Serial Data Format $Revision: 4.41

WattsUp? communicates with the outside world over a standard RS232 serial port. The following section gives technical specification for the communication protocol. This can be useful to programmers or other people who have special needs. Ordinary users can skip over this section.

I. DATA FORMAT. All commands and data packets have the following format.

1. Serial Format. Data is transmitted and received as standard RS232 data at 115200 baud, 8 data bits, 1 stop bit and no parity bit.

2. Character Format. Data consists of standard ASCII characters. Numbers are represented as strings of ASCII characters with no binary numbers.

3. Packets. All information is contained in packets. All characters outside of a packet are ignored. Packets begin with a pound sign "#" and end with a semicolon ";".

4. Arguments. Inside the packet are a series of arguments. Arguments can be strings or ASCII representations of numbers. Arguments are separated by commas ",". The last argument is terminated by the semicolon ";". No empty arguments are allowed. Each argument must have at least one character. For numerical arguments, the argument must be a valid ASCII Number (hex 30 through hex 39). String arguments must contain at least one character, even if it is just a space. Control character such as Carriage Return, Line Feed and Tab characters are ignored within a packet. Spaces are valid characters inside string arguments.

Any argument other than Subcommand (see next) which intentionally has no value is represented by an underscore "_". For example data record arguments following "#d" will contain "_" for every value that was unlogged.

5. Commands. The first two arguments are single ASCII characters representing a Command and Subcommand. Commands and Subcommand specify an action to be carried out or specify the type of data in the packet. Commands to the WattsUp meter from the host (PC, Mac, etc.) are generally upper case. Replies from the WattsUp meter to the host are generally lower case. If there is no Subcommand, any ASCII character can be used, but a dash "-" is generally preferred.

6. Argument Count. The third argument is a numerical value that represents the number of arguments that follow. The two commands and the count itself are not included in the count.

7. Transactions. All data transfers are initiated by the host. In other words, WattsUp never sends data to the host unless the host makes a specific request. There is just one exception to this general rule: at reset (normally occurring only at power-on startup), the WattsUp meter emits an unsolicited announcement heralding its firmware name, version, compilation date, and the principal electrical configuration detected: nominal line frequency Hz and nominal voltage.
Example:
EED05F01 $Revision: 4.41 $ 200612212301 60Hz 120V

8. Syntax Diagram.

#<C1>,<C2>,<Nm>,<A1>,<A2>,...<An>;

<C1> = First Command

<C2> = Second Command

<Nm> = Number of Arguments

<A1> = Individual Arguments

8. Examples.

#U,R,0;
#u,-,3,80,100,0;

9. Timeouts. The Meter will be expected to respond within two seconds of a request. If no response is received by the PC within two seconds, the software will timeout and it will be assumed that the Meter is not connected or is malfunctioning.

10. Aborting Meter Communication. Sending Control-X character (hex 18) to the meter will cause it to abort any pending communication.

II. Commands and Replies, and <argument> formats

#C,R,0;
#c,-,18,<0|1>,...;
        Command: chosen fields request.
        Reply: chosen fields now set to be logged. Each argument is either 0 or 1. The arguments correspond to the Header Record reply list.

#C,W,n,<0|1>,<0|1>,...;
#n,-,1,<Limit>;
        Command: Choose list of fields to be logged from now on.
        Note: clears logging memory, even if change was "no change".
        Reply: record count limit using these chosen fields.

<Limit> integer may be approximately 1,700 - 262,000 depending upon choices and model features.

#D,R,0;
#n,-,3,<Reserved>,<Interval>,<Count>;
#d,-
,18,<W>,<V>,<A>,<WH>,<Cost>,<WH/Mo>,<Cost/Mo>,<Wmax>,<Vmax>,<Amax>,<Wmin>,<Vmin>,<Amin>,<PF>,<DC>,<PC>,<HZ>,<VA>;
#l,-,2,<Reserved>,<Interval>;
        Command: Data request for all data logged in the WattsUp data memory.
        Reply: Data Preamble Record. This preamble record is sent before the main data record.
        Reply: Data records. Non-chosen fields reported as "_" (underscore).
        Reply: Last record of data transfer.

<Interval> integer seconds between logged values

#F,R,0;
#f,-
,48,<Edition>,<NonvolatileStructureVersion>,<50HzIsCalibrated>,<50HzVRMSperVolt>,<50HzIRMSOS>,<50HzVRMSOS>,<50HzCH1OS>,<50HzCH2OS>,<50HzWattsPerAENERGYslope>,<50HzWattsPerAENERGYoffset>,<50HzVoltampsPerVAENERGYslopeUpper>,<50HzVoltampsPerVAENERGYoffsetUpper>,<50HzMidPowerVAENERGYperSecond>,<50HzVoltampsPerVAENERGYslopeLower>,<50HzVoltampsPerVAENERGYoffsetLower>,<50HzLowPowerVAENERGYperSecond>,<50HzIRMSperAmpSlopeUpper>,<50HzIRMSperAmpOffsetUpper>,<50HzMidPowerIRMS>,<50HzIRMSperAmpSlopeLower>,<50HzIRMSperA

```
mpOffsetLower>,<50HzLowPowerIRMS>,<50HzIRMSnoLoadBias>,<50Hzs24LAENERGYperSecondNoLoa
dBias>,<50Hzs24LVAENERGYperSecondNoLoadBias>,<60HzIsCalibrated>,<60HzVRMSperVolt>,<60
HzIRMSOS>,<60HzVRMSOS>,<60HzCH1OS>,<60HzCH2OS>,<60HzWattsPerAENERGYslope>,<60HzWattsP
erAENERGYoffset>,<60HzVoltampsPerVAENERGYslopeUpper>,<60HzVoltampsPerVAENERGYoffsetUp
per>,<60HzMidPowerVAENERGYperSecond>,<60HzVoltampsPerVAENERGYslopeLower>,<60HzVoltamp
sPerVAENERGYoffsetLower>,<60HzLowPowerVAENERGYperSecond>,<60HzIRMSperAmpSlopeUpper>,<
60HzIRMSperAmpOffsetUpper>,<60HzMidPowerIRMS>,<60HzIRMSperAmpSlopeLower>,<60HzIRMSper
AmpOffsetLower>,<60HzLowPowerIRMS>,<60HzIRMSnoLoadBias>,<60Hzs24LAENERGYperSecondNoLo
adBias>,<60Hzs24LVAENERGYperSecondNoLoadBias>;
Example:
#f,-
,48,13,0,0,3690,0,0,0,0,252,919,252,919,1,252,919,0,100,0,1234,100,0,0,0,0,0,0,3690,0
,0,0,0,252,919,252,919,1,252,919,0,100,0,1234,100,0,0,0,0,0;
        Command: Request WattsUp calibration factors.
        Reply: all calibration factor values.

#H,R,0;
#h,-,18,W,V,A,WH,Cost,WH/Mo,Cost/Mo,Wmax,Vmax,Amax,Wmin,Vmin,Amin,PF,DC,PC,Hz,VA;
        Command: Header record request.
        Reply: Header record with the shown text.

#L,W,3,E,<Reserved>,<Interval>;
#d,...
        Command: Set the WattsUp to external Logging with this interval.
        Reply: logging output records

#L,W,3,I,<Reserved>,<Interval>;
#s,-,3,<Reserved>,<Interval>,1;
        Command: Set the WattsUp to internal Logging with this interval.
        Reply: interval used for logging.
        Note: clears logging memory.
        Note: Interval is ignored if memory-full option is automatically-condensing

<Interval> integer seconds between logged values

#N,R,0;
#n,-,1,<Limit>;
        Command: Request record count limit.
        Reply: record count limit using fields chosen in #C,W command.

<Limit> integer may be approximately 1,700 - 262,000 depending upon choices and model
features.

#O,R,0;
#o,-,1,<FullHandling>;
        Command: Request the memory-full handling option for when logging memory is
filled
        Reply: memory-full handling option in effect.

#O,W,1,<FullHandling>;
        Command: Set the memory-full handling option for when logging memory is
filled
        Note: clears logging memory, even if change was "no change".

<FullHandling> is logging procedure when logging memory is filled
 = 0 stop/suspend means logging suspends until memory is cleared
```

```
 = 1 wrap/overwrite means logging overwrites oldest (wraparound); most-recent N
records always available
 = 2 automatic/condensing means interval always starts at 1 second; at every memory-
full condition interval is doubled (frequency is halved), and logged data is
condensed by one-half, repeatably; oldest (earliest) record always available.

#R,W,0;
        Command: Reset (clear) the WattsUp data memory.
        Note: logging immediately restarts into empty memory space whenever memory is
cleared.
        Note: whenever automatic/condensing mode is in effect, any memory clear
resets logging interval to 1 second

#S,R,0;
#s,-,3,<Reserved>,<Interval>,<LoggingNow>;
        Command: Request current sampling interval.
        Reply: interval used logging, current logging state.

<LoggingNow> current status of logging
 = 0 means internal logging suspended because memory is full
 = 1 means internal logging proceding
 = 2 means external logging proceding

#S,W,2,<Reserved>,<Interval>;
        Command: Set sampling interval.
        Note: clears logging memory, even if change was "no change".

<Interval> integer seconds between logged values

#U,R,0;
#u,-,3,<Rate>,<Threshold>,<Euro>;
        Command: User parameters request.
        Reply: User parameters in effect.

#U,W,3,<Rate>,<Threshold>,<Euro>;
        Command: User parameters set.


<Rate> Cost per KWH, mils (tenths of penny); 0 - 65500.
<Threshold> Duty cycle threshold, watts (units); 0-5000.
<Euro> currency symbol displayed in LCD
 = 0 means dollar
 = 1 means Euro

#V,R,0;
#v,-
,8,<Model>,<Memory>,<HWMajor>,<HWminor>,<FWMajor>,<FWminor>,<FWtimestamp>,<Checksum>;
        Command: Version request.
        Reply: Version information, including raw memory capacity (see #c,r reply).
        Note: this is also the reply to any well-formatted but unrecognized command,
e.g., #X,R,0;.
Example:
#v,-,8,1,65206,5,2,3,14,200612211910,0;

<Model>
 0=Standard
 1=PRO
```

 2=ES
 3=Ethernet
 4=Blind Module
<Memory> Integer number of bytes of nonvolatile logging memory (EEPROM less calibration memory).
<HWMajor> Integer Hardware major version
 5 = WattsUp redesign with ADE7763 and PIC18F45J10
<HWminor> Integer Hardware major version
 0 = no USB
 1 = 4MHz oscillator with USB serial at 19,200bps
 2 = 3.6864MHz oscillator PLL'd x4 with USB serial at 115,200bps
<FWMajor> Integer Hardware major version, e.g. the 13 in 13.5
<FWminor> Integer Hardware major version, e.g., the 127 in 9.127
<FWtimestamp> Integer year, month, date, hour, minute, second of compilation, e.g., 200610312322.
<Checksum> Integer program memory checksum calculated on demand. (Unimplemented.)

#V,W,0;
        Command: Soft restart; like power off/on; firmware program starts over leaving configuration memory intact.

IV. UNITS, FORMAT, AND RANGE FOR THE VALUES SENT FROM THE DATA LOG:

A. All arguments are integer values.
B. Most electrical values such as watts, volts, etc. are represented as tenths of the basic unit. For example, watts would be represented as the number of tenths of watts.

<W> Integer, watts * 10. (Tenths of watts); 0 - 50000 [2 bytes].
<V> Integer, volts * 10. (Tenths of volts); 900 - 2800 [2 bytes].
<A> Integer, amps * 10. (Thousandths of amps); 0 - 20000 [2 bytes].
<WH> Integer, watt-hours * 10. (Tenths of watt-hours); 0 - 2398800000 (= 5000W * 24 hr/day * 1999 days * 10 tenths/unit) [4 bytes].
<Cost> Integer, mils. (Tenths of cents or other currency); 0 - 4294967296 (= 2^32-1; 5kW * 65500 mils/kW-hr * 24 hr/day * 546 days, or 5kW * 17900 mils/kW-hr * 24 hr/day * 1999 days, or 1366W * 65500 mils/kW-hr * 24 hr/day * 1999 days) [4 bytes].
<WH/Mo> Integer, Watt Hours (Units of WHrs); 0 - 3600000 (= 5000W * 24 hr/day * 30 day/mo) [3 bytes].
<Cost/Mo> Integer, mils (Tenths of cents or other currency); 0 - 235800000 (= 5kW * 65500 mils/kW-hr * 24 hr/day * 30 days) [4 bytes].
<PF> Power factor, percent ratio of Watts versus Volt Amps; 0 - 100 [1 byte].
<DC> Duty cycle, percent of the time on versus total time; 0 - 100 [1 byte].
<PC> Power Cycle: Integer, number of power-on events indicates that power was removed at some point during this sampling interval since last memory clear; 0 - 255 [1 byte].
<HZ> Line Frequency, hertz * 10 (tenths of hertz), 400 - 700 [2 bytes].
<VA> Volt-Amps, VA * 10 (tenths of volt-amps); 0 - 50000 [2 bytes].