

Universal Powerline Bus



UPB Powerline Interface Module (PIM) Description

Preliminary

V 1.4

04/01/03

Table Of Contents

1. PIM Overview	1
2. PIM/Host Serial Communications	2
2.1. Overview	2
2.2. PIM/Host Serial Communications Philosophy	2
2.3. Receiving UPB Messages	3
2.4. Transmitting UPB Messages	3
2.5. Reading/Writing PIM Setup Registers	3
2.6. PIM/Host Serial Communication Messages	4
2.6.1. UPB Pulse Reports	4
2.6.2. Host-To-PIM Commands	6
2.6.3. PIM-To-Host Responses	8
3. PIM/Host Serial Communication Examples	8
3.1. Example 1: Receiving A UPB Communication Packet	9
3.2. Example 2: Transmitting A UPB Communication Packet	11
3.3. Example 3: Writing/Reading PIM Setup Registers	13
3.4. Example 4: Receiving UPB Noise	15
4. PIM Setup Registers	17
5. PIM Pushbutton	18
6. PIM LED Indicator	19
6.1. LED Indication Of Firmware Version	19
6.2. LED Indication Of UPB Status	19
6.3. LED Indication Of Special Modes	19

1. Scope

The document describes the Powerline Control Systems' (PCS) Powerline Interface Module (PIM). Included in this document are descriptions of PIM/Host Serial Communications, PIM Setup Registers, PIM Pushbutton operation, and PIM Status LED operation. Examples of PIM/Host Serial Communication are also included.

In order to fully understand some of the concepts discussed herein, the reader should have a good understanding of the UPB Communication Method (as presented in the [UPB Communication Method](#) document), and the UPB System Model (as presented in the [UPB System Model Description](#) document), and the UPB Message Structure (as presented in the [UPB Message Structure](#) document).

2. PIM Overview

The Powerline Interface Module (PIM) is a plug-in module designed to interface a host computer device to the powerline communications of the Universal Powerline Bus (UPB). UPB communication is a method, developed by PCS, to reliably communicate command and control information on a standard 60Hz AC electrical powerline. The PIM communicates to a host computer device via a standard RS-232 serial cable running at 4800 baud.

The PIM can be used to both transmit information onto the UPB (powerline) and to receive information from the UPB.



Figure 1: The Powerline Interface Module (PIM)

3. PIM/Host Serial Communications

3.1. Overview

The PIM can be used to both transmit information onto the UPB (powerline) and to receive information from the UPB. The PIM uses RS-232 serial communications between it and the host device to both receive commands from the host and to send responses and reports to the host (see Figure 2).

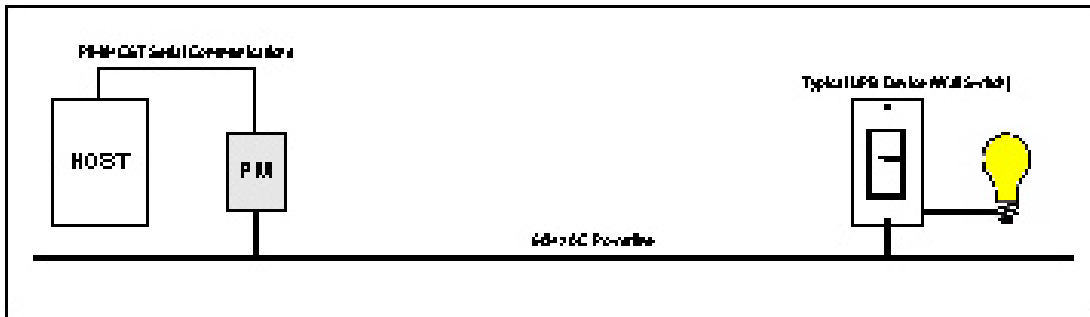


Figure 2: PIM/Host Communication Diagram

All PIM/Host communications are done serially at 4800 baud in the “N-8-1” format. All PIM/Host communications shall consist of ASCII characters. All PIM/Host communications shall end with a Carriage Return (<cr>) character. This section describes the PIM/Host serial communications in detail.

3.2. PIM/Host Serial Communications Philosophy

The general philosophy of the PIM/Host Serial Communications is to not “hide” any UPB information from the host. Instead of only communicating with the host device when a valid UPB Message has been received, the PIM informs the host of every UPB Pulse that it detects on the UPB (powerline). The PIM not only informs the host of the value (0, 1, 2, or 3) of the UPB Pulse, but it also informs the host of the relative strength (or energy) of that UPB Pulse. This, of course, places a much higher burden on the host device (to assemble the received UPB Pulses into a complete UPB Message and to validate it) but it also allows the host device the freedom (and power) to make intelligent decisions based on what is currently happening on the UPB.

To aid the host in determining the start and end of a UPB Communication Packet, the PIM sends special messages when it receives the Preamble Pulses and the ACK Pulse.

3.3. 3.4. Receiving UPB Messages

In its normal mode of operation, the PIM sends “UPB Pulse Reports” to inform the host whenever it has detected a valid UPB Pulse. The host can use these reports to build them into a received UPB Message and take action accordingly. It is up to the host device to validate the checksum and other fields of the received UPB Communication Packet (see section 4.1 - Example 1: Receiving A UPB Communication Packet).

A special “Dropped UPB Message “ report is sent by the PIM in the case where it begins receiving a

UPB Message but then the message ends prematurely (loss of signal, noise, etc.).

3.5. Transmitting UPB Messages

The host can send a “Transmit UPB Message” command (Table 2) to request the PIM to transmit a valid UPB Message onto the UPB. It is up to the host device to pass a valid UPB Message to the PIM for transmission. The PIM will validate this message and send back either a “PIM Accept” or a “PIM Error” response. The PIM will then transmit the UPB Message and send “UPB Pulse Reports” back to the host to indicate each UPB Pulse that was transmitted and also to inform the host of the presence or absence of an ACK Pulse at the end (see section 4.2 - Example 2: Transmitting A UPB Communication Packet).

3.6. Reading/Writing PIM Setup Registers

In addition, the PIM (like all UPB devices) has a block of 128 non-volatile registers, known as Setup Registers, which can be read from and/or written to using PIM/Host Communications (see section 4.3 - Example 3: Writing/Reading PIM Setup Registers).

Section 5 of this document describes the function of each of the PIM’s Setup Registers.

3.7. 3.8. PIM/Host Serial Communication Messages

The following section describes the PIM/Host Serial Communication Messages that have been implemented. These messages have been broken up into three separate classifications:

1. UPB Pulse Reports
2. Host-To-PIM Commands
3. PIM-To-Host Responses

3.8.1. UPB Pulse Reports

UPB Pulse Reports (Table 1) are serial messages that travel from the PIM to the Host to inform the Host of what is currently happening on the UPBus. These reports are sent in real-time (i.e. as each UPB Pulse is received). All UPB Pulse Reports are four characters long with a carriage return as the last (4th) character.

UPB Pulse Report	Report Syntax				Description
	1	2	3	4	
Received a Start Pulse (or Noise)	'X'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the UPBus is no longer Idle (A UPB Pulse is present). This can be due to either the start of a UPB Communications Packet or noise.
Received a Sync Pulse	'R'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a Sync Pulse and is about to start receiving a UPB Message.
Received a Data Pulse 0	'0'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a Data Pulse in Position #0 (Value = 0).
Received a Data Pulse 1	'1'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a Data Pulse in Position #1 (Value = 1).
Received a Data Pulse 2	'2'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a Data Pulse in Position #2 (Value = 2).
Received a Data Pulse 3	'3'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a Data Pulse in Position #3 (Value = 3).
Received an ACK Pulse	'A'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM received a valid ACK Pulse at the end of the UPB Message.
Received a NAK Pulse	'N'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM did not receive a valid ACK Pulse at the end of the UPB Message.
Dropped UPB Message	'D'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the PIM did not receive a valid Data Pulse when one was expected.
UPBus Idle	'_'	E ^[1]	S ^[2]	<cr>	Used to inform the Host that the UPBus is Idle (no Pulses are present). This report, by default, is not generated but can be enabled by writing the value 0x00 to Setup Register 0x70. Once enabled, writing any non-zero value to Setup Register 0x70 will disable this report.
Transmitted a Pulse	'T'	V	S ^[2]	<cr>	Used to inform the Host that the PIM transmitted a UPB Pulse. V = Pulse Value ('0' – '3').

Table 1: UPB Pulse Reports

^[1]The E in the 2nd character in Table 1 is a representation of the relative Energy of the UPB Pulse. It is represented as an ASCII character between '0' and 'F' where '0' represents a non-existent Pulse, '1' represents a very weak Pulse and 'F' represents a very strong Pulse.

[2] The S in the 3rd character in Table 1 is a representation of the relative Sequence of the UPB Pulse. This character is not implemented at this current time and serves only as a placeholder. This character will always be sent as an ASCII space character (' ') at this present time.

3.8.2.3.8.3. Host-To-PIM Commands

Host-To-PIM Commands (Table 2) are serial messages that travel from the Host to the PIM to request that the PIM perform some sort of action. Host-To-PIM Commands vary in length depending on how much information needs to be transferred. The PIM can be commanded to either transmit a UPB Message or to read or write its Setup Registers.

NOTE: All characters in Table 2 are in ASCII Hex encoding except for the following:

- <Cr> = Carriage Return character (0x0D)
- <Ctl-T> = DC4 character (0x14)
- <Ctl-R> = DC2 character (0x12)
- <Ctl-W> = ETB character (0x17)

Host-To-PIM Command	Command Syntax	Description
Transmit UPB Message	<Ctl-T>UU..KK<Cr>	<p>Used to request the PIM to transmit a given UPB Message onto the UPBus.</p> <p>UU..KK = A Valid UPB Message (see Note #1 below).</p> <p>The PIM will check this message for a proper checksum. If it fails the checksum test it will respond to the Host with a "PIM Error" response. If it passes the checksum test, the PIM will respond to the Host with a "PIM Accept" response and transmit the message onto the UPBus.</p> <p>The PIM will start sending "Transmitted a Pulse" reports to the Host until the UPB message ends followed by either a "Received an ACK Pulse" report or a "Received a NAK Pulse" report.</p>

Read PIM Registers	<Ctl-R>RRNNKK<Cr>	<p>Used to read one or more PIM Setup Registers.</p> <p>RR = Starting Register Number, NN = Number Of Registers, KK = Checksum.</p> <p>The PIM will check this message for a proper checksum. If it fails the checksum test it will respond to the Host with a "PIM Error" response. If it passes the checksum test, the PIM will read the specified registers and report them to the Host in a "PIM Register Report" response.</p>
Write PIM Registers	<Ctl-W>RRVV..KK<Cr>	<p>Used to write one or more PIM Setup Registers with the given value(s).</p> <p>RR = Starting Register Number, VV = Value(s) To Write (Note #2), KK = Checksum.</p> <p>The PIM will check this message for a proper checksum. If it fails the checksum test it will respond to the Host with a "PIM Error" response. If it passes the checksum test, the PIM will write the given Value(s) to the specified Register(s) and indicate completion with a "PIM Accept" response.</p>

Table 2: Host-To-PIM Commands

Note #1: The format for a valid UPB Message (UU..KK) is described in the [UPB Message Structure](#) document.

Note #2: Any amount (from one to sixteen) of Values (VV) can be supplied to write a block of Setup Registers.

3.8.4.3.8.5. PIM-To-Host Responses

PIM-To-Host Responses (Table 3) are serial messages that travel from the PIM to the Host to inform the Host of the status of the previous Host-To-PIM Command.

PIM-To-Host Response	Syntax	Description
PIM Accept	'P"A'<cr>	Used to inform the Host that the PIM accepted the previous command.
PIM Busy	'P"B'<cr>	Used to inform the Host that the PIM rejected the previous command because it was busy.
PIM Error	'P"E'<cr>	Used to inform the Host that the PIM rejected the previous command because it found an error in it.
PIM Register Report	'P"R'RRVV<cr>	Used to inform the Host of the contents of the PIM Registers previously requested by a "Read PIM Registers" command. RR = Starting Register Number, VV = Value(s) Read [Note #1].

Table 3: PIM-To-Host Responses

Note #1: Any amount (from one to sixteen) of Values (VV) can be returned when reporting a block of Setup Registers.

4. PIM/Host Serial Communication Examples

In order to better understand PIM/Host Serial Communication, some examples have been provided below. In these examples, the PIM/Host Serial Communications have been captured on a PC running Procomm Plus Terminal. Screen shots of the Procomm Plus Terminal have been provided to help get an understanding of what messages the PIM sends to the host. The Procomm Plus Terminal converts all serial communications to ASCII representation and converts all carriage returns to a combination of a carriage return and a line feed. This results in each PIM/Host Serial Communication message being displayed on a separate line of the Procomm Plus Terminal.

4.1. 4.2. Example 1: Receiving A UPB Communication Packet

When the PIM detects a UPB Communication Packet on the powerline it will send UPB Pulse Reports to the host device indicating exactly what pulses were seen on the powerline. In this example, the following UPB Message was transmitted by a remote UPB device and received by the PIM.

UPB Message = 83 88 01 03 03 20 FF FF D0

The screen shot in Figure 3 shows the UPB Pulse Reports that the PIM sent to the host as it received this UPB Message. In studying Figure 3, we will pay particular attention to the first character of each UPB Pulse Report. This first character indicates what type (value) of UPB Pulse was received by the PIM. The second character indicates the relative strength (energy) of the received UPB Pulse and is not discussed in this example.

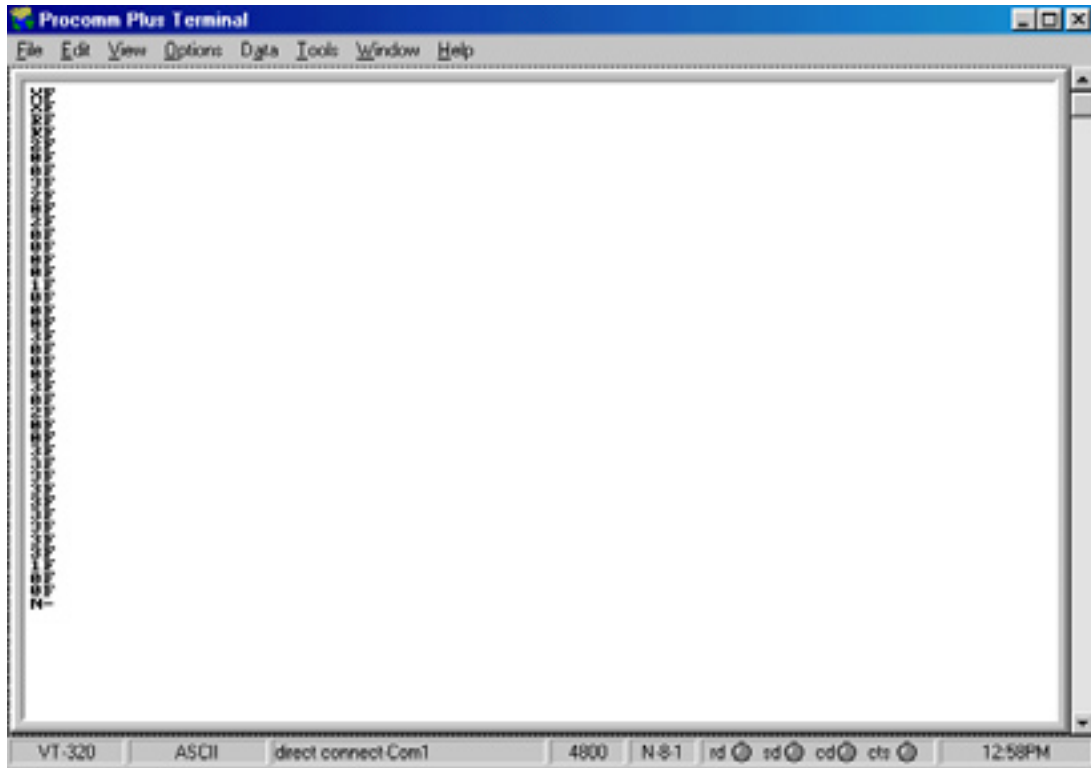


Figure 3: PIM/Host Communications - Receiving A UPB Communication Packet

As seen in the screen shot, the first four UPB Pulse Reports that the PIM sent were X-X-R-R. These UPB Pulse Reports indicate that the PIM received a valid Preamble to a UPB Communications Packet. The Preamble is always transmitted as 2-1-1-2 but, since the PIM is not yet synchronized to the transmitter, the PIM represents it as characters X-X-R-R. In fact, the sequence of two UPB Pulse Reports in a row with a value of 'R' can be used as an indication of the start of a UPB Communications Packet.

NOTE: The actual Preamble is always X-X-R-R but since the R-R portion will never occur without the preceding X-X portion it is sufficient to look for an R-R to indicate a valid UPB Communications Packet start. The fact that there is a valid Preamble sequence is in no way assuring that the UPB Communications Packet will conclude successfully. These four Pulse Reports come in "real-time" and a UPB Communications Packet can terminate improperly for a variety of reasons. When a UPB Communications Packet terminates prematurely, the PIM will send a "Dropped UPB Message" Pulse Report to the host.

The next 36 UPB Pulse Reports represent the 9 bytes of the UPB Message. Each UPB Pulse Report represents two bits of the UPB Message. For example, looking at the first 4 of these 36 UPB Pulse Reports, we see that the PIM received a 2-0-0-3 which, represented in hexadecimal, is 0x83. This is the first byte of the UPB Message shown above.

The last UPB Pulse Report that the PIM sent was an 'N', which indicates that the PIM did not see an ACK Pulse on the powerline after the UPB Message ended. Had there been a receiving device that sent an ACK Pulse the PIM would have sent an 'A' in the UPB Pulse Report instead. In fact, a UPB Pulse Report with a value of either 'N' or 'A' can be used as an indication of the end of a UPB Communications Packet.

4.3. 4.4. Example 2: Transmitting A UPB Communication Packet

The host can request the PIM to transmit a UPB Communication Packet onto the powerline by sending it a valid "Transmit UPB Message" command (see Table 2). In this example, the PIM was sent a "Transmit UPB Message" command to request it to send the following UPB Message.

UPB Message = 83 88 01 03 03 20 FF FF D0

The screen shot in Figure 4 shows the PIM-To-Host Responses and the UPB Pulse Reports that the PIM sent back to the host as it received this command and transmitted the UPB Communication Packet onto the powerline.

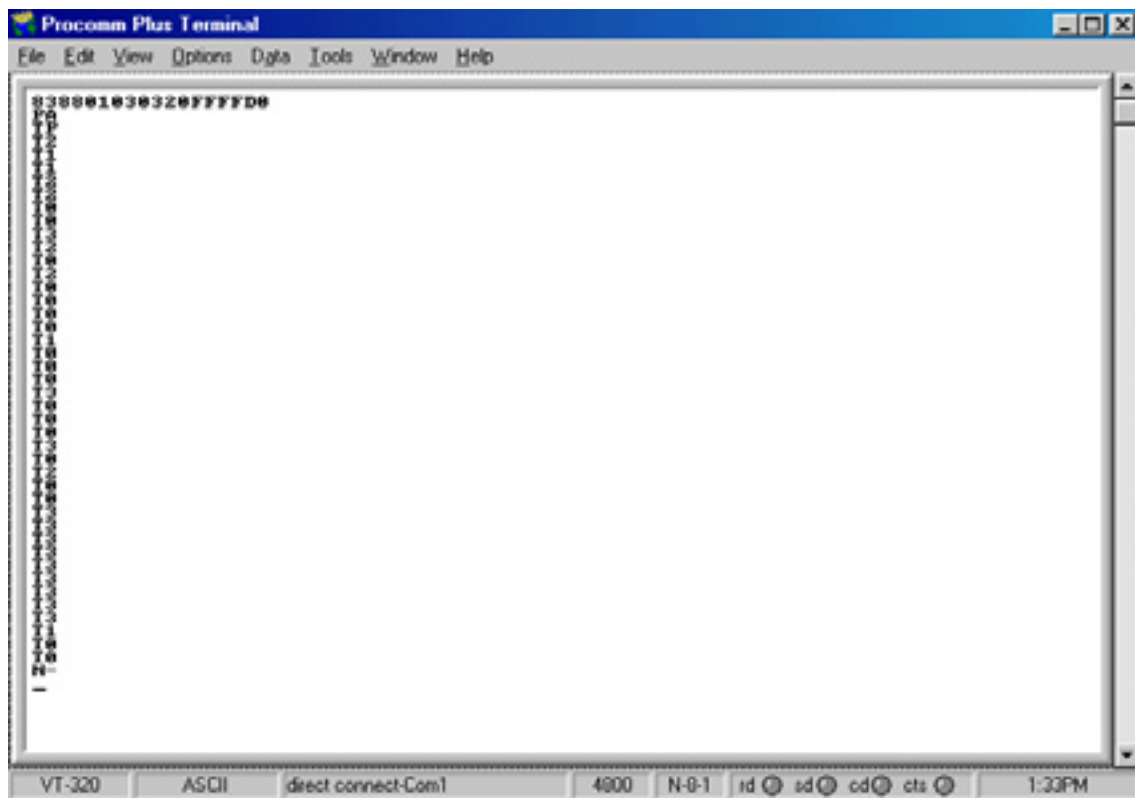


Figure 4: PIM/Host Communications - Transmitting A UPB Communication Packet

As seen in the screen shot, the first line after the "Transmit UPB Message" command is a "PA" (PIM Accept) response indicating that the PIM accepted the command (checksum was good). If there were something wrong with the command (checksum, etc.), the PIM would have responded with a "PE" (PIM Error) instead.

The next line (“TP”) is a UPB Pulse Report indicating that the PIM pre-charged its UPB Transmit circuit in preparation to transmit the UPB Communication Packet. This is not a particularly important message but it is sent by the PIM anyway.

The next four lines (2-1-1-2) are UPB Pulse Reports that indicate that the PIM transmitted a valid Preamble byte of the UPB Communications Packet.

The next 36 UPB Pulse Reports represent the 9 bytes of the UPB Message. Each UPB Pulse Report represents two bits of the UPB Message. For example, looking at the first 4 of these 36 UPB Pulse Reports, we see that the PIM transmitted a 2-0-0-3 which, represented in hexadecimal, is 0x83. This is the first byte of the UPB Message shown above.

The last UPB Pulse Report that the PIM sent was an ‘N’, which indicates that the PIM did not see an ACK Pulse on the powerline after the UPB Message ended. Had there been a receiving device that sent an ACK Pulse the PIM would have sent an ‘A’ in the UPB Pulse Report instead. In fact, a UPB Pulse Report with a value of either ‘N’ or ‘A’ can be used as an indication of the end of a UPB Communications Packet.

4.5. 4.6. Example 3: Writing/Reading PIM Setup Registers

The PIM (like all UPB devices) has a block of 128 non-volatile registers, known as Setup Registers, which can be read from and/or written to. In this example, a value of 00 will be written to Setup Register #0 (using the “Write PIM Registers” command) and then all 128 Setup Registers will be read (using the “Read PIM Registers” command).

The “Write PIM Registers” command (see Table 2) has the form: <ctl-W> 00 00 00.

The “Read PIM Registers” command (see Table 2) has the form: <ctl-R> 00 80 80.

The screen shot in Figure 5 shows the PIM-To-Host Responses that the PIM sent back to the host as it received the two commands. The first line after receiving the “Write PIM Registers” command shows that the PIM sent back a “PA” (PIM Accept) response indicating that the PIM accepted the command. If there were something wrong with the command (checksum, etc.), the PIM would have responded with a “PE” (PIM Error) instead.

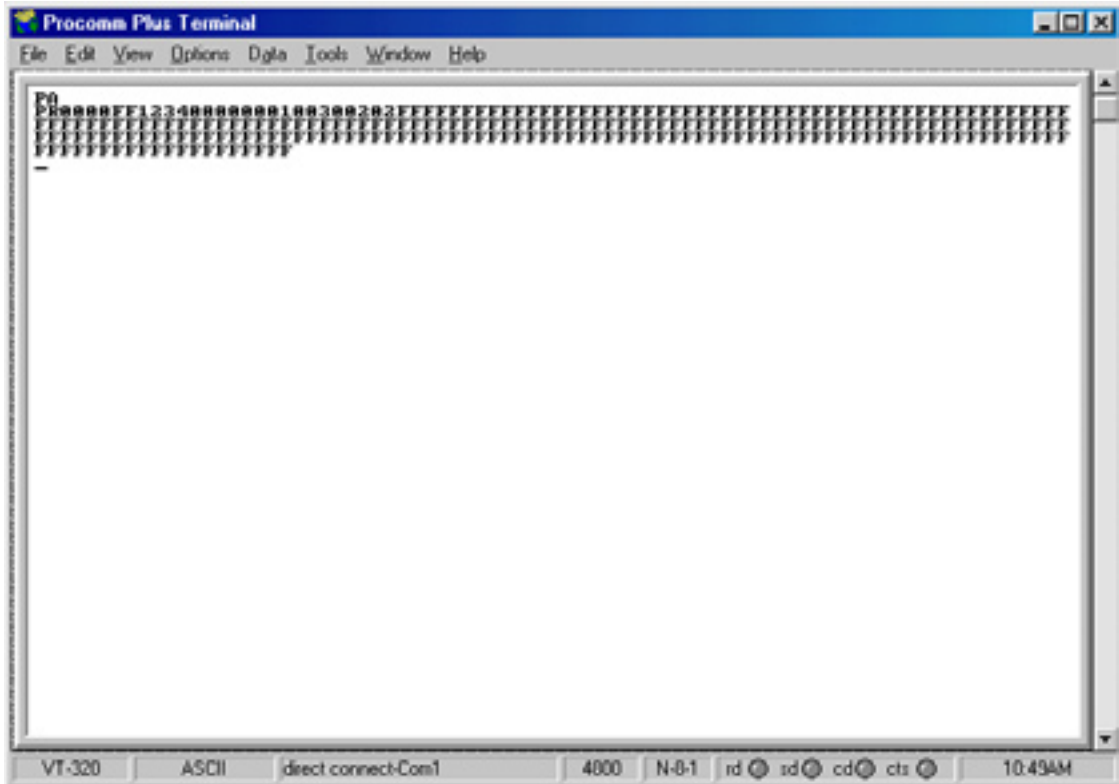


Figure 5: PIM/Host Communications – Writing/Reading PIM Setup Registers

The following lines after the PIM Accept response shows that the PIM sent back a “PIM Register Report” (see Table 3) that included the values of all 128 Setup Registers starting at Register #0.

Note that Setup Register #0 contains the value 0x00 indicating that the previous “Write PIM Registers” command was successful. Note also that Setup Registers 0x0A and 0x0B contain the value 0x0202 indicating that the PIM firmware version is 2.02.

4.7. Example 4: Receiving UPB Noise

The host can use the PIM to make measurements of the amount of UPB Noise that is currently on the powerline. UPB Noise is defined as any UPB Pulse that is on the powerline that is not part of an actual UPB Communications Packet. This powerline noise can be caused by many sources including fluorescent ballasts, electric motors, light switches, etc.

The PIM will identify a UPB Noise Pulse as a UPB Pulse Report with a value of 'X'. Since this same UPB Pulse Report could be sent as an indication of the start of a valid UPB Communications Packet, the host should make the distinction that it is actually noise by verifying that the rest of a UPB Communications Packet does not follow it.

UPB Noise measurements can be performed in many different fashions. One possible method is for the host to monitor for UPB noise over a set period of time (e.g. 1 second). A count of noise reports in that set period of time can then be an indication as to the relative noise on the powerline. Another method would be to enable UPBus Idle Reports and count the ratio of the number of noise reports to the number of idle reports as an indication as to the relative noise on the powerline. The E value (signal strength) of the noise reports can also factor in to your noise calculations.

In this example writing 0x00 to the PIM's Setup Register 0x70 enabled UPBus Idle Reports. While UPBus Idle Reports are enabled, the PIM sends one UPB Pulse Report every AC half-cycle (or 8.33mS). The host can now easily identify UPB Noise as any UPB Pulse Report with a value of 'X' that is not followed by a UPB Communications Packet.

The screen shot in Figure 6 shows the UPB Pulse Reports that the PIM sent to the host as it detected one UPB Noise Pulse on the powerline. As seen in the screenshot, the PIM was sending the host many UPBus Idle Reports ("-") and then one UPB Pulse Report with a value of 'X' that was followed by another UPBus Idle Report. The E value for the noise pulse in this example was 1, which is a relatively weak pulse.

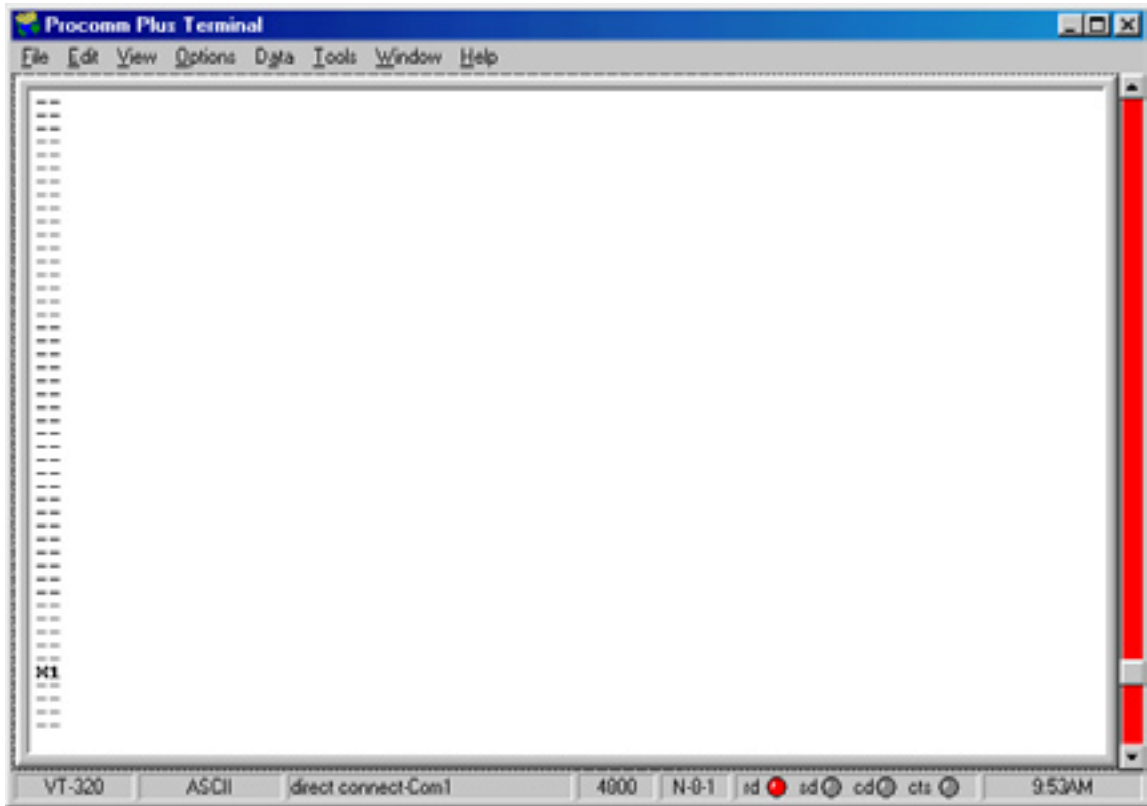


Figure 6: PIM/Host Communications – Receiving Powerline Noise

5. 6. PIM Setup Registers

The PIM, like all UPB devices, has a block of 128 non-volatile registers, known as Setup Registers, which can be read from and/or written to. Currently, the PIM only uses a small portion of its available Setup Registers as defined in Table 4. All other Setup Registers are reserved at this time (set to 0xFF) and should not be written to.

Setup Registers Field Name	Reg #	Factory Defaults (Hexadecimal)	Description
Network ID (NID)	0x00	FF	Unique identifier for the UPB Network this device communicates on.
Module ID (MID)	0x01	FF	Unique identifier for this UPB device.
Network Password	0x02 – 0x03	1234	Password designed to keep unauthorized users from accessing the Setup Registers of this device.
UPB Class ID	0x04	00	TBD
UPB Type ID	0x05	00	TBD
Manufacturer ID	0x06 – 0x07	0001	Unique identifier of the manufacturer of this UPB device (PCS).
Product ID	0x08 – 0x09	0030	The manufacturer's unique product identifier for this UPB device.
Version Number	0x0A – 0x0B	Current Firmware Version (i.e. 0202)	The manufacturer's unique version number of this Product ID.
Reserved1	0x0C – 0x6F	All FF's	Unused. Reserved for future.
Idle Report Filter	0x70	FF	Indicates if UPBus Idle Reports should be filtered.
Reserved2	0x71 – 0x7F	All FF's	Unused. Reserved for future.

Table 4: PIM Setup Registers

. PIM Pushbutton

The PIM has a momentary pushbutton switch on its front panel (labeled “Program”) that is used to take the PIM in and out of certain modes as defined in Table 5 below.

Number of Clicks	PIM Operation	Description
1	Stop SETUP Mode	Takes the PIM out of the SETUP or Factory Default Mode.
5	Start SETUP Mode	Puts the PIM into the SETUP Mode for 5 minutes. While in this mode, the PIM blinks its LED green and automatically responds to special setup commands from a UPB setup tool.
10	Set Factory Defaults	Puts the PIM into the Factory Default Mode. Once in this mode, the PIM’s Setup Registers have been written to their factory default values and the PIM blinks its LED red.

Table 5: PIM Pushbutton Operation

8. 9. PIM LED Indicator

The PIM has a bi-color (red/green) LED indicator on its front panel (labeled “Status”) that is used to indicate the current status of the PIM.

9.1. LED Indication Of Firmware Version

When power is first applied, the PIM will use the status LED to indicate the version of the firmware it is running. It will do this by blinking the LED four times. Each blink will either be red or green. The four blinks will be such as to indicate the binary value that matches the least significant digit of the firmware version. A blink of red shall indicate a binary “zero” and a blink of green shall indicate a binary “one”. As an example, if the firmware version is 2.05, then the LED will indicate the binary value for 5, which is zero-one-zero-one. The PIM will therefore blink its LED four times upon power-up: red-green-red-green.

9.2. LED Indication Of UPB Status

The PIM will also use its LED to indicate the status of the UPB network. Whenever the PIM transmits a UPB message, it will indicate so by pulsing the status LED red. Whenever the PIM receives a valid UPB message, it will indicate so by pulsing the status LED green. Whenever the PIM receives a rejected UPB message, it will indicate so by pulsing the status LED black (off).

9.3. LED Indication Of Special Modes

The PIM has two special modes that it can be put into: SETUP Mode and Factory Default Mode. When the PIM is in SETUP Mode it will indicate so by blinking its status LED alternately between green and black. When the PIM is in Factory Default Mode it will indicate so by blinking its status LED alternately between red and black.