RACKLINK™ SERIES

SERIAL COMMUNICATIONS PROTOCOL

Middle Atlantic Products

Middle Atlantic Products
I-00472 Rev C

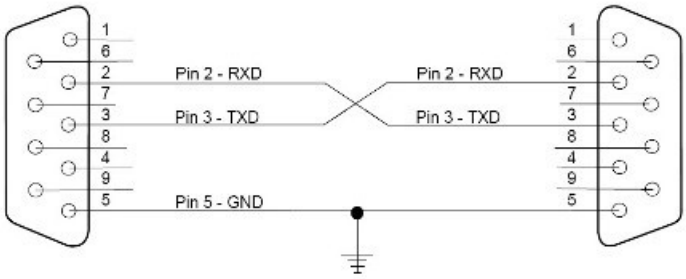
Table of Contents

Protocol Transports

RS232 Specifications

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

RS232 Cabling



TCP/IP

- TCP Port: 60000

Protocol Document Notes

The general format of this protocol is hexadecimal, although the Data Envelope portion of the protocol will contain some ASCII characters for the purposes of reporting the varied data values.

Throughout this document, all hexadecimal values will be displays using 0x in front of the actual value. All ASCII string literal examples will be show with double quotes at the beginning and end of the series of characters. Neither of these identifiers are actually part of the protocol and are used just for clarification.

The symbols < and > will be used to indicate separation of protocol sections and are also not part of the protocol and will be used throughout this document for clarity purposes.

The term “You” that is used throughout this document refers to the client side programmer.

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, accept 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 0x14 0x00 0x02 0x01 “UserName|Password” 0x04 0xff

In order to calculate the checksum correctly, you must add up all of 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.

Elongated Example

sum = 0 ; (clear your variable)

sum = sum + 0xfe; (Header)
sum = sum + 0x14; (Length)
sum = sum + 0x00;
sum = sum + 0x02;
sum = sum + 0x01;
sum = sum + 0x55; // “U” == 0x55
sum = sum + 0x73; // “s” == 0x73
sum = sum + 0x65; // “e” == 0x65
sum = sum + 0x72; // “r” == 0x72
sum = sum + 0x4e; // “N” == 0x4e
sum = sum + 0x61; // “a” == 0x61
sum = sum + 0x6d; // “m” == 0x6d
sum = sum + 0x65; // “e” == 0x65
sum = sum + 0x7c ; // “|”== 0x7c
sum = sum + 0x50; // “P” == 0x50
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

sum = sum & 0x7f (Bitwise “and” to keep least significant 7 bits)

Answer: sum = 0x04

We elongated the checksum example 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;
}
```

Escaped Data

(The protocol has been developed to limit the chances of actually needing escaping. This section may not be needed, but added for the off chance that it will be needed.)

To protect key portions of this protocol we have three protected byte values that cannot be used anywhere in the protocol except for their designated purpose. If the value must be used, than the value must get escaped.

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. So when calculating both length and checksum, you must remove and revert the escape sequences prior to your calculations. Examples will be provided.

Protected Values

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

Escaping a Data Value

When escaping a data value the value must equal one of the three protected values, see 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)

So we add the escape byte (0xfd) before the value that needs escaping, and invert the bits of the value to complete the escaped process. You will notice that the length and the checksum did not change.

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

Data Envelope Message Structure

Format

<address><cmd><sub cmd><data_n0 … data_n(x)>

- Address
  - Future
  - Values: 0-127 (0 will be used for the initial release)
  - Data Length: 1 Byte
- Cmd
  - Values: 0x0-0xfc (See Command Tables)
  - Data Length: 1 Byte
- Sub Cmd
  - Values: 0x0-0xfc (See Sub-Command Tables)
  - Data Length: 1 Byte
- Data Bytes
  - Valid Values: 0-255 (See Data Tables, and Escaped Section)
  - Data Length: 0 – 247 Bytes

Command Table

Command	Byte Value	Valid Sub Command
Nack	0x10	Response
Ping, Pong	0x01	Set, Response
Login	0x02	Set, Response
Power Outlet	0x20	Set, Get, Response, Status Change
Power Outlet Name	0x21	Set, Get, Response
Power Outlet Count/Controlled	0x22	Get, Response
Power Outlet Device Energy Management State	0x23	Set, Get, Response
Dry Contacts	0x30	Set, Get, Response, Status Change

Dry Contacts Name	0x31	Set, Get, Response
Dry Contacts Count /Controlled	0x32	Get, Response
*Input Sense	0x33	Get, Response, Status Change
*Input Sense Name	0x34	Set, Get, Response
*Input Sense Count /Controlled	0x35	Get, Response
Sequence Power Outlets	0x36	Set, Get, Response, Status Change
(EPO) Emergency Power Off	0x37	Set, Get, Response, Status Change
Register Log Alerts	0x40	Set, Get, Response
Register Status Change	0x41	Set, Get, Response
Current Kilowatt Hours	0x50	Get, Response, Status Change
Current Peak Voltage	0x51	Get, Response, Status Change
Current RMS Voltage	0x52	Get, Response, Status Change
Current Peak Load	0x53	Get, Response, Status Change
Current RMS Load	0x54	Get, Response, Status Change
Current Temperature	0x55	Get, Response, Status Change
Current Wattage	0x56	Get, Response, Status Change
Current Power Factor	0x57	Get, Response, Status Change
Current Thermal Load	0x58	Get, Response, Status Change
Current Surge Protection State	0x59	Get, Response, *Status Change
Current Energy Management State	0x60	Get, Response
Occupancy State	0x61	Set, Get, Response
Low Voltage Threshold	0x70	Set, Get, Response, Status Change
High Voltage Threshold	0x71	Set, Get, Response, Status Change
Maximum Load	0x73	Set, Get, Response, Status Change
Minimum Load	0x74	Set, Get, Response, Status Change
Maximum Temperature	0x76	Set, Get, Response, Status Change
Minimum Temperature	0x77	Set, Get, Response, Status Change
Log Entry	0x80	Get, Response, Alert
Log Count	0x81	Get, Response, *Status Change
Clear Log	0x82	Set, Response
Product Part Number	0x90	Get, Response
Product Rating	0x91	Get, Response
Product Surge Protection	0x93	Get, Response
Product Current IP Address	0x94	Get, Response
Product MAC Address	0x95	Get, Response

*Future

Sub-Command Table

Sub-Command	Byte Value
Set	0x01
Get	0x02
Response	0x10
Status Change	0x12
Log Alert	0x30

Data Bytes – NACK (0x10)

Sub-Commands	Quantity Data Bytes Required
Set	N / A
Get	N / A
Response	1 Bytes
Status Change	N / A
Log Alert	N / A

Nack (1 Byte)

- Byte 1
  - Error Code
    - 0x01 Bad Checksum
    - 0x02 Bad Length
    - 0x03 Escaped Error
    - 0x04 Invalid Command
    - 0x05 Invalid Sub-Command
    - 0x06 Invalid Qty Data Bytes
    - 0x07 Invalid Data Byte Values
    - 0x08 Access Denied (Credentials)
    - 0x10 Unknown
    - 0x11 Access Denied (EPO)

Data Bytes – Ping/Pong (0x01)

Sub-Commands	Quantity Data Bytes Required
Set	0 Bytes
Get	0 Bytes
Response	0 Bytes
Status Change	N / A
Log Alert	N / A

Data Bytes – Login (0x02)

Sub-Commands	Quantity Data Bytes Required	
Set	Variable	50 Max
Get	N / A	
Response	1 Byte	
Status Change	N / A	
Log Alert	N / A	

Login (Set)

- Bytes 1 – x (Variable - 50 Bytes Max)
  - Format: (Pipe separating the name and password)
    - "UserName|Password"

Login (Response)

- Byte 1
  - Value: 0x00 (Denied)
  - Value: 0x01 (Accepted)

Data Bytes – Power Outlet (0x20)

Sub-Commands	Quantity Data Bytes Required
Set	6 Bytes
Get	1 Byte
Response	6 Bytes
Status Change	6 Bytes
Log Alert	N / A

Power Outlet (Get)

- Byte 1:
  - Outlet Number: 0x1 – 0x8 (1-8)

Power Outlet (Set, Response, Status Change)

- Byte 1:
  - Outlet Number: 0x1 – 0x8 (1-8)
- Byte 2:
  - State:
    - 0x00 = OFF
    - 0x01 = ON
    - 0x02 – Cycle
    - 0x03 – Not Controllable (Response Only)
- Byte 3 – Byte 6:

- On/Off
  - ASCII Encoded Decimal
  - Valid Value: “0000”
- Cycle Time:
  - ASCII Encoded Decimal
  - “0000” - “3600”

Data Bytes – Power Outlet Name (0x21)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	Variable	51 Bytes
Get	1 Byte	
Response	Variable	51 Bytes
Status Change	N / A	
Log Alert	N / A	

Power Outlet Name (Get)

- Byte 1:
  - Outlet Number 0x01 – 0x08 (1-8)

Power Outlet Name (Set, Response)

- Byte 1:
  - Outlet Number 0x01 – 0x08 (1-8)
- Byte 2 – x: ASCII Name (50 Bytes Max)

Data Bytes – Power Outlet Count /Controlled (0x22)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	8 Bytes	
Status Change	N / A	
Log Alert	N / A	

Power Outlet Count – Controlled (Response)

- Byte 1:
  - Outlet 1
    - C = Controllable
    - N = Non-Controllable
    - X = Does not Exist
- Byte 2:
  - Outlet 2
    - C = Controllable

        - N = Non-Controllable
        - X = Does not Exist
- Byte 3:
    - Outlet 3
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 4:
    - Outlet 4
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 5:
    - Outlet 5
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 6:
    - Outlet 6
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 7:
    - Outlet 7
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 8:
    - Outlet 8
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist

Example: “CCNNNNXX”

Data Bytes – Power Outlet Device Energy Management State (0x23)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	2 Bytes	
Get	1 Byte	
Response	2 Bytes	
Status Change	N / A	
Log Alert	N / A	

Power Outlet Device Energy Management State (Get)

- Byte 1:
  - Dry Contact Number: 0x1 – 0x8

Power Outlet Device Energy Management State (Set, Response)

- Byte 1:
  - Dry Contact Number: 0x1 – 0x8
- Byte 2:
  - Energy Management State
    - Value: "D" (Disconnected)
    - Value: "S" (Standby)
    - Value: "I" (On)
    - Value: "O" (Off)
    - Value: "U" (Unknown)

Data Bytes – Dry Contacts (0x30)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	6 Bytes	
Get	1 Byte	
Response	6 Bytes	
Status Change	6 Bytes	
Log Alert	N / A	

Dry Contacts (Get)

- Byte 1:
  - Dry Contact Number: 0x1 – 0x8

Dry Contacts (Set, Response, Status Change)

- Byte 1:
  - Dry Contact Number: 0x1 – 0x8
- Byte 2:
  - State:
    - 0x00 = OFF

        - 0x01 = ON
        - 0x02 – Cycle
        - 0x03 – Not Controllable (Response Only)
- Byte 3 – Byte 6:
    - On/Off
        - ASCII Encoded Decimal
        - Valid Value: “0000”
    - Cycle Time:
        - ASCII Encoded Decimal
        - “0000” - “3600”

Data Bytes – Dry Contacts Name (0x31)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	Variable	51 Bytes
Get	1 Byte	
Response	Variable	51 Bytes
Status Change	N / A	
Log Alert	N / A	

Dry Contact Name (Get)

- Byte 1:
  - Outlet Number 0x01 – 0x08 (1-8)

Dry Contact Name (Set, Response)

- Byte 1:
  - Outlet Number 0x01 – 0x08 (1-8)
- Byte 2 – x: ASCII Name (50 Bytes Max)

Data Bytes – Dry Contact Count - Controlled (0x32)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	8 Bytes	
Status Change	N / A	
Log Alert	N / A	

Dry Contact Count – Controlled (Response)

- Byte 1:
  - Dry Contact 1
    - C = Controllable
    - N = Non-Controllable
    - X = Does not Exist
- Byte 2:
  - Dry Contact 2
    - C = Controllable
    - N = Non-Controllable
    - X = Does not Exist
- Byte 3:
  - Dry Contact 3
    - C = Controllable

        - N = Non-Controllable
        - X = Does not Exist
- Byte 4:
    - Dry Contact 4
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 5:
    - Dry Contact 5
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 6:
    - Dry Contact 6
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 7:
    - Dry Contact 7
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist
- Byte 8:
    - Dry Contact 8
        - C = Controllable
        - N = Non-Controllable
        - X = Does not Exist

Example: “CCNNNNXX”

Data Bytes – Sequence Power Outlets (0x36)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	5 Bytes	
Get	0 Bytes	
Response	5 Bytes	
Status Change	5 Bytes	
Log Alert	N / A	

Sequence Power Outlets (Set)

- Byte 1:
  - State:
    - 0x01 = UP
    - 0x03 = DOWN
- Byte 2 – Byte 5
  - Delay Time (Delay Time Between Outlets)
    - ASCII Encoded Decimal
    - "0000" – "0999"

Sequence Power Outlets (Response, Status Change)

- Byte 1:
  - State:
    - 0x00 = No Sequence Status (fallback value)
    - 0x01 = Sequencing Up
    - 0x02 = Sequence Up Complete
    - 0x03 = Sequencing Down
    - 0x04 = Sequence Down Complete
- Byte 2 – Byte 5
  - Delay Time (Delay Time Between Outlets)
    - ASCII Encoded Decimal
    - "0000" – "0999"

Data Bytes – Emergency Power Off (EPO) (0x37)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	1 Byte	
Get	0 Bytes	
Response	1 Bytes	
Status Change	1 Bytes	
Log Alert	N / A	

Emergency Power Off (Set)

- Byte 1:
  - State:
    - 0x00 = EPO Recover
    - 0x01 = EPO Initiate

Emergency Power Off (Response, Status Change)

- Byte 1:
  - State:
    - 0x00 = Normal State
    - 0x01 = Emergency Power Off Mode

Data Bytes – Register Log Alerts (0x40)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	2 Bytes	
Get	0 Bytes	
Response	2 Bytes	
Status Change	N / A	
Log Alert	N / A	

Register Log Alerts (Set, Response)

- Byte 1:
  - BITMASK
    - 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
- Byte 2:
  - BITMASK

- 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

Data Bytes – Register Status Change (0x41)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	6 Bytes	
Get	0 Bytes	
Response	6 Bytes	
Status Change	N / A	
Log Alert	N / A	

Register Status Change (Set, Response)

- Byte 1:
  - BITMASK
    - 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
    -
- Byte 2:
  - BITMASK
    - 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
- Byte 3:
  - BITMASK
    - 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
- Byte 4:
    - BITMASK
        - 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
- Byte 5:
    - BITMASK
        - 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 6:
    - BITMASK
        - 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

*Future

Data Bytes – Current Kilowatt Hours (0x50)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	12 Bytes	
Status Change	12 Bytes	
Log Alert	N / A	

Current Kilowatt Hours (Response, Status Change)

- Byte 1 – Byte 12:
  - ASCII Encoded Double Number
    - Ten Digits before the dot, and one digit after.
  - Example: “0000010200.1”

Data Bytes – Current Peak Voltage (0x51)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Current Peak Voltage (Response)

- Byte 1 – Byte 3:
  - ASCII Encoded Decimal Number
  - Example: “010”

Data Bytes – Current RMS Voltage Changes (0x52)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Current RMS Voltage Changes (Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal Number
  - Example: “010”

Data Bytes – Current Peak Load (0x53)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	4 Bytes	

Status Change	4 Bytes
Log Alert	N / A

Current Peak Load (Response, Status Change)

- Byte 1 – Byte 4
  - ASCII Encoded Float Number
  - Example: “66.1” (Two Digits before the dot and one digit after dot)

Data Bytes – Current RMS Load (0x54)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	4 Bytes	
Status Change	4 Bytes	
Log Alert	N / A	

Current RMS Load (Response, Status Change)

- Byte 1 – Byte 4
  - ASCII Encoded Float Number
  - Example: “66.1” (Two Digits before the dot and one digit after dot)

Data Bytes – Current Temperature (0x55)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Current Temperature (Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal Number
  - Example: “098”

Data Bytes – Current Wattage (0x56)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	4 Bytes	

Status Change	4 Bytes
Log Alert	N / A

Current Wattage (Response, Status Change)

- Byte 1 – Byte 4
  - ASCII Encoded Decimal Number
  - Example: "1234"

Data Bytes – Current Power Factor (0x57)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	4 Bytes	
Status Change	4 Bytes	
Log Alert	N / A	

Current Power Factor (Response, Status Change)

- Bytes 1 – Bytes 4
  - ASCII Encoded Float Number
  - Example: "0.98" (One Digit before the dot and two digits after dot)

Data Bytes – Current Thermal Load (BTU) (0x58)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	6 Bytes	
Status Change	6 Bytes	
Log Alert	N / A	

Current Thermal Load (BTU) (Response, Status Change)

- Byte 1 – Byte 6
  - ASCII Encoded Float Number
  - Example: "1234.5" (Four Digits before the dot and one digit after dot)

Data Bytes – Current Surge Protection State (0x59)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	1 Bytes	

*Status Change	1 Bytes
Log Alert	N / A

*Future

Current Surge Protection State (Response, Status Change)

- Byte 1
  - Value: 0x00 (Not Supported)
  - Value: 0x01 (Protected)
  - Value: 0x02 (Compromised)

Data Bytes – Current Energy Management State (0x60)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	9 Bytes	
Status Change	N / A	
Log Alert	N / A	

*Future

Current Energy Management State (Response)

- Byte 1
  - Occupancy State
    - Value: “U” (Unoccupied)
    - Value: “O” (Occupied)
- Byte 2
  - Outlet 1 Energy Management State
    - Value: “D” (Disconnected)
    - Value: “S” (Standby)
    - Value: “I” (On)
    - Value: “O” (Off)
    - Value: “U” (Unknown)
- Byte 3
  - Outlet 2 Energy Management State
    - Value: “D” (Disconnected)
    - Value: “S” (Standby)
    - Value: “I” (On)
    - Value: “O” (Off)
    - Value: “U” (Unknown)
- Byte 4
  - Outlet 3 Energy Management State
    - Value: “D” (Disconnected)
    - Value: “S” (Standby)
    - Value: “I” (On)
    - Value: “O” (Off)

        - Value: “U” (Unknown)
- Byte 5
    - Outlet 4 Energy Management State
        - Value: “D” (Disconnected)
        - Value: “S” (Standby)
        - Value: “I” (On)
        - Value: “O” (Off)
        - Value: “U” (Unknown)
- Byte 6
    - Outlet 5 Energy Management State
        - Value: “D” (Disconnected)
        - Value: “S” (Standby)
        - Value: “I” (On)
        - Value: “O” (Off)
        - Value: “U” (Unknown)
- Byte 7
    - Outlet 6 Energy Management State
        - Value: “D” (Disconnected)
        - Value: “S” (Standby)
        - Value: “I” (On)
        - Value: “O” (Off)
        - Value: “U” (Unknown)
- Byte 8
    - Outlet 7 Energy Management State
        - Value: “D” (Disconnected)
        - Value: “S” (Standby)
        - Value: “I” (On)
        - Value: “O” (Off)
        - Value: “U” (Unknown)
- Byte 9
    - Outlet 8 Energy Management State
        - Value: “D” (Disconnected)
        - Value: “S” (Standby)
        - Value: “I” (On)
        - Value: “O” (Off)
        - Value: “U” (Unknown)

Data Bytes – Occupancy State (0x61)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	1 Bytes	
Get	0 Bytes	
Response	1 Bytes	
Status Change	N / A	
Log Alert	N / A	

Occupancy State (Set, Response)

- Byte 1
  - Occupancy State
    - Value: “U” (Unoccupied)
    - Value: “O” (Occupied)

Data Bytes – Low Voltage Threshold (0x70)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	3 Bytes	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Low Voltage Threshold (Set, Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal Number
  - Example: “105”
  - Valid Range: “105” – “110”

Data Bytes – High Voltage Threshold (0x71)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	3 Bytes	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

High Voltage Threshold (Set, Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal
  - Example: “128”
  - Valid Range: “123” – “128”

Data Bytes – Maximum Load Current (0x73)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	4 Bytes	
Get	0 Bytes	
Response	4 Bytes	

Status Change	4 Bytes
Log Alert	N / A

Maximum Load Current (Set, Response, Status Change)

- Byte 1 – Byte 4
  - ASCII Encoded Float
  - Example: "15.0"
  - Valid Range: "00.0" – "15.0" (15 amp models)
  - Valid Range: "00.0" – "20.0" (20 amp models)

Data Bytes – Minimum Load Current (0x74)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	4 Bytes	
Get	0 Bytes	
Response	4 Bytes	
Status Change	4 Bytes	
Log Alert	N / A	

Minimum Load Current (Set, Response, Status Change)

- Byte 1 – Byte 4
  - ASCII Encoded Float
  - Example: "00.0"
  - Valid Range: "00.0" – "15.0" (15 amp models)
  - Valid Range: "00.0" – "20.0" (20 amp models)

Data Bytes – Maximum Temperature (0x76)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	3 Bytes	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Maximum Temperature (Set, Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal
  - Example: "100"
  - Valid Range: "000" – "250"

Data Bytes – Minimum Temperature (0x77)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	3 Bytes	
Get	0 Bytes	
Response	3 Bytes	
Status Change	3 Bytes	
Log Alert	N / A	

Minimum Temperature (Set, Response, Status Change)

- Byte 1 – Byte 3
  - ASCII Encoded Decimal
  - Example: “65”
  - Valid Range: “000” – “250”

Data Bytes – Log Entry (0x80)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	7 Bytes	
Response	60 Bytes	
Status Change	N / A	
Log Alert	60 Bytes	

Log Entry (Get)

- Byte 1 – Byte 4
  - Beginning Log Entry
  - ASCII Encoded Decimal
  - Valid Range: “0001” – “2000”
- Byte 5:
  - Valid Value: “|” (Pipe)
- Byte 6 – Byte 7
  - Return Count
  - ASCII Encoded Decimal
  - Valid Range: “00” – “99”

Log Entry (Response, Log Alert)

- Byte 1 – Byte 2
  - Request Item Number (referenced from the count)
  - ASCII Encoded Decimal
  - Valid Range: “00” – “99”
- Byte 3:
  - Valid Value: “|” (Pipe)

- Byte 4– Byte 5
  - Quantity Items to Receive
  - ASCII Encoded Decimal
  - Valid Range: “00” – “99”
- Byte 6
  - Valid Value: “|” (Pipe)
- Byte 7 – Byte 10
  - Log Entry Number
  - ASCII Encoded Decimal
  - Valid Range: “0000” – “2000”
- Byte 11
  - Valid Value: “|” (Pipe)
- Byte 12 – Byte 13
  - Log Entry Category
  - ASCII Encoded Decimal
  - Valid Range: “00” - “09”
    - 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)
- Byte 14
  - Valid Value: “|” (Pipe)
- Byte 15 – Byte 78
  - Log Data
  - Various ASCII Data
  - Description:
    - MM/DD/YYYY HH:MM:SS,TTT,WWWW,F.F,VVR,CC.R,LLLL.L,O,1,2,3,4,5,6,7,8
      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
      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
      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

Data Bytes – Log Count (0x81)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	4 Bytes	
*Status Change	4 Bytes	
Log Alert	N / A	

*Future

Log Count (Response)

- Byte 1 – Byte 4
  - ASCII Encoded Decimal
  - Valid Range: “0000” – “2000”

Data Bytes – Clear Log (0x82)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	0 Bytes	
Get	N/A	
Response	0 Bytes	
Status Change	N / A	
Log Alert	N / A	

Data Bytes – Product Part Number (0x90)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	Variable	50 Bytes
Status Change	N / A	
Log Alert	N / A	

Product Part Number (Response)

- Bytes 1 – x (50 Bytes Max)
  - ASCII Alpha Numeric

Data Bytes – Product Rating (0x91)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes

Set	N / A
Get	0 Bytes
Response	2 Bytes
Status Change	N / A
Log Alert	N / A

Product Rating (Amp hours) (Response)

- Byte 1 – Byte 2
  - ASCII Encoded Decimal
  - Valid Range: "00" – "99"

Data Bytes – Product Surge Protection (0x93)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	1 Bytes	
Status Change	N / A	
Log Alert	N / A	

Product Surge Protection (Does this product have surge protection) (Response)

- Byte 1
  - Valid Values
    - "Y" (Yes)
    - "N" (No)

Data Bytes – Product Current IP Address (0x94)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	
Get	0 Bytes	
Response	Variable	15
Status Change	N / A	
Log Alert	N / A	

Product Current IP Address (Response)

- Byte 1 – x (Max 15)
  - Example: "192.168.100.10"

Data Bytes – Product MAC Address (0x95)

Sub-Commands	Quantity Data Bytes Required	Max Data Bytes
Set	N / A	

Get	0 Bytes
Response	17 Bytes
Status Change	N / A
Log Alert	N / A

Product MAC Address (Response)

- Byte 1 – Byte 17
    - Example: “58:b0:35:6a:24:35”

Examples

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

Example - Initial Login Sequence

You Send: “Set” Login: (replace “UserName|Password” with actual values)

0xfe 0x14 0x00 0x02 0x01 “UserName|Password” 0x04 0xff

You Get: “Response” Login: (Accepted)

0xfe 0x04 0x00 0x02 0x10 0x01 0x15 0xff

You Get: “Set” Ping

0xfe 0x03 0x00 0x01 0x01 0x03 0xff

You Send: “Response” Ping (Pong)

0xfe 0x03 0x00 0x01 0x10 0x12 0xff

Example - Get Log Entries

You Send: Get Log Enties (Starting at Item 1 and return 10 items).

0xfe 0x0b 0x00 0x80 0x02 “0001|05” 0x2c 0xff

You Get: Response Log Entries (1 of 05, Log Entry 1)

0xfe 0x3c 0x00 0x80 0x10 "01|05|0001|00|06/25/11 10:59:47,075,0013,0.8,119,00.2,0046.4" 0x04 0xff

You Get: Response Log Entries (1 of 05, Log Entry 2)

0xfe 0x3c 0x00 0x80 0x10 "02|05|0002|00|06/29/11 11:37:02,076,0006,0.6,114,00.0,0023.2" 0x71 0xff

You Get: Response Log Entries (1 of 05, Log Entry 3)

0xfe 0x3c 0x00 0x80 0x10 "03|05|0003|00|06/29/11 11:36:19,076,0003,0.5,115,00.0,0012.2" 0x75 0xff

You Get: Response Log Entries (1 of 05, Log Entry 4)

0xfe 0x3c 0x00 0x80 0x10 "04|05|0004|00|06/29/11 11:31:04,076,0003,0.5,115,00.0,0012.2" 0x6c 0xff

You Get: Response Log Entries (1 of 05, Log Entry 5)

0xfe 0x3c 0x00 0x80 0x10 "05|05|0005|00|06/29/11 11:26:48,076,0003,0.5,115,00.0,0012.2" 0x7a 0xff

Example - Log Alert

You Get: Log Alert (Item 1 of 1 Log Entry 1777, Reason: RS232 Ping Failed)

0xfe 0x3c 0x00 0x80 0x30 "01|01|1777|09|06/29/11 11:31:04,076,0003,0.5,115,00.0,0012.2" 0x4a 0xff

Example – Control Outlet

You Send: "Set" Power Outlet (Turn on Outlet 1)

0xfe 0x09 0x00 0x20 0x01 0x01 0x01 "0000" 0x6a 0xff

You Get: "Response" Power Outlet

0xfe 0x09 0x00 0x20 0x10 0x01 0x01 "0000" 0x79 0xff

You Send: "Set" Power Outlet (Turn off Outlet 2)

0xfe 0x09 0x00 0x20 0x01 0x02 0x00 "0000" 0x6a 0xff

You Get: "Response" Power Outlet

0xfe 0x09 0x00 0x20 0x10 0x02 0x00 "0000" 0x79 0xff

You Send: "Set" Power Outlet (Cycle Outlet 3 for 5 Seconds)

0xfe 0x09 0x00 0x20 0x01 0x03 0x02 “0005” 0x72 0xff

You Get: “Response” Power Outlet (Outlet 3 is off)

0xfe 0x09 0x00 0x20 0x010 0x03 0x00 “0005” 0x7f 0xff

Example - Register Status Change

(Unsolicited Status Change)

You Send: “Set” Register Status (Example: Outlet and Dry Contact Status Change Only)

0xfe 0x09 0x00 0x41 0x01 0x01 0x01 0x00 0x00 0x00 0x00 0x4b 0xff

You Get: “Response” Register Status

0xfe 0x09 0x00 0x41 0x10 0x01 0x01 0x00 0x00 0x00 0x00 0x5a 0xff

Example - Showing Unsolicited Response

(Based on Status Registration Above)

You Send: “Set” Power Outlet (Turn on Outlet 1)

0xfe 0x09 0x00 0x20 0x01 0x01 0x01 “0000” 0x6a 0xff

You Get: “Response” Power Outlet

0xfe 0x09 0x00 0x20 0x10 0x01 0x01 “0000” 0x79 0xff

(You will NOT get a “Status Change” message in this case because the RackLink device already told you in the above message that the outlet is on)

You Send: “Set” Power Outlet (Cycle Outlet 3 for 5 Seconds)

0xfe 0x09 0x00 0x20 0x01 0x03 0x02 “0005” 0x72 0xff

You Get: “Response” Power Outlet (Outlet 3 is cycling, this response also indicates the outlet is currently off)

0xfe 0x09 0x00 0x20 0x010 0x03 0x02 “0005” 0x01 0xff

(Five Seconds Later)

You Get: “Status Change” Power Outlet (Outlet 3 is on, also this response indicates that the cycling period is over)

0xfe 0x09 0x00 0x20 0x012 0x03 0x001“0005” 0x02 0xff