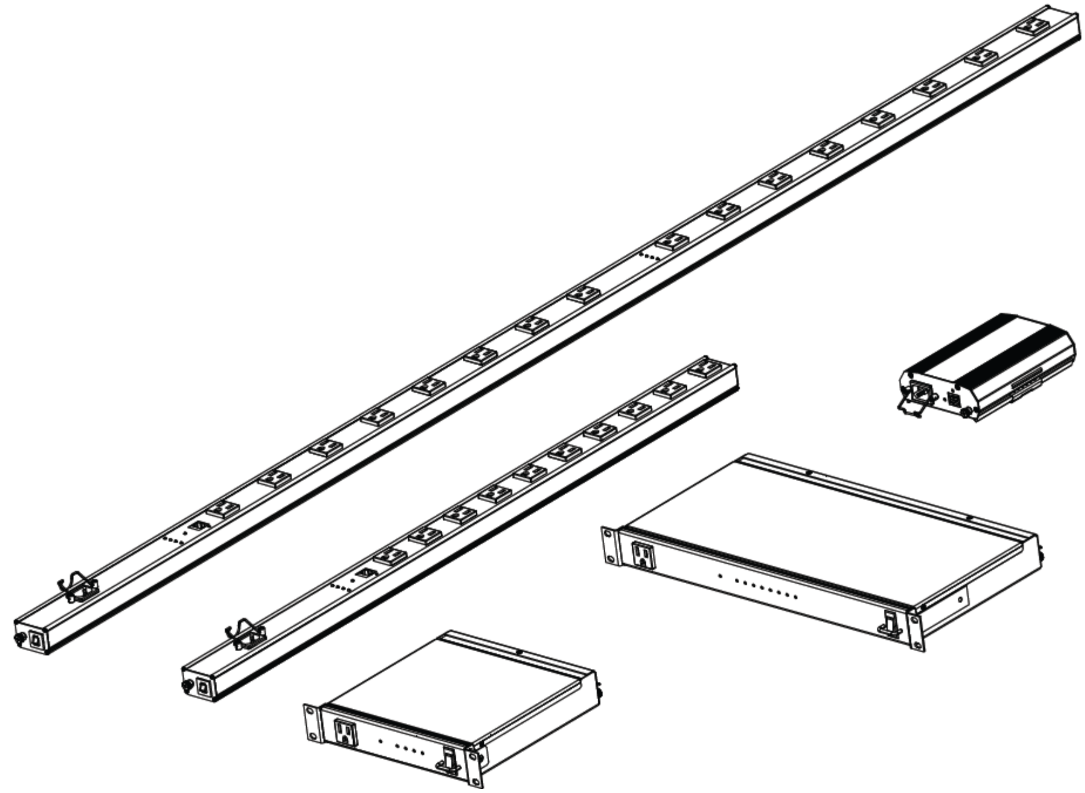
Control System Communication Protocol Manual

RackLink™

Monitor | Control | Alert | Report | Analyze



THANK YOU

Thank you for purchasing a RackLink™ product.

MIDDLE ATLANTIC®

A brand of legrand

22 PAP

I-00472 Rev G

TABLE OF CONTENTS

INTRODUCTION 3

PROTOCOL TRANSPORTS 3

RS232 Specifications (RackLink Premium Only) 3

TCP/IP (RackLink Select, Premium, Premium+) 3

ENABLING CONTROL PROTOCOL ACCESS 3

PROTOCOL DOCUMENT NOTES 3

COMMUNICATION 3

PROTOCOL MESSAGE STRUCTURE 4

Checksum 5
Checksum calculation elongated example 5
Checksum masking visual description 6
C Style example 6

Escape Characters 6
Protected Values 7
Escaping a Data Value 7
Escaping Example 7

BASIC COMMAND AND MESSAGE DEFINITIONS (ALL RACKLINK SERIES) 7
NACK (0x10) 7
Ping/Pong (0x01) 8
Login (0x02) 9
Power Outlets(0x20)/Dry Contacts(0x30) 10
Power Outlet Name (0x21)/ Dry Contact Name (0x31) 12
Power Outlet Count(0x22)/Dry Contact Count (0x32) 13
Sequence Power Outlets (0x36) 14

ADVANCED COMMAND AND MESSAGE DEFINITIONS (PREMIUM/PREMIUM+ ONLY) 15
Power Outlet Device Energy Management State (0x23) 15
Emergency Power Off (EPO) (0x37) 16
Register Log Alerts (0x40) 17
Register Status Change (0x41) 19
Sensor Values (0x50 to 0x61) 21
Thresholds (0x70 to 0x77) 23
Read Log Entry (0x80) 25
Log Count (0x81), Clear Log (0x82) 27
Log Clear and Count Commands 27
Product Part Number(0x90), Rating (0x91), Surge (0x93), IP Address (0x94), MAC Address (0x95) 28

Introduction

NOTE: The Control Protocol interface is only enabled after you log in for the first time (using the Administrator account) and change the passwords for Administrator, User, and Control Systems accounts.

Protocol Transports

RS232 Specifications (RackLink Premium Only)

- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1
- Parity: None

TCP/IP (RackLink Select, Premium, Premium+)

- TCP Port: 60000

Enabling Control Protocol Access

Access to the control protocol is disabled by default, to block unauthorized access until properly set up. In the case of Select and Premium models, it will be unlocked at first webpage login, when the password is set. The login command will use the user of “user” with the password established for these models. When using the Premium+, the protocol must be manually enabled via the checkbox in Device Setting -> Network Services -> Control Protocol. Being that the Premium+ has configurable accounts, there is no default “user” login for the control protocol. Instead, the login command’s username and password can reference any created user account that has admin rights and the control protocol enabled.

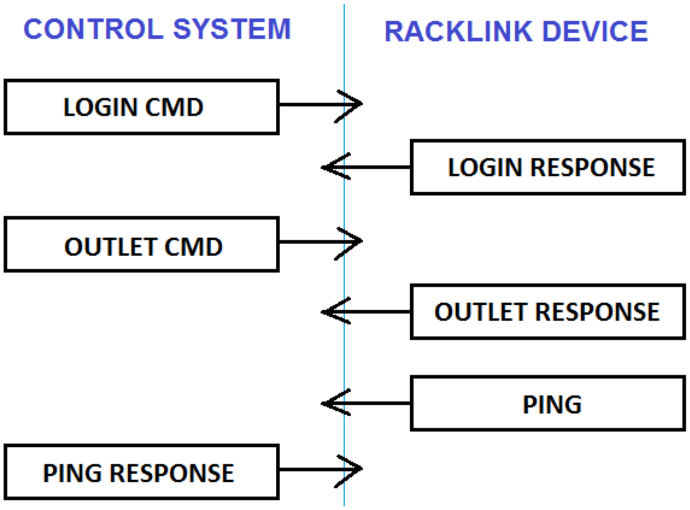
Protocol Document Notes

The general format of this protocol is hexadecimal, although the Data Envelope portion of the protocol may contain some ASCII characters for the purposes of clarifying data values. All ASCII string literal examples will be shown with double quotes at the beginning and end of the series of characters. The double quotes and 0x identifiers are used for clarification. These characters are not part of the actual protocol.

Communication

When you send any command using this protocol, you will receive a “NACK” when the RackLink™ device was unable to carry out your request. The “NACK” will contain an error code indicating what failed. If your message is correct you will receive a “RESPONSE” to the command indicating the current status. If you have registered status changes updates, you will receive those unsolicited when items change. You are NOT responsible for sending NACKS on message that you receive. You are responsible for responding to PING Messages, if you fail to respond to three ping messages in a row, you will be considered disconnected and will stop receiving any unsolicited messages. Once you are considered disconnected, you will also receive a NACK for any message you send, except for the Login command. A log entry will be entered into the RackLink device indicating you have lost communication. You must reestablish communication at this point.

When you first connect you must send a "SET" login message to the RackLink device. You will get a login "RESPONSE" indicating if your credentials have been accepted. You will than get a "SET" ping message from the RackLink device. You respond with a "Response" message. Once you do that you can use any of the other valid commands in this document. If you lose communication to the RackLink device, you must follow these same steps to "reestablish" communication.



Protocol Message Structure

General Message Format

<Header><Length><data envelope><Checksum><Tail>

- Header
  - Value: 0xfe
  - Data Length: 1 Byte
- Length
  - Total Length of the data envelope.
    - Value Example 1: 0x3c (60 bytes)
    - Value Example 2: 0x06 (6 bytes)
  - Data Length: 1 Byte
- Data Envelope
  - Contents: Varied (See Remainder of Document)
  - Data Length: 3 – 250 Bytes
- Checksum
  - Summation of all the bytes starting and including the header byte all the way to the end of the data envelope. The checksum will only include the least significant 7 bits. The eighth bit gets set to zero.
  - Data Length: 1 Byte
- Tail
  - Value: 0xff
  - Data Length: 1 Byte

Checksum

Here is an example message with the correct checksum.

0xfe 0x10 0x00 0x02 0x01 “user|password” 0x3F 0xff

In order to calculate the checksum correctly, you must add up all of the bytes starting and including the header byte(0xfe) all the way to the end of the data envelope. The checksum will only include the least significant 7 bits. The eighth bit gets set to zero, and all higher bits discarded.

Checksum calculation elongated example

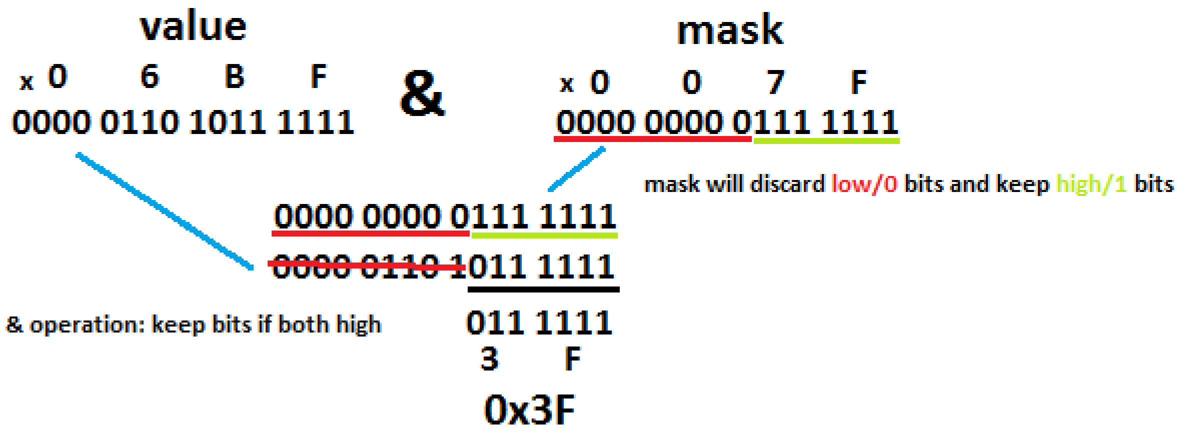
```
sum = 0 ; (clear your variable)

sum = sum + 0xfe;  (Header)
sum = sum + 0x10; (Length)
sum = sum + 0x00;
sum = sum + 0x02;
sum = sum + 0x01;
sum = sum + 0x75;    // “u” == 0x75
sum = sum + 0x73;    // “s” == 0x73
sum = sum + 0x65;    // “e” == 0x65
sum = sum + 0x72;    // “r” == 0x72
sum = sum + 0x7c ;   // “|”== 0x7c
sum = sum + 0x70;    // “p” == 0x70
sum = sum + 0x61;    // “a” == 0x61
sum = sum + 0x73;    // “s” == 0x73
sum = sum + 0x73;    // “s” == 0x73
sum = sum + 0x77;    // “w” == 0x77
sum = sum + 0x6f ;   // “o” == 0x65
sum = sum + 0x72;    // “r” == 0x72
sum = sum + 0x64;    // “d” == 0x64

//note: sum is 0x6BF at this point

checksum = sum & 0x7f (Bitwise “and” to keep least significant 7 bits)
Answer: checksum = 0x3F
```

Checksum masking visual description



The checksum example is lengthened for clarification, normally a while loop is used to iterate through the bytes to perform the math. See the example below:

C Style example

```
int calculateChecksum(char *msg, int length)
{
	int sum = 0;

	for (int i = 0; i < length; i++)
	{
		sum += msg[i];
	}

	sum &= 0x7f;

	return sum;
}
```

Escape Characters

(The protocol has been designed to limit the possible escaping needs. While this section may not be needed, it is provided in case it ends up being required.)

The header(0xFE) and tail (0xFF) characters are used to control messaging processing, indicating the start and end of message packets. These values cannot be used in the body of messages, as they would prematurely end the current message or start a new message. If these values are needed for a byte within the message, we need to use a special escape character to indicate that the character is a piece of the body and not a header or tail byte. For this purpose, the value of 0xFD is designated as the escape character, and must be used whenever 0xFF, 0xFE, or 0xFD is encountered in the message body.

It is important to understand that escaping the data values adds bytes to the message. These escaped bytes (0xfd) will not be included in the length or the checksum calculations. Thus, when calculating both length and checksum, you must remove and revert the escape sequences prior to your calculations.

Protected Values

- Header: 0xfe
- Tail: 0xff
- Escape Byte: 0xfd

Escaping a Data Value

When escaping a data byte, the value must equal one of the three protected values as seen above. You insert the "Escape Byte" into the message prior to the value that needs escaping. Then you invert the bits on the value that needs escaping. This should be the last thing you do while assembling your command message before you transmit the message.

Escaping Example

The example contains a fictitious message and is only used for the purpose of this discussion. On a side note the checksum and length are calculated correctly for the message.

If you notice below in the "Data Envelope" section that a value of 0xff is needed. Since this value is protected this message needs to get escaped.

<Header><Length><data envelope><Checksum><Tail>

<0xfe><0x04><0x00 0x06 0x00 0xff><0x07><0xff> (Not Escaped)

The escape character (0xfd) is added before the value that needs escaping. The bits of the data value are then inverted to complete the escape process. Note that the length and the checksum did not change.

<0xfe><0x04><0x00 0x06 0x00 0xfd 0x00><0x07><0xff> (Escaped)

Address

Basic Command and Message Definitions (All RackLink Series)

NACK (0x10)

NACK (from device upon error)

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x10 - NACK
4	Subcommand	0x10 - Response

Byte Index	Description	Note
5	Data	0x01 - Bad CRC on previous command 0x02 - Bad Length on previous command 0x03 - Bad Escape sequence on previous command 0x04 - Previous command invalid 0x05 - Previous sub-command invalid 0x06 - Previous command incorrect byte count 0x07 - Invalid data bytes in previous command 0x08 - Invalid Credentials (note: need to login again) 0x10 - Unknown Error 0x11 - Access Denied (EPO)
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF

Sample Command: 0xFE040010100123FF (bad CRC on previous command)

Ping/Pong (0x01)

Note: Pings are periodically sent to the Control System from the RackLink device to confirm connection status. The RackLink will close the connection after three unanswered pings.

Ping

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03
2	Destination	Always 0x00
3	Command	0x01 - PING
4	Subcommand	0x01 - SET
6	Checksum	Always 0x03
7	Tail Character	Always 0xFF

Sample Request: 0xFE0300010103FF (Device sending PING request to Control System)

Ping Response / Pong

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03
2	Destination	Always 0x00
3	Command	0x01 - PING
4	Subcommand	0x10 - Response

Byte Index	Description	Note
6	Checksum	Always 0x12
7	Tail Character	Always 0xFF

Sample Response: 0xFE0300011012FF (Control System sending PING response to device)

Login (0x02)

Login Command

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Variable length . Length of Bytes 2 to X
2	Destination	Always 0x00
3	Command	0x02 - Login
4	Subcommand	0x01 - Set
5 to X	Data	Variable String representing login info. Format: “Username|Password” Default: "user|password" Default: 0x757365727C70617373776F7264
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Sample Command: 0xFE10000201757365727C70617373776F72643FFF

Login Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x02 - Login
4	Subcommand	0x10 - Response
5	Response	0x00 - Rejected 0x01 - Accepted

Byte Index	Description	Note
6	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
7	Tail Character	Always 0xFF

Sample Response: 0xFE040002100115FF (login accepted)

Power Outlets(0x20)/Dry Contacts(0x30)

Read Outlet/Contact Status

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x20 - Power Outlets 0x30 - Dry Contact
4	Subcommand	0x02 - Get State
5	Outlet/Contact Number	0x01 to 0x10 (1 to 16)
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF

Example: 0xfe040020000325ff (Read Outlet 3 state)

Write Outlet/Contact Status

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x09
2	Destination	Always 0x00
3	Command	0x20 - Power Outlets 0x30 - Dry Contact
4	Subcommand	0x01 - Set State
5	Outlet/Contact	0x01 to 0x10 (1 to 16 for outlets) 0x1 to 0x08 (1 to 8 for contacts)
6	Desired State	0x00 - OFF 0x01 - ON 0x02 - Cycle (set time in next field) 0x03 - Not Controllable (response only)

Byte Index	Description	Note
7, 8, 9, 10	Cycle Time	Cycle time . Range is 0000 to 3600 seconds . Sent as ACII characters (whereas 0 is 0x30, 9 is 0x39, etc) . example: 0000 seconds would be encoded as 0x30303030 . example: 3600 seconds would be encoded as 0x33363030 not used for on/off cmds. Send 0000 (0x30303030)
11	Checksum	Sum of all previous bytes (0 to 10), masked with 0x007F
12	Tail Character	Always 0xFF

Example 1: 0xfe0900200101013030303030306aff (Outlet 1 ON)
Example 2: 0xfe09002001020030303030306aff (Outlet 2 OFF)
Example 3: 0xfe0900200102023030303571ff (Outlet 2 Cycle, 5 seconds)
Example 3: 0xfe0900200108023030303874ff (Outlet 8 Cycle, 8 seconds)

Outlet Status Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x09
2	Destination	Always 0x00
3	Command	0x20 - Power Outlets 0x30 - Dry Contact
4	Subcommand	0x10 - Response to a command 0x12 - Status Update from local events or changes (non-cmds) 0x30 - Log Update (scheduled log reports)
5	Outlet/Contact	0x01 to 0x10 (1 to 16 for outlets) 0x1 to 0x08 (1 to 8 for contacts)
6	Current State	0x00 - Outlet is OFF 0x01 - Outlet is ON
7, 8, 9, 10	Cycle Time	Currently saved cycle time for this outlet . Range is 0000 to 3600 seconds . Sent as ACII characters (whereas 0 is 0x30, 9 is 0x39, etc) . example: 0000 seconds would be encoded as 0x30303030 . example: 3600 seconds would be encoded as 0x33363030 not used for on/off cmds. Send 0000 (0x30303030)
11	Checksum	Sum of all previous bytes (0 to 10), masked with 0x007F
12	Tail Character	Always 0xFF

Example: 0xfe090020010030030303030357fff (Outlet Changed, Outlet 3 OFF, 5 sec cycle)

Power Outlet Name (0x21)/ Dry Contact Name (0x31)

Read Outlet/Contact Name

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x21 - Outlet Name 0x31 - Contact Name
4	Subcommand	0x02 - Get Name
5	Outlet/Contact Number	0x01 to 0x10 (1 to 16 for outlets) 0x1 to 0x08 (1 to 8 for contacts)
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF

Example: 0xFE040021020126FF (Read Name of Outlet 1)

Write Outlet/Contact Name

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Variable length . Length of Bytes 2 to X
2	Destination	Always 0x00
3	Command	0x21 - Outlet Name 0x31 - Contact Name
4	Subcommand	0x01 - Set Name
5	Outlet/Contact Number	0x01 to 0x10 (1 to 16)
6 to X	Outlet/Contact Name	Variable String representing outlet name. Example: "name" Example: 0x6E616D65
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Example: FE08002110016E616D6559FF (Set Outlet 1 name to "name")

Outlet/Contact Name Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Variable length . Length of Bytes 2 to X
2	Destination	Always 0x00
3	Command	0x21 - Outlet Name 0x31 - Contact Name
4	Subcommand	0x10 - Response
5	Outlet/Contact Number	0x01 to 0x10 (1 to 16)
6 to X	Outlet/Contact Name	Variable String representing outlet name. Example: "name" Example: 0x6E616D65
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Example: FE08002101016E616D654AFF (Response of Outlet 1 named "name")

Power Outlet Count(0x22)/Dry Contact Count (0x32)

Read Outlet/Contact Count

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03
2	Destination	Always 0x00
3	Command	0x22 - Outlet Count 0x32 - Dry Contact Count
4	Subcommand	0x02 - Get Count
5	Checksum	Always 0x25 for Outlet count or Always 0x35 for contact count
6	Tail Character	Always 0xFF

Example: 0xFE0300220225FF (Read outlet count)

Outlet/Contact Count Response

Byte Index	Description	Note
0	Start Character	Always 0xFE

Byte Index	Description	Note
1	Length	. Always 0x13 for Outlet Count Response . Always 0x0B for Contact Count response
2	Destination	Always 0x00
3	Command	0x22 - Outlet Count 0x32 - Dry Contact Count
4	Subcommand	0x10 - Response
5 to X	Outlet/Contact Name	. 16 bytes of status for Outlet Count . 8 bytes of status for Contact Count . Each byte for each outlet/contact indicating: . "C"/0x43 - Controllable . "N"/0x4E - Non-Controllable . "X"/0x58 - Does not exist
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Sequence Power Outlets (0x36)

Sequencing Command

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x08
2	Destination	Always 0x00
3	Command	0x36 - Power Sequencing
4	Subcommand	0x01 - Set
5	Sequencing Direction	0x01 - Sequence UP 0x03 - Sequence DOWN
6 to 9	Delay Time	Delay time between outlets . Always 0000 for Premium+ (uses configured delay settings) . Range is 0000 to 0999 seconds . Sent as ACII characters (whereas 0 is 0x30, 9 is 0x39, etc) . example: 0000 seconds would be encoded as 0x30303030 . example: 999 seconds would be encoded as 0x30393939
10	Checksum	Sum of all previous bytes (0 to 9), masked with 0x007F
11	Tail Character	Always 0xFF

Example: 0xFE08003601013030303301FF (Sequence up, 3 second delay)
Example: 0xFE08003601033030303505FF (Sequence down, 5 second delay)
Example: 0xFE0800360101303030307EFF (Sequence up, Premium+, use saved settings)

Sequencing Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x08
2	Destination	Always 0x00
3	Command	0x36 - Power Sequencing
4	Subcommand	0x01 - Set
5	Sequencing Direction	0x00 - Not Sequencing (Fallback value on error) 0x01 - Sequencing up 0x02 - Sequencing up complete 0x03 - Sequencing down 0x04 - Sequencing down complete
6 to 9	Delay Time	Delay time between outlets . Range is 0000 to 0999 seconds . Sent as ACII characters (whereas 0 is 0x30, 9 is 0x39, etc) . example: 0000 seconds would be encoded as 0x30303030 . example: 999 seconds would be encoded as 0x30393939
10	Checksum	Sum of all previous bytes (0 to 9), masked with 0x007F
11	Tail Character	Always 0xFF

Advanced Command and Message Definitions (Premium/Premium+ Only)

Power Outlet Device Energy Management State (0x23)

Energy Management Command

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Get/Read subcommand: Always 0x04 . Set/Write Subcommand: Always 0x05
2	Destination	Always 0x00
3	Command	0x23 Energy Management State
4	Subcommand	0x01 - Set/Write 0x02 - Get
5	Outlet Number	0x01 to 0x10 (1 to 16)
6 (set only)	Energy State	Desired State (byte included for SET only) Desired Energy Management State (1 byte ASCII letter per outlet, 16 bytes total) "D"/0x44: Disconnected "S"/0x53: Standby "I"/0x49: ON "O"/0x4F: OFF "U"/0x55: Unknown

Byte Index	Description	Note
7	Checksum	Sum of all previous bytes (0 to 6), masked with 0x007F
8	Tail Character	Always 0xFF

Emergency Management Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x05
2	Destination	Always 0x00
3	Command	0x23 Energy Management State
4	Subcommand	0x10 - Response
5	Outlet Number	0x01 to 0x10 (1 to 16)
6	Energy State	Current Energy Management State (1 byte ASCII letter per outlet, 16 bytes total) "D"/0x44: Disconnected "S"/0x53: Standby "I"/0x49: ON "O"/0x4F: OFF "U"/0x55: Unknown
7	Checksum	Sum of all previous bytes (0 to 6), masked with 0x007F
8	Tail Character	Always 0xFF

Emergency Power Off (EPO) (0x37)

Emergency Power Off Command

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x37 - EPO
4	Subcommand	0x01 - Set
5	EPO Command	0x00 - EPO Recover 0x01 - EPO Initiate
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF

Emergency Power Off Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x04
2	Destination	Always 0x00
3	Command	0x37 - EPO
4	Subcommand	0x10 - Response 0x12 - Status Change
5	EPO Response	0x00 - Normal Operation State 0x01 - Emergency Power Off Mode
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF

Register Log Alerts (0x40)

Log Alerts Commands

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Always 0x03 for get subcommand . Always 0x05 for set subcommand
2	Destination	Always 0x00
3	Command	0x40 - Register Log Alerts
4	Subcommand	0x01 - Set 0x02 - Get

Byte Index	Description	Note
5, 6	Data (SET subcommand)	. Two bytes of data, each bit representing an option. . Set a bit high to enable logging of that option. First Byte: BIT 1: Normal Log Alerts BIT 2: Over Voltage BIT 3: Under Voltage BIT 4: Over Temperature BIT 5: Under Temperature BIT 6: Surge Fault BIT 7: <Future> BIT 8: RESERVED Second Byte: BIT 1: Auto Ping Timeout BIT 2: RS232 Ping Timeout BIT 3: Over Current BIT 4: Under Current BIT 5: EPO BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
7	Checksum	Sum of all previous bytes (0 to 6), masked with 0x007F
8	Tail Character	Always 0xFF

Log Alerts Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x05
2	Destination	Always 0x00
3	Command	0x40 - Register Log Alerts
4	Subcommand	0x10 - Response

Byte Index	Description	Note
5, 6	Data	. Two bytes of data, each bit representing an option. . Bit is set high if logging of that option is enabled First Byte: BIT 1: Normal Log Alerts BIT 2: Over Voltage BIT 3: Under Voltage BIT 4: Over Temperature BIT 5: Under Temperature BIT 6: Surge Fault BIT 7: <Future> BIT 8: RESERVED Second Byte: BIT 1: Auto Ping Timeout BIT 2: RS232 Ping Timeout BIT 3: Over Current BIT 4: Under Current BIT 5: EPO BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
7	Checksum	Sum of all previous bytes (0 to 6), masked with 0x007F
8	Tail Character	Always 0xFF

Register Status Change (0x41)

Log Status Commands

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Always 0x03 for get subcommand . Always 0x06 for set subcommand
2	Destination	Always 0x00
3	Command	0x41 - Register Log Status
4	Subcommand	0x01 - Set 0x02 - Get

Byte Index	Description	Note
5	Data Byte 1	Bytes of data, each bit representing an option. . Set a bit high to enable logging of that option. BIT 1: Outlet Status Changes BIT 2: <Future> BIT 3: <Future> BIT 4: <Future> BIT 5: <Future> BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
6	Data Byte 2	BIT 1: Dry Contact Status Changes BIT 2: Input Status Change BIT 3: Sequence Status Change BIT 4: EPO Status Change BIT 5: <Future> BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
7	Data Byte 3	BIT 1: Low Voltage Threshold Changes BIT 2: High Voltage Threshold Changes BIT 3: <Future> BIT 4: Maximum Load Changes BIT 5: Minimum Load Changes BIT 6: <Future> BIT 7: Maximum Temperature Changes BIT 8: RESERVED
8	Data Byte 4	BIT 1: Minimum Temperature Changes BIT 2: <Future> BIT 3: <Future> BIT 4: <Future> BIT 5: <Future> BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
9	Data Byte 6	BIT 1: Current Kilowatt Hours Changes BIT 2: Current Peak Voltage Changes BIT 3: Current RMS Voltage Changes BIT 4: Current Peak Load Changes BIT 5: Current RMS Load Changes BIT 6: Current Temperature Changes BIT 7: Current Wattage Changes BIT 8: RESERVED

Byte Index	Description	Note
10	Data Byte 7	BIT 1: Current Power Factor Changes BIT 2: Current Thermal Load Changes BIT 3: *Current Log Count Changes BIT 4: *Current Surge Protection State Changes BIT 5: <Future> BIT 6: <Future> BIT 7: <Future> BIT 8: RESERVED
11	Checksum	Sum of all previous bytes (0 to 10), masked with 0x007F
12	Tail Character	Always 0xFF

Log Status Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Always 0x05
2	Destination	Always 0x00
3	Command	0x41 - Register Log Status
4	Subcommand	0x10 - Response
5 to 10	Data Bytes	. Six Bytes, see definition in command above
11	Checksum	Sum of all previous bytes (0 to 10), masked with 0x007F
12	Tail Character	Always 0xFF

Sensor Values (0x50 to 0x61)

Sensor Value Commands

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Get/Read subcommand: Always 0x03 . Set/Write Subcommand: Cmd dependent length example: "##.#" format for value, length is 0x07 example: "###" format for value, length is 0x06
2	Destination	Always 0x00

Byte Index	Description	Note
3	Command	0x50 - Kilowatt Hours (##########.#) 0x51 - Peak Voltage (###) 0x52 - RMS Voltage Changes (###) 0x53 - Peak Load (##.#) 0x54 - RMS Load (##.#) 0x55 - Temperature (###) 0x56 - Wattage (####) 0x57 - Power Factor (#.#) 0x58 - Thermal Load (BTU) (####.#) 0x59 - Surge Protection State (0x00-n/a, 0x01-protected, 0x02- compromised) 0x60 - Energy Management State (1 byte ASCII letter per outlet, 16 bytes total) "D"/0x44: Disconnected "S"/0x53: Standby "I"/0x49: ON "O"/0x4F: OFF "U"/0x55: Unknown 0x61 - Occupancy State "U"/0x55: Unoccupied "O"/0x4F: Occupied
4	Subcommand	0x01 - Set/Write 0x02 - Get
5 to X (Set Only)	Sensor Value (Set Only)	. Value in ASCII representation. . See markings next to options in command for format. . Note: decimal point is an included character (0x2E)
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Sensor Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Command dependent length example: "##.#" format for value, length is 0x07 example: "###" format for value, length is 0x06
2	Destination	Always 0x00

Byte Index	Description	Note
3	Command	0x50 - Kilowatt Hours (##########.#) 0x51 - Peak Voltage (###) 0x52 - RMS Voltage Changes (###) 0x53 - Peak Load (##.#) 0x54 - RMS Load (##.#) 0x55 - Temperature (###) 0x56 - Wattage (####) 0x57 - Power Factor (#.#) 0x58 - Thermal Load (BTU) (####.#) 0x59 - Surge Protection State (0x00-n/a, 0x01-protected, 0x02- compromised) 0x60 - Energy Management State (1 byte ASCII letter per outlet, 16 bytes total) "D"/0x44: Disconnected "S"/0x53: Standby "I"/0x49: ON "O"/0x4F: OFF "U"/0x55: Unknown 0x61 - Occupancy State "U"/0x55: Unoccupied "O"/0x4F: Occupied
4	Subcommand	0x10 - Response 0x12 - Status Change (unsolicited, local change/event)
5 to X	Sensor Value	. Value in ASCII representation. . See markings next to options in command for format. . Note: decimal point is an included character (0x2E)
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Thresholds (0x70 to 0x77)

Threshold Commands

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Get/Read subcommand: Always 0x03 . Set/Write Subcommand: Cmd dependent length example: "##.#" format for value, length is 0x07 example: "###" format for value, length is 0x06
2	Destination	Always 0x00

Byte Index	Description	Note
3	Command	0x70 - Low Voltage Threshold (###) 0x71 - High Voltage Threshold (###) 0x73 - Max Load Current (##.#) 0x74 - Min Load Current (##.#) 0x76 - Max Temperature (###) 0x77 - Min Temperature (###)
4	Subcommand	0x01 - Set/Write 0x02 - Get
5 to X (Set only)	Sensor Value (Set Only)	. Value in ASCII representation. . See markings next to options in command for format. . Note: decimal point is an included character (0x2E)
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Threshold Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Command dependent length example: "##.#" format for value, length is 0x07 example: "###" format for value, length is 0x06
2	Destination	Always 0x00
3	Command	0x70 - Low Voltage Threshold (###) 0x71 - High Voltage Threshold (###) 0x73 - Max Load Current (##.#) 0x74 - Min Load Current (##.#) 0x76 - Max Temperature (###) 0x77 - Min Temperature (###)
4	Subcommand	0x10 - Response 0x12 - Status Change (unsolicited, local change or event)
5 to X	Sensor/Reading Value	. Value in ASCII representation. . See markings next to options in command for format. . Note: decimal point is an included character (0x2E)
X+1	Checksum	Sum of all previous bytes (0 to X), masked with 0x007F
X+2	Tail Character	Always 0xFF

Read Log Entry (0x80)

Log Entry Read Command

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x0A
2	Destination	Always 0x00
3	Command	0x80 - Read Log Entry/Entries
4	Subcommand	0x02 - Get
5 to 8	Index of First Log Entry	ASCII Encoded . Example(ASCII): "0002", start on 2nd entry . Example(HEX): 0x30303032, start on 2nd entry
9	Separator Character	Always "|" (0x7C)
10 to 11	Number of Entries	ASCII Encoded . Example(ASCII): "10", read 10 entries . Example(HEX): 0x3130, read 10 entries
12	Checksum	Sum of all previous bytes (0 to 11), masked with 0x007F
13	Tail Character	Always 0xFF

Log Entry Read Response (Multiple Responses if Multiple Entries)

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x0A
2	Destination	Always 0x00
3	Command	0x80 - Read Log Entry/Entries
4	Subcommand	0x02 - Get
5 to 6	Log Request Number	ASCII Encoded . Example(ASCII): "01", entry 1 of 10 requested . Example(HEX): 0x3031
7	Separator Character	Always "|" (0x7C)
8 to 9	Number of Entries Expected	ASCII Encoded . Example(ASCII): "09", 9 more entries pending . Example(HEX): 0x3039
10	Separator Character	Always "|" (0x7C)
11 to 14	Actual Log Entry Number	ASCII Encoded . Example(ASCII): "0002", 2nd entry in log . Example(HEX): 0x30303032
15	Separator Character	Always "|" (0x7C)

Byte Index	Description	Note
16 to 17	Log Entry Type	Format (ASCII): "##" Value: “00” (Normal) Value: “01” (Over Voltage) Value: “02” (Under Voltage) Value: “03” (Over Current) Value: “04” (Under Current) Value: “05” (Over Temperature) Value: “06” (Under Temperature) Value: “07” (Surge Fault) Value: “08” (Auto Ping Fault) Value: “09” (RS-232 Ping Fail) Value: “10” (EPO Initiate) Value: “11” (EPO Recovery)
18	Separator Character	Always "|" (0x7C)
19 to 30	Log Date/Time with trailing comma	Format (ASCII): "MM/DD/YYYY HH:MM:SS," 1. MM is Month 2. DD is day 3. YYYY is 4 digit year 4. HH is hour 5. MM is Minutes 6. SS is seconds
31 to 51	Log Sensors, with trailing comma	Format (ASCII): "TTT,WWWW,F.F,VVR,CC.R,LLLL.L,O," 7. TTT is temperature in Fahrenheit 8. WWWW is Wattage 9. F.F is Power Factor 10. VVR is RMS Voltage 11. CC.R is RMS Current 12. LLLL.L is Thermal Load 13. O is Occupancy

Byte Index	Description	Note
52 to 82	Log of Energy Management States	Format (ASCII): "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16" 14. 1 is Outlet 1 Energy Management State 15. 2 is Outlet 2 Energy Management State 16. 3 is Outlet 3 Energy Management State 17. 4 is Outlet 4 Energy Management State 18. 5 is Outlet 5 Energy Management State 19. 6 is Outlet 6 Energy Management State 20. 7 is Outlet 7 Energy Management State 21. 8 is Outlet 8 Energy Management State 22. 9 is Outlet 9 Energy Management State 23. 10 is Outlet 10 Energy Management State 24. 11 is Outlet 11 Energy Management State 25. 12 is Outlet 12 Energy Management State 26. 13 is Outlet 13 Energy Management State 27. 14 is Outlet 14 Energy Management State 28. 15 is Outlet 15 Energy Management State 29. 16 is Outlet 16 Energy Management State
83	Checksum	Sum of all previous bytes (0 to 11), masked with 0x007F
84	Tail Character	Always 0xFF

Log Count (0x81), Clear Log (0x82)

Log Clear and Count Commands

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03
2	Destination	Always 0x00
3	Command	0x81 - Get Log Count 0x82 - Clear Log
4	Subcommand	0x01 - Set (Clear Log) 0x02 - Get (Get Log Count)
5	Checksum	Sum of all previous bytes (0 to 4), masked with 0x007F
6	Tail Character	Always 0xFF

Log Clear and Count Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03 for Clear Log Response Always 0x07 for Log Count Response
2	Destination	Always 0x00
3	Command	0x81 - Get Log Count 0x82 - Clear Log
4	Subcommand	0x10 - Response
5 to 8	Data (for log count only)	ASCII Encoded Size of Data . Example(ASCII): "0120", 120 entries . Example(HEX): 0x30313230, 120 entries
9	Checksum	Sum of all previous bytes (0 to 4), masked with 0x007F
10	Tail Character	Always 0xFF

Product Part Number(0x90), Rating (0x91), Surge (0x93), IP Address (0x94), MAC Address (0x95)

Product Rating and Information

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	Always 0x03
2	Destination	Always 0x00
3	Command	0x90 - Part Number 0x91 - Product Amp Hour Rating 0x93 - Product Surge Existence 0x94 - Current IP Address 0x95 - MAC Address
4	Subcommand	0x02 - Get Info
5	Checksum	Sum of all previous bytes (0 to 4), masked with 0x007F
6	Tail Character	Always 0xFF

Product Rating and Information Response

Byte Index	Description	Note
0	Start Character	Always 0xFE
1	Length	. Variable length . Length of Bytes 2 to X
2	Destination	Always 0x00
3	Command	0x90 - Part Number (string of up to 50 chars) 0x91 - Product Amp Hour Rating (###) 0x93 - Product Surge ("Y"/"N") 0x94 - Current IP Address (###.###.###.###) 0x95 - MAC Address (##:##:##:##:##:##)
4	Subcommand	0x10 - Response
5	Response Data	Variable size ASCII String representing requested data. Note: IP Address is variable length, no leading zeroes.
6	Checksum	Sum of all previous bytes (0 to 5), masked with 0x007F
7	Tail Character	Always 0xFF