

UDP Local Communication Interface for the PVM1010

1.0 Overview

This document explains the UDP local communication interface for the PVM1010 data communication module for PV Powered inverters. The PVM1010 uses a connectionless protocol to handle direct on-site communication. The protocol is standard UDP. Communication is not persistent; when the user initiates a communication request, the PVM1010 gives one reply to that user. This document will cover: the real time data transmission, inverter scale factors and end user license agreement.

2.0 Terms and Definitions

UDP – User Datagram Packet. A connectionless protocol that works over the top of standard IP.

Port – Internet UDP port 14917. UDP port numbers range from 1 – 65535.

Endian – The order of the most significant bytes in a word. PVM1010 uses little endian.

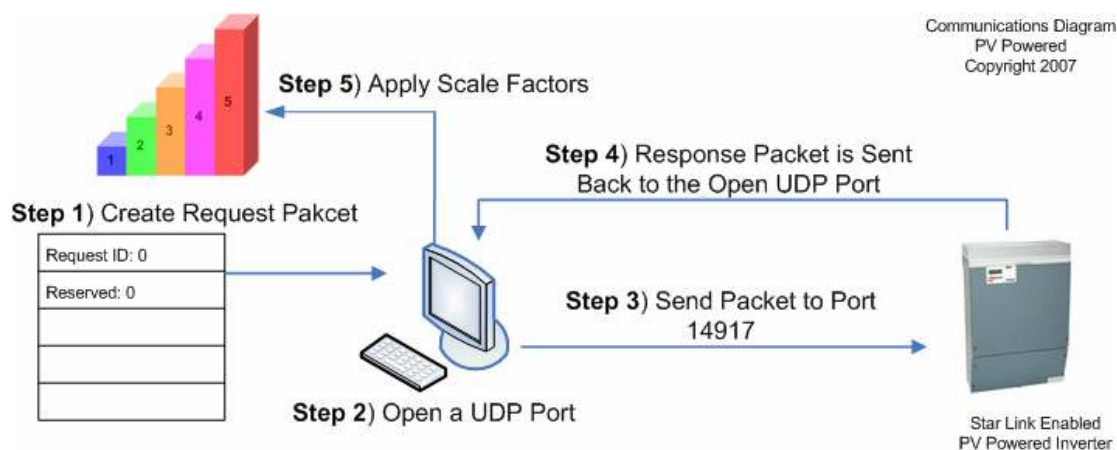
Ex. word = (byte0) * 256 + byte1, The number 1234 = (4) * 256 + 210. byte0 = 4 and byte1 = 210.

Word – A two byte set of data. A word can range in value from 0 to 65535.

Packet – A collection of words. The request packet is 2 words, and the response packet is 18 words.

3.0 Communication Flow

- 1.) A request packet is created on the users local computer
- 2.) A UDP port is opened on the users computer
- 3.) The packet is sent to a PVM1010 over port 14917 from the open UDP port on the user's computer
- 4.) The PVM1010 interfaces with the inverter over the RS232 interface to collect the requested data
- 5.) A response containing the requested data packet is sent from the PVM1010 over the IP interface
- 6.) The response is received on the users computer
- 7.) The data in the packet gets scale factors applied to it



(Figure 3.0)

UPD Local Communication Interface for the PVM1010

4.0 Packet Structure

4.1 Request Packet (sent by the user)

- Word 0: Echo - This word will echo back any value you give it. This is useful when talking to multiple PVM1010's at the same time.
- Word 1: Reserved - Must be Zero

4.2 Response Packet (received by the user)

- Word 0: Echo – This is a number passed by the user on request.
- Word 1: Reserved
- Word 2: State – The inverters state (0x15 = Online, 0x80 = Error, All others are sleeping)
- Word 3: Fault Code2 – The inverters main fault field
- Word 4: AC Power, unscaled (If not faulted)
- Word 5: DC Voltage, unscaled
- Word 6: AC Current (AMPS), unscaled (If not faulted)
- Word 7: Temperature (Celsius), unscaled
- Word 8: AC Voltage, unscaled
- Word 9: Empty
- Word 10: Inverter Board serial number
- Word 11: Inverter Board type
- Word 12: Firmware version. - The first byte is the major version, the second byte is the minor version.
- Word 13: Inverter Model Type – This is the true type of the inverter
- Word 14: Inverter Model Serial Number (High Word) – True inverter serial number
- Word 15: Inverter Model Serial Number (Low Word) – True inverter serial number
- Word 16: Total kWh (High Word)
- Word 17: Total kWh (Low Word)

UPD Local Communication Interface for the PVM1010

5.0 Scale Factor

The values coming out of the inverter are un-scaled integers. To make these values useful, a scale factor must be applied. The scale factors change depending on the inverter the values come from. To apply the scale factors, use the following equation: $\text{Real Value} = \text{Scale Factor} * \text{Raw Value}$

* Total power's scale factor is always 1, Temperature's scale factor is always 0.01

Family	Inverter	ID	AC Power	AC Voltage	AC Current	DC Voltage
Legacy						
	PVP 1100	1	0.261016846	0.012420654	0.001495361	0.012329102
	PVP 1800	2	0.37487793	0.012420654	0.001495361	0.012329102
	PVP 2800-208	3	0.453491211	0.018493652	0.001495361	0.015441895
	PVP 2800-240	4	0.453491211	0.018493652	0.001495361	0.015441895
XV						
	PVP 2800XV	5	0.366210938	0.018493652	0.001495361	0.015441895
PVP Generation 2						
	PVP 2000	6	0.404418945	0.012420654	0.001495361	0.012329102
	PVP 2500	16	0.393005371	0.018493652	0.001495361	0.015441895
	PVP 2900	8	0.519958496	0.018493652	0.001495361	0.015441895
	PVP 3200	7	0.519958496	0.018493652	0.001495361	0.015441895
	PVP 3500	17	0.601043701	0.018493652	0.001495361	0.015441895
	PVP 4600	13	0.726318359	0.018493652	0.001495361	0.015441895
	PVP 4800	18	0.741699219	0.018493652	0.001495361	0.015441895
	PVP 5200	12	0.726318359	0.018493652	0.001495361	0.015441895
SPR Generation 2						
	SPR 2000	9	0.404418945	0.012420654	0.001495361	0.012329102
	SPR 2500	19	0.393005371	0.018493652	0.001495361	0.015441895
	SPR 2900	11	0.519958496	0.018493652	0.001495361	0.015441895
	SPR 3200	10	0.519958496	0.018493652	0.001495361	0.015441895
	SPR 3500	20	0.601043701	0.018493652	0.001495361	0.015441895
	SPR 4600	15	0.726318359	0.018493652	0.001495361	0.015441895
	SPR 4800	21	0.741699219	0.018493652	0.001495361	0.015441895
	SPR 5200	14	0.726318359	0.018493652	0.001495361	0.015441895
PVP Generation 3						
	PVP 2500	26	0.393005371	0.018493652	0.001495361	0.015441895
	PVP 3500	27	0.601043701	0.018493652	0.001495361	0.015441895
	PVP 4600	23	0.871185303	0.018493652	0.001495361	0.015441895
	PVP 4800	28	0.741699219	0.018493652	0.001495361	0.015441895
	PVP 5200	22	0.871185303	0.018493652	0.001495361	0.015441895
SPR Generation 3						
	SPR 2500	29	0.393005371	0.018493652	0.001495361	0.015441895
	SPR 3500	30	0.601043701	0.018493652	0.001495361	0.015441895
	SPR 4600	25	0.871185303	0.018493652	0.001495361	0.015441895
	SPR 4800	31	0.741699219	0.018493652	0.001495361	0.015441895
	SPR 5200	24	0.871185303	0.018493652	0.001495361	0.015441895

UPD Local Communication Interface for the PVM1010

6.0 Java Code Examples

Below is a Java code example showing all the steps in action.

```
import java.io.*;
import java.net.*;
import java.util.*;

public class Client {
    public static void main(String[] args) throws IOException {

        //Required program startup code
        if (args.length != 1) {
            System.out.println("Usage: java Client <Host>");
            return;
        }

        //Step 1) Create the request packet
        byte[] buffer = new byte[48];

        //Set all entries in buffer to zero
        for ( int i = 0; i < buffer.length; i++ )
            buffer[i] = 0;

        //Step 2) Create a datagram socket and open UDP port
        DatagramSocket socket = new DatagramSocket();

        //Step 3) Send the 4 byte request packet to port 14917
        InetAddress address = InetAddress.getByName(args[0]);
        DatagramPacket packet = new DatagramPacket(buffer, 4, address, 14917);
        socket.send(packet);
        System.out.println( "" );
        System.out.println( "Sending Packet to: "+ args[0] );

        //Step 4) Get the response packet from the PVM1010
        packet = new DatagramPacket(buffer, buffer.length);
        socket.receive(packet);
        System.out.println( "" );
        System.out.println( "Response Packet size: "+ packet.getLength() );
        System.out.println( "" );

        //Step 5) Apply scale factors (These scale factors are for a PVP 5200)
        int words[] = new int[18];
        buffer = packet.getData();
        socket.close();

        //Print output from the response packet scaled to a PVP 5200
        //Fill my words array
        for ( int w = 0; w < words.length; w++ )
            words[w] = getWord( buffer, w );

        System.out.println( "Inverter's Information" );
```

UPD Local Communication Interface for the PVM1010

```
//Inverter's Type
System.out.println( "\tType: "+ words[13] );

//Print out the inverter's status
if ( words[2] == 0x15 )
    System.out.println("\tStuats: Online");
else if ( words[2] == 0x80 )
    System.out.println("\tStuats: Faulted");
else
    System.out.println("\tStuats: Sleeping");

//Print the DC voltage
System.out.println( "\tDC Voltage: "+ ((double)words[5] * 0.015459) );
//Print the AC voltage
System.out.println( "\tAC Voltage: "+ ((double)words[8] * 0.018497) );
//Print the AC Power
if ( words[2] == 0x15 )
    System.out.println( "\tAC Power: "+ ((double)words[4] * 0.726318) );

//Print Temperature
System.out.println( "\tTemperature: "+ ((double)words[7] * 0.01) + " C");
//Print the total kWh
System.out.println( "\tTotal kWh: "+ ((words[16] << 16) + words[17]) );
}

// Get a single word from a buffer
static int getWord( byte buffer[], int word )
{
    int word2 = word * 2;

    //Make sure word is valid
    if ( word2 + 1 > buffer.length || word2 < 0 )
        return 0;

    //Return the requested word
    return ((buffer[word2 + 1] & 255) << 8) +
        (buffer[word2 + 0] & 255);
}
}
```

6.1 Running the example

- 1) Ensure JDK 1.5 or greater is installed on your machine
- 2) Save this program: Client.java
- 3) Compile the program: javac Client.java
- 4) Run the program: java Client 192.168.0.121



UPD Local Communication Interface for the PVM1010

7.0 License Agreement

IMPORTANT READ CAREFULLY:

This End-User License Agreement (EULA) is a legal agreement between you (either an individual or a single entity) and PV Powered for use of the UDP Local Communications Interface INSTRUCTIONS described above. By following, copying, or otherwise using the INSTRUCTIONS, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this Agreement, you are not authorized to use the INSTRUCTIONS to collect or use data from the inverter.

GRANT OF LICENSE

This EULA grants you the following rights and obligations:

USE AND COPIES

PV Powered grants to you the right to use the INSTRUCTIONS on a single installation site with properly installed equipment. Even though the UDP interface does not require an internet connect to function, you are still expected to create an end user account on www.mypvpower.com and register your inverters for warranty records and when possible enable an internet connection to the inverter for backup data monitoring service and support. Copies of these INSTRUCTIONS shall not be produced or distributed for other installations or used for commercial purposes without express written consent from PV Powered.

RESTRICTIONS

You may not rent or lease the INSTRUCTIONS or the functionality of, or benefits derived from, the use of the INSTRUCTIONS without the express written consent of PV Powered.

You may permanently transfer all of your rights under this EULA only in conjunction with a permanent transfer of your ownership of the PV Powered inverter from which you are gathering data.

TERMINATION

PV Powered may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the INSTRUCTIONS and cease to access inverter data over the UDP interface.

COPYRIGHT

All title and copyrights and intellectual property rights in and to the INSTRUCTIONS and any copies thereof are owned by PV Powered.

NO WARRANTY

ANY USE OF THE INSTRUCTIONS IS AT YOUR OWN RISK. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, PV POWERED DISCLAIMS ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PV POWERED BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INSTRUCTIONS, EVEN IF PV POWERED HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE



UPD Local Communication Interface for the PVM1010

EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

MISCELLANEOUS

The laws of the State of Oregon, U.S.A, govern this EULA.

Should you have any questions concerning this EULA, or if you desire to contact PV Powered for any reason, please send US Mail to:

PV Powered, Inc.
150 S.W. Scalehouse Loop #101
Bend, OR 97702