File DownloadPeriodicData3.doc
May 18, 2011

Purpose of document

This document describes how <u>programmers</u> can interpret the raw "Periodic Logged Data" that can be downloaded from the PentaMetric using their own programs. If you are using the PMComm program to download this data you do not need this document—because PMComm downloads it into a spreadsheet form with the data already interpreted as described here.

This document is believed to be accurate. If you discover errors please notify us at Bogart Engineering so we can correct them.
www.bogartengineering.com

1. How to interpret PentaMetric Periodic data.

For progammers that wish to use their own programs to download and interpret the PentaMetric Periodic Logged data.

Description of the type of logged data:

The "PentaMetric Periodic logged data" allows logging data at uniform periodic intervals. It is possible to select many sampling interval lengths from as small as 1 minute to 24 hours. Some types of data represent a "snapshot" of at the moment of the data measurement. Other type data is recorded continuously for the entire duration of each single interval.

The relative time of recording is included in the data file for each data point. In other words, the absolute time is not recorded, but the time interval between recorded events is recorded.

When selected, the following data will be recorded for the duration of each interval:
Amp hours 1, Amp hours2, Amp hours 3, Watt hours 1, Watt hours 2, amp hours 1 measures, for example, the total Amp-hours1 between two successive recorded times. If one hour measurement intervals are chosen, this is the same as "average current from channel 1 in the interval between two successive recorded events".

The following data represents a "snapshot" at the time of each recording: **Max Temperature, Min Temperature, Filtered Volts1, Filtered Volts2, Filtered Amps1**, **Battery % full.** For filtered items, the filter time is selected by PentaMetric program mode P16. Filter time choices may be 0, 0.5, 2 or 8 minutes.

It is possible to record all, or just some of the above data. Which data to be recorded is selected using the PMComm program (while connected to the PentaMetric), by choosing the "Program" option, using programs P40, P41 andP42. The desired sample interval is also selected here. If less data per record is selected, then more records may be recorded before the memory is filled. If it becomes filled, then the oldest data will be overwritten by the newest data. Information on how many records can be stored (before download) is described in section 6.c.1 of the PentaMetric instructions, as well as in the PMComm program in the program window in which you choose which data above you wish to record.

How to download the logged data:

The data can, of course, be downloaded into a comma delineated computer file by the "PMComm" program. Alternatively, for programmers the logged data can be downloaded using a program, such as "C", using several "Long Reads" of the data. "Long reads" are described in the following document on the Bogart Engineering web site: PentaMetricRS232SerialSpecWeb.doc.

The **entire PentaMetric memory** consists of 64 "pages" of 256 bytes each. The "long read" is used to access one or more full pages of this memory. Therefore, each "long read" will cause an integer multiple of 256 bytes, plus one byte checksum at the end to emerge as output at the RS232 port.

The **Periodic Data section** of memory is a subset of the above memory, and begins at page 3 (location 0x300) and goes through end of page 31 (decimal) through location 0x1fff, or 29 decimal pages total. That's 7424 decimal bytes. We will now shift into "hex" gear, and refer (unless otherwise qualified) to hex numbers.

The Periodic data is further grouped into "sections" of 0x40 bytes each, as will be soon described. Each 0x40 byte "section" begins at values in memory = 0 modulo 40(hex) In other words at (hex values) 0x300, 0x340, 0x380, 0x400, etc. up to 0x1fc0. Each "section" is therefore 64 decimal bytes each.

Beginning, when the memory is "empty" the first data is written at location 0x300, then successive data is written going up from there. When the entire memory up to 0x1fff is filled (end of page 31) then data continues to be written at 0x300 and going up, thus writing over the oldest data.

How to interpret logged data:

Memory pointer: There is a 14 bit "pointer" which, in between periodic data reads, describes the first byte (base) location of the last data record that was stored. The "pointer" is accessed at location 0xd2 using a **4 byte** "short read" of location **d2**. The pointer is (oddly enough!) the top two bytes. (Ignore the lower two bytes). Form an address using the top byte as bits 8-15 (decimal) and the bottom byte as bits 0-7. The top two bits should be 00. The pointer reads from 0x0300 to 1fff.

Record length: The record length, consisting of all the data corresponding to a single read time will vary from 5 to 19 bytes, depending on the amount data which has been selected to be recorded.

As mentioned before, the user may choose which of the possible types of data to be regularly logged (for example they might choose only to record: Amp hr 1, Volts1, % full,) . This choice can possibly change at any time if, after recording some logged data, someone decides to change the PentaMetric log data choices using programs P41 and P42, using PMComm. If at any time someone does change these choices, the memory pointer will immediately advance to the beginning of the next section, and if the memory in the present section hasn't been used up it will remain unwritten.

After the data at one measurement time has been taken, if insufficient space remains in that section to write the entire record, then the pointer will advance to the beginning of the next section.

So within each 0x40 section, the data types will be the same, and each record length will be the same. All data on one single page has data formatted the same way.

The bottom 3 bytes on every (0x40) section has the "format" information for its page. However data on different pages may be formatted differently.

The beginning byte of each section—at **location 0 mod 40** has "top data address" information for the previous section. It has the base (beginning) address of the highest "record" on the previous page--but only the least significant 6 bits of the address (which is all that is necessary, since each section has only 0x40 bytes). If this value is 0, the data on the previous page is not valid. This is how to tell the highest significant location of the previous section..

The method of determining whether or not the memory space is completely full or not, is determined by examining the highest value of 0 mod 40 within the periodic data memory space from page 3 to 31.. If the value is 0, that means that the memory is not full. This is the byte on page 31 which is located at location c0 on this page. (1fc0). It would typically be read after downloading pages 3-31, then examining the value at location 0c0. If the value is 0 then the oldest data must begin at 0x300. Otherwise it will be just above the location identified by the pointer.

The locations 1 and 2 mod 40 have 16 bit "format information" , that describes what data is to be recorded in that section.. Byte 1 has bits 0-7. Byte 2 has bits 8-15.
Bit 0 high selects Amp Hour1.
Bit1 high selects Amp Hour2.
Bit 2 high selects Amp Hour3.
Bit 3 high selects Watt Hour 1.
Bit 4 high selects Watt hour2.
Bit 5 high selects "Max and min temp" .
Bit 6 selects "volts1"
Bit 7 selects "Amp1"**.**
Bit 8 high selects "volts2"
Bit 9 high selects "Batt% full, Batt 1 and 2"
Bits10-15 Not used.

How the data is arranged on each section: The data is written starting at 3 mod 40 on each section, going up in sequence. The first 3 bytes (3 thru 5 mod 40) are the date and time of the data to be written just above it. The next two bytes are the data "amp hour 1" if this has been selected by format "bit 0" . If not selected, it will have the data which is represented by lowest bit data that is selected in the format byte. The two bytes above that is the next lowest bit data that is selected, etc. If all ten bits are selected then each "record" will be 23 (decimal) bytes in length. This is 3 date/time bytes, plus 2 bytes each for each 10 data selected=20 +3=23 . In this "maximum" data case only two records per section can be recorded, and the last 15 bytes in each section will not be unused (unwritten). (3[formatting]+2*23[data]+15[unused]=64)

Date/Time: 3 bytes. [For some stupid reason, the date/time here is formatted a little differently than for the "efficiency cycle data"] The low two bytes consist of the number of days in 1/8 day units. The higher of the two bytes is the most significant byte. Thus these show the "date time" to within 3 hours (1/8 day) up to a maximum of 65535 x 1/8 days =8192 days. . The highest byte is the number of (additional) minutes 0-179. Therefore it is possible to specify days from 0 to 8191 days, and the time to within 1 minute. This is how relative time between record events is determined.

Amp-hours, watt hours and amps: 2 bytes: The LOW 10 bits (bits 0-9) represent the 3 digit number 000-999,(hex from 0-0x3e7) absolute value. The high bit (bit 15)=sign. 1=sign is minus.0=sign is +. Bits 12-14: decimal point placement as follows: 001 = a.bc, 010= ab.c, 011 = abc 100 = abc0, 101 = abc00, 110 = abc000. 111 = a,bc0,000.

Volts: 2 bytes. Bits 0 thru 10 represent the number from 000-2047. Divide this by 20 to get volts..

Temperature, 2 bytes.: Max/min. Show temperature (degrees C) as signed 8 bit number (two's complement). (-128° to +127° , however PentaMetric range is -20 to +65C)) 0 means 0°. . ff means -1°. 1 means 1°, Lower byte is "minimum temp", upper byte is "maximum temp".

Batt% full, and "charged" information: 2 bytes: **Low byte**: **7 low bits**: Percent for battery 1. **Bit 7**: (high bit) is high if during that interval the batteries reached the "charged" criteria. **High byte**, 8 bits: Same except for battery 2. Note that when selecting the "% full" option to be recorded the data for both batteries is always recorded. However in Program mode 39-42 the user should be given a chance to select either Batt1 or Batt2 or both. Depending on whether a battery is being measured in channel 1 or 2, this data will or will not be valid or meaningful.

Step by step procedure for reading data: (log data is from 300 to1fff. First this data must be downloaded into a file)

This will read starting from earliest to most recent data:

1A. Read the **4 bytes** of the "last written address base pointer" at location **d2,** using a "short read" at that location.. Use only the top two bytes: Form the high two bytes expressed as a 16 bit number (high address=bits (8-15, low address=bits0-7) The **bottom 14 bits (bits 0-13) is an absolute pointer address ("P")** pointing to the base address of the most recently written data. **Bit 14- 16 are not used**.

1B. Read data at location 1fc0. If this byte is 0, then the memory is NOT FULL. Otherwise it is full.(Set or clear the "**memory full**" bit.)

1B. if memory full bit =0 and P=1fc0, quit. (No data yet). Otherwise proceed:

2. Find the base (oldest record location) **reading** address ("R") location of the data as follows
—If "memory full"=0, R=303. Go to step 3. (If memory not full, begin reading at bottom of memory)

If "memory full"=1 (If memory full, read starting on next page above last page written)
if (P & ffc0) => 1fc0 then R=303, Go to step 3.
if (P & ffc0) < 1fc0 then R = (P & ffc0) +43 Go to step 3.

3. On the page R , read byte 1 and 2 (location (R & ffc0)+1) to read the "format" bytes F1 and F2. The meaning of this format byte is described above under "format byte".

3B.(If R is not on the same page as P, then go to step 4 to find upper limit on this page. If R is on same page, then upper limit to read is P).
If (R & ffc0) NOT= (P & ffc0) Go to step 4.
If (R & ffc0) = (P & ffc0) Go to step 6. (or H=(3f & P)

4. (Find the base of the next page N to read to get "top of page" data.) .
If R= 1fc3, then N=300. Else N=(R & ffc0)+40

5. At location N, read byte 0 (at location N) which gives the least significant 6 bits of the base of the topmost "record" on the (previous) page R. This byte will have a value which is between 0 and 3f hex. Call this top value "H.". This will indicate when you are through reading page R, and will be used in step 8.

NOW R POINTS TO CORRECT BOTTOM DATA LOCATION TO READ.

6. The format bytes F1 and F2 from step 3 will describe the data to be read. Start reading data from location R. The first 3 bytes (R thru R+2) will be the "date/time" bytes as described above.(in " **how to interpret the data**"). The next 2 bytes will be the data represented by "bit 0" of the "format" byte if bit0 is high. If bit0 low, then look at bit1 of the format byte. If this bit is high, this will be the data to read. If low, then go to bit2, etc. Consult " **how to interpret the data**" above, for how to translate each two (or 3 bytes) into date/times/amps/volts/etc. Thus write all the data for one row, above the last row written of the table.

7 Altogether, the number of bytes read in one "record" will be :3+K*2,

{where K=number of data items selected (for that page): i.e count the number
of bits that are high in format bytes F1 and F2.

8. (Advance the read pointer to get the next base address to read the next "record". What to do depends on whether P and R are on the same page or not)
if (P& ffc0) not = (R & ffc0)
if (R& 003f) < H, Set R =R+ 3+K*2 ; go to step 6.
if (R & 003f) => H
if (R&ffc0) => 1fc0 then R=303 Go to step 3
if (R& ffc0) < 1fc0 then R= (R & ffc0) +43. Go to step 3.
if (P& ffc0) = (R & ffc0)
if R < P . Set R =R+ 3+K*2; go to step 6

if R >= P Quit. Finished reading all data