A	38	EmiS
User Calibration Enable 0 Factory 1 User	110	110	CAL
Transducer Lo Cal enable 0: No 1: Yes	N/A	109	CAL.L
Linear Correction Low	N/A	145	Adj.L
Transducer Hi Cal enable 0: No 1: Yes	108	108	CAL.H
Linear Correction High	N/A	144	Adj.H
ADC Converter millivolts	202	202	MV.1
Second PV millivolt input	N/A	208	MV.2
CJC Temperature	215	215	CJC.1

OP List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Low Power Limit	31	31	OP.Lo
High Power Limit	30	30	OP.Hi
Remote Low Power Limit	N/A	33	rOP.L
Remote High Power Limit	N/A	32	rOP.H
Output Rate Limit (%/second) 0: Off	N/A	37	OPrr
Forced Output Power	N/A	84	FOP
Sensor Break Power	34	34	Sb.OP
Heat Cycle Time	10	10	CYC.H
Heat channel Minimum On Time 0: Auto	45	45	ont.H
Cool Cycle Time	20	20	CYC.C
Cool channel Minimum On Time 0: Auto	89	89	ont.C

CMS List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Comms address. Changes are effective immediately.	131	131	Addr

Info List

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Custom Display Type 0: Standard 1: Load Current 2: Output Power 3: Status 4: Program Time 5: None	106	106	diSP
SPC Minimum PV	N/A	134	LoG.L
SPC Maximum PV	N/A	133	LoG.H
SPC Mean PV	N/A	135	LoG.A
SPC Time above TimeTrigger	N/A	139	LoG.t
PV Threshold for Timer Log	N/A	138	LoG.v
SPC reset 0: Not Reset 1: Reset	N/A	140	rES.L
Control task execution time high water mark	N/A	201	MCt
Working Output	N/A	4	w.OP
PDSIO SSr Status 0 OK 1 Load Fail 2 Open 3 Heater Fail 4 SSr Fail 5 Sn Fail	N/A	79	SSr
Feedforward Output	N/A	209	FF.OP
Proportional Output	N/A	214	P OP
Integral Output	N/A	55	I OP
Derivative Output	N/A	116	d OP
VP Velocity signal	N/A	219	VEL
VP Motor Calibration State 0: Start 1: Waiting 2: Open Valve 3: BLUp/InDn 4: TT Up 5: Overshoot 6: InUp/BLDn 7: TT Down 8: Open 9: Low Lim 10: Stopping 11: Raise 12: Inert UP	N/A	210	vP S

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
13: Lower 14: Low Lim 15: Stopping 16: Lower 17: InDn/BL 99: Abort			

Miscellaneous Status and Comms-only Parameters

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Remote Input Comms access parameter	26	26	-
Process Error	39	39	-
Setpoint Rate Limit Holdback Status 0: Inactive 1: Active	N/A	41	-
System Error Logged Flag 0 No Error 1 Error	N/A	73	-
Ramp Rate disable	N/A	78	-
Slave Instrument Target Setpoint	N/A	92	-
Slave Instrument Ramp Rate	N/A	93	-
Slave Instrument Sync signal	N/A	94	-
Remote SRL Hold	N/A	95	-
BCD Input Value	N/A	96	-
Instrument Version Number in format: >XXYY (hex) where XX is major version number, and YY is minor version number. For example >0304 corresponds to V3.04	107	107	-
CNOMO Manufacturers Identifier	121	121	-
Instrument Identifier in format >ABCD (hex), A = 2 (series 2000) B = Range number 2: 2200 4: 2400 C = Size 3: 1/32 din 6: 1/16 din 8: 1/8 din 4: 1/4 din D = Type 0: PID/ON-OFF 2: VP	122	122	-
Bisynch Comms Status 0: No Error 1: Invalid Mnemonic 2: Parameter is read only 7: Incorrect message 8: Limit error	-	-	-

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Reserved remote par	N/A	151	-
VP Low Limit switch	N/A	120	-
VP High Limit Switch - Open	N/A	119	-
VP Motor Calibrate Enable 0: Off 1: On	N/A	46	-

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Instrument Mode. **NB: WRITING OTHER VALUES TO THIS PARAMETER MAY CAUSE DAMAGE TO CALIBRATION OR INSTRUMENT CONFIGURATION!** 0: Normal 1: Standby 2: Config Note that the instrument address for EI-Bisynch changes to '00' when instrument mode is changed to configuration	199	199	-
PV Millivolts from Comms	203	203	-
Input test point enable	205	205	-
Sensor Break sourced from Test	206	206	-
Filter Initialization Flag	207	207	-
Maximum Number of Segments (8 or 16): Read Only	N/A	211	-
Edit Program	-	-	-
Freeze Control flag 0: Controlling 1: Hold	N/A	257	-
Sensor break status flag 0: OK 1: Sensor Break	258	258	-
Power Failed Flag 0: OK 1: Power Fail detected	N/A	259	-
Loop Break Status Flag 0: OK 1: Loop Break	N/A	263	-

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Integral Hold Status Flag 0: OK 1: Integral Hold	N/A	264	-
Acknowledge All Alarms 0: OK 1: Acknowledge All Alarms	274	274	-

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Setpoint Rate Limit Active Status 0: No SRL 1: SRL Active	N/A	275	-
SRL Complete Status 0: SRL Incomplete 1: SRL Complete	N/A	277	-
Holdback Disable 0: Holdback Enabled 1: Holdback Disabled	N/A	278	-
Disable keys 0: Keys Enabled 1: Keys Disabled	279	279	-
Remote input status 0: OK 1: Fault	N/A	280	-
Sync/Continue flag 0: Continue 1: Awaiting Sync	N/A	281	-
DC Input Remote Fault 0: OK 1: Fault	N/A	283	-
Maximum Input value in Engineering Units	N/A	548	-
Minimum Input Value in Engineering Units	N/A	549	-
Setpoint Span	N/A	552	-

Status Words

Note that status words are generally read only. Individual parameters exist for all status indicators that may be changed over the communications link, and these should be used for 'write operations'.

The exception is the digital output telemetry status word, which may be written to set digital outputs, provided their function is configured to 'No Func'.

Parameter	***Modbus 220x***	***Modbus 240x***	***Display***
Fast Status byte. Read Only (Also available via Modbus Function 7)	74	74	-

BIT	**DESCRIPTION**
Bit 0	Alarm 1 State (0 = Safe 1 = Alarm)
Bit 1	Alarm 2 State (0 = Safe 1 = Alarm)
Bit 2	Alarm 3 State (0 = Safe 1 = Alarm)
Bit 3	Alarm 4 State (0 = Safe 1 = Alarm)
Bit 4	Manual Mode (0 = Auto 1 = Manual)
Bit 5	Sensor Break (0 = Good PV 1 = Sensor Broken)
Bit 6	Loop Break (0 = Good closed loop 1 = Open Loop)
Bit 7	Heater Fail (0 = No Fault 1 = Load fault detected)

Parameter	***Modbus 220x***	***Modbus 240x***	***Display***
Summary Output Status Word	N/A	75	-

BIT	**DESCRIPTION**
0	Alarm 1 State (0 = Safe 1 = Alarm)
1	Alarm 2 State (0 = Safe 1 = Alarm)
2	Alarm 3 State (0 = Safe 1 = Alarm)
3	Alarm 4 State (0 = Safe 1 = Alarm)
4	Manual Mode (0 = Auto 1 = Manual)
5	Sensor Break (0 = Good PV 1 = Sensor Broken)
6	Loop Break (0 = Good closed loop 1 = Open Loop)
7	Heater Fail (0 = No Fault 1 = Load fault detected)
8	Tune Active (0 = Auto Tune disabled 1 = Auto Tune active)
9	Ramp/Program Complete (0 = Running/Reset 1 = Complete)
10	PV out of range (0 = PV within table range 1 = PV out of table range)
11	DC control module fault (0= Good. 1= BAD)
12	Programmer Segment Synchronize (0 = Waiting, 1 = Running)
13	Remote input sensor break (0 = Good, 1 = Bad)
14	Reserved
15	Reserved

Parameter		*Modbus 220x*	*Modbus 240x*	*Display*
Control Status Word		N/A	76	-
BIT	**DESCRIPTION**			
0	Control algorithm Freeze			
1	PV input sensor broken			
2	PV out of sensor range			
3	Self Tune failed			
4	PID servo signal			
5	PID debump signal			
6	Fault detected in closed loop behavior			
7	Freezes the integral accumulator			
8	Indicates that a tune has completed successfully			
9	Direct/reverse acting control			
10	Algorithm Initialisation flag			
11	PID demand has been limited.			
12	Autotune enabled			
13	Adaptive tune enabled			
14	Automatic Droop compensation enabled			
15	Manual / Auto mode switch			

Parameter		*Modbus 220x*	*Modbus 240x*	*Display*
Instrument Status Word		N/A	77	-
BIT	**DESCRIPTION**			
0	Config/Oper mode switch			
1	Disables limit checking			
2	SRL ramp running (Read Only)			
3	Remote setpoint active			
4	Alarm acknowledge switch.			
5	Reserved			
6	Reserved			
7	Reserved			
8	Reserved			
9	Reserved			
10	Reserved			
11	Reserved			
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Digital Input Status Word Note that the order of LA and LB is reversed relative to what might be expected.	N/A	87	-

BIT	DESCRIPTION
0	H Interface module (0 = Off, 1 = On)
1	J Interface module (0 = Off, 1 = On)
2	1A module (0 = Off, 1 = On)
3	LB logic input (0 = Off, 1 = On)
4	LA logic input (0 = Off, 1 = On)
5	1B module telemetry (0 = Off, 1 = On)
6	1C module (0 = Off, 1 = On)
7	2A module (0 = Off, 1 = On)
8	2B module (0 = Off, 1 = On)
9	2C module (0 = Off, 1 = On)
10	3A module (0 = Off, 1 = On)
11	3B module (0 = Off, 1 = On)
12	3C module (0 = Off, 1 = On)
13	Reserved
14	Reserved
15	Reserved

Parameter	*Modbus 220x*	*Modbus 240x*	*Display*
Digital Output Telemetry Parameter Note that the order of LA and LB is reversed relative to what might be expected.	N/A	551	-

BIT	DESCRIPTION
0	H Interface module telemetry (0 = Off, 1 = On)
1	J Interface module telemetry (0 = Off, 1 = On)
2	1A module telemetry (0 = Off, 1 = On)
3	LB logic telemetry (0 = Off, 1 = On)
4	LA logic telemetry (0 = Off, 1 = On)
5	1B module telemetry (0 = Off, 1 = On)
6	1C module telemetry (0 = Off, 1 = On)
7	2A module telemetry (0 = Off, 1 = On)
8	2B module telemetry (0 = Off, 1 = On)
9	2C module telemetry (0 = Off, 1 = On)
10	3A module telemetry (0 = Off, 1 = On)
11	3B module telemetry (0 = Off, 1 = On)
12	3C module telemetry (0 = Off, 1 = On)
13	AA relay telemetry (0 = Off, 1 = On)
14	Reserved
15	Reserved

Parameter		*Modbus 220x*	*Modbus 240x*	*Display*
Program Logic Outputs		N/A	162	-
BIT	**DESCRIPTION**			
0	Program Output 1 (0 = OFF 1 = ON)			
1	Program Output 2 (0 = OFF 1 = ON)			
2	Program Output 3 (0 = OFF 1 = ON)			
3	Program Output 4 (0 = OFF 1 = ON)			
4	Program Output 5 (0 = OFF 1 = ON)			
5	Program Output 6 (0 = OFF 1 = ON)			
6	Program Output 7 (0 = OFF 1 = ON)			
7	Program Output 8 (0 = OFF 1 = ON)			
8	Reserved			
9	Reserved			
10	Reserved			
11	Reserved			
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			

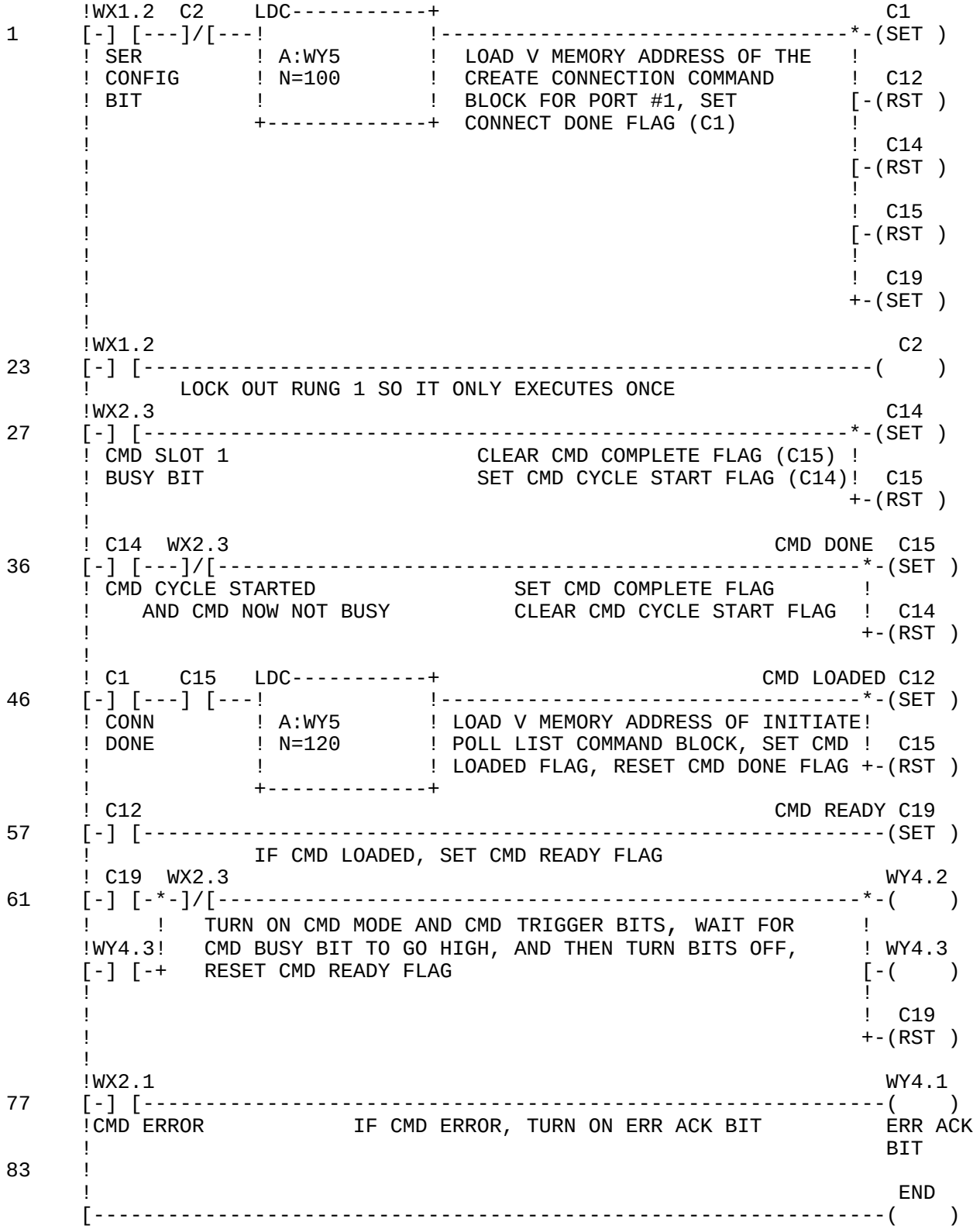
Modbus Bit Addressable Parameters

A few bit addressable parameters are provided to conform to the CNOMO Modbus standard, but in general status information should be obtained via the status words or single status parameters in the Modbus word address space.

Parameter	*Modbus Bit (Coil) Address*
Auto/Manual Mode 0: Auto 1: Manual	2
Alarm 1 Status 0: No Alarm 1: Alarm	5
Sensor Break Status 0: OK 1: Sensor Break	10

Appendix B.

505 CPU Ladder Logic Example



APPENDIX C.

2573-MOD WX/WY REFERENCE CHART

WX/WY OFFSET	BIT #1	BIT #2	BIT #3	BIT #4	BIT #5	BIT #6	BIT #7	BIT #8	BIT #9	BIT #10	BIT #11	BIT #12	BIT #13	BIT #14	BIT #15	BIT #16
WX1	MOD FAIL	SER CFG	N/A	DIAG ERR	CFG ERR	N/A	N/A	N/A	TIMER OR ERROR VALUE							
WX2	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 Bits				CMD 2 Status Bits				CMD 3 Status Bits				CMD 4 Status Bits			
WY3	MOD Reset	Bits 2 - 16 are reserved - Set to 0														
WY4	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	ERR ACK
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	CMD 1 Control Bits				CMD 2 Control Bits				CMD 3 Control Bits				CMD 4 Control Bits			

WY5	CMD 1 V Memory Address of Command Block
WY6	CMD 2 V Memory Address of Command Block
WY7	CMD 3 V Memory Address of Command Block
WY8	CMD 4 V Memory Address of Command Block

NOTE:

WX1 offset relates to the starting word address where the 2573-MOD is logged into the CPU's word I/O. The table below illustrates how the WX/WY offsets relate to various 2573-MOD login addresses.

WX/WY OFFSET	LOGIN ADDRESS WX1	LOGIN ADDRESS WX9	LOGIN ADDRESS WX57	LOGIN ADDRESS WX385
WX1	WX1	WX9	WX57	WX385
WX2	WX2	WX10	WX58	WX386
WY3	WY3	WY11	WY59	WY387
WY4	WY4	WY12	WY60	WY388
WY5	WY5	WY13	WY61	WY389
WY6	WY6	WY14	WY62	WY390
WY7	WY7	WY15	WY63	WY391
WY8	WY8	WY16	WY64	WY392

2573-MOD Word and Bit Descriptions

WX1 - Module Status Word - *Used to communicate overall status of the module.*

MOD FAIL Bit - The module sets this bit if diagnostics detects a severe fault.

SER CFG Bit - The module sets this bit high on power up or reset. The bit stays ON until all comm ports are configured, either by dipswitch or PLC Logic.

DIAG ERR - Set ON if module diagnostics detects a Ram or Rom error.

CFG ERR - Set ON if module detects invalid switch setting at startup/reset.

TIMER/ERROR - Displays value of a module timer (increments approx. once per second) when module processor is operating. If an error occurs, an error code will written to this field, if possible.

WX2 - COMMAND STATUS WORD - *Used by module to indicate status of commands for the 4 command slots.*

CMD ERR Bit - This bit is set when an error is encountered while attempting to execute the command block pointed to in the corresponding command word.

PLC ERR Bit - Set to indicate that a valid command block could not be read from V memory.

CMD BUSY Bit - Indicates the module is in the process of executing a command.

ABORT BUSY Bit - Indicates the module is attempting to abort a command.

WY3 - MODULE CONTROL WORD - *Module control bits are contained in WY3.*

MOD RESET Bit - When this bit is set *and* all the abort trigger bits 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 and error or aborting a command does not clear the error.

WY4 - COMMAND CONTROL WORD - *The Command Control Word contains bits that are set by the PLC logic for command execution and control.*

ERR ACK Bit - The PLC sets this bit to acknowledge the corresponding error bit set by the module in WX2.

CMD MODE Bit - 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 Bit - The PLC sets this bit to initiate a command.

ABORT TRIG Bit - The PLC sets this bit to abort a command in process.

WY5 - COMMAND SLOT 1 - *Contains V memory address of command block for CMD 1.*

WY6 - COMMAND SLOT 2 - *Contains V memory address of command block for CMD 2.*

WY7 - COMMAND SLOT 3 - *Contains V memory address of command block for CMD 3.*

WY8 - COMMAND SLOT 4 - *Contains V memory address of command block for CMD 4.*

Trademarks:

SIMATIC®, SIEMENS® and SINEC are registered trademarks of Siemens AG.
505 and TISOFT are registered trademarks of Siemens Energy & Automation, Inc.
Modbus® is a registered trademark of AEG-Modicon.
Allen-Bradley is a registered trademark of Allen-Bradley.
PLC is a trademark of Allen-Bradley.
Series 2000 is a registered trademark of Eurotherm.
PDSIO™ is a registered trademark of Eurotherm.
Instant Accuracy™ is a registered trademark of Eurotherm.

Eurotherm has used all reasonable resources and efforts to indicate and supply information regarding trademarks used in this presentation. The absence of a trademark identifier is not a representation that a particular word or technology is not a trademark.

DOCUMENT DISCLAIMER STATEMENTS

Eurotherm Controls, Inc.

 Every effort has been taken to ensure the accuracy of this presentation. However, in order to maintain our technological lead, we are continuously improving our products which could, without notice, result in amendments or omissions to this presentation. We cannot accept responsibility for damage, injury, loss, or expenses resulting therefrom.

Control Technology Inc.

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 and error, please call us toll-free at 1-800-537-8398.