Cincinnati Sub-Zero

EZT-570L

User Communication Reference Manual

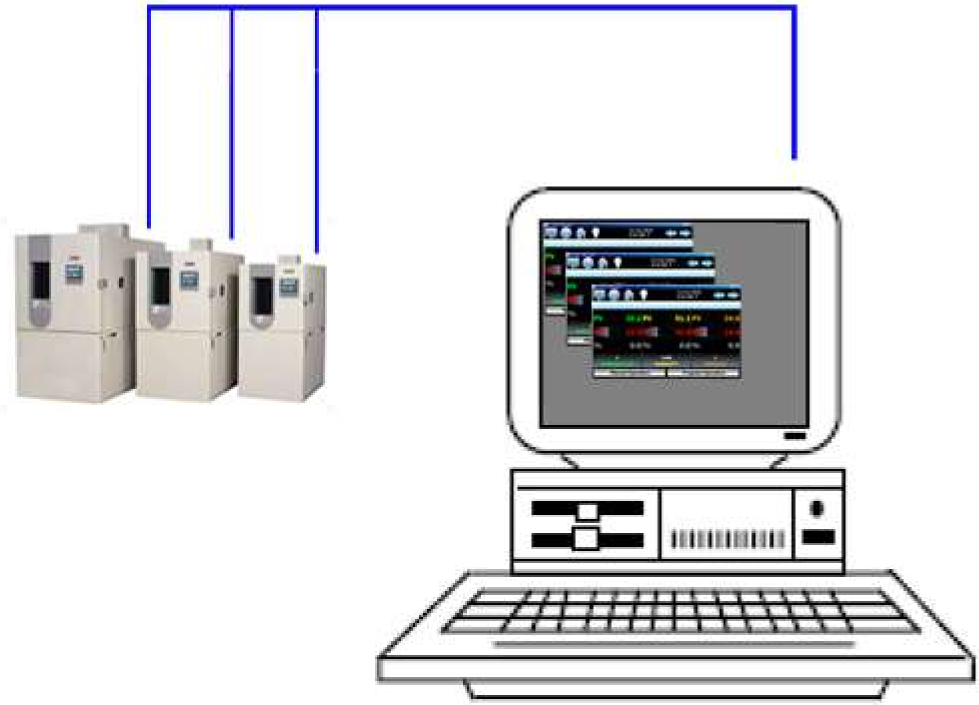


TABLE OF CONTENTS

1. Introduction

This document is targeted towards users interested in using data communications with EZT-570L chamber control. The purpose of this manual is to enable users to:

1. Understand the basics of data communications via standard definitions, interfaces and protocols.
2. Set up and use a simple network of one or more EZT-570L controller(s).

In this manual, numbers in the format 0x00 represent values in hexadecimal. Numbers in the format 0 represent values in decimal and finally, numbers in the format 00000000 represent values in binary unless otherwise stated.

1.1 Definition of Terms

Machine-to-Machine Communication
In order for machines to communicate with each other, they need a code called a character format or character set. They need rules called protocol to govern their conversation and prevent confusion and errors. Computers need a connecting interface over which to communicate. They may use one pair of wires to send information in one direction and another pair to send in the opposite direction (full duplex). Or they may use one pair to send in both directions (half duplex).

Character Format
The code or character format for the EZT-570L data communication is shared by virtually everyone in the electronics industry. This code defines a computer stream of 1's and 0's that are created by varying a voltage signal in a regular manner. This code is the American Standard Code for Information Interchange, called ASCII.

Bits and Bytes
The word bit is simply the contraction of the words binary digit. A bit is the basic unit in ASCII. It is either a "1" or a "0". A byte is a string of eight bits that a computer treats as a single character. ASCII can use a single byte to represent each letter of the alphabet, each digit and each punctuation mark we use.

ASCII
The ASCII code defines 128 separate characters, one for each letter, digit and punctuation mark. ASCII also includes control characters similar to those we find on computer keys, such as backspace, shift and return. It also has nine communications control characters for identification, enquiry (inquiry), start of text, end of text, end of transmission, acknowledge, negative acknowledge and escape. The ASCII code is sometimes written in a base 16 number system that is called hexadecimal or "hex" for short. The numbers 0 through 9 represents the first ten digits of this system, and the letters A through F represents the final six digits. The 128 ASCII character codes with the decimal, binary and hexadecimal equivalents are listed in the following table.

ASCII Control Codes
ASCII Control Codes are used to give instructions to remote device and result in specific actions, such as a line feed instruction on a printer. ASCII Control Codes, the first 33 ASCII characters (non-printable), are important for the operation of communicating equipment. Holding down the keyboard control key while pressing the appropriate keyboard key is what sends these values.

ASCII Character Chart

Char	Code	Decimal	Binary	Hex	Char	Code	Decimal	Binary	Hex
NUL	Ctrl @	0	00000000	00	**@**	Shift 2	64	01000000	40
SOH	Ctrl A	1	00000001	01	**A**	Shift A	65	01000001	41
STX	Ctrl B	2	00000010	02	**B**	Shift B	66	01000010	42
ETX	Ctrl C	3	00000011	03	**C**	Shift C	67	01000011	43
EOT	Ctrl D	4	00000100	04	**D**	Shift D	68	01000100	44
ENQ	Ctrl E	5	00000101	05	**E**	Shift E	69	01000101	45
ACK	Ctrl F	6	00000110	06	**F**	Shift F	70	01000110	46
BEL	Ctrl G	7	00000111	07	**G**	Shift G	71	01000111	47
BS	Ctrl H	8	00001000	08	**H**	Shift H	72	01001000	48
TAB	Ctrl I	9	00001001	09	**I**	Shift I	73	01001001	49
LF	Ctrl J	10	00001010	0A	**J**	Shift J	74	01001010	4A
VT	Ctrl K	11	00001011	0B	**K**	Shift K	75	01001011	4B
FF	Ctrl L	12	00001100	0C	**L**	Shift L	76	01001100	4C
CR	Ctrl M	13	00001101	0D	**M**	Shift M	77	01001101	4D
SO	Ctrl N	14	00001110	0E	**N**	Shift N	78	01001110	4E
SI	Ctrl O	15	00001111	0F	**O**	Shift O	79	01001111	4F
DLE	Ctrl P	16	00010000	10	**P**	Shift P	80	01010000	50
DC1	Ctrl Q	17	00010001	11	**Q**	Shift Q	81	01010001	51
DC2	Ctrl R	18	00010010	12	**R**	Shift R	82	01010010	52
DC3	Ctrl S	19	00010011	13	**S**	Shift S	83	01010011	53
DC4	Ctrl T	20	00010100	14	**T**	Shift T	84	01010100	54
NAK	Ctrl U	21	00010101	15	**U**	Shift U	85	01010101	55
SYN	Ctrl V	22	00010110	16	**V**	Shift V	86	01010110	56
ETB	Ctrl W	23	00010111	17	**W**	Shift W	87	01010111	57
CAN	Ctrl X	24	00011000	18	**X**	Shift X	88	01011000	58
EM	Ctrl Y	25	00011001	19	**Y**	Shift Y	89	01011001	59
SUB	Ctrl Z	26	00011010	1A	**Z**	Shift Z	90	01011010	5A
ESC	Ctrl [	27	00011011	1B	**[**	[	91	01011011	5B
FS	Ctrl \	28	00011100	1C	\	\	92	01011100	5C
GS	Ctrl]	29	00011101	1D	**]**	]	93	01011101	5D
RS	Ctrl ^	30	00011110	1E	**^**	Shift 6	94	01011110	5E
US	Ctrl _	31	00011111	1F	**_**	Shift -	95	01011111	5F
SP	SPACE	32	00100000	20	**`**	`	96	01100000	60
!	Shift 1	33	00100001	21	**a**	A	97	01100001	61
"	Shift ‘	34	00100010	22	**b**	B	98	01100010	62
#	Shift 3	35	00100011	23	**c**	C	99	01100011	63
$	Shift 4	36	00100100	24	**d**	D	100	01100100	64
%	Shift 5	37	00100101	25	**e**	E	101	01100101	65
&	Shift 7	38	00100110	26	**f**	F	102	01100110	66
‘	‘	39	00100111	27	**g**	G	103	01100111	67
(	Shift 9	40	00101000	28	**h**	H	104	01101000	68
)	Shift 0	41	00101001	29	**I**	I	105	01101001	69
*****	Shift 8	42	00101010	2A	**j**	J	106	01101010	6A
+	Shift =	43	00101011	2B	**k**	K	107	01101011	6B
,	,	44	00101100	2C	**l**	L	108	01101100	6C
-	-	45	00101101	2D	**m**	M	109	01101101	6D
.	.	46	00101110	2E	**n**	N	110	01101110	6E
/	/	47	00101111	2F	**o**	O	111	01101111	6F
0	0	48	00110000	30	**p**	P	112	01110000	70
1	1	49	00110001	31	**q**	Q	113	01110001	71
2	2	50	00110010	32	**r**	R	114	01110010	72
3	3	51	00110011	33	**s**	S	115	01110011	73
4	4	52	00110100	34	**t**	T	116	01110100	74
5	5	53	00110101	35	**u**	U	117	01110101	75
6	6	54	00110110	36	**v**	V	118	01110110	76
7	7	55	00110111	37	**w**	W	119	01110111	77
8	8	56	00111000	38	**x**	X	120	01111000	78
9	9	57	00111001	39	**y**	Y	121	01111001	79
:	Shift ;	58	00111010	3A	**z**	Z	122	01111010	7A
;	;	59	00111011	3B	**{**	Shift [	123	01111011	7B
<	Shift ,	60	00111100	3C	**|**	Shift \	124	01111100	7C
=	=	61	00111101	3D	**}**	Shift]	125	01111101	7D
>	Shift .	62	00111110	3E	**~**	Shift `	126	01111110	7E
?	Shift /	63	00111111	3F	**DEL**	Delete	127	01111111	7F

1.2 Protocol

Protocol describes how to initiate a data exchange. It also prevents two machines from attempting to send data at the same time. There are a number of different data communications protocols, just as there are different human cultural protocols that vary according to the situation.

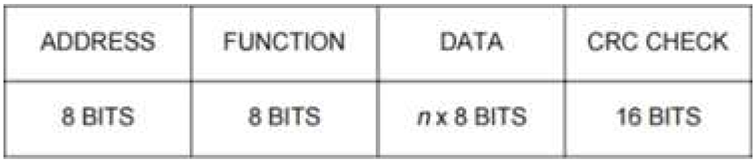
The protocol portion of the EZT-570L communications is very important, because it provides a quality of communication that others often don't have. Protocol-driven communications are more accurate, because they are less prone to both operator and noise errors. Protocol maintains system integrity by requiring a response to each message. It's like registered mail — you know that your letter has been received because the post office sends you a signed receipt.

In EZT-570L data communications, a dialog will continue successfully as long as the messages are in the correct form and responses are returned to the protocol leader. If the operator enters an incorrect message, or interference comes on to the data line, there will be no response. In that case the master must retransmit the message or go to a recovery procedure.

CSZ has selected Modbus as the protocol of choice. Modbus enables a PC to read and write directly to registers containing the EZT-570L parameters. With it, you can read all of the controller's parameters with only a couple of read commands.

Modbus

Gould Modicon, now called Schneider Electric, created this protocol for process control systems called "Modbus". It has the advantage over other protocols of being extremely reliable in exchanging information. This protocol works on the principle of packet exchanges. The packet contains the address of the controller to receive the information, a command field that says what is to be done with the information and several fields of data. The last item sent in the packet is a field to ensure the data is received intact. This is called a cyclic redundancy check-sum. All information exchanged is in hex numbers.



ADDRESS	FUNCTION	DATA	CRC CHECK
8 BITS	8 BITS	*n* x 8 BITS	16 BITS

MODBUS RTU Frame Structure

Each 8 bit byte is framed by 1 start bit, 8 data bits, 0 or 1 parity bit and 1 stop bit. The message itself starts after a silent period of 3.5 character times. Messages are transmitted in a continuous stream format where:

Address: Modbus messages start with an 8 bit target address. This can take any value from 0 to 247, where 0 may be used as a broadcast address and the rest are used as unique device addresses.

Function: The function code is an 8 bit value representing a specific command for reading/writing specific data.

Data: The data field conveys application level information as defined by the function code. If the function contains variable size of data, it begins with a "bye count".

CRC: Modbus messages end with a cyclical redundancy checksum (CRC) which is a 16 bit calculated value derived from the values of all previous bytes in the message.

2. Serial Communication Interface

The EZT-570L provides an optional EIA-232 serial communication interface, which exchanges data in a one-bit-at-a-time, sequential manner on a single data line or channel. Serial contrasts with parallel communication, which sends several bits of information simultaneously over multiple lines or channels. Not only is serial data communication simpler than parallel, it is also less costly.

Baud Rate

The baud unit is named after Jean Maurice Emile Baudot, who was an officer in the French Telegraph Service. He is credited with devising the first uniform-length 5-bit code for characters of the alphabet in the late 19th century. Baud refers to the modulation rate or the number of times per second that a line changes state. This is not always the same as bits per second (BPS). However, if you connect two serial devices together using direct cables then baud and BPS are in fact the same. Thus, if you are running at 9600 BPS, then the line is also changing states 9600 times per second.

Typical baud rates for computers are 9600, 19200, 38400 and 57600 baud. As the baud rate increases, so does the transmission rate of data. Thus, you get more information in a shorter period of time. However, the faster the transmission rate, the more susceptible it is to error due to the quality of the cable and sources of electrical “noise” in the environment. In order to balance throughput with reliability, CSZ has chosen to use 9600 baud as the data rate for the EZT-570L. *Thus, a device used to communicate with the EZT-570L must have its serial port set for 9600 baud in order to for data communications to work properly.*

Start and Stop Bits

The start bit informs the receiving device that a character is coming, and a stop bit tells it that a character is complete. The start bit is always a 0. The stop bit is always a 1. The human speech equivalent of these bits could be a clearing of the throat to get someone’s attention (start bit); and a pause at the end of a phrase (stop bit). Both help the listener understand the message.

A stop bit has a value of 1 - or a mark state - and it can be detected correctly even if the previous data bit also had a value of 1. This is accomplished by the stop bit's duration. Stop bits can be 1, 1.5, or 2 bit periods in length. CSZ has chosen to use the default – and most common – length of 1 period for the EZT-570L. *Thus, a device used to communicate with the EZT-570L must also have its serial port set to use a stop bit of 1 in order for data communications to work properly.*

Parity Bit

Besides the synchronization provided by the use of start and stop bits, an additional bit called a parity bit may optionally be transmitted along with the data. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose even parity, odd parity, mark parity, space parity or none at all. When even or odd parity is being used, the number of marks (logical 1 bits) in each data byte are counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bits just sent is even or odd.

For example, when even parity is chosen, the parity bit is transmitted with a value of 0 if the number of preceding marks is an even number. For the binary value of 0110 0011 the parity bit would be 0. If even parity was in effect and the binary number 1101 0110 was sent, then the parity bit would be 1. Odd parity is just the opposite, and the parity bit is 0 when the number of mark bits in the preceding word is an odd number. Mark parity means that the parity bit is always set to the mark signal condition and likewise space parity always sends the parity bit in the space signal condition. Since these two parity options serve no useful purpose whatsoever, they are almost never used. *The EZT-570L can be set for even, odd or no parity. Therefore, a device used to communicate with the EZT-570L must also have its serial port set to use the same parity setting as the EZT-570L in order for data communications to work properly.*

2.1 Interface Standards

An interface is a means for electronic systems to interact. It's a specific kind of electrical wiring configuration.

EIA-232 (Full Duplex)

An EIA-232 (formerly RS-232C) interface uses three wires: a single transmit wire; a single receive wire; and a common line. Only two devices can use an EIA-232 interface. A -3 to -24 volt signal indicates a 1 and a +3 to +24 volt signal indicates a 0. The EIA-232 signal is referenced to the common line rather than to a separate wire, as in EIA-485. Thus, an EIA-232 cable is limited to a maximum of 50 feet, due to noise susceptibility.

EIA-485 (Half Duplex)

An EIA-485 interface uses two wires: a T+/R+, a T-/R- line. A -5-volt signal is interpreted as a 1, a +5-volt signal as a 0. Multiple devices can be connected to a master on a multi-drop network up to 4000 feet long. The maximum number of devices depends on many factors including cable quality, sources of electrical interference in the area as well as the impedance of the serial port of each device.

Wiring

For PCs with a standard EIA-232 port (usually referred to as RS-232), you must use an interface converter to connect to EIA-485. These interface standards are required to have a multi-drop system (more than one EZT-570L on the link). See section 2.1.1 Interface Converters.

The standards do not specify the wire size and type. Use of AWG 24 twisted pair provides excellent results. If shielded cable is used, terminate the shield at one end only. Always follow the manufacturer's instructions supplied with the interface converter. See Biasing of Buses next.

Biasing of Buses

The EIA-485 standard requires the bus to be biased for reliable communication. This requires termination resistors to be placed across the T+/R+ and T-/R- wires. One resistor is placed at the PC where it connects to the EIA-485 bus. The second resistor is placed at the last controller on the network. Do not place resistors at each controller. The impedance of the wires used for the bus determines the resistor value. For twisted pair, the value is typically 120 ohms.

Check the documentation that came with your interface adapter. Biasing the bus reduces reflection of signals sent down the bus. These reflections are sometimes referred to as a standing wave. This condition is most notable when communicating at high baud rates over longer distances.

2.1.1 Interface Converters

The purpose of an interface converter is to allow two different buses to be connected together. Interface converters are required when connecting an EIA-232 port to an EIA-485 bus. The EIA-485 bus is a half-duplex bus. This means that it can only send or receive data at any given time. Some interface converters on the market provide the ability to have full duplex with the EIA-485 bus. This is accomplished by using two receivers and transmitters tied in tandem.

Another consideration when selecting an interface converter is how the converter handles switching between transmit and receive. Typically, it is accomplished via a handshake line from the PC. When data flows into the converter from the PC, a handshake line is placed high. When data flows out of the converter to the PC, the handshake line is placed low. In this way, the handshake line controls the direction of information. Another method of achieving this is to use a built-in timer. The converter switches to transmit when a character is sent to it from the PC. After a period of time when the PC has not transmitted, the converter switches to a receive mode.

It is important that you understand how your converter accomplishes this task. You are required to wire this feature or make settings on the converter to enable this function. The PC will not talk to the controller correctly without properly setting this. Your converter may also require settings through dip switches to set up communications parameters like baud rate, data bits, start bits, stop bits and handshaking. The converter may also require a separate power supply. Some converters get their power from the handshake lines of the PC. If you rely on this method, you will need to wire these additional lines. In addition, your software must set these lines high. A more reliable method is to use the external power supply. This is especially necessary when using a laptop computer. See the documentation that is provided with your converter for more detail.

Not all converters are equal in performance. If your chamber operates in a harsh, electrically noisy environment, this can cause less robust converters to work intermittently or not at all. CSZ has only tested the converters listed below; however, CSZ makes no claims as to the performance or compatibility of these converters with your PC. These converters are equipped with automatic send data control circuits, driver control in the converter hardware, so you don't have to work with software at all. The circuit monitors data flow and enables the driver during transmission and automatically disables it when no data is being sent. There is no need to rework software or install new drivers.

B&B Electronics
707 Dayton Road
PO Box 1040
Ottawa, IL 61350
Phone 815-433-5100
http://www.bb-elec.com

Part # **485OI9TB** for EIA-232 to EIA-85
Part # **485PS2** (external power supply – required if handshake lines unavailable for power)

RESmith
4311 Smith Drive
Hamilton, OH 45011
Phone 513-874-4796
http://www.RS485.com

Part # **ASC24T-B9FPS** for EIA-232 to EIA-485 (provided with adapter cables and power supply)

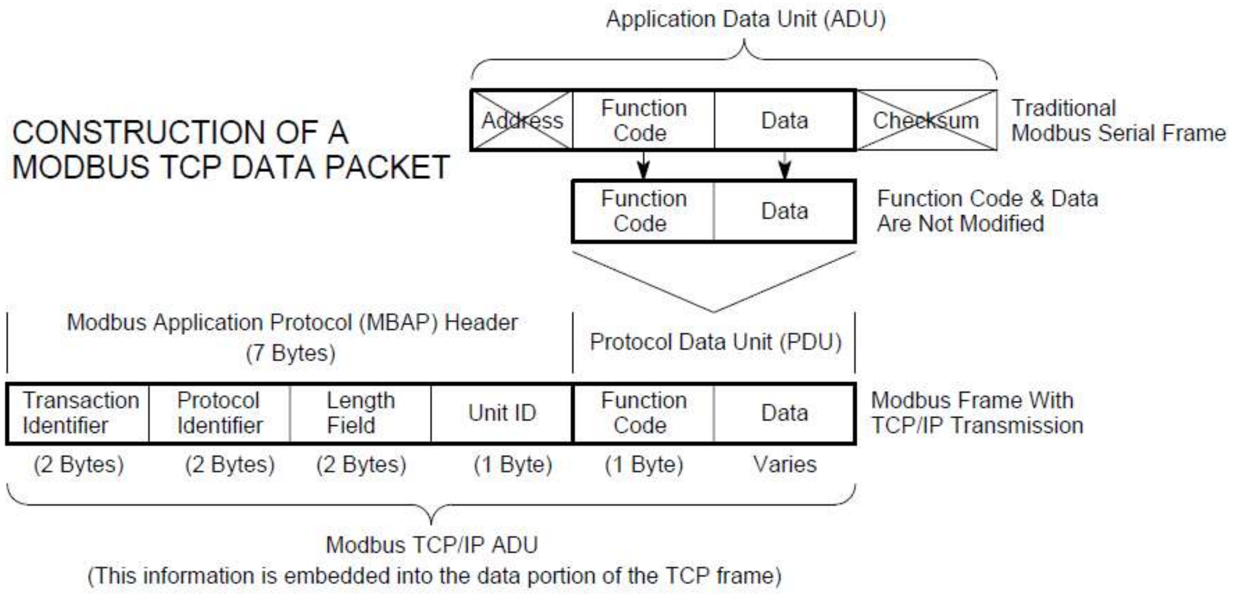
3. Ethernet Communication Interface

The EZT-570L provides a standard 10/100 Base-T Ethernet interface which utilizes Modbus TCP (also known as Modbus TCP/IP) protocol. This is simply the Modbus protocol (described in section 2.2) running on Ethernet. TCP/IP refers to the Transmission Control Protocol and Internet Protocol which provide the transmission medium for the Modbus messaging.

TCP/IP allows blocks of binary data to be exchanged between computers. It is also a world-wide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. Note that the TCP/IP combination is merely a *transport protocol, a*nd does not define what the data means or how the data is to be interpreted (this is the job of the application protocol, Modbus in this case).

Therefore, Modbus TCP/IP uses TCP/IP and Ethernet to carry the data of the Modbus message structure between the EZT-570L and another device, i.e., a computer. In other words, Modbus TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (Modbus as the application protocol). A Modbus TCP/IP message is simply a Modbus communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, Modbus TCP embeds a standard Modbus data frame into a TCP frame without using the Modbus checksum. The Modbus commands and user data are encapsulated into the data container of a TCP/IP telegram without being modified as shown below.


Application Data Unit (ADU)
CONSTRUCTION OF A MODBUS TCP DATA PACKET
Address
Function Code
Data
Checksum
Traditional Modbus Serial Frame
Function Code
Data
Function Code & Data Are Not Modified
Modbus Application Protocol (MBAP) Header (7 Bytes)
Protocol Data Unit (PDU)
Transaction Identifier
Protocol Identifier
Length Field
Unit ID
Function Code
Data
Modbus Frame With TCP/IP Transmission
(2 Bytes)
(2 Bytes)
(2 Bytes)
(1 Byte)
(1 Byte)
Varies
Modbus TCP/IP ADU
(This information is embedded into the data portion of the TCP frame)

The standard Ethernet TCP/IP link layer checksum is automatically applied to guaranty data integrity. The address field of the standard Modbus message is replaced by the unit identifier which is part of the Modbus Application Protocol (MBAP) header. This is the same Modbus Address entered on the Communications screen of the EZT-570L that is used with serial communications.

All Modbus TCP data frames are exchanged between devices over port 502. This is the industry standard port reversed for Modbus TCP and is the port open on the EZT-570L for receiving and transmitting the Modbus TCP messages on the Ethernet interface.

4. Modbus Communications

The EZT-570L utilizes Modbus as its standard protocol and offers both Modbus RTU serial and Modbus TCP forms. This section provides information on the supported functions and data available over the Modbus interfaces. It is assumed that the user is already familiar with Modbus protocol and has a basic understanding of its implementation and use. For additional information and support, modbus.org (http://modbus.org/tech.php) provides an abundance of technical documents, standards and support applications.

Listed below are a few of the more common software packages that claim to support Modbus protocol. This list is provided as informational only. Contact the software manufacturer for more information on applying their software.

LabView by National Instruments
11500 N Mopac Expwy
Austin, TX 78759-3504
Phone 800-683-8411
http://www.natinst.com

Wonderware by Wonderware
26561 Rancho Pkwy. South
Lake Forest, CA 92630
Phone 949-727-3200
http://www.wonderware.com

SpecView by SpecView Corporation
13409 53rd Ave NW
Gig Harbor, WA 98332
Phone 253-853-3199
http://www.specview.com

4.1 Supported Function Codes

Modbus bases its data model on a series of tables, each of which defines a particular range of values. In the case of the EZT-570L, all data is contained within the Holding Register data table. This means that all communication transactions between the EZT-570L and the master (client) are performed using three command types. These include Read Holding Registers, Write Single Register and Write Multiple Registers.

		Function Codes	
		code	(hex)
16 bits access (word)	Read Holding Registers	03	03
	Write Single Register	06	06
	Write Multiple Registers	16	10

Function code 03 (read holding registers) is used to read the contents of a contiguous block of data registers from the EZT-570L. The request from the master specifies the starting register address and the number of registers. The response message from the EZT-570L contains the value of each register requested (two bytes per register).

Function code 06 (write holding register) is used to write to a single data register in the EZT-570L. The request from the master specifies the address of the register to be written and the value. The response message from the EZT-570L is an exact duplicate of the sent message (if completed successfully). This command is used for setting all control parameters in the EZT-570L

Function code 16 (write multiple registers) is used to write to a contiguous block of data registers in the EZT-570L. The request from the master specifies the starting register address to be written, the number of registers to write and two bytes of data for each register. The normal response message from the EZT-570L returns the function code, starting address and quantity of data registers written. This command is used only for downloading program steps to the EZT-570L.

4.2 EZT-570L Control Registers

The EZT-570L is capable of utilizing up to three control loops (or two control loops and one integrated limit) and up to eight monitor inputs. The register list in this section of the manual lists the associated values for all of the loops, inputs and their associated alarms by the loop or monitor input number, i.e., 1 - 3 and 1 - 8. While the monitor inputs will be easy to decipher, since they are shipped from the factory with the relative number in their tag name, the loops are not. The loop names are defined by the chamber process they control, i.e., temperature, humidity, etc., thus the number of control loops required and their function can vary between different chamber models.

The EZT-570L displays all control loops and monitor inputs in sequential order. The loop/monitor order can be viewed from the Overview screen. Starting at the top of the list and counting down, the first entry is loop 1, the second is loop 2, and so on. The following chart provides a loop number to controlled process reference for use in selecting the desired parameter from the register list.

	Temperature Chamber	Temperature and Humidity Chamber
Loop #		
1	Temperature	Temperature
2	Product (or integrated Limit)	Humidity
3	(integrated Limit - optional)	Product (or integrated Limit)

The events also vary based on the model of chamber and options present. In order to turn the dedicated chamber events on and off, it is necessary to set the proper event. The chart below provides the chamber event number and its associated function.

	Dedicated Chamber Events
Event #	
1	CHAMBER
2	HUMIDITY
3	AUX COOL
4	PURGE
5	-
6	-
7	-
8	-
9	INITIATE DEFROST
10	PRODUCT CONTROL
11	-
12	-
13	-
14	-
15	-
16	-

Depending upon the EZT-570L configuration and chamber model, some events may be defined as “dedicated chamber events”. This can be determined by going to the Event Names screen under Device Settings and checking the event descriptions shown under each event name. If listed as “Chamber Event #”, they are dedicated chamber events and must be set in the first event register (400023).

Any optional customer events are then set in sequential order in the second event register (400024).

If the event descriptions contain only the loop or expansion output assignment numbers (no chamber event # descriptions), then the events are in sequential order as shown on the Overview screen and are set in the first event register (400023) only. The second event register (400024) is not used.

NOTE: *See the control register list legend item B9 for more information on setting specific events.*

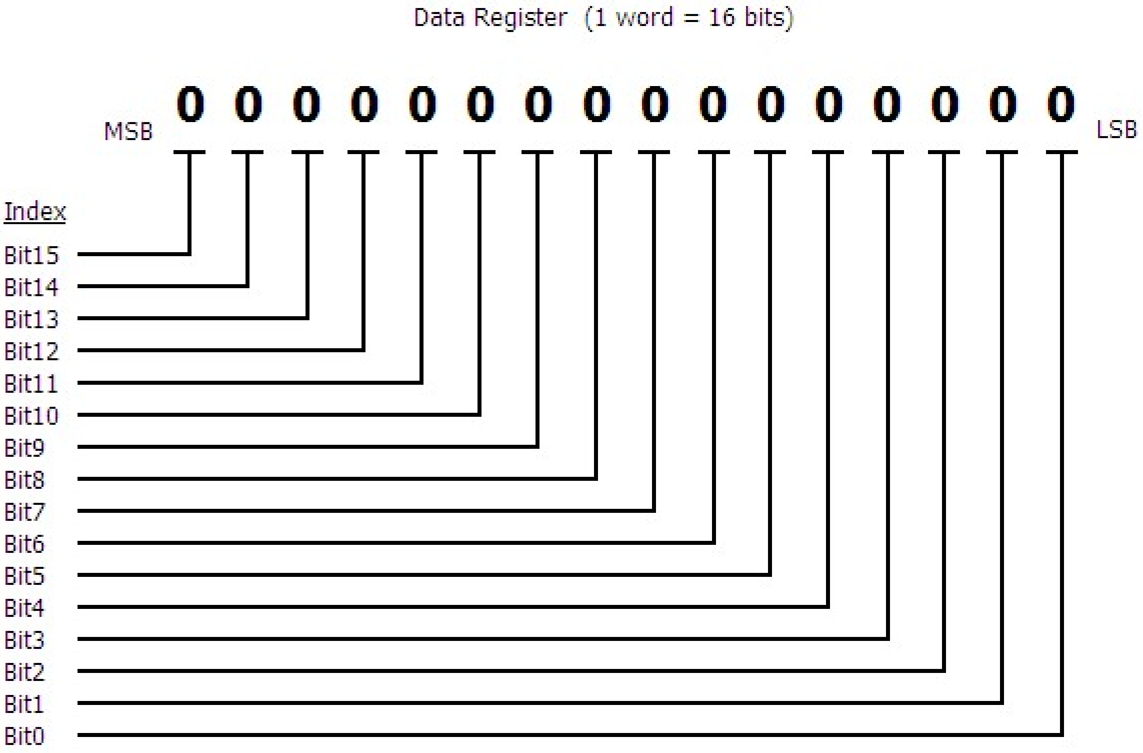
The control registers are grouped into three blocks of 60 (for a total of 180) registers relating to the specific types of data they contain. The first group of 60 registers (0 – 59) contains the configuration settings for various options on the EZT-570L as well as all of the alarm status, ramp/soak program status and manual on/off settings for the chamber. The second group of 60 registers (60 – 119) contains all of the loop control/monitor settings which include the set point and alarm settings for each loop. The third group of 60 registers (120 – 179) contains all of the optional monitor input settings including the individual alarm settings for each.

Bit Oriented Parameters

Some of the values contained in the EZT-570L register base contain bit-oriented values. This means that each bit of the word indicates an on/off status for a specific setting or condition. In handling these values, it is recommended that the word be converted to its binary equivalent.

By converting the value to its binary equivalent, it produces a Boolean array of true [bit on (1)] and false [bit off (0)] values. This allows each bit to be examined individually. In the same manner, creating a Boolean array of 16 bits produces an equivalent decimal value that can be sent to the EZT-570L in order to set a control value.

For the purpose of this manual, parameters defined as bit oriented will have the function of each bit associated with the bit's index number in the data word. The index number is equal to that of a typical array function. Thus, an index number of zero, selects the first bit in the word. An index number of 1 selects the second bit in the word, and so on. This helps eliminate offset selection errors that may occur when coding software and using array functions to select which bit in the word that is required for examination.



Adhere to the following lists of registers and their allowable data ranges. DO NOT attempt to write to any other register number than those listed. DO NOT write to registers that are for options your controller does not have. Failure to adhere to this requirement can result in erratic control and/or damage to equipment.

Modbus Address	Register Number		Parameter Description	Data Type *A	Range *B		*C Unit
					Low	High	
400001	0	**(0x0000)**	System Mode	R/W	*B1	*B1	-
400002	1	**(0x0001)**	Clock (Year/Month)	R	*B2	*B2	-
400003	2	**(0x0002)**	Clock (Day/DOW)	R	*B3	*B3	-
400004	3	**(0x0003)**	Clock (Hours/Minutes)	R	*B4	*B4	-
400005	4	**(0x0004)**	Clock (Seconds)	R	0	59	seconds
400006	5	**(0x0005)**	Power Recovery Mode	R/W	*B5	*B5	-
400007	6	**(0x0006)**	Power Recovery Time	R/W	0	32767	seconds
400008	7	**(0x0007)**	Defrost Operating Mode	R/W	*B6	*B6	-
400009	8	**(0x0008)**	Defrost Temperature Set point	R/W	-32768	32767	*C1
400010	9	**(0x0009)**	Defrost Interval	R/W	0	32767	minutes
400011	10	**(0x000A)**	Defrost Status	R	*B7	*B7	-
400012	11	**(0x000B)**	Time Remaining Until Next Defrost	R	0	32767	minutes
400013	12	**(0x000C)**	Product Control	R	*B8	*B8	-
400014	13	**(0x000D)**	Product Control Upper Set Point	R/W	-32768	32767	*C1
400015	14	**(0x000E)**	Product Control Lower Set Point	R/W	-32768	32767	*C1
400016	15	**(0x000F)**		R			
400017	16	**(0x0010)**		R			
400018	17	**(0x0011)**		R			
400019	18	**(0x0012)**		R			
400020	19	**(0x0013)**		R			
400021	20	**(0x0014)**		R			
400022	21	**(0x0015)**	Chamber Light Control (Off/On)	R/W	0	1	-
400023	22	**(0x0016)**	Chamber Events	R/W	*B9	*B9	-
400024	23	**(0x0017)**	Customer Events	R/W	*B9	*B9	-
400025	24	**(0x0018)**	Program Control/Status	R/W	*B10	*B10	-
400026	25	**(0x0019)**	Program Advance Previous/Next Step	R/W	*B11	*B11	-
400027	26	**(0x001A)**	Program Name Characters 1 & 2	R	*B12	*B12	-
400028	27	**(0x001B)**	Program Name Characters 3 & 4	R	*B12	*B12	-
400029	28	**(0x001C)**	Program Name Characters 5 & 6	R	*B12	*B12	-
400030	29	**(0x001D)**	Program Name Characters 7 & 8	R	*B12	*B12	-
400031	30	**(0x001E)**	Program Name Characters 9 & 10	R	*B12	*B12	-
400032	31	**(0x001F)**	Year/Month Program Started	R	*B2	*B2	-
400033	32	**(0x0020)**	Day/DOW Program Started	R	*B3	*B3	-
400034	33	**(0x0021)**	Hour/Minute Program Started	R	*B4	*B4	-
400035	34	**(0x0022)**	Year/Month Estimated Program End	R	*B2	*B2	-
400036	35	**(0x0023)**	Day/DOW Estimated Program End	R	*B3	*B3	-
400037	36	**(0x0024)**	Hour/Minute Estimated Program End	R	*B4	*B4	-
400038	37	**(0x0025)**	Program Start Step Number	R/W	1	99	-
400039	38	**(0x0026)**	Current Step of Program	R	1	99	-
400040	39	**(0x0027)**	Last Step of Program	R	1	99	-

Modbus Address	Register Number		Parameter Description	Data *A	Range *B		*C
				Type	Low	High	Unit
400041	40	**(0x0028)**	Hours Left in Current Step	R	0	999	hours
400042	41	**(0x0029)**	Minutes/Seconds Left in Current Step	R	*B13	*B13	-
400043	42	**(0x002A)**	Program Wait Status	R	*B14	*B14	-
400044	43	**(0x002B)**	Wait Set Point (Digital Input Number)	R	-32768	32767	-
400045	44	**(0x002C)**	Current Step Jump Step Number	R	1	99	-
400046	45	**(0x002D)**	Current Step Cycles Remaining	R	0	999	-
400047	46	**(0x002E)**	Program Loop 1 Target Set Point	R	-32768	32767	*C1
400048	47	**(0x002F)**	Program Loop 2 Target Set Point	R	-32768	32767	*C1
400049	48	**(0x0030)**	Program Loop 3 Target Set Point	R	-32768	32767	*C1
400050	49	**(0x0031)**		R			
400051	50	**(0x0032)**		R			
400052	51	**(0x0033)**	Last Jump Made from Step	R	1	99	-
400053	52	**(0x0034)**	Last Jump Made to Step	R	1	99	-
400054	53	**(0x0035)**	Total Jumps Made	R	0	32767	-
400055	54	**(0x0036)**	Alarm Reset	R/W	*B15	*B15	-
400056	55	**(0x0037)**	Input Alarm Status	R	*B16	*B16	-
400057	56	**(0x0038)**	Loop/Monitor Alarm Status	R	* B20	*B17	-
400058	57	**(0x0039)**	Chamber Critical Alarm Status	R	*B18	*B18	-
400059	58	**(0x003A)**	Refrigeration Alarm Status	R	*B19	*B19	-
400060	59	**(0x003B)**	System Status Monitor	R	*B20	*B20	-
400061	60	**(0x003C)**	Loop 1 Set Point (SP)	R/W	-32768	32767	*C1
400062	61	**(0x003D)**	Loop 1 Process Value (PV)	R	-32768	32767	*C1
400063	62	**(0x003E)**	Loop 1 Percentage of Output (%Out)	R	-100.00	100.00	*C1
400064	63	**(0x003F)**	Loop 1 Autotune Status	R/W	*B21	*B21	-
400065	64	**(0x0040)**	Loop 1 Upper Set Point Limit	R/W	-32768	32767	*C1
400066	65	**(0x0041)**	Loop 1 Lower Set Point Limit	R/W	-32768	32767	*C1
400067	66	**(0x0042)**	Loop 1 Alarm Type	R/W	*B22	*B22	-
400068	67	**(0x0043)**	Loop 1 Alarm Modes	R/W	*B23	*B23	-
400069	68	**(0x0044)**		R			
400070	69	**(0x0045)**	Loop 1 High Alarm Set Point	R/W	-32768	32767	*C1
400071	70	**(0x0046)**	Loop 1 Low Alarm Set Point	R/W	-32768	32767	*C1
400072	71	**(0x0047)**	Loop 1 Alarm Differential	R/W	0	32767	*C1
400073	72	**(0x0048)**	Loop 2 Set Point (SP)	R/W	-32768	32767	*C1
400074	73	**(0x0049)**	Loop 2 Process Value (PV)	R	-32768	32767	*C1
			Integrated Limit Process Value (PV)	R	-19999	45536	*C2
400075	74	**(0x004A)**	Loop 2 Percentage of Output (%Out)	R	-100.00	100.00	*C1
400076	75	**(0x004B)**	Loop 2 Autotune Status	R/W	*B21	*B21	-
400077	76	**(0x004C)**	Loop 2 Upper Set Point Limit	R/W	-32768	32767	*C1
400078	77	**(0x004D)**	Loop 2 Lower Set Point Limit	R/W	-32768	32767	*C1
400079	78	**(0x004E)**	Loop 2 Alarm Type	R/W	*B22	*B22	-

Modbus Address	Register Number	Parameter Description	Data Type *A	Range *B		*C Unit
				Low	High	
400080	79 **(0x004F)**	Loop 2 Alarm Modes	R/W	*B23	*B23	-
400081	80 **(0x0050)**		R			
400082	81 **(0x0051)**	Loop 2 High Alarm Set Point	R/W	-32768	32767	*C1
400083	82 **(0x0052)**	Loop 2 Low Alarm Set Point	R/W	-32768	32767	*C1
400084	83 **(0x0053)**	Loop 2 Alarm Differential	R/W	0	32767	*C1
400085	84 **(0x0054)**	Loop 3 Set Point (SP)	R/W	-32768	32767	*C1
400086	85 **(0x0055)**	Loop 3 Process Value (PV)	R	-32768	32767	*C1
		Integrated Limit Process Value (PV)	R	-19999	45536	*C2
400087	86 **(0x0056)**	Loop 3 Percentage of Output (%Out)	R	-100.00	100.00	*C1
400088	87 **(0x0057)**	Loop 3 Autotune Status	R/W	*B21	*B21	-
400089	88 **(0x0058)**	Loop 3 Upper Set Point Limit	R/W	-32768	32767	*C1
400090	89 **(0x0059)**	Loop 3 Lower Set Point Limit	R/W	-32768	32767	*C1
400091	90 **(0x005A)**	Loop 3 Alarm Type	R/W	*B22	*B22	-
400092	91 **(0x005B)**	Loop 3 Alarm Modes	R/W	*B23	*B23	-
400093	92 **(0x005C)**		R			
400094	93 **(0x005D)**	Loop 3 High Alarm Set Point	R/W	-32768	32767	*C1
400095	94 **(0x005E)**	Loop 3 Low Alarm Set Point	R/W	-32768	32767	*C1
400096	95 **(0x005F)**	Loop 3 Alarm Differential	R/W	0	32767	*C1
400097	96 **(0x0060)**		R			
400098	97 **(0x0061)**		R			
400099	98 **(0x0062)**		R			
400100	99 **(0x0063)**		R			
400101	100 **(0x0064)**		R			
400102	101 **(0x0065)**		R			
400103	102 **(0x0066)**		R			
400104	103 **(0x0067)**		R			
400105	104 **(0x0068)**		R			
400106	105 **(0x0069)**		R			
400107	106 **(0x006A)**		R			
400108	107 **(0x006B)**		R			
400109	108 **(0x006C)**		R			
400110	109 **(0x006D)**		R			
400111	110 **(0x006E)**		R			
400112	111 **(0x006F)**		R			
400113	112 **(0x0070)**		R			
400114	113 **(0x0071)**		R			
400115	114 **(0x0072)**		R			
400116	115 **(0x0073)**		R			
400117	116 **(0x0074)**		R			
400118	117 **(0x0075)**		R			

Modbus Address	Register Number	Parameter Description	Data Type *A	Range *B		*C
				Low	High	Unit
400119	118 **(0x0076)**		R			
400120	119 **(0x0077)**		R			
400121	120 **(0x0078)**	Monitor 1 Process value (PV)	R	-32768	32767	*C1
400122	121 **(0x0079)**	Monitor 1 Alarm Type	R/W	*B22	*B22	-
400123	122 **(0x007A)**	Monitor 1 Alarm Modes	R/W	*B23	*B23	-
400124	123 **(0x007B)**		R			
400125	124 **(0x007C)**	Monitor 1 High Alarm Set Point	R/W	-32768	32767	*C1
400126	125 **(0x007D)**	Monitor 1 Low Alarm Set Point	R/W	-32768	32767	*C1
400127	126 **(0x007E)**	Monitor 1 Alarm Differential	R/W	0	32767	*C1
400128	127 **(0x007F)**	Monitor 2 Process value (PV)	R	-32768	32767	*C1
400129	128 **(0x0080)**	Monitor 2 Alarm Type	R/W	*B22	*B22	-
400130	129 **(0x0081)**	Monitor 2 Alarm Modes	R/W	*B23	*B23	-
400131	130 **(0x0082)**		R			
400132	131 **(0x0083)**	Monitor 2 High Alarm Set Point	R/W	-32768	32767	*C1
400133	132 **(0x0084)**	Monitor 2 Low Alarm Set Point	R/W	-32768	32767	*C1
400134	133 **(0x0085)**	Monitor 2 Alarm Differential	R/W	0	32767	*C1
400135	134 **(0x0086)**	Monitor 3 Process value (PV)	R	-32768	32767	*C1
400136	135 **(0x0087)**	Monitor 3 Alarm Type	R/W	*B22	*B22	-
400137	136 **(0x0088)**	Monitor 3 Alarm Modes	R/W	*B23	*B23	-
400138	137 **(0x0089)**		R			
400139	138 **(0x008A)**	Monitor 3 High Alarm Set Point	R/W	-32768	32767	*C1
400140	139 **(0x008B)**	Monitor 3 Low Alarm Set Point	R/W	-32768	32767	*C1
400141	140 **(0x008C)**	Monitor 3 Alarm Differential	R/W	0	32767	*C1
400142	141 **(0x008D)**	Monitor 4 Process value (PV)	R	-32768	32767	*C1
400143	142 **(0x008E)**	Monitor 4 Alarm Type	R/W	*B22	*B22	-
400144	143 **(0x008F)**	Monitor 4 Alarm Modes	R/W	*B23	*B23	-
400145	144 **(0x0090)**		R			
400146	145 **(0x0091)**	Monitor 4 High Alarm Set Point	R/W	-32768	32767	*C1
400147	146 **(0x0092)**	Monitor 4 Low Alarm Set Point	R/W	-32768	32767	*C1
400148	147 **(0x0093)**	Monitor 4 Alarm Differential	R/W	0	32767	*C1
400149	148 **(0x0094)**	Monitor 5 Process value (PV)	R	-32768	32767	*C1
400150	149 **(0x0095)**	Monitor 5 Alarm Type	R/W	*B22	*B22	-
400151	150 **(0x0096)**	Monitor 5 Alarm Modes	R/W	*B23	*B23	-
400152	151 **(0x0097)**		R			
400153	152 **(0x0098)**	Monitor 5 High Alarm Set Point	R/W	-32768	32767	*C1
400154	153 **(0x0099)**	Monitor 5 Low Alarm Set Point	R/W	-32768	32767	*C1
400155	154 **(0x009A)**	Monitor 5 Alarm Differential	R/W	0	32767	*C1
400156	155 **(0x009B)**	Monitor 6 Process value (PV)	R	-32768	32767	*C1
400157	156 **(0x009C)**	Monitor 6 Alarm Type	R/W	*B22	*B22	-
400158	157 **(0x009D)**	Monitor 6 Alarm Modes	R/W	*B23	*B23	-

Modbus Address	Register Number	Parameter Description	Data *A Type	Range *B		*C
				Low	High	Unit
400159	158 **(0x009E)**		R			
400160	159 **(0x009F)**	Monitor 6 High Alarm Set Point	R/W	-32768	32767	*C1
400161	160 **(0x00A0)**	Monitor 6 Low Alarm Set Point	R/W	-32768	32767	*C1
400162	161 **(0x00A1)**	Monitor 6 Alarm Differential	R/W	0.0	32767	*C1
400163	162 **(0x00A2)**	Monitor 7 Process value (PV)	R	-32768	32767	*C1
400164	163 **(0x00A3)**	Monitor 7 Alarm Type	R/W	*B22	*B22	-
400165	164 **(0x00A4)**	Monitor 7 Alarm Modes	R/W	*B23	*B23	-
400166	165 **(0x00A5)**		R			
400167	166 **(0x00A6)**	Monitor 7 High Alarm Set Point	R/W	-32768	32767	*C1
400168	167 **(0x00A7)**	Monitor 7 Low Alarm Set Point	R/W	-32768	32767	*C1
400169	168 **(0x00A8)**	Monitor 7 Alarm Differential	R/W	0.0	32767	*C1
400170	169 **(0x00A9)**	Monitor 8 Process value (PV)	R	-32768	32767	*C1
400171	170 **(0x00AA)**	Monitor 8 Alarm Type	R/W	*B22	*B22	-
400172	171 **(0x00AB)**	Monitor 8 Alarm Modes	R/W	*B23	*B23	-
400173	172 **(0x00AC)**		R			
400174	173 **(0x00AD)**	Monitor 8 High Alarm Set Point	R/W	-32768	32767	*C1
400175	174 **(0x00AE)**	Monitor 8 Low Alarm Set Point	R/W	-32768	32767	*C1
400176	175 **(0x00AF)**	Monitor 8 Alarm Differential	R/W	0.0	32767	*C1
400177	176 **(0x00B0)**		R			
400178	177 **(0x00B1)**	Expansion Module Input Status	R	*B26	*B26	-
400179	178 **(0x00B2)**	Expansion Module Output Status	R	*B27	*B27	-
400180	179 **(0x00B3)**	Program Step Time Addition	R/W	0	32767	minutes
400181	180 **(0x00B4)**	EZT-570L Program Download/Offline	R	*B24	*B24	-

Legend:

*A R/W Specifies readable / writable data, R specifies read only data and W specifies a write only control value.

*B The range of certain parameters is dependent upon system options. Consult the following range tables for information regarding the use of these parameters.

Reading bit-oriented parameters

The value contained in these parameters is dependent upon the combination of “on” bits (1). Therefore, only the individual status of each bit has meaning, not the value of the parameter.

Setting bit oriented parameters

The value that must be written to these parameters is dependent upon the combination of “on” bits. Therefore, it is necessary to know the current value of the parameter before setting it so that only the bit status you want to update is changed. Otherwise, sending a value derived from only the bit you wish to set will turn off all other functions related to the other bits in the parameter.

*B1

Parameter Value	Description
Bit0	EZT-570L Online
Bit1 - Bit15	Not Assigned

*B2

Parameter Value	Range Low	Range High	Description
High Byte	0	99	Year
Low Byte	1	12	Month

*B3

Parameter Value	Range Low	Range High	Description
High Byte	1	31	Day
Low Byte	0	6	Day of Week**

**The days of the week are represented as numbers:
0=Sun, 1=Mon, 2=Tue, 3=Wed, 4=Thu, 5=Fri, 6=Sat

*B4

Parameter Value	Range Low	Range High	Description
High Byte	0	23	Hour
Low Byte	0	59	Minutes

Example

Read of registers 1 to 4 for the current time from the EZT-570L returns the following values:

Register Values:	0x0A 0B	0x04 04	0x0A 1D	0x00 20
Decimal Equivalent:	10 11	4 4	10 29	32

Translating the values into an actual date and time provides a date and time of Thursday November 4, 2010 at 10:29:32 am.

*B5

Parameter Value	Description
0	Continue
1	Hold
2	Off
4	Start Over
8	Resume

*B6

Parameter Value	Description
0	Disabled
1	Manual
2	Auto

*B7

Parameter Value	Description
0	Inactive
1	In Defrost
2	In Prechill

*B8

Parameter Value	Description
0	Off
1	Deviation
2	Process
5	Deviation with Event Enable/Disable
6	Process with Event Enable/Disable

*B9

Parameter Value	Description
Bit0	Event 1
Bit1	Event 2
Bit2	Event 3
Bit3	Event 4
Bit4	Event 5
Bit5	Event 6
Bit6	Event 7
Bit7	Event 8
Bit8	Event 9
Bit9	Event 10
Bit10	Event 11
Bit11	Event 12
Bit12	Event 13
Bit13	Event 14
Bit14	Event 15
Bit15	Event 16

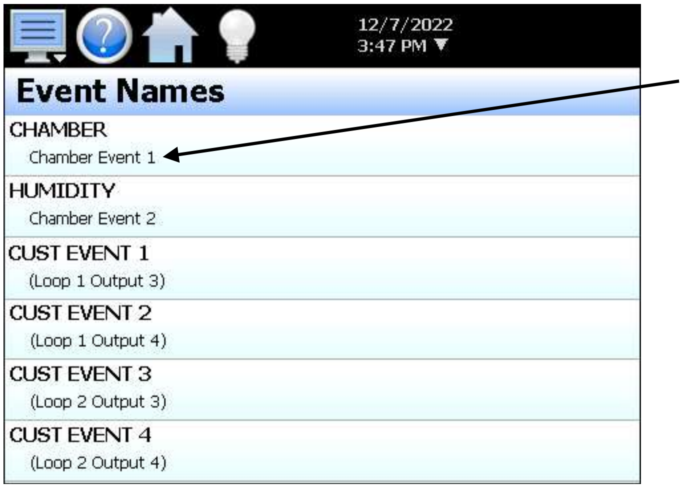
Setting the individual bits in the word turns on (1) or turns off (0) the event. If an event is for controlling an option that is not available on your chamber, the associated bit should be set off.

Depending upon the EZT-570L configuration and chamber model, some events may be defined as “dedicated chamber events”. This can be determined by going to the Event Names screen under Device Settings and checking the event descriptions shown under each event name. If listed as “Chamber Event #”, they are dedicated chamber events and must be set in the first event register (400023).

Any optional customer events are then set in sequential order in the second event register (400024).

If the event descriptions contain only the loop or expansion output assignment numbers (no chamber event # descriptions), then the events are in sequential order as shown on the Overview screen and are set in the first event register (400023) only. The second event register (400024) is not used.

Example 1
Turn on the chamber and customer events 2 and 3 on an EZT-570L with dedicated chamber events.



12/7/2022
3:47 PM ▼

Event Names

CHAMBER
Chamber Event 1

HUMIDITY
Chamber Event 2

CUST EVENT 1
(Loop 1 Output 3)

CUST EVENT 2
(Loop 1 Output 4)

CUST EVENT 3
(Loop 2 Output 3)

CUST EVENT 4
(Loop 2 Output 4)

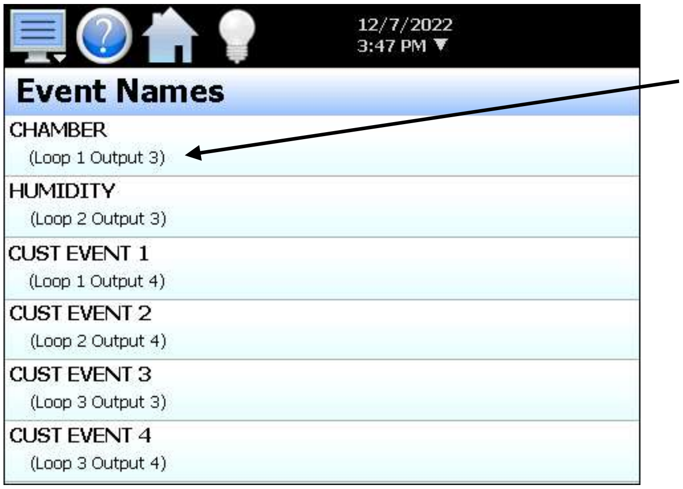
When the system is configured with dedicated chamber events, one or more events will be shown on the Event Names screen under Device Settings with the text "Chamber Event #".

In this case, all available chamber events will be associated with the first chamber event register. All customer events are then associated with the second customer event register.

According to the event table at the beginning of this section, the dedicated chamber event is event 1. The bit number for event 1 is zero, thus the bit at position (index) zero of the word should be set. The bit values of the word then become: 00000000 00000001. The decimal equivalent of the binary array is 1 (0x0001). By setting register 22 (Modbus address 400023) to a value of 1, the chamber will turn on.

The customer events are then set in the second event register. By comparing the event numbers to their bit positions, set the bits in the word accordingly: 0000000000000110. The decimal equivalent is 6 (0x0006). Setting register 23 (Modbus address 400024) to a value of 6, will turn on customer events 2 and 3.

Example 2
Turn on the chamber and customer events 2 and 3 on an EZT-570L without dedicated chamber events.



12/7/2022
3:47 PM ▼

Event Names

CHAMBER
(Loop 1 Output 3)

HUMIDITY
(Loop 2 Output 3)

CUST EVENT 1
(Loop 1 Output 4)

CUST EVENT 2
(Loop 2 Output 4)

CUST EVENT 3
(Loop 3 Output 3)

CUST EVENT 4
(Loop 3 Output 4)

When the system is NOT configured with dedicated chamber events, all of the events will be will be shown on the Event Names screen under Device Settings with text contained in parenthesis indicating the loop or expansion output they are assigned to.

In this case, all available events will be associated with the first chamber event register only, and will be in sequential order.

By assigning the event numbers to the events sequentially down the list, the chamber event is event 1 and the customer events 2 and 3 are events 4 and 5 in the event list. The bit values of the word then become: 00000000 00011001. The decimal equivalent of the binary array is 25 (0x0019). By setting register 22 (Modbus address 400023) to a value of 25, the chamber and customer events 2 and 3 will be turned on.

*B10

Parameter Value	Description
0	Stop/Off
1	Stop/All Off
2	Hold
4	Run/Resume
8	Autostart**
16	Wait **
32	Ramp**
64	Soak**
128	Guaranteed Soak**

**These values are set by the EZT-570L to indicate the operating status of the program and cannot be set directly.

*B11

Parameter Value	Description
1	Program Advance to Previous Step
2	Program Advance to Next Step

This parameter only performs its function when the program is in hold. Once the set function is executed, this parameter automatically resets to zero (0).

*B12

Parameter Value	High Order Byte	Low Order Byte	Description
Range Low	32	32	Program Name Character (ASCII Table)
Range High	126	126	Program Name Character (ASCII Table)

See the ASCII character chart in Section 1.1 for the character representation of these values.

Example

Read command of registers 26 to 30 from the EZT-570L returns the following values:

Register Values:	0x74 53	0x72 6F	0x20 65	0x65 54	0x74 73
ASCII Equivalent:	t S	r o	e	e T	t s

Assemble the ASCII characters in order from low to high byte starting with register 26 (Modbus address 400027) in order to assemble the Program name "Store Test". Note that null characters are not used in the Program name. A space (0x20) will be used in place of a null character to maintain the 10 character name length if the Program name is not ten characters long.

*B13

Parameter Value	Range Low	Range High	Description
High Byte	0	59	Minutes
Low Byte	0	59	Seconds

*B14

Parameter Value	Description
0	Not Waiting
1	Input 1
2	Input 2
4	Input 3
8	Input 4
16	Input 5
32	Input 6
64	Input 7
128	Input 8
256	Input 9
512	Input 10
1024	Input 11
2048	Input 12
4096	Input 13

*B15

Parameter Value	Description
1	Alarm Reset (Silence)
2	Pumpdown Reset

Once the set function is executed, this parameter automatically resets to zero (0).

*B16

Parameter Value	Description
Bit0	Input 1 Sensor Break
Bit1	Input 2 Sensor Break
Bit2	Input 3 Sensor Break
Bit3	Input 4 Sensor Break
Bit4	Input 5 Sensor Break
Bit5	Input 6 Sensor Break
Bit6	Input 7 Sensor Break
Bit7	Input 8 Sensor Break
Bit8	Input 9 Sensor Break
Bit9	Input 10 Sensor Break
Bit10	Input 11 Sensor Break
Bit11	Input 12 Sensor Break
Bit12	Input 13 Sensor Break
Bit13	Not Assigned
Bit14	Communications Failure
Bit15	Not Assigned

The individual bits of the word indicate specific alarm conditions. When the bit is on (1) the alarm is present. More than one alarm can be present at a time.

*B17

Parameter Value	Description
Bit0	Input 1 Alarm
Bit1	Input 2 Alarm
Bit2	Input 3 Alarm
Bit3	Input 4 Alarm
Bit4	Input 5 Alarm
Bit5	Input 6 Alarm
Bit6	Input 7 Alarm
Bit7	Input 8 Alarm
Bit8	Input 9 Alarm
Bit9	Input 10 Alarm
Bit10	Input 11 Alarm
Bit11	Input 12 Alarm
Bit12	Input 13 Alarm
Bit13-15	Not Assigned

The individual bits of the word indicate specific alarm conditions. When the bit is on (1) the alarm is present. More than one alarm can be present at a time.

*B18

Parameter Value	Description
Bit0	Chamber High Limit
Bit1	External Product Safety
Bit2	Boiler Over Temperature
Bit3	Boiler Low Water
Bit4	Not Assigned
Bit5	Motor Overload
Bit6	Not Assigned
Bit7	Not Assigned
Bit8	Not Assigned
Bit9	Door Open
Bit10	Not Assigned
Bit11	Not Assigned
Bit12	Emergency Stop
Bit13	Power Failure
Bit14-15	Not Assigned

The individual bits of the word indicate specific alarm conditions. When the bit is on (1) the alarm is present. More than one alarm can be present at a time.

*B19

Parameter Value	Description
Bit0	System 1 High/Low Pressure
Bit1	System 1 Oil Pressure/Protection Module
Bit2	System 1 Discharge Temperature
Bit3	Not Assigned
Bit4	Pumpdown Disabled
Bit5	Not Assigned
Bit6	Not Assigned
Bit7	Not Assigned
Bit8	System 2 High/Low Pressure
Bit9	System 2 Oil Pressure/Protection Module
Bit10	System 2 Discharge Temperature
Bit11-15	Not Assigned

The individual bits of the word indicate specific alarm conditions. When the bit is on (1) the alarm is present. More than one alarm can be present at a time.

*B20

Parameter Value	Description
Bit0	Humidity Water Reservoir Low
Bit1	Humidity Disabled (temp out-of-range)
Bit2	Humidity High Dewpoint Limit
Bit3	Humidity Low Dewpoint Limit
Bit4	Door Open
Bit5-15	Not Assigned

The individual bits of the word indicate specific alarm conditions. When the bit is on (1) the alarm is present. More than one alarm can be present at a time.

*B21

Parameter Value	Description
0	Autotune Off**
1	Start Autotune
2	Autotune in Progress**
4	Cancel Autotune

**These values are set by the EZT-570L to indicate the autotune status and cannot be set directly.

*B22

Parameter Value	Description
3	Absolute High
5	Absolute Low
7	Absolute Both
24	Deviation High
40	Deviation Low
56	Deviation Both

For monitor input alarms, only the absolute high, low or both selections are valid. Monitor inputs are not associated with set points so the deviation alarm modes cannot be used for the monitor input alarms. Deviation mode selections are only valid for loop alarms.

*B23

Parameter Value	Description
Bit0	Latching Alarm
Bit1-3	Not Assigned
Bit4	Audible Alarm On (Silent Off)
Bit5	Shut Down on Alarm
Bit6-15	Not Assigned

Only the bits listed perform the control actions as specified. The state of the other bits does not affect alarm operation.

*B24

Parameter Value	Description
0	Online
1	Offline/Downloading Program

When the EZT-570L is offline or downloading a program, refrain from writing to any control registers. In offline mode, there are no updates made to any control registers.

*C1 The units of measure and range of a loop or monitor input is dependent upon the configuration of the input and/or the units of temperature selection (Celsius or Fahrenheit) of the EZT-570L. The decimal point position for the loop or monitor input is an implied value. Thus, depending upon the decimal resolution set at the EZT-570L, a register value of 345 could represent an actual process value of 345 if the decimal resolution is zero, or a value of 34.5 if the decimal resolution is one.

CAUTION: *When setting set points and alarm values for loops and monitor points, it is imperative that the value sent is scaled properly according to the decimal resolution set in the EZT-570L.*

Example:
To set a set point of 40 degrees with temperature scaled for one decimal resolution, multiply the set point by a factor of 10 to offset the value for one decimal position, i.e., 40 (or 40.0) X 10 = 400. When 400 is written to register 60 (Modbus address 400061) for loop 1, the EZT-570L will scale it with the assumed decimal point for a set point of 40. If 40 was written to the EZT-570L, it would scale it to a value of 4.0 according to the assumed decimal resolution it is set for.

*C2 The units of measure and range of the integrated limit control is dependent upon the input type configured. For temperature inputs (TC or RTD) the PV is always scaled by one decimal regardless of the resolution displayed on the EZT-570L. For linear input types (Vdc or mA), the PV is scaled by the implied decimal resolution shown on the EZT-570L. The value read from the EZT-570L is also offset by a value of -19999, so to convert to an actual process value, you must first subtract 19999 from the value read from the EZT-570L and then scale it by the implied decimal resolution.

Example:
The value read from the EZT-570L for the integrated limit process value is an unsigned 16-bit integer with a value between 0 and 65535. Thus, if a value of 20999 is read from the EZT-570L, you must first subtract 19999 from the value. Therefore, 20999 – 19999 = 1000. With the implied decimal resolution of a temperature input for the limit, 1000 / 10 = 100.0 degrees for the actual PV.

4.3 EZT-570L Automatic Ramp/Soak Program Registers

The program parameters are a separate group of registers that are used for sending ramp/soak programs to the EZT-570L. The manner in which the program steps are sent to the EZT-570L is specific and must be followed exactly.

Each program step consists of 15 data registers. Programs must be written one step at a time, using a multiple write command (0x10) to write the data for all 15 registers at once. This allows programs to be stored as two-dimensional arrays, of which code can be written to simply index through the array step-by-step, and transmit the program to the EZT-570L.

The first 15 registers of the Program contain specific settings related to the program. These include AutoStart settings, the program name, the length of the program (number of steps), and guaranteed soak band settings. These values are always transmitted as the first "step" of the program.

Modbus Address	Register Number		Parameter Description	Data *D Type	Range *E		*F Unit
					Low	High	
400201	200	**(0x00C8)**	Autostart On/Off	W	*E1	*E1	-
400202	201	**(0x00C9)**	Year/Month for Autostart	W	*E2	*E2	-
400203	202	**(0x00CA)**	Day/DOW for Autotstart	W	*E3	*E3	-
400204	203	**(0x00CB)**	Time of Day for Autostart	W	*E4	*E4	-
400205	204	**(0x00CC)**	Program Name (Chars 1 & 2)	W	*E5	*E5	-
400206	205	**(0x00CD)**	Program Name (Chars 3 & 4)	W	*E5	*E5	-
400207	206	**(0x00CE)**	Program Name (Chars 5 & 6)	W	*E5	*E5	-
400208	207	**(0x00CF)**	Program Name (Chars 7 & 8)	W	*E5	*E5	-
400209	208	**(0x00D0)**	Program Name (Chars 9 & 10)	W	*E5	*E5	-
400210	209	**(0x00D1)**	Total Number of Steps in Program	W	1	99	-
400211	210	**(0x00D2)**	Guaranteed Soak Band Loop 1	W	0	32767	PV
400212	211	**(0x00D3)**	Guaranteed Soak Band Loop 2	W	0	32767	PV
400213	212	**(0x00D4)**	Guaranteed Soak Band Loop 3	W	0	32767	PV
400214	213	**(0x00D5)**	(not currently used – can be left at 0)	W	-32768	32767	-
400215	214	**(0x00D6)**	(not currently used – can be left at 0)	W	-32768	32767	-

The following 15 registers of the Program contain the data for step 1 of the Program.

Modbus Address	Register Number	Parameter Description	Data *D	Range *E		*F
			Type	Low	High	Unit
400216	215 **(0x00D7)**	Step Time Hours	W	0	9999	-
400217	216 **(0x00D8)**	Step Time Minutes/Seconds	W	*E6	*E6	-
400218	217 **(0x00D9)**	Chamber Events	W	*E7	*E7	-
400219	218 **(0x00DA)**	Customer Events	W	*E7	*E7	-
400220	219 **(0x00DB)**	Guaranteed Soak/Wait for Digital	W	*E8	*E8	-
400221	220 **(0x00DC)**	Wait for Loop	W	*E9	*E9	-
400222	221 **(0x00DD)**	Wait for Monitor	W	*E10	*E10	-
400223	222 **(0x00DE)**	Wait for Loop/Monitor Set Point	W	-32768	32767	-
400224	223 **(0x00DF)**	Jump Step	W	1	99	-
400225	224 **(0x00E0)**	Cycle Count	W	0	999	-
400226	225 **(0x00E1)**	Loop 1 Set Point	W	-32768	32767	PV
400227	226 **(0x00E2)**	Loop 2 Set Point	W	-32768	32767	PV
400228	227 **(0x00E3)**	Loop 3 Set Point	W	-32768	32767	PV
400229	228 **(0x00E4)**	(not currently used – can be left at 0)	W	-3276.8	3276.7	-
400230	229 **(0x00E5)**	(not currently used – can be left at 0)	W	-3276.8	3276.7	-

All remaining steps of the Program follow the same format and data structure as is represented for step one above. Up to the following 1470 registers are used to contain the additional step data of the Program as required for steps 2 through 99. Since few if any programs will contain the maximum of 99 steps, it is only necessary to write the step data for the number steps used in the Program.

Modbus Addresses	**Register Numbers**	
400231 – 400245	230 (0x00E6) – 244 (0x00F4)	Program Step 2 Data Registers
400236 – 400260	245 (0x00F5) – 259 (0x0103)	Program Step 3 Data Registers
400261 – 400275	260 (0x0104) – 274 (0x0112)	Program Step 4 Data Registers
400276 – 400290	275 (0x0113) – 289 (0x0121)	Program Step 5 Data Registers
400291 – 400305	290 (0x0122) – 304 (0x0130)	Program Step 6 Data Registers
400306 – 400320	305 (0x0131) – 319 (0x013F)	Program Step 7 Data Registers
400321 – 400335	320 (0x0140) – 334 (0x014E)	Program Step 8 Data Registers
400336 – 400350	335 (0x014F) – 349 (0x015D)	Program Step 9 Data Registers
Through		
401686 – 401700	1685 (0x0695) – 1699 (0x06A3)	Program Step 99 Data Registers

Legend:

*D W Specifies writable data.

*E1

Parameter Value	Description
0	Autostart Off
1	Autostart by Date
2	Autostart by Day

*E2 See *B2 in Section 4.2 for information on the range of this parameter.

*E3 See *B3 in Section 4.2 for information on the range of this parameter.

*E4 See *B4 in Section 4.2 for information on the range of this parameter.

*E5 These parameters contain data which represent up to ten ASCII characters in order to display the name of the currently loaded (or operating) program in the EZT-570L.

See *B12 in Section 4.2 for information on the range of these parameters.

*E6

Parameter Value	Range Low	Range High	Description
High Byte	0	59	Minutes
Low Byte	0	59	Seconds

*E7 See *B9 in Section 4.2 for information on the range of this parameters.

NOTE: The customer events register is only used for customer events if dedicated chamber events are configured on the EZT-570L.

*E8

Parameter Value	Description
Bit0	Guaranteed Soak Loop 1
Bit1	Guaranteed Soak Loop 2
Bit2	Guaranteed Soak Loop 3
Bit3-15	Not Assigned

If an event is for controlling an option not available on your chamber, the associated bit should be set to zero.

Multiple guaranteed soak events can be enabled at a time. The guaranteed soak and "wait for" events can be used concurrently on the same step.

*E9

Parameter Value	Description
0	Wait for Disabled (no loop selected)
1	Wait For Loop 1
2	Wait For Loop 2
4	Wait For Loop 3

If an event is for controlling an option not available on your chamber, the associated bit should be set to zero.

Only one "wait for" event can be used concurrently on the same step. The "wait for" and guaranteed soak events can be used concurrently on the same step.

*E10

Parameter Value	Description
0	Wait for Disabled (no input selected)
1	Wait For Monitor 1
2	Wait For Monitor 2
4	Wait For Monitor 3
8	Wait For Monitor 4
16	Wait For Monitor 5
32	Wait For Monitor 6
64	Wait For Monitor 7
128	Wait For Monitor 8

4.3.1 Sending a Ramp/Soak Program to the EZT-570L

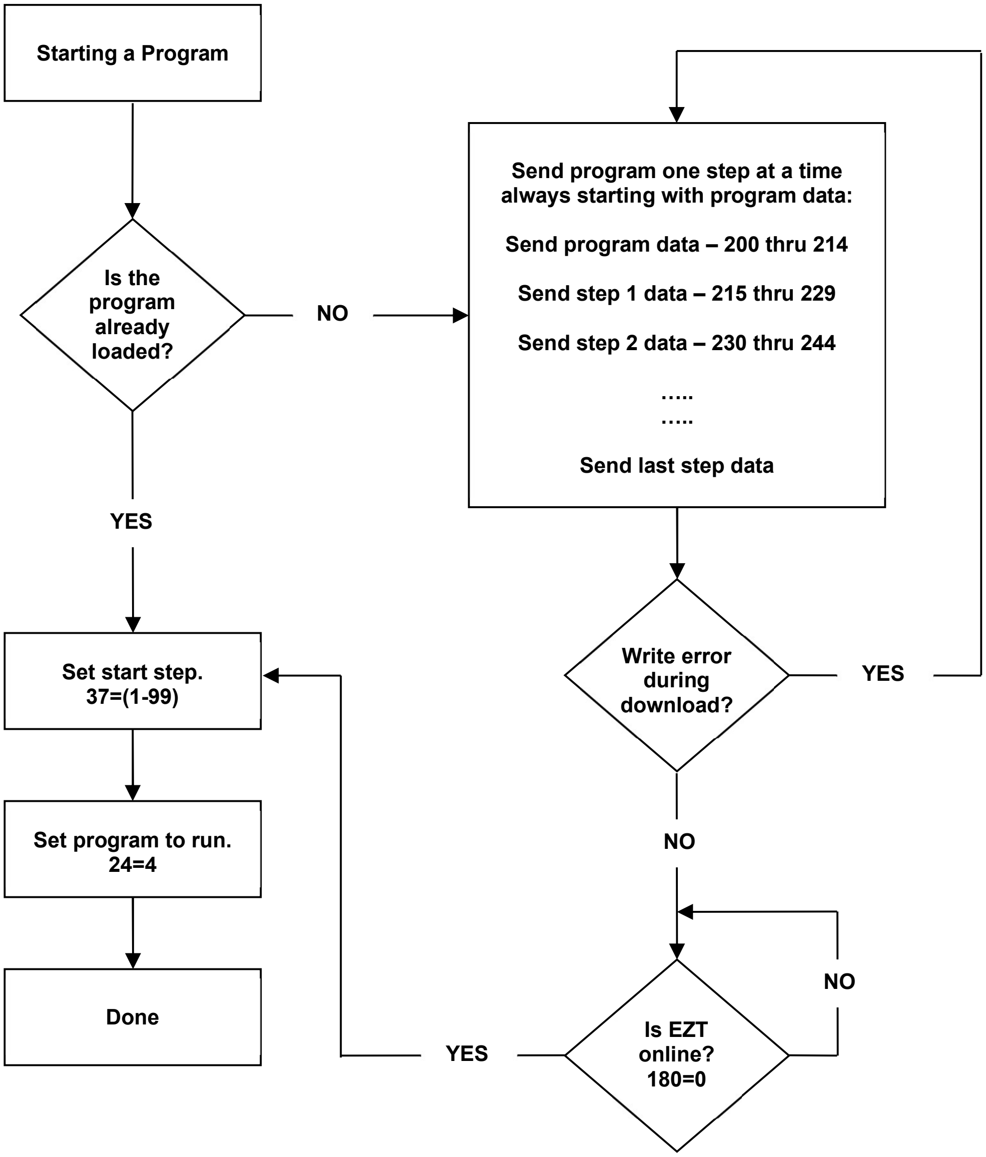
Programs are sent to the EZT in a step-by-step process. The download sequence must be followed in order and must complete without errors to be valid. If a write error is detected during the transfer of a program from a PC to the EZT (no response from EZT or NACK returned), the program download must be aborted and restarted.

The EZT-570L is put into program transfer mode when the first group of registers containing the program specific data is sent (registers 200-214). The EZT then begins looking for the number of steps of the program to be sent as was set in register 209. As each step is received, it increments the count. Once all steps have been received, the EZT transfers the program into internal memory. During this transfer, register 180 will be set to 1 to indicate that the process is taking place. Once the register value returns to zero, the program is ready to be started.

IMPORTANT: *If a program is stored on the EZT-570L with the same name as set in the program header data of the program downloaded, it will overwrite the existing program on the EZT-570L unless that program is also currently running.*

If the program currently operating in the EZT-570L has the same name as the downloaded program, the downloaded program will be saved to the EZT-570L under the name "pcDownload". It can then be opened as saved under another name at the EZT-570L.

4.3.2 Starting a Ramp/Soak Program in the EZT-570L


Starting a Program
Is the program already loaded?
NO
YES
Send program one step at a time always starting with program data:
Send program data – 200 thru 214
Send step 1 data – 215 thru 229
Send step 2 data – 230 thru 244
…..
…..
Send last step data
Write error during download?
YES
NO
Set start step. 37=(1-99)
Set program to run. 24=4
Done
Is EZT online? 180=0
NO
YES