App Note 128

Using the PulseWorx Gateway with your own applications

The PulseWorx Gateway, like the PIM and the PIM-IP can be used to send UPB commands to as well as receive commands from the power line. While the PIM-IP and The Gateway (PGW) both have a network and not a serial connection, the PulseWorx Gateway is a much different product than the PIM-IP. It is possible to ignore all the Gateway's features and use it only as a means for sending and receiving UPB messages but its commands are different than the PIM. It is not a direct replacement for the PIM or PIM-IP.

This app note focuses only on what you need to know to connect to the Gateway, to send UPB messages, and to receive UPB messages. For all the other features that the Gateway has you should refer to the Gateway Development Kit that is available by request from PCS technical support.

Note: This is an advanced topic and is only needed by users who are interfacing their own applications to the Gateway.

Proper preparation prevents poor performance
The first step is to make sure that you have downloaded the complete documentation about the PIM, UPB and the Gateway. These documents can be found on the PCS web site under the following titles in the resources section.

- Powerline Interface Module PIM Description
- UPB Technology Description
- Gateway Firmware design

Before starting on this note, please review the app note titled "Applications that use the PIM to send and receive messages" as much of that note applies to the Gateway and will not be duplicated here.

It also would be a major help to have a recent version of the Windows PulseWorx Application installed on your computer as it contains a tool – described later - that is useful to this work.

Differences between the Gateway and the PIM
There are several major differences between the Gateway and the PIM. These are outlined here.

IP vs Serial
The most obvious difference is that the Gateway has a network connection and not a serial connection. To connect to the Gateway, a TCP/IP connection to the Gateway's IP address and port must be made. There is a method where you can discover the Gateway using a UDP message and that is documented in the firmware design document. For this app note it is assumed that you know the Gateway's IP address and port and you can create a connection to that from your application.

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

Secure Connection
When you first open the connection to the Gateway it uses a connection protocol where you must authenticate your application. Once past that point, all messages sent and received use a command protocol described in the next section.

After the TCP/IP is made to the Gateway, the application begins the connection protocol by sending a message called the "Client Hello" message and is defined as follows:

```
ClientName/FWVer/ProtocolVers\0

For Example: "UPSTART/7.0.16/1:2:3"\0
```

Note: Messages sent and received during the connection are in ascii printable test and are terminated with a zero byte.

ClientName is any string with a reasonable length. It is used to differentiate one client app from another. As of the date of this document, the Gateway ignores this field. In the future, it may be desired to know when a particular app is connecting. Following the client name is a slash and the client's software version, separated by another slash, and the protocol versions it supports. Each protocol version (if more than one) is separated by a colon. The Gateway command protocol will be versioned to allow for future enhancements. If the protocol changes, it will be assigned a new protocol version number. Some reasons to do this are: adding new commands, changing the format of messages, and adding/improving features. The Gateway will select the highest protocol version it supports from the list provided and reply with that version. If the Gateway supports none of the protocols requested, it will reply with protocol version set to zero. The connection will then be terminated. As of the date of this document, only protocol version 1 exists.

As an example, here is what the Windows PulseWorx-App sends to the Gateway after the socket is open.

50 75 6C 73 65 57 6F 72 78 20 41 70 70 2F 31 2E 30 2E 31 2F 31 00

This decodes to "PulseWorx App/1.0.1/1<0>" and is the name of the connecting application, the application version major dot minor dot build, and the protocol wanted.

There are also several possible errors the Gateway may send in response to the client hello message. These are – in these exact text strings.

- MAX CONNECTIONS REACHED
- PULSE MODE ACTIVE

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

- PIM NOT INITIALIZED
- FIRMWARE CORRUPT – FLASH WITH UPSTART

If the Gateway sends one of these messages it then disconnects the client. The only message that signals that the connection should be retried is the "PIM NOT INITIALIZED" and within a few seconds your application should attempt the connection again. If "PULSE MODE ACTIVE" is returned it means that an application like UPStart is currently using the Gateway and it can be the only connection. It is important that all these errors get reported to the user in some form.

If the message is incomplete or the Gateway is unable to parse it for any reason, it replies with "INCOMPLETE MESSAGE"\0 and disconnect. Otherwise, it replies with the "Server Hello" message as follows:

```
PCS PIM-IP2/{FWver}/{ProtocolVer}/{SomeOtherStuff}\0
```

```
For example: PCS PIM-IP2/1.0/1/ . . . \0
```

In this example, the Gateway's firmware version is 1.0, and it supports protocol version 1.

The "SomeOtherStuff" field indicates whether authentication is required or not. If it is, then the message includes a 64-byte MD-5 challenge, encoded in ASCII. What determines whether authentication is needed is the presence of at least one username/password pair in the users table programmed into the Gateway by UPStart. If there are no username/password pairs, or if the table doesn't exist, then authentication is not required.

When authentication is not required, the "SomeOtherStuff" field contains "AUTH NOT NEEDED/x CLIENTS", where x is 1-8 and is the number of clients already connected. If authentication is needed, it contains "AUTH REQUIRED/" and followed by the 128 characters representing the 64-byte challenge.

```
For example: PCS PIM-IP2/1.0/1/AUTH NOT NEEDED/2 CLIENTS\0
```

This example shows authentication is not needed and two clients are already connected. This connection is the third. At this point, the client is connected and may send/receive data using the command protocol. Another example:

```
PCS PIM-IP2/1.0/1/AUTH REQUIRED/30E2FED3 . . . 783EA827\0
```

This example shows authentication is needed. "30E2FED3 . . . 783EA827" is 128 chars long. The client converts the ASCII challenge to binary (64 bytes) and computes an MD-5 hash using the password. Details for doing are in the Gateway Firmware Design document. The client returns the MD-5 response as shown in this example:

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

```
username/46B3D1F8D93C450572682DC48203FFE0
```

Where “username” is the name of the user making the connection, and must be one of the users in the user table in the Gateway as configured by UPStart. The characters following the slash are 32 ASCII characters encoding the 16-byte MD-5 response.

If the response does not match what the Gateway expected, the device will issue an “AUTHENTICATION FAILED\0” message and disconnect. If it succeeds, the device will issue an “AUTH SUCCEEDED/x CLIENTS\0” message, where x is the number of clients already connected.

The Gateway allows up to eight usable TCP connections simultaneously. A ninth connection is allowed just long enough to tell the client the limit of eight has been reached. The message is “MAX CONNECTIONS REACHED\0”. That connection is then terminated.

If any client is already connected and has put the Gateway in Pulse Mode, no new connections will be allowed. Other clients attempting to connect will receive the message “PULSE MODE ACTIVE\0.”

Here is an example of a connection that required authentication. This is what the client sends to the Gateway after the socket is open.

50 75 6C 73 65 57 6F 72 78 20 41 70 70 2F 31 2E 30 2E 31 2F 31 00

This decodes to “PulseWorx App/1.0.1/1<0>” and is the name of the connecting application, the application version major dot minor dot build, and the protocol wanted. This is what comes back from the Gateway.

50 43 53 20 50 49 4D 2D 49 50 32 2F 31 2E 30 2F 31 2F 41 55 54 48 20 52 45 51 55 49 52 45 44 2F 34 39 32 36 44 31 30 35 32 31 30 35 30 35 46 43 37 31 33 33 37 42 39 31 30 30 44 46 42 41 37 43 33 30 44 45 32 45 34 33 32 38 39 41 36 42 42 41 34 38 44 35 45 44 39 43 32 35 41 33 33 33 31 45 33 39 31 34 46 41 46 35 34 31 37 36 34 42 36 42 31 44 45 31 38 36 30 32 37 46 33 37 33 42 46 38 34 32 30 30 34 41 43 41 30 37 42 39 30 45 37 31 34 31 46 31 33 37 36 45 34 33 46 39 41 39 37 43 00

Here is the decode
PCS PIM-IP2/1.0/1/AUTH REQUIRED/4926D105210505FC71337B9100DFBA7C30DE2E43289A6BBA48D5ED9C25A3331E3914FAF541764B6B1DE186027F373BF842004ACA07B90E7141F1376E43F9A97C

The “PCS PIM-IP2” is stock phase it always sends. The “1.0“ is the gateway firmware version, the “1” is the protocol and then the phase “AUTH REQUIRED” followed by a bunch of stuff which is the challenge key.

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

To process this, convert the challenge key from 128 characters to 64 bytes. Then hash it with the password. The "Gateway Firmware Design" doc describes how this is done.

The next message sent to the Gateway is the username, a slash, and then the 16 bytes from the hash rendered into 32 characters. Again, terminates by a zero byte.

6B 69 6D 62 65 72 6C 79 2F 38 45 33 43 39 39 37 42 44 44 39 30 37 45 37 41 30 43 43 39 44 44 38 37 37 33 42 41 41 42 45 44 00

This decodes to:
kimberly/8E3C997BDD907E7A0CC9DD8

If all goes well a message comes back a message from the Gateway:
41 55 54 48 20 53 55 43 43 45 45 44 45 44 2F 30 20 43 4C 49 45 4E 54 53 00

This decodes to:
AUTH SUCCEEDED/0 CLIENTS

And the connection is made.

Command Protocol

Once a connection is established and authenticated then all subsequent messages sent and received from the Gateway use the Gateway command protocol.

Messages sent to and received from the Gateway are enclosed in a packet similar but different in important ways from those used by the PIM.

The message structure is:

- 1 byte command
- 2 bytes data length
- Command data from 0 bytes to 65534 bytes
- 1 byte checksum

Important differences than the PIM:

- The messages are binary and not in "hex text" like the PIM message are.
- There is no message delimiter. The length of the message tells you when the complete message has been received.

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

- The data length is the count of the number of bytes to follow so it does not count the three bytes that precede it – the command byte and the length bytes.
- The Gateway message checksum is the 1s complement of the eight-bit sum of the bytes of the message starting at the command byte and ending at the last data byte.

The 2 bytes of the length are encoded with the high byte first. For example, a message that has a length of 20 bytes would have the length bytes encoded as 00 14.

A reply is generated for all commands received by the Gateway. The reply indicates success or failure. A success reply may or may not include data. There are two replies that indicate failure: one is a NAK and the other contains an error code. The first byte of any reply that is not a NAK is the original received command code plus one, followed by a two-byte data length, a single byte success or error code, optional data, and a checksum. The checksum is calculated in the same way as described above. The data portion of the message contains a zero for success or any other number to indicate a specific error.

A NAK is sent when the message is not understood, such as an incomplete message or a bad checksum. The reason NAKs are separate from other error replies is that the command received may be invalid. Therefore, the Gateway cannot reply with the command +1. NAK replies will include an error code indicating the problem.

For successful replies with no extra data but only the 1 byte showing success:
`(CMD+1)(0001)(00)(Checksum)`

For successful replies with a success code plus data:
`(CMD+1)(length)(00)(x x x … x)(Checksum)`

For error replies:
`(CMD+1)(0001)(Error Code)(Checksum)`

NAK replies:
`(FF)(0001)(NAK Reason)( Checksum)`

The NAK codes are:

- 0x01 BAD CHECKSUM
- 0x02 INCOMPLETE MESSAGE
- 0x03 UNKNOWN COMMAND

Sending UPB Messages

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

To send a UPB message you must first construct the exact string of bytes that you would send if you were using a serial PIM. This includes the <ctrl-t> character, the message bytes, the UPB message checksum, and the concluding <cr>. As in the case of the serial PIM, all except the <ctrl-t> and the<cr> are in "hex text".

Then take that string of characters and enclose it in a Gateway command packet that has command code 0x30. The Gateway message would start with 0x30, be followed by the 2 length bytes, then the UPB message that goes to the PIM and then concludes with the Gateway message checksum. Then send that to the Gateway.

The Gateway then replies that it received the command and it is a valid command – has a known command code and the checksum is correct. The message that comes back has command code 0x31 which is the command code for "send a message to the PIM" plus 1.

Once the Gateway passes the message on to the PIM, the PIM replies with the PA, PB, or PE message as usual and those are received by the Gateway and passed on to your application wrapped in a Gateway message as described in the next section.

Receiving UPB Messages

Messages that come from the PIM are sent to the application wrapped in a Gateway packet with command code 0xE0. It is important to know that there may be more than one PIM message in the data portion of the Gateway message so your application should plan for that and process as many PIM messages as are found there.

Disconnections due to Pulse Mode applications

While the Gateway supports multiple connections, if at any time an application connects and places the PIM into pulse mode, any other connections are dropped. All those other connections are sent a message before being disconnected. That message contains a command code of 0xf2 and a message length of zero, followed by the checksum byte. All an application can do upon receiving this message is to report to the user that this has happened and clean up its internal state.

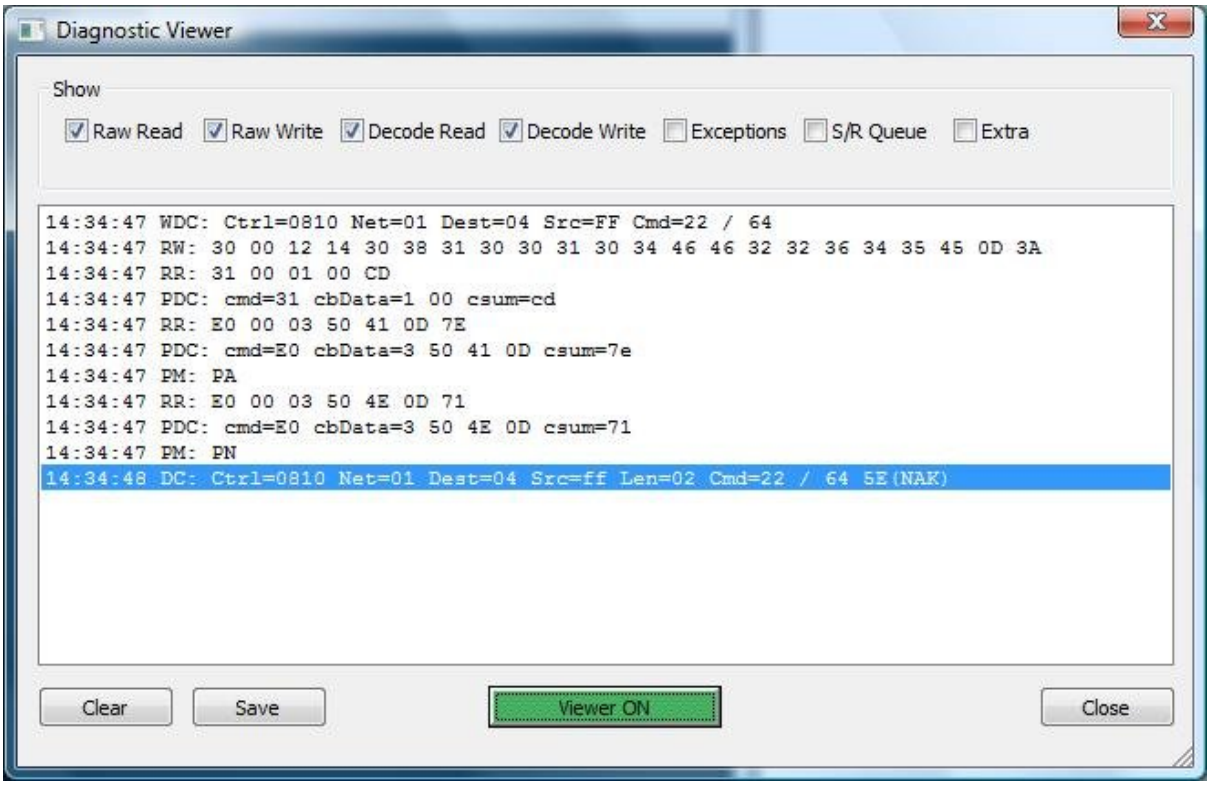
Tools for learning more about the PulseWorx Gateway

The best method to learn the Gateway protocol – aside from this documentation – is to perform actions with the PulseWorx-App and watch the data sent to and received from the Gateway.

To do this, press the Comm Viewer button in the Development category. This opens a viewer.

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

App Note 128



This trace shows what happens when the application controlled a device to 100%. The view shows:

Line 1, tagged “WDC”, is a Write Decode of the UPB packet to be sent.

Line 2, tagged “RW” is the Raw Write of the bytes sent to the Gateway.

Line 3, tagged “RR” is a Raw Read from the Gateway.

Line 4, tagged “PDC” is the Packet Decode of the Gateway message. In this case it is responding to say that the message was received and processed without error.

Line 5, tagged “RR” is again a Raw Read from the Gateway.

Line 6, tagged “PDC” is the Packet Decode of the Gateway message. In this case it is a message from the PIM.

Line 7, tagged “PM” is the PIM Message decoded. In this case it is “PA” which tells that the PIM accepted the message for transmission.

Line 8, tagged “RR” is the Raw Read of a message from the Gateway.

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com

Line 9, tagged “PDC” is a decode of that message. In this case another message from the PIM.

Line 10, tagged “PM” is the PIM message. In this case PN which means that the device didn’t ACK the command. This is ok. While having the ACK bit set in the message control word, this network being used for this test has a Split Phase Repeater (SPR) which renders the ACK bit meaningless.

Line 11, tagged “DC” is a Decode of the complete UPB message and the ACK/NAK.
As you can see this trace mechanism, while useful it can be very verbose. As such you can control the type of messages logged by using the checkboxes in the dialog.

Note: If when you install the PulseWorx0-App it doesn’t have the Development category do this:

1. Shutdown the application.
2. Start RegEdit and go to HKEY_CURRENT_USER – Software – PCSLighting – PulseWorxApp – UserInterface.
3. Look for the key DeveloperMode. Change to a value of 1.
4. Exit RegEdit.
5. Restart PulseWorx-App and the Development ribbon category will be shown.

##end##

19201 Parthenia St. Suite J
Northridge, CA 91324
P: 818.701.9831 F: 818.701.1506
pcssales@pcslighting.com
www.pcslighting.com