# WORK IN PROGRESS PentaMetricRS232-PCSpecWeb.doc
# September 5, 2005, Ralph Hiesey
# How to access data via the RS232 port from the PentaMetric battery monitor.

**Note: This information is intended for computer programmers that wish to control the PentaMetric using their own software program. Although the information here is mostly correct, it hasn't been thoroughly checked because of time constraints--therefore there are very likely to be some errors. Please call and let us know if something doesn't seem to be correct, or if certain parts are confusing so we can correct or improve this document. Bogart Engineering: 831 338-0616.**

1.This document describes the communication protocol used to communicate via the RS232 port. This one covers how to read the "real time" data in the PentaMetric (such as present "volts", "amps", "watt hour" etc.) and also how to change the "programmable data" that is used to control the operation of the PentaMetric.:

There are also three "logged data" files that reside within the Pentametric. They are on the order of 4 kbyte each. By giving appropriate commands this data can be sent as RS232 serial output data with file lengths of a multiple of 256 bytes. . The information for downloading and interpreting this data will be explained in separate documents.

3.Communication is at 2400 baud, 8 bits, no parity. The RS232 RTS and CTS handshaking lines are not needed.

4 Communication of the "real time" and "programmed" data takes place by "reading" or "writing" data registers that reside in the PentaMetric "input unit".. Each type of data, for example "volts 1", , "amp hours" or "charged voltage setpoint" may be read (or written to) by giving commands consisting of several sequential bytes of data:

**How to read data:** There are two ways to read data from the PentaMetric: a SHORT READ or LONG READ. The SHORT READ is used for accessing just a few bytes from the tables below used for reading real time data or programmed data (described in detail here). The LONG READ reads entire pages (of 256 bytes) so the data read has the number of bytes equal to a multiple of 256, used when downloading the logged data (described in detail in other documents.)**.**

**SHORT READ: This following is used to read the "real time data" shown in tables 1 and the programmed data in Table 2.** To READ a particular item in Table 1: Send the bytes in this order: First send hex 81 ("read" command) Then send the hex "address" as shown in Table 1 or 2. Next send the "no. bytes" (N) shown in the table. Finally send a one byte checksum. The checksum must be the hex number, which when added to the previous 3 bytes results in a sum with "FF" (hex) as the least significant byte. When this is done the Pentametric will respond in a few hundred milliseconds by sending the N data bytes, followed by the Checksum. The checksum is such that when the N+1 bytes are added the sum will have "FF: as the two least significant bytes. The data emerges with the LOWEST BYTE FIRST, with successive bytes, ending with the HIGHEST BYTE followed by the checksum.

Example: if you want to read **Average Battery1 volts** (see table1 below: under display D3) you would send the following 4 (hex) bytes 81, 3, 2, 79. Note that the sum of these 4 bytes is "0FF". The PentaMetric would respond with 2 bytes of data followed by 1 byte of checksum. The LOW BYTE comes first followed by the HIGH BYTE, then the checksum. To confirm correct reception of the data you could add all (3) bytes together and see if they add up to a hex number with "ff" as the low byte.

Send: (all hex digits): 81,03,02,79. Response (for example): FA, 01, 04. The high byte is 01, the low byte FA. Checksum=04. So the voltage is 1FA= 506 decimal. Divide by 20 to get 25.3 volts. The sum of FA + 01 + 04=0ff, which confirms that the transmission is OK

**LONG READ of 256 byte blocks:** (For reference) This shows how to read the logged data. This is shown for completeness but is used only when downloading the logged data. The memory space is divided into 64 "pages" of 256 bytes each. To **READ** N **pages** (where N is between 1 and 4) starting with the base location of "page"=P (where P=0 through 0f hex) Send the following 4 (hex) bytes in sequence: C1(hex), P, N, X, where X is the checksum . The Pentametric then responds by sending the N*256 data bytes, followed by the Checksum, which is the hex number, which when added to all the bytes sent causes the least significant two bytes to be FF.

**SHORT WRITE: Writing the data as shown in Table 2 (or even Table 1) is done as described here.**
**This command will write a maximum of 16 bytes of data at a time. To WRITE N bytes (up to 16 sequential bytes) for a particular item in Table 2 (or table 1): send N+4 bytes in this order: First send hex 01 ("write" command) Then send the hex "address" as shown in Table 2 (or 1): . Next send the "no. bytes" (N) shown in the table, then send each desired byte to write, beginning with the LOW BYTE and ending with the HIGH BYTE. Finally send a one byte checksum. The checksum must be the hex number, which when added to the previous 3 bytes results in a sum with "FF" (hex) as the two least significant hex digits. When this is done** the Pentametric then responds in a few hundred milliseconds by sending back the same checksum if it successfully received and wrote the data.

**Example: To write the "Batt1 Capacity" equal to 1000 amp hours (see Table 2, "Batt1 Capacity") you would send the following 6 hex digits: 01,f2, 02, e8, 03, 1f. The decimal 1000 number is 03e8 hex, which is where the e8, 03 comes from. Note that the sum of all 6 bytes is 1ff. The PentaMetric will respond with the checksum originally sent (1f) if it receives the data OK. If you then read back the location it should contain the new data.**

**Table 1, below is the Display table**: Data here is ordinarily only read, not written to. The **first column** shows the "display number" which is used to identify the type of data as described in the PentaMetric instructions. The **second column** shows the description of the data. The 5th column shows how to decode the data.

**RESET commands**:. A few of the display functions (in Table 1) have a "reset" command to easily allow the user to reset these values to 0. These are D13,14,15 (Amp hours), D16,D17 (cumulative amp hours), D20,21 (Watt hours), D24,25 Days since charged D26,27 (days since equalized). To "reset" these, write the data shown in the "Reset command" column to decimal location 39 (hex 27). (They could also be reset by just writing 0's to the address location, however.)
**Example**: To reset the "amp hours 1" to 0, note that the "Reset command" for "amp hours1" is 09. Write the data 09 to location hex 26 as follows: 01,27,09,ce. The PentaMetric should respond with "ce".

## TABLE 1.

I**MPORTANT NOTE: When data is read from serial port, the LOW byte comes out first, and the HIGH byte comes out last.**

| Display number | Display Data Displayed name | decimal Address | No Bytes | Display format | Display units | Reset command |
|---|---|---|---|---|---|---|
| **D1** | **Battery1 Volts** | 1 | 2 | **FORMAT1:** Take the low 11 bits of (binary) data, divide it by (decimal) 20, | Volts | |
| **D2** | **Battery2 Volts:** | 2 | 2 | **FORMAT1** | Volts | |
| **D3** | **Average Battery 1 Volts** | 3 | 2 | **FORMAT1** | Volts | |
| **D4** | **Average Battery2 Volts** | 4 | 2 | **FORMAT1** | Volts | |
| **D7** | **Amps 1:** | 5 | 3 | **FORMAT2 :** The 3 bytes of data consist of 24 bits of data which we label from B0 (LSB) to B23 (Hi bit**). First** look at the highest bit (B23). **If B23 is high**, then COMPLEMENT all the bits B0 - B22, and strip B23 off. (So you have 23 bits left.). **If B23 is low,** just leave the number as is. This is the binary data in 1/100 amp units: so next divide by 100 (decimal) to get amps. **Finally**, | Amps | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | multiply the result by -1 for correct sign. **Note** : when using the 100 amp/100mV shunt it's OK to regard the lowest digit as significant. However when using the 500amp/50mV shunt you should only use the digits down to 1/10 amp, as the 1/100 amp digit is beyond the resolution of the PentaMetric. | | |
| **D8** | **Amps 2** | 6 | 3 | **FORMAT2 :** | Amps | |
| **D9** | **Amps 3:** | 7 | 3 | **FORMAT2.** | Amps | |
| **D10** | **Average Amp1s** | 8 | 3 | **FORMAT2.** | Amps | |
| **D11** | **Average Amp2** | 9 | 3 | **FORMAT2.** | Amps | |
| **D12** | **Average Amp3:** | 10 | 3 | **FORMAT2.** | Amps | |
| **D13** | **Amp Hours 1** | 12 | 3 | **FORMAT 3:** Same as **FORMAT2,** except disregard the **Note.** | Amp-hours | 09 |
| **D14** | **Amp Hours 2** | 13 | 3 | **FORMAT3** | Amp-hours | 0a |
| **D15** | **Amp Hours 3** | 14 | 4 | **FORMAT4:** The4 bytes of data consist of 32 bits of data which we label from B0 (LSB) to B31 (Hi bit**). First** look at the highest bit (B31). **If B31 is high**, then COMPLEMENT all the bits B0 - B31,.. **If B31 is low** leave as is. **Next** strip off the low 7 bits: B0 -B6, leaving 24 bits (B7 -B30) This is the binary data in 1/100 amp hr units: so next divide by 100 (decimal) to get amp-hr.. | Amp-hours | 0b |
| **D16** | **Cumulative Amp Hours 1** | 18 | 3 | **FORMAT2B.** Same as **FORMAT2** except don't divide the result by 100.. | Amp-hours | b0 |
| **D17** | **Cumulative Amp Hours2** | 19 | 3 | **FORMAT2B.** | Amp-hours | b1 |
| | | | | | Amp-hours | |
| **D18** | **Watts 1:** | 23 | 3 | **FORMAT2** | Watts | |
| **D19** | **Watts 2** | 24 | 3 | **FORMAT2** | Watts | |
| **D20** | **Watt Hours 1:** | **21** | 4 | **FORMAT5 :** The4 bytes of data consist of 32 bits of data which we label from B0 (LSB) to B31 (Hi bit**). First** look at the highest bit (B31). **If B31 is high**, then COMPLEMENT all the bits B0 - B31,.. **If B31 is low** leave as is. This is the 31 bit binary data in 1/100 amp hr units: so next divide by 100 (decimal) to get watt-hr.. Finally multiply by -1 to get correct sign. | Watt-hours | 11 |
| **D21** | **Watt Hours 2** | 22 | 4 | **FORMAT5** | Watt-hours | 12 |
| **D22** | **Battery1 Percent Full** | 26 | 1 | **FORMAT6:** Just use the number "as is". | % | |
| **D23** | **Battery 2 Percent Full:** | 27 | 1 | **FORMAT6** | % | |
| **D24** | **Days since Battery 1 charged** | 28 | 2 | **FORMAT7 .** Divide 2 byte result by 100 (decimal). Result has 1/100 day resolution.) | Days | 19 |
| **D25** | **Days since Battery 2 charged** | 29 | 2 | **FORMAT7** | Days | 1a |
| **D26** | **Days since Battery 1 equalized** | 30 | 2 | **FORMAT7** | Days | 1b |
| **D27** | **Days since Battery 2 equalized** | 31 | 2 | **FORMAT7** | Days | 1c |
| **D28** | **Temperature** | 25 | 1 | **FORMAT8** 1 byte. Signed 1 byte number: 2's complement: ie fe=-2 , ff=-1, 0=0, 1=1, etc thru 127. 80(hex)= -128. So it can express number from -128 to +127. | Degrees C. | |
| **D29thru34 .** | **Battery1 efficiency data** | 218 | 12 | Byte 3 is 1 cycle batt effeceincy: like **FORMAT6**<br>Byte 7 is 4 cycle batt efficiency like **FORMAT6**<br>Byte 11 is 15 cycle batt efficiency like **FORMAT6**<br>Byte 0-1 is 1 cycle self disch current (amps): Take low 10 bits and divide by 100 to get value of amps. Sign is indicated by a "1" or "0" in bit 15. means -, 0 means +. Format is ±X.XXby 100 | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | The "number of cycles"(Bits 11-14 not used)1<br>Byte 0-1 is 4 cycle self disch current (amps): (Like 1 ccle self disch current)<br>Byte 0-1 is 15 cycle self disch current (amps): (Like 1 ccle self disch current) | | |
| **D35thru40** | **Battery 2 efficiency data** | 215 | 12 | Same as D29 thru 34 | | |

# TABLE 2: PROGRAMMED DATA TABLE

**Program number** is the "P" number. Note that not all numbers will be used at first. Later they may be added.
**Address** and **No of bytes** are the numbers that are sent to the Pentametric to read the data.
**Data Format** points to the program that converts the received bytes to the screen display. It utilizes the data in "Display name".
**Allowed Data Limits** describes the lower and upper bound of acceptable data to store.

| Description | Program number | (hex) Address1 | No. of bytes 1 | Address2 | No of bytes2 | **Data format:** To understand this: Refer to section 6A, PentaMetric instructions corresponding to program number | Allowed data limits) |
|---|---|---|---|---|---|---|---|
| | 8 bit Integer | 8 bit Integer | 8 bit Integer | 8 bit integer | 8 bit integer | | |
| Sw1-select | P1 | 0ff | 5 | | 0 | **FormatA**: Refer to PentaMetric instructions, section 6 under P1-P5. Each of the 5 bytes specifies the "AD" number of the display items to be displayed, starting from byte1 to byte 5. Put 0 if it is desired to not show any data. For example, if the 5 bytes of data are 08,01,0,0, 0, switch 1 would display .Amps2 and Battery 1 volts. . | Limits of data: 0-28 (decimal). "0" means "don't display" |
| Sw2-select | P2 | 0fe | 5 | | 0 | **FormatA** | |
| Sw3-select | P3 | 0fd | 5 | | 0 | **FormatA** | |
| Sw4-select | P4 | 0fc | 5 | | 0 | **FormatA** | |
| Sw5-select | P5 | 0fb | 5 | | 0 | **FormatA** | |
| Batt2 Label | P6 | e1 | 3 | | 0 | Bit 4 indicates label. bit 4= "1" is "Battery".Bit 4= 0 makes "Battery 1" | Bit 4 hi or low. |
| Amp1 thru 3 Labels | P7-9 | e1 | 3 | | 0 | Byte1=label for amps 1. Byte2=label for amps 2. Byte3=label for amps3. The label is indicated by bits 0-3 as follows:<br>0=Amps#. 1=Solar 2=Wind 3=Hydro 4=Load 5=Battery 6=Battery 1. | For all 3 bytes: bits0-3 must be from 0 thru 6. |
| Shunt Select | P11-13 | f6 | 3 | | 0 | Byte1=shunt for Amps 1.<br>Byte2=shunt for Amps 2<br>Byte3=shunt for Amps3.<br>Bit 4 hi=100A/100mV shunt<br>Bit 4 lo=500A/50mV shunt. | For all 3 bytes, bit 4 either hi or low. |
| | | | | | | | |
| Batt1Capacity | P14 | 0f2 | 2 | | 0 | **FormatD:** Using 2 bytes, put capacity in amp hours 1-9999. | Limits of data: 0-9999. "0" indicates "no battery" |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Batt2Capacity | P15 | 0f1 | 2 | | 0 | **FormatD:** | |
| Filter Time | P16 | 0f3 | 1 | | 0 | 0=Time constant=0, 1:TC= .5 min, 2:TC=2min. 3=TC=8min | Bits 0-1 can be number from 0-3. |
| Ch control(not used) | P17-20 | 0f0 | 7 | | 0 | | |
| AlarmLevlBat1 | P22-23 | 0ce | 2 | | 0 | **Byte 1** : Bits 0-1 : Lo Batt alarm. Bits2-3:Hi batt alarm, bits 4-5: Batt charged, bits 6-7: **Byte 2** : Time to charge, bits0-1 For each of above: 0=alarm off. 1=visual alarm only, 3=visual/audible alarm. | see column to left. |
| AlarmLevlBat2 | P24-25 | 0cd | 2 | | 0 | **Byte 1** : Bits 0-1 : Lo Batt alarm. Bits2-3:Hi batt alarm, bits 4-5: Batt charged, bits 6-7: **Byte 2** : Time to charge, bits0-1 For each of above: 0=alarm off. 1=visual alarm only, 3=visual/audible alarm. | |
| Bat1LoAlarmSetpoint | P26 | 0ec | 3 | | 0 | **FormatG:** Bytes1-2: Take low 10 bits. Divide by (decimal) 10 to get volts. Byte3=%, from 0-100 . | |
| Bat1HiAlarmSetpoint | P27 | 0ea | 2 | | 0 | **FormatF:** Bytes1-2: Take low 10 bits. Divide by (decimal) 10 to get volts. : | |
| Bat2LoAlarmSetpt | P28 | 0eb | 3 | | 0 | **FormatG:** | |
| Bat2HiAlarmSetpt | P29 | 0e9 | 2 | | 0 | **FormatF** | |
| Relay ON/OFF Setpt | P30-31 | 0d4 | 6 | | 0 | **Low3 bytes** Relay on; Bytes 1-2: voltage setting, byte3: %Full setting . **High 3 bytes:** Relay off; Bytes 1-2: voltage setting, byte3: %Full setting | |
| Batt1charged" criteria | P32 | 0e8 | 3 | | 0 | **FormatG:** Bytes1-2: Take low 10 bits. Divide by (decimal) 10 to get volts. Byte3=Amps, from 0-100?? . | Bytes 1-2 |
| Batt2charged" criteria | P33 | 0e7 | 3 | | 0 | **FormatG** | |
| BattEfficy setpoints | P34-35 | 0e5 | 3 | | 0 | Low byte: %, next two bytes: amps expressed in 1/100 amp units.: 0-999 (decimal) | Amps: 0-999 decimal. |
| TimeBetween Equalize | P36 | e3 | 1 | | 0 | Days 0-255. (0=off) | |
| TimeBetween Charge- | P37 | e2 | 1 | | 0 | Days 0-255. (0=off) | |
| Day: 1/8 day units | P38 | f9 | 2 | | | To get day and time you need to read two | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Time: minutes 0-179 | P38 | 24 | 1 | | | different locations: The two bytes at f9 give the DAYS in 1/8 day units (0-65535). This gives the time within 1/8 day (=180 minutes). The remaining time (in minutes) is read in location 24, (0-179)and is added to other data to give date/time to 1 minute. See Note 5. | |
| Time of One Periodic data measurement | P39 | cf | 3 | | | Take byte 1, multiply by 180. Add this to byte 3. (Ignore byte 2) This gives number of minutes after midnight that measurement is taken. (a number from 0-1439) | |
| Periodic data: number of measurements/day | P40 | d0 | 1 | | | Each bit can be either 0 or 1. Calculate [(bit0 +1) * (bit1+1) * (bit2+1) * (bit3+1) * (bit4+1) * (2* bit5+1) * (2* bit6+1) * (4 * bit7+1)]. (The * represents multipication) . This is the number of times/day that the measurement will occur. | |
| Periodic data: items to log: | 39-42 | d2 | 2 | | | Select measurement when bit is =1. **Byte 1:** Bit0=Amp hr 1. Bit1=AmpHr2. Bit2=Amp hr3. Bit 3=Watt Hr1 Bit4=WattHr2. Bit 5=Min/Max Temp. Bit6=Volts1 Bit 7=Amps1. **Byte 2**: Bit 0=Volts2. Bit1=Batt%Full | Change41-50 |
| 5%Data-Options | 43 | d1 | 1 | | | Only bits 5 through 7 are used: But bits 0-4 must not be disturbed. So before writing bits 5-7 be sure to read the data and write back 0-4 the same way so as to leave them unchanged. Bit5=Record Batt 1 data: Bit 6=Record batt 2 data. : Bit7:=Record every 5% (otherwise every 10%) : | Change51-54 |
| Backlight | 44 | | | | | Not yet implemented | |
| Erase periodic data | 45 | | | | | To erase: write hex 72. to location (hex)27, (or decimal 39.) | |
| Erase Batt discharge voltage profile data | 46 | | | | | To erase: write hex 82 to location (hex)27, (or decimal 39.) | |
| Erase battery 1 efficiency data | 47 | | | | | To erase: write hex 90 to location (hex)27, (or decimal 39.) | |
| Erase battery 1 efficiency data | 48 | | | | | To erase: write hex 91 to location (hex)27, (or decimal 39.) | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Erase & Initialize All programmed data | 49 | | | | | To erase: write hex a5 to location (hex)27, (or decimal 39.) | Change55-59??? |

**NOTE 5** : This gives a "relative present time" in "days and minutes". Using the PC computer's time system you can determine the "actual present date time". The "log data" times in the PentaMetric are stamped with the time according to this "relative time". By subtracting the "logged data" date/time from the date/time you read here you can determine how long ago a particular data was logged. This can be subtracted from the computer's time to determine its actual time/date.

.

**(For reference) The following describe the 3 types of logged data which can be downloaded. The procedure for downloading and intre preting the result is described in other documents.**

(1)**The PERIODIC data**, consists of periodic recording of any or all of the following: AmpHours1, AmpHours2, AmpHours3, Watt-hr1, Watt-hr2, (Filtered)Volts1 and (FilteredAmps1). For stage 1, this just needs to download all the data from 300 hex to 1fff. This just requires that the PAGE data be retrieved, 50h starting from Page "0C" be loaded into a excel friendly file (7360d bytes) call **GetDataPages(0ch, 73h,M)** and then put data in a comma delimited file

(2) **The BATTERY DISCHARGE CURVE data** shows (filtered) volts and amps for each increment of 5% (or 10%) battery full. This is used to check batteries for possible capacity loss.This requires that the PAGE data be retrieved, 40h pages, starting from Page "80h" be loaded into a excel friendly file.(total 1000bytes-=4096 d bytes)Call **GetDataPages(80h,40h,M)** and then put data in a comma delimited file

(3)**The battery cycle efficiency data** tracks the time between successive "full charge" points, the amp-hours charging and amp hours discharging. This is used to check the batteries for their ability to retain charge. This requires that the PAGE data be retrieved, 40h pages, starting from Page "C0" be loaded into a excel friendly file.(total 1000bytes-=4096 d bytes)Call **GetDataPages(c0h,40h,M)** and then put data in a comma delimited file

**Alarm data** : . There are 10 different possible alarm conditions--5 for each battery. For each one there are 3 choices: 1.NO ALARM, 2.VISUAL ONLY ALARM, 3.AUDIBLE AND VISUAL. These can be read by reading location 25 (hex). See details below in Table 3 , under "Alarm Status"

Table 3

| Description | Address | No bytes | detailed description | Where used |
|---|---|---|---|---|
| Amps and Battery 1 labels | e1 | 3 | low byte: bits 0-3: Amps 1 label 0= "**Amps 1**" . 1= "**Solar**". 2= "**Wind**" 3= "**Hydro**" 4= "Load". 5= "Battery".6=Battery 1. Bit 4=Battery 1 label. When Bit4=0, "**Battery 1**". When Bit4=1, "**Battery**" Bits 3, 5-7 not used.<br><br>Middle byte: bits 0-3: Amps 2 label 0= "**Amps 2**" . 1= "**Solar**". 2= "**Wind**" 3= "**Hydro**" 4= "Load". 5= "Battery".6=**Battery 2**<br><br>High byte: bits 0-3: Amps 3 label 0= "**Amps 3**" . 1= "**Solar**". 2= "**Wind**" 3= "**Hydro**" 4= "**Load**". 5= "**Battery**".6=**Battery** | When displaying |
| Alarm (enable) level | ce | 4 | Each alarm is specified as one of 3 levels: 0=**alarm off**, 1=**visual alarm only**. 2=**visual and audio alarm**.<br>Low byte: Bits0-1 "**Low Batt 1 alarm**" Bits2-3: "**Battery 1 meets charged criteria**", Bits 4-5: "**high Batt 1 alarm**":. Bits 6-7: "**Time to Charge Batt 1 full**"<br><br>2nd byte: Bits0-1 "**Time to Equalize Batt 1**"" Bits2-7- not used (or actually, not needed: bit 7 is "relay on" bit) :<br><br>3rd byte: Bits0-1 "**Low Batt 2 alarm**" Bits2-3: "**Battery 2 meets charged criteria**", Bits 4-5: "**high Batt 2 alarm**":. Bits 6-7: "**Time to Charge Batt 2 full**"<br><br>Highest byte: Bits0-1 "**Time to Equalize Batt 2**" Bits2-7- not used: | |
| Alarm status | 25 (hex) | 2 | Low byte: Bit0: **Low Batt 1 alarm**. Bit1: **Battery 1 meets charged criteria** Bit2: "**high Batt 1 alarm**" Bit3: "**Time to Charge Batt 1 full**" Bit4: "**Time to Equalize Batt 1**" bits 5-6 not used. Bit7=relay on, but not used here.<br><br>High byte: Bit0: **Low Batt 2 alarm**. Bit1: **Battery 2 meets charged criteria** Bit2: "**high Batt 2 alarm**" Bit3: "**Time to Charge Batt 2 full**" Bit4: "**Time to Equalize Batt 2**" bits 5-6 not used. Bit7=relay on, but not used here. | |

| | | | | |
|---|---|---|---|---|
| Shunt type | f6 | 3 | Low byte: bit 4: Shunt for **Amps1** channel: When 0=100A/100mV shunt. When 1= 500A/50mV shunt. Other bits not used.<br><br>Middle byte: bit 4: Shunt for **Amps2** channel: When 0=100A/100mV shunt. When 1= 500A/50mV shunt. Other bits not used.<br><br>Hi byte: bit 4: Shunt for **Amps3** channel: When 0=100A/100mV shunt. When 1= 500A/50mV shunt. Other bits not used. | |
| Minutes | 24 | 1 | Minutes 0-179 (to be added to days, below) | |
| Days | f9 | 2 | 1/8 day units. | |
| | | | | |
| | | | | |