ADVANCED COMMUNICATIONS PACKAGE

LME & LU7 USER’S MANUAL

VERSION 2.11
MAY 26, 2016

PATLITE (USA) CORPORATION
20130 S WESTERN AVE.
TORRANCE, CA 90501
PHONE: +1-310-328-3222
FAX: +1-310-328-2676

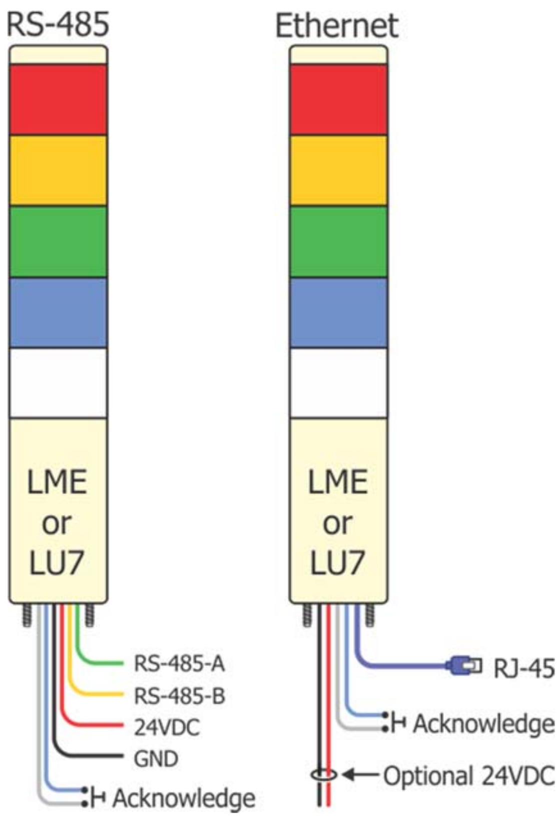
Table of Contents

Revision History

Name	Date	Reason For Changes	Version
W. Petroff	03-JAN-16	Initial Release	1.00
W. Petroff	08-MAY-16	General Updates and Installation Guide Added	2.00
W. Petroff	23-MAY-16	Added Instructions for new PA Dashboard software	2.10
W. Petroff	26-MAY-16	Added Documentation for Modbus Command 16 (0x10)	2.11
S. Khurana	14-JUNE-16	Spelling and Grammar Check	2.12

INTRODUCTION

This document provides detailed information for the installation, configuration and operation of the Advanced Communications Controllers which can be installed in Patlite's popular LME and LU7 light towers.



Two types of controllers are available. One communicates via Standard TCP/IP over Ethernet and the other uses an RS-485 balanced multi-drop network. The model numbers for these controllers are PA-06 for Ethernet and PA-05 for RS-485.

Various commands that control the light towers are generated by the host PC or PLC and sent to the towers over the network connections. The formats of the commands are the same for both Ethernet and RS-485 methods. The details of the commands and the message formats are provided later in this manual.

The Ethernet tower is normally powered via Power over Ethernet (PoE). It adheres to the 802.3af standard version "B," where pins 4 & 5 have +24VDC power applied and pins 7 & 8 provide the power return. This differs significantly from the normal 802.3at standard which is normally powered by 45V. (See more details on this subject in the section on PoE.)

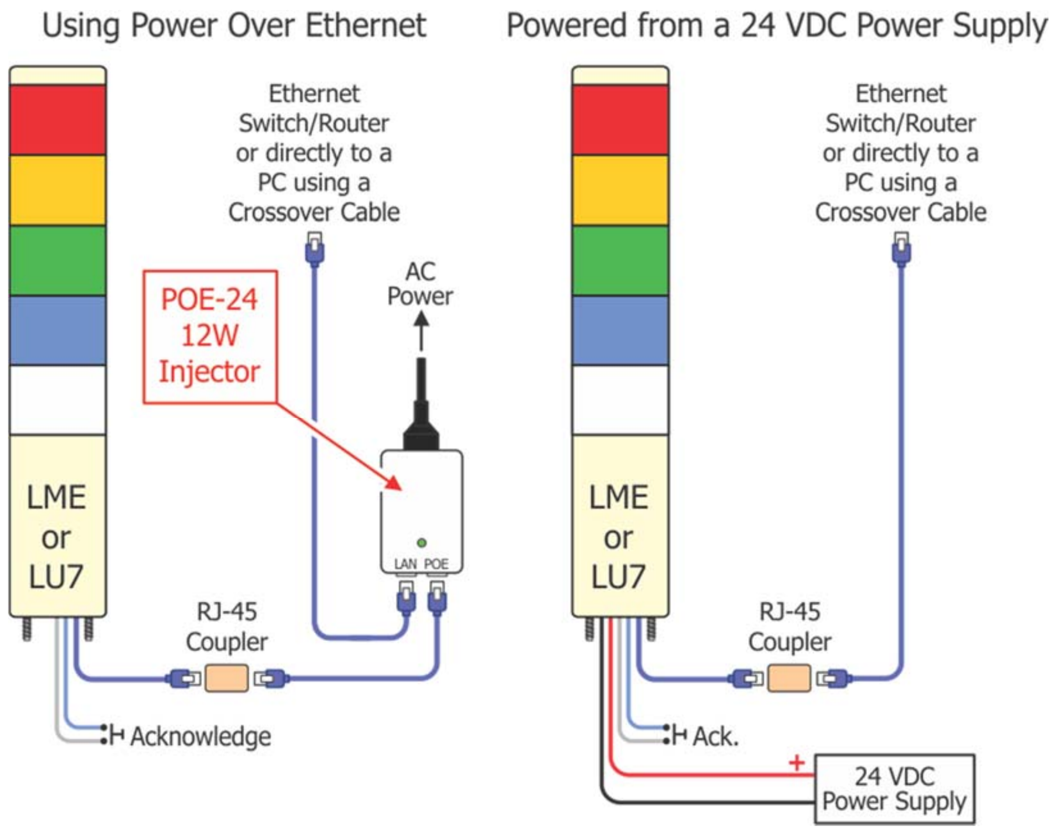
Two communications protocols are used to control the operation of the light towers. One is a sequence of ASCII characters that begins with a special character ($ or #) followed by a specific command character and terminated with a carriage return character (CR). This protocol is described in detail in the section marked ASCII COMMUNICATIONS.

The other protocol is Modbus. This is the easiest to implement and most widely used protocol in existence today. Instead of using a variety of commands to perform specific functions, this protocol uses just a few functions to read and write values from and to "registers." The registers are numbered from 0 to 25 (18 hex) and contain the setting or the return value for a specific function of the light tower.

ACKNOWLEDGE PUSHBUTTON

The towers for both communications packages contain audible alarms. A pair of wires that can be attached to a simple pushbutton switch can be used to silence the alarm. **<u>The pushbutton is not provided</u>**, but almost any SPST normally open momentary switch will work. Since logic levels are being controlled, a pushbutton with gold contacts is preferred.

EHERNET TOWER WIRING



POWER OVER ETHERNET (POE)

Two standards for Power over Ethernet are currently defined. One meets the IEEE 802.3af-2003 specification which is in wide use today. The other is a derivative of the original specification to become IEEE 802.3at. The Ethernet Advanced Communications Controller (ACC) described in this manual uses 802.3at Mode B for POE and supports 10BASE-T and 100BASE-TX communications.

Pin	Cable		Function
1	White with Orange Stripe		Rx+
2	Solid Orange		Rx-
3	White with Green Stripe		Tx+
4	Solid Blue		Power+
5	White with Blue Stripe		
6	Solid Green		Tx-
7	White with Brown Stripe		Power-
8	Solid Brown		

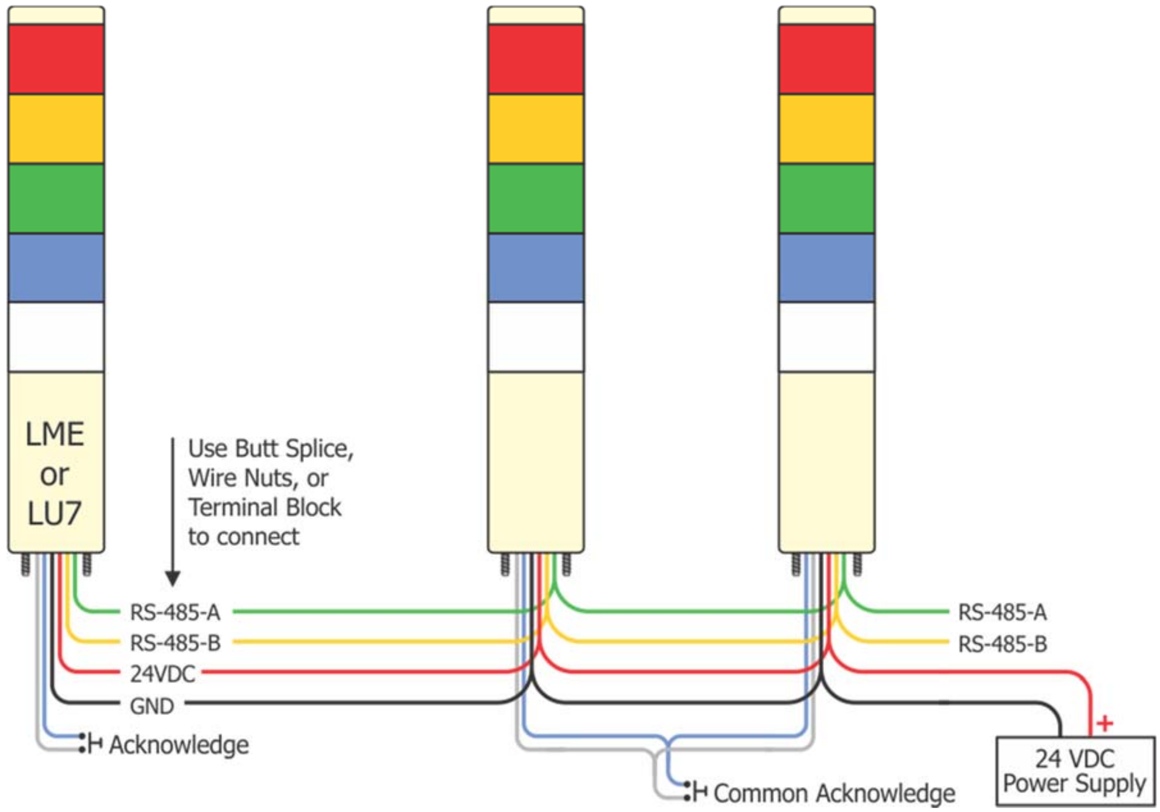
When POW is used, a power injector capable of supplying 24 VDC with a pin-out shown in the table to the left must be inserted between the light tower and the Ethernet Switch as shown in the diagram in the previous section.

Notice that 4-wires are used to carry the DC power. Consequently Gigabit Ethernet

(1000BASE-T) is not supported because it uses all cable conductors to carry data.

IMPORTANT: Use only special Ethernet crossover cables with the POE powered light towers so the polarity of the DC supply on pins 4-5 and 7-8 are not reversed.

RS-485 WIRING



Multiple ACC RS-485 equipped LME and LU7 light towers can communicate with a computer or PLC over a common 4-wire network. Two of the wires are used for data and the other pair used for 24 VDC power. Since multiple towers operate over the same pair of wires, each has a unique address which is set using the configuration utility described later in this document.

CONNECTING THE OPTIONAL ACKNOWLEDGE PUSHBUTTON SWITCH

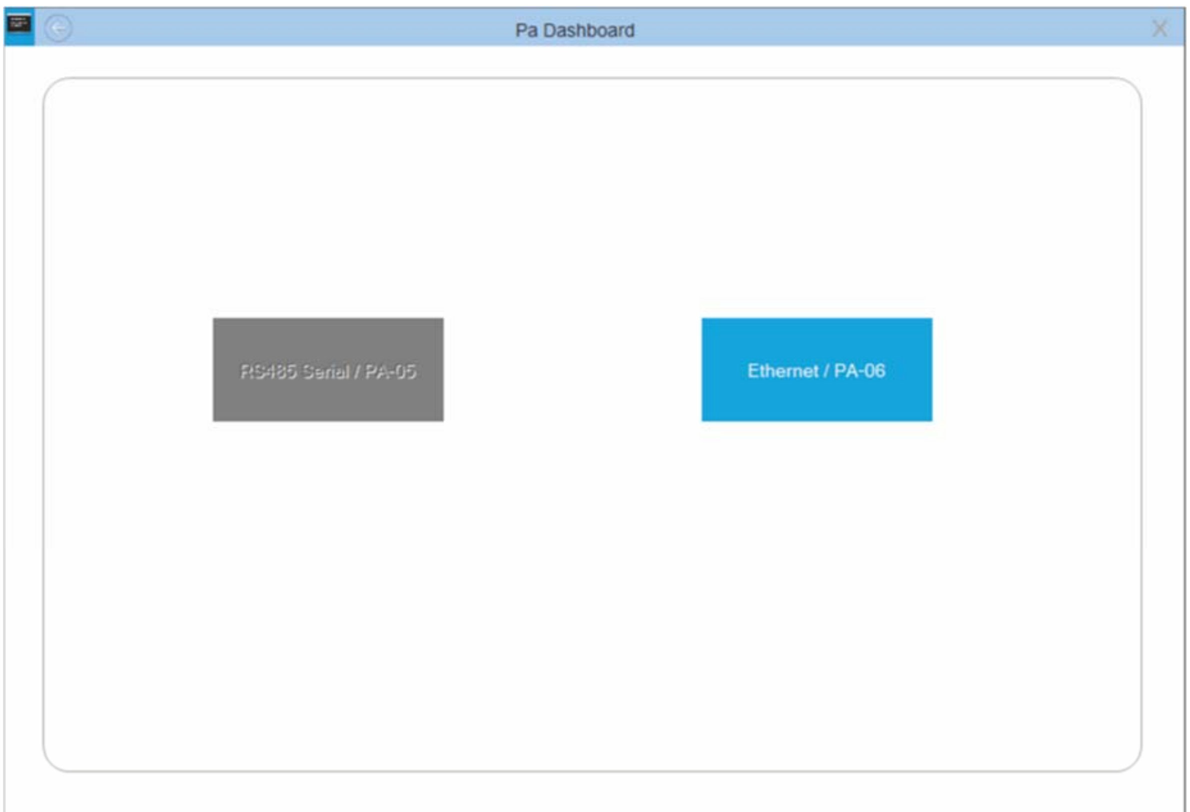
An audible alarm is available in LME or LU7 light towers. The alarm can be turned on or off and the volume adjusted using commands sent to the tower over the communications network. Once set, the audible alarm will continue to sound until it is turned off by a command or by using a normally open pushbutton switch connected to the "Acknowledge" inputs. Operation of the pushbutton switch is described later in this manual.

ACC CONFIGURATION UTILITY

The Advanced Communications Controller configuration utility software is available on Patlite's website (http://www.patlite.com). The package can be easily downloaded and installed on any modern Windows computer.

PREPARATION

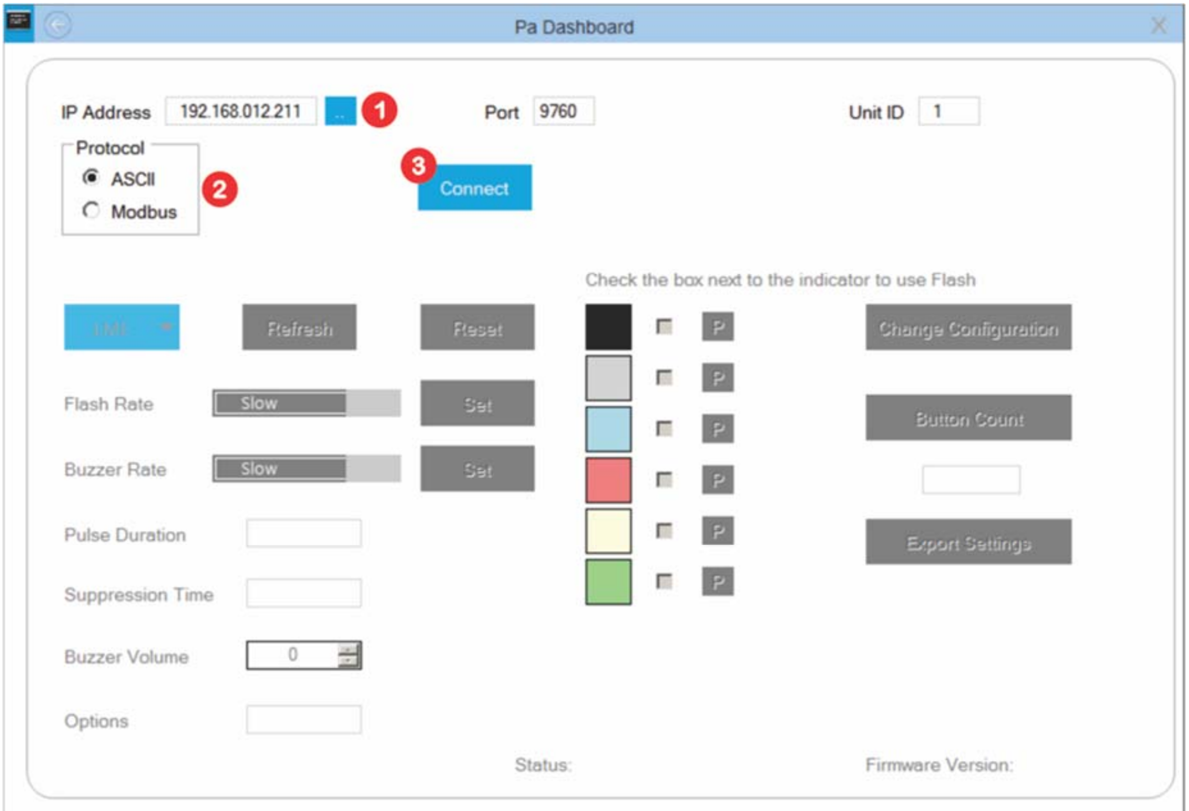
The configuration utility supports both Ethernet and RS-485 based towers. Make sure the utility is installed on a suitable Windows computer. Next, connect an LME or LU7 light tower to the computer as described in the previous sections. Once this has been completed, start the program and select the type of communications PA-05 RS-485 Serial or PA-06 Ethernet.



Then, follow the steps contained in the remainder of this section to configure tower.

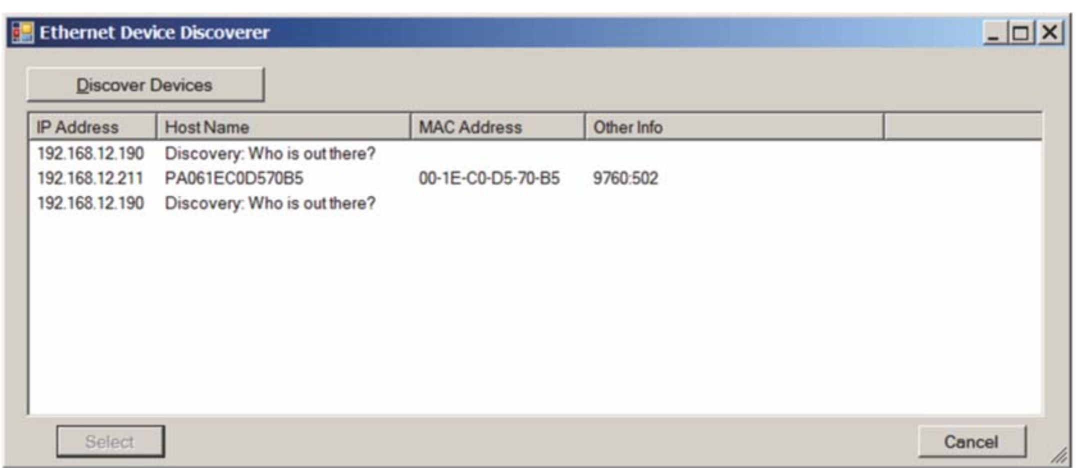
CONNECTING THE DASHBOARD TO AN ETHERNET TOWER

After selecting PA-06 Ethernet the following interactive dashboard will be displayed.



Follow the steps below to get started.

1. DISCOVER DEVICES – Clicking on the button containing the ellipsis will display the dialog below that will allow you to select an Ethernet enabled tower on the network.

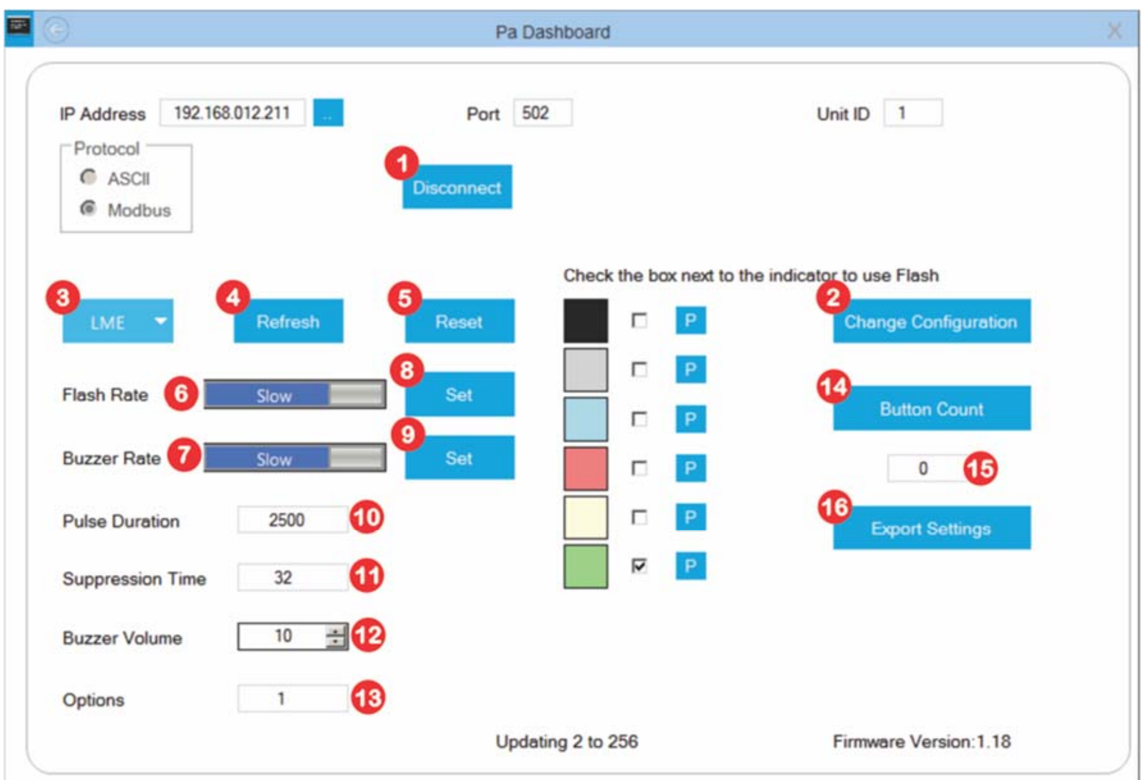


Each device detected will be displayed in the dialog's window along with it static IP address or DHCP assigned IP address. The list also contains the MAC Address and its Host Name. Port numbers and other useful information will also be display under "Other Info". The list can be updated with additional devices by clicking "Discover Devices" at the top. Once the desired device is located, click on it in the list, then click "Select" at the bottom. The window will close and the IP address of the PA-06 device will now be placed in the IP Address field of the utility.

2. SELECT PROTOCOL – Selection switches between ASCII commands PodNet) and Modbus Protocols.
3. CONNECT - At this point communication between a PA dashboard and the device's ACC controller board should be configured. Clicking "Connect" will initiate communication, and the remaining controls on the dashboard will become available.

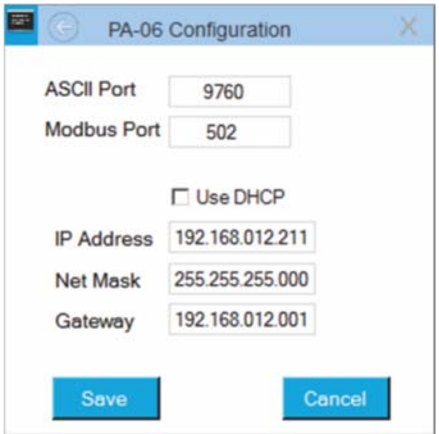
DASHBOARD OPERATION

Once the dashboard has been connected to an ACC equipped light tower, users can easily exercise any of the functions available with the tower. This includes turning any of the lights on or off or flashing them, turning the buzzer on or off and setting the volume, just to name a few. The annotated screen shot on the next page describes the function of each of the buttons, selection lists, and text boxes.



1. DISCONNECT – Clicking on this button will disconnect the Dashboard from the device and the button caption will change to "CONNECT." A second click will reconnect the device.

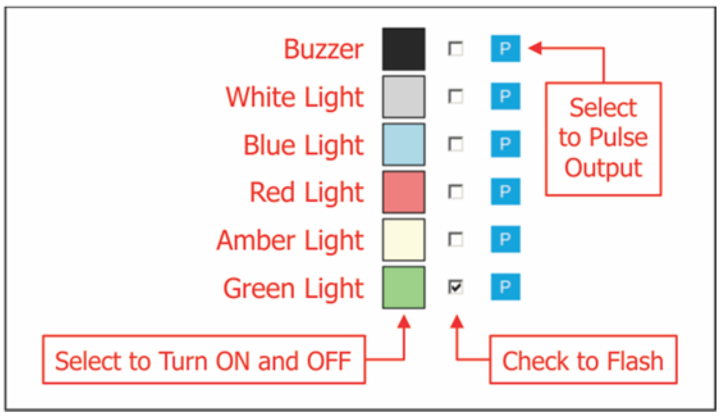
2. CHANGE CONFIGURATION – Clicking on this button displays the dialog shown below that will allow you to change the network settings.



   The "Port" fields at the top of the dialog contain the Network Port Numbers that the device will use for both ASCII and Modbus communications. If you want to use a static IP address for the device, enter it in the "IP Address" field and make the appropriate "Net Mask" and "Gateway" settings. If you want the IP address to be automatically assigned by the network, check "DHCP." You will also need to set the "Gateway" address when using DHCP. After the appropriate values have been entered, click on the "Save" button. At this point the device must be reset, and this is done by selecting the "Reset" button on the home screen. After the device completely resets and the LEDs flash, a scan can be performed for the new IP address."
3. DEVICE TYPE – Displays a dropdown that gives options for the type of device (LME, RT, or EHV).
4. REFRESH – Clicking on this button refreshes the utility.
5. RESET – Clicking on this button resets the device (must be used when making network changes).
6. FLASH RATE – Clicking on the button switches the Indicator flash rate from slow to fast.
7. BUZZER RATE – Clicking here switches the buzzer repetition rate from slow to fast.
8. SET RATE (FLASH) – Clicking on this button will display the dialog shown below. This will be used to set the rate of the slow and fast flash rates in 25 millisecond intervals.

9. SET RATE (BUZZER) – Clicking on this button displays the same dialog used to set the indicator flash rate in 25 millisecond intervals.
10. PULSE DURATION – Sets the duration of the pulse in milliseconds (not used on the LME or LU7 Towers.
11. SUPPRESSION TIME – Sets the duration of time to suppress the buzzer when triggered (measured in seconds).
12. BUZZER VOL – Sets the percentage (0-100%) of the internal buzzer's volume.
13. OPTIONS – Enables (1) or disables (0) the display of the connection status during startup.
14. BUTTON COUNT – Updates the current count of suppression button presses.
15. COUNT DISPLAY – Displays the current push button count value.
16. EXPORT SETTINGS – Clicking on this button will display the computer's file system dialog box which will allow you to select or create a text file to copy the current settings to.

To turn on the LEDs and buzzer, select the colored icon as shown in the screenshot below. (The top indicator triggers the buzzer to sound)



To trigger the LEDs to flash or to place the buzzer in a suppressible mode, check the box next to the indicator. This way, specific LEDs can flash while others remain solid. To control the rate of the LEDs' flash or the Buzzer repetition, use the set rate buttons and define both the slow and fast rates. The Rate switch to the left can then be used to switch between the rates.

Once the buzzer has been triggered with its check box checked, the Buzzer suppression feature can be utilized by pressing the suppressing button attached to the LME. This will silence the button for the duration of the value set in suppression time field, measured in seconds.

Each time the button has been pressed, a count will be incremented and stored. To view this count, select the "Button Count" button and the value will be displayed in the field to its right.

After the suppression time period has elapsed, the buzzer will again sound unless it is deselected. During the suppression period, the suppression can be cancelled/overridden if an additional alarm occurs or if the buzzer is triggered again.

In addition to controlling outputs, setting timing, and configuring the PA-05/06's settings, the user can also access the engineering view of the utility, which gives a breakdown of the registers and their current values. To view all values, select "Poll" and the list will be updated. To write a value to any register, click in the Value column of that register and enter the value you wish to write to it. If done correctly, the right arrow in the left column will change to a pencil icon. After entering the values, clicking on any other part of the utility should trigger this value to be "sent," or written and saved to the register. Note: some registers are read-only and cannot be written to.

A list of available registers for both the PA-05 and PA-06 can be found in the next section of this manual.

Finally, each device contains a "Known State" feature which is enabled by holding down the suppression button within the first 5 seconds of suppling power, and continuing to hold for 5 seconds as the LEDs illuminate. If performed correctly, the communication configuration will be reset to a known state.

In the case of the Ethernet accessible device, the device will switch to DHCP, Podnet protocol, and search for an IP address. If a DHCP server is not available or the address assigned to it is unusable, then the device will try to acquire it by DHCP regularly once every 5 minutes. Meanwhile, an auto configuration address in the 169.254.x.x. subnet, will be used.

During this state, the device can be connected via a crossover cable, and manually configured for a network. After all the settings have been entered, restart the device and allow it to skip the known state in order to reach the normal state, and use the setting configured during the known state.

CONTROLLER BOARD OPERATION

Operation of either the Ethernet or RS-485 equipped towers is virtually the same as is described in this section.

COMMON COMMAND SET

The table below contains the value of the command character for ASCII commands and the register number for Modbus.

<table>
<tr><th>Reg/Cmd</th><th>Hex</th><th colspan="2">Description for All Communications</th></tr>
<tr><td>0</td><td>0</td><td colspan="2">Set/Read Device Being Controlled (0=LME,1=RT,2=EVH)</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Write to Low order byte directly to control output (no buzzer or flashing control)</td></tr>
<tr><td>2</td><td>2</td><td colspan="2">Same as #1, but use high order byte to control flashing and allow buzzer suppression to occur.</td></tr>
<tr><td>3</td><td>3</td><td colspan="2">Turn on Outputs for "Pulse output Duration" (no Buzzer or flashing)</td></tr>
<tr><td>4</td><td>4</td><td colspan="2">Set/Read the Pulse Output Duration (msec)</td></tr>
<tr><td>5</td><td>5</td><td colspan="2">Set/Read the Slow Flash Rate (see format below)</td></tr>
<tr><td>6</td><td>6</td><td colspan="2">Set/Read the Fast Flash Rate (see format below)</td></tr>
<tr><td>7</td><td>7</td><td colspan="2">Set/Read the Slow Buzzer Repetition Rate (see format below)</td></tr>
<tr><td>8</td><td>8</td><td colspan="2">Set/Read the Fast Buzzer Repetition Rate (see format below)</td></tr>
<tr><td>9</td><td>9</td><td colspan="2">Set/Read the Buzzer Suppression Time (sec), A negative value will never timeout</td></tr>
<tr><th>Reg/Cmd</th><th>Hex</th><th>For RS-485 Communications</th><th>For Ethernet Communications</th></tr>
<tr><td>10</td><td>A</td><td>Set/Read Communications Protocol(0=Modbus, 1=ASCII)</td><td>ENET ASCII Port</td></tr>
<tr><td>11</td><td>B</td><td colspan="2">Read Model ID</td></tr>
<tr><td>12</td><td>C</td><td colspan="2">Read Firmware Version</td></tr>
<tr><td>13</td><td>D</td><td>Set/Read Baud Rate(RS485)</td><td rowspan="2">ENET IP address</td></tr>
<tr><td>14</td><td>E</td><td>Set/Read Address</td></tr>
<tr><td>15</td><td>F</td><td rowspan="6">Not used on 485</td><td>Set/Read Modbus Port</td></tr>
<tr><td>16/17</td><td>10/11</td><td>Set/Read Net Mask for Ethernet</td></tr>
<tr><td>18/19</td><td>12/13</td><td>Set/Read Gateway for Ethernet</td></tr>
<tr><td>20</td><td>14</td><td>Set/Read Ethernet Flags (0=static, 1=DHCP), setting top byte to 0xff will Update Ethernet settings and reset unit.</td></tr>
<tr><td>21/22</td><td>15</td><td>Read MAC address on Ethernet</td></tr>
<tr><th>Reg/Cmd</th><th>Hex</th><th colspan="2">Description for All Communications</th></tr>
<tr><td>23</td><td>16</td><td colspan="2">Push Button Count</td></tr>
<tr><td>25</td><td>18</td><td colspan="2">Buzzer Volume (0 to 100%)</td></tr>
<tr><td>26</td><td>19</td><td colspan="2">Startup Sequence enabled or disabled</td></tr>
</table>

Commands that are only used for Ethernet or RS-485 are highlighted in the table above.

Each command or Modbus write instruction has a 16-bit integer value associated with it. The high and low order bytes in commands (registers) 5 through 8 contain the following information.

Rates Format	
High Order Byte	Time ON in 25 msec Increments
Low Order Byte	Time OFF in 25 msec Increments

The low order byte of Registers/Commands 1 through 3 are used to control the light output as shown in the table below.

b0	OK (Green)
b1	WARN (Orange)
b2	ALARM (Red)
b3	ERROR (Blue)
b4	USER (White)

The high order byte for Cmd/Reg 2 contains the bit flags that will indicate the rate at which an indicator will flash. The table below shows the bit assignment.

b0	OK (Green)
b1	WARN (Orange)
b2	ALARM (Red)
b3	ERROR (Blue)
b4	USER (White)
b5	
b6	Flash (Fast/Slow)
b7	Buzzer (Fast/Slow)

SPECIAL FUNCTIONS

Cmd/Reg 0 allows the user to select other Patlite products that can be controlled with the Advanced Communications Controller. These devices generally need a pulsed output; consequently, CMD/REG 3 is used to set this duration, and writes to REG/CMD 4 will send all 8 bits of the low order data byte directly to the output. The outputs will turn on for the specified amount of time to activate the device and then turn off.

AUDIBLE ALARM SUPPRESSION

The audible alarm is only used in LME or LU7 light towers. To use the audible alarm suppression capability, attach a simple normally open pushbutton switch to the "Acknowledge" input on either controller board. Whenever the switch is pressed, the firmware will turn off the audible alarm (buzzer) for the time contained in register 0x09 (Buzzer Suppression Time). If a new output command is received with the buzzer off, the suppression timer will be reset and rearmed. Entering a negative suppression time will permanently turn off the audible alarm until it is rearmed.

TURNING THE LIGHT TOWER INDICATORS ON AND OFF

Commands/Registers 1-3 control the outputs which drive the indicators in the light tower. The following describes how each command or register operates.

1. Write the low order byte directly to output. Do not activate the buzzer or flash for any of the outputs. This will normally be used to control the RT and EVH devices.
2. Same as #1, but use high order byte to control flashing. This will normally be used with the LME light tower.
3. Turn on outputs for "Pulse Output Duration." This may be used to control the RT and EVH device and no flashing will be activated (ignore high order byte).

STARTUP IN A KNOWN STATE

Whenever the 485 version of the controller board powers up, it will come up in a known state and will remain that way for 5 seconds as long as no valid messages are received during that period. During the startup period, the RS-485 interface board will respond to messages that are sent to address 0 at 9600 baud with the ASCII protocol. The board will remain in this mode until a communications configuration command has been received, which will change the address and/or baud rate or the module is power cycled.

The Ethernet board will use its configured IP address on startup. If configured for DHCP mode and no address is acquired, or if the configured IP address cannot be used for some reason (such as another host using it), it will continue to attempt DHCP at a regular interval, no more frequently than once every 5 minutes. In the meantime, it will use an auto configuration address in the 169.254.x.x subnet, as specified in RFC 3927. The board will listen for messages via UDP to identify PA-06s on the network, and assign them a new IP address and port. The broadcasts will be on port 30303. The board can be reset to default configuration (DHCP enabled) by using the "Audible Defeat" input held active for 4 seconds during power up. Each protocol will run on independent ports. All changes to Ethernet settings have to be "SET" by writing to the upper byte of the Ethernet flags register. The board will reset and start back up with the new Ethernet settings active.

SIMPLE BINARY COMMUNICATIONS

The following simple binary protocol has been implemented to maintain compatibility with the early versions of the Ethernet controller board. This is an extension form of the ASCII protocol.

Incoming Message Format …

0xAA (start of message)
Data Byte
1's complement of Data Byte
0x55 (end if message)

If the above message is received correctly (data bytes are the complement of each other and the 0xAA exists at the beginning and a 0x55 at the end), the Data Byte will be stored in register 1 which will send it to the output drivers. The controller board will respond with an echo of the message.

If an error was detected, the controller board will remain silent (no response).

ASCII COMMUNICATIONS

This section describes a single master, multi-dropped, command-response protocol for communicating with either the RS-485 or the Ethernet towers. Eight (8) bit ASCII characters are used with no parity. Each command and all responses begin with a command delimiter character and end with a carriage return character (0x0D). A checksum is always included just before the terminating carriage return.

GENERAL CONSIDERATIONS

RETURNED DATA FORMAT

Hexadecimal numbers always have the high order nibbles transmitted first.

RESPONSE TIME

The tower communications controller will wait for at least 10 ms before responding to a command.

CHECKSUM

Checksums will be calculated and appended to all messages transferred using this protocol. The checksum is calculated by summing the values of all the characters from and including the command delimiter through the character just before the checksum. The low order byte of the sum is appended to the message as two hexadecimal characters just before the terminating carriage return.

DEFAULT VALUES

The following are the default configuration values.

Address = 0
Baud Rate = 9600 (code = 06)
Protocol = ASCII commands

RESPONSE TO INVALID COMMANDS

The modules will respond with a question mark followed by the module's address and the checksum (?AACC).

RESPONSE IF CHECKSUM ERRORS ARE DETECTED

The modules will not respond if a checksum error had been detected in the command.

RS-485 COMMUNICATIONS COMMANDS

The following sections contain the commands that will be implemented when communicating via RS-485.

WRITE INTEGER VALUE COMMAND

The following command will write a 16-bit value to the register defined by "NN."

COMMAND: #AANNVVVVCC(CR)
RESPONSE: >PPCC(CR)

Where: "AA" is the device address in hex
"NN" contains the command ID as a hex character ('00' through '18')
"VVVV" contains value as 4 hex characters
"PP" a number which is incremented each time the pushbutton is pressed
"CC" is the checksum in hex

READ INTEGER VALUE COMMAND

The following command will return the 16-bit value contained in register "NN."

COMMAND: $AANNCC(CR)
RESPONSE: !VVVVCC(CR)

Where: "AA" is the device address in hex
"NN" contains the command ID as a hex character ('00' through '1B')
"VVVV" contains value as 4 hex characters
"CC" is the checksum in hex

ETHERNET COMMUNICATIONS COMMANDS

The following sections contain the commands that will be implemented when doing Ethernet communications. An escape character will be sent when closing this connection.

WRITE INTEGER VALUE COMMAND

The following command will write a 16-bit value to the register defined by "NN."

COMMAND: #NNVVVVCC(CR)
RESPONSE: >PPCC(CR)

Where: "NN" contains the command ID as a hex character ('00' through '1B')
"VVVV" contains value as 4 hex characters
"PP" a number which is incremented each time the pushbutton is pressed
"CC" is the checksum in hex

READ INTEGER VALUE COMMAND

The following command will return the 16-bit value contained in register "NN."

COMMAND: $NNCC(CR)
RESPONSE: !VVVVCC(CR)

Where: "NN" contains the command ID as a hex character ('00' through '1B')
"VVVV" contains value as 4 hex characters
"CC" is the checksum in hex

MODBUS RTU COMMUNICATIONS

In addition to ASCII commands, the RS-485 controller board will support standard Modbus RTU communications. Only function codes 0x03 and 0x06 need to be supported.

IT IS IMPORTANT TO NOTE that all Modbus messages begin with at least 3.5 character times of inactivity on the RS-485 communications line.

FUNCTION CODES SUPPORTED

Request	Returns
Read Analog Output (input reg.)	Returns…
Function = 0x03	Function = 0x03
Register (hi)	Bytes Returned
Register (lo)	Values as 16-bit Registers
Number of Registers (hi)	
Number of Registers (lo)	
Write Single Analog Output (reg.)	Returns…
Function = 0x06	Function = 0x06
Register (hi)	Register (hi)
Register (lo)	Register (lo)
Data to Write (hi)	Data to Write (hi)
Data to Write (lo)	Data to Write (lo)
Write "n" Analog Output (reg.)	Returns…
Function = 0x10	Function = 0x10
Register (hi)	Register (hi)
Register (lo)	Register (lo)
Quantity to Write (hi)	Quantity Written (hi)
Quantity to Write (lo)	Quantity Written (lo)
Byte Count (2 x n)	
Register Values (n x 2 bytes)	

Output values are mapped low order bit to high in successive bytes.

EXCEPTIONS

Except for broadcast messages, when a master device sends a query to a slave device, it expects a normal response. One of four possible events can occur from the master's query:

- If the slave device receives the query without a communication error and can handle the query normally, it returns a normal response.
- If the slave does not receive the query due to a communication error, no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query, but detects a communication error (parity, LRC, or CRC), no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query without a communication error, but cannot handle it (for example, if the request is to read a nonexistent coil or register), the slave will return an exception response informing the master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:
Function Code Field: In a normal response, the slave echoes the function code of the original query in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the slave sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimals higher than the value would be for a normal response.

With the function code's MSB set, the master's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the slave may return data or statistics in the data field (any information that was requested in the query). In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

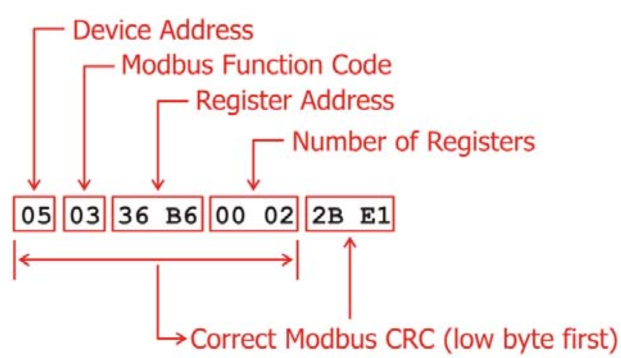
The exception codes are listed below.

01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master

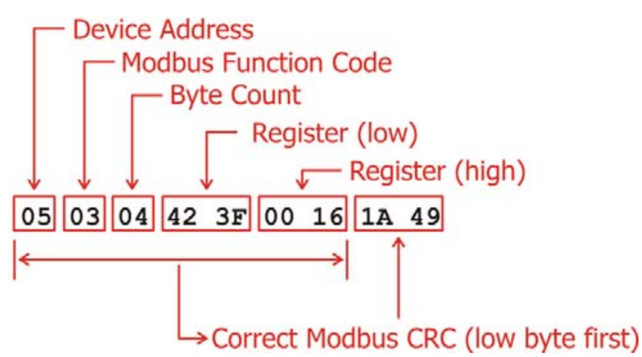
	can next issue a Poll Program Complete message to determine if processing is completed.
06 SLAVE DEVICE BUSY	The slave is engaged in processing a long-duration program command. The master will retransmit the message later when the slave is free.
07 NEGATIVE ACKNOWLEDGE	The slave cannot perform the program function received in the query. This code is returned for an unsuccessful programming request using function code 13 or 14 decimal. The master will request diagnostic or error information from the slave.
08 MEMORY PARITY ERROR	The slave attempted to read extended memory, but detected a parity error in the memory. The master can retry the request, but service may be required on the slave device.

EXAMPLE MODBUS MESSAGE TRANSACTION

Poll



Response



MODBUS/TCP COMMUNICATIONS

The Ethernet version of the communications controller board also supports Modbus in addition to the simple ASCII commands. Modbus TCP is a Modbus RTU message transmitted with a TCP/IP wrapper and sent over a network instead of serial lines. The Server does not have a SlaveID since it uses an IP Address instead.

The official Modbus specification can be found at www.modbus-ida.org. The main differences between Modbus RTU and Modbus TCP are outlined here.

ADU & PDU

Aside from the main differences between serial and network connections stated above, there are a few differences in the message content.

Starting with the Modbus RTU message and removing the SlaveID from the beginning and the CRC from the end results in the PDU, Protocol Data Unit.

Here is an example of a Modbus RTU request for the content of analog output holding registers # 40108 to 40110 from the slave device with address 17.

11 03 006B 0003 7687

11: The SlaveID Address (17 = 11 hex)
03: The Function Code (read Analog Output Holding Registers)
006B: The Data Address of the first register requested. (40108-40001 = 107 =6B hex)
0003: The total number of registers requested. (read 3 registers 40108 to 40110)
7687: The CRC (cyclic redundancy check) for error checking.

Removing the SlaveID and CRC gives the PDU:

03 006B 0003

MBAP HEADER

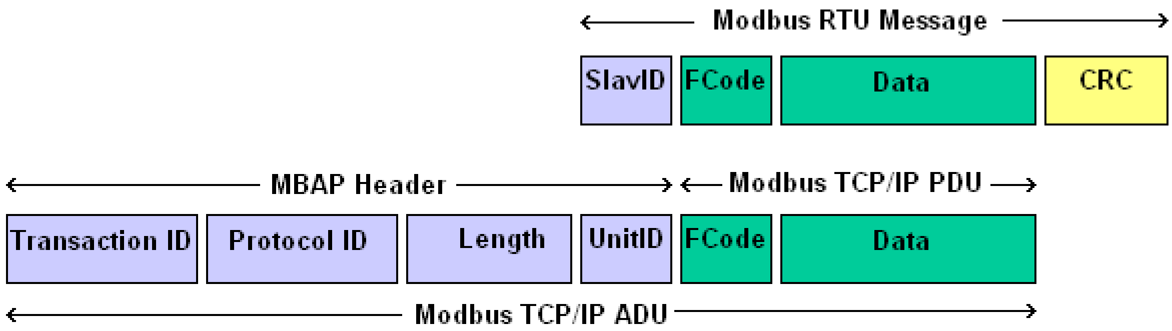
A new 7-byte header called the MBAP header (Modbus Application Header) is added to the start of the message. This header has the following data:

Transaction Identifier: 2 bytes set by the Client to uniquely identify each request. These bytes are echoed by the Server since its responses may not be received in the same order as the requests.

Protocol Identifier: 2 bytes set by the Client, always = 00 00

Length: 2 bytes identifying the number of bytes in the message to follow.

Unit Identifier: 1 byte set by the Client and echoed by the Server for identification of a remote slave connected on a serial line or on other buses.



SUMMARY

The equivalent request to this Modbus RTU example

11 03 006B 0003 7687

in Modbus TCP is:

0001 0000 0006 11 03 006B 0003

0001: Transaction Identifier
0000: Protocol Identifier
0006: Message Length (6 bytes to follow)
11: The Unit Identifier (17 = 11 hex)
03: The Function Code (read Analog Output Holding Registers)
006B: The Data Address of the first register requested. (40108-40001 = 107 =6B hex)
0003: The total number of registers requested. (read 3 registers 40108 to 40110)

DEFAULT SETTINGS VALUES

The PA-06 controllers inside the LME and LU7 Towers are shipped with the following settings.

DEFAULT CONFIGURATION SETTINGS

Device Selection = 0 (LME Tower)
Output Pulse Duration = 2500
Slow Flash Rate On Time = 125
Slow Flash Rate Off Time = 125
Fast Flash Rate On Time = 20
Fast Flash Rate Off Time = 20
Slow Buzzer Rate On Time = 125
Slow Buzzer Rate Off Time = 125
Fast Buzzer Rate On Time = 50
Fast Buzzer Rate Off Time = 50
Buzzer Supression Time = 30
MODBUS Port Number = 502
SERVER Port Number = 9760
Buzzer Volume = 100
Options = 0
DHCP Enabled = 1 (DHCP Enabled)

DEFAULT REGISTER SETTINGS

Reg/Cmd	Value	Description for All Communications
0	0	Set/Read Device Being Controlled (0=LME,1=RT,2=EVH)
1	0	Write to Low order byte directly to control output (no buzzer or flashing control)
2	0	Same as #1, but use high order byte to control flashing and allow buzzer suppression to occur.
3	0	Turn on Outputs for "Pulse output Duration" (no Buzzer or flashing)
4	2500	Set/Read the Pulse Output Duration (msec)
5	32125	Set/Read the Slow Flash Rate (see format on page 11)
6	5140	Set/Read the Fast Flash Rate (see format on page 11)
7	32125	Set/Read the Slow Buzzer Repetition Rate (see format on page 11)
8	12850	Set/Read the Fast Buzzer Repetition Rate (see format on page 11)
9	30	Set/Read the Buzzer Suppression Time (sec), A negative value will never timeout

Reg/Cmd	Value	For RS-485 Communications	For Ethernet Communications
10	9760	Set/Read Communications Protocol(0=Modbus, 1=ASCII)	ENET ASCII Port
11	Read Only	Read Model ID	
12		Read Firmware Version	
13	DHCP Driven	Set/Read Baud Rate(RS485)	ENET IP address
14		Set/Read Address	
15	502	Not used on 485	Set/Read Modbus Port
16/17	DHCP Driven		Set/Read Net Mask for Ethernet
18/19			Set/Read Gateway for Ethernet
20	1		Set/Read Ethernet Flags (0=static, 1=DHCP), setting top byte to 0xff will Update Ethernet settings and reset unit.
21/22	Read Only		Read MAC address on Ethernet
Reg/Cmd	**Value**	**Description for All Communications**	
23	Read Only	Push Button Count	
25	100	Buzzer Volume (0 to 100%)	
26	1	Startup Sequence enabled or disabled	