
Serial Data Format \$Revision: 1.8 \$

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.

1. 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 are some exceptions 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: WattsUp.NET \$ Version: 3.23 \$ 200712212301 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>, <50HzCH10S>, <50HzCH20S>, <50HzWattsPerAENERGYslope>, <50HzWattsPerAENERGYoffset>, <50HzVolampsPerVAENERGYslopeUpper>, <50HzVolampsPerVAENERGYoffsetUpper>, <50HzMidPowerVAENERGYperSecond>, <50HzVolampsPerVAENERGYslopeLower>, <50HzVolampsPerVAENERGYoffsetLower>, <50HzLowPowerVAENERGYperSecond>, <50HzIRMSperAmpSI slopeUpper>, <50HzIRMSperAmpOffsetUpper>, <50HzMidPowerIRMS>, <50HzIRMSperAmpSI slopeLower>, <50HzIRMSperAmpOffsetLower>, <50HzLowPowerIRMS>, <50HzIRMSnoLoadBias>, <50Hzs24LAENERGYperSecondNoLoadBias>, <50Hzs24LVAENERGYperSecondNoLoadBias>, <60HzIsCalibrated>, <60HzVRMSperVolt>, <60HzIRMSOS>, <60HzVRMSOS>, <60HzCH10S>, <60HzCH20S>, <60HzWattsPerAENERGYslope>, <60HzWattsPerAENERGYoffset>, <60HzVolampsPerVAENERGYslopeUpper>, <60HzVolampsPerVAENERGYoffsetUpper>, <60HzMidPowerVAENERGYperSecond>, <60HzVolampsPerVAENERGYslopeLower>, <60HzVolampsPerVAENERGYoffsetLower>, <60HzLowPowerVAENERGYperSecond>, <60HzIRMSperAmpSI slopeUpper>, <60HzIRMSperAmpOffsetUpper>, <60HzMidPowerIRMS>, <60HzIRMSperAmpSI slopeLower>
```

wer>, <60HzIRMSperAmpOffsetLower>, <60HzLowPowerIRMS>, <60HzIRMSnoLoadBias>, <60Hzs24LAE
ENERGYperSecondNoLoadBias>, <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, 3
690, 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

#L, W, 3, T, <Reserved>, <Interval>;

#i, -, 0;

Command: Set the WattsUp to TCP/IP Logging with this interval.

<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 proceeding

= 2 means external logging proceeding

#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

.....
#T, D, 1, <Diagnostic Level>;

Command: Set diagnostic output level

<Diagnostic Level> how much extra serial port output desired

= 0 is Default, no extra output

= 1 standard diagnostic messages #T, D, 1, 1;

= 2 all messages (overkill)

#T, T, 1, <Test Number>;

Command: Execute a System Test

<Test Number> the number of a pre-defined system test

= 1 Flash sector random write/read/verify

.....
#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 (a.k.a. 'dot-Net')

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
6 = WattsUp redesign with ADE7763 and PIC16F87J60 with on-chip Ethernet
<HWMi nor> Integer Hardware minor 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
3 = Ethernet, Normally-Closed relay, PLL 31.2500_MHz
<FWMajor> Integer Firmware major version, e.g. the 13 in 13.5
<FWmi nor> Integer Firmware minor 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 (= 2³²-1; 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].

II. Commands related to TCP/IP

A. Query Basic Network/Internet Configuration:

#I, Q, 0;

#i, q, 7, <IP Address>, <Gateway>, <Name Server1>, <Name Server2>, <Net Mask>, <Do DHCP>, <MAC Address>;

Defi ni ti ons:

1. <IP Address> - IP address in octet format.
2. <Gateway> - IP address in octet format.
3. <Name Server1> - IP address in octet format.
4. <Name Server2> - IP address in octet format.
5. <Net Mask> - Mask as IP address in octet format.
6. <Do DHCP> - Boolean - character '0' = False, character '1' = True
7. <MAC Address> - Address as 48-bit, unsigned integer, READ-ONLY

=====

B. Set Basic Network/Internet Configuration:

#I, S, 6, <IP Address>, <Gateway>, <Name Server1>, <Name Server2>, <Net Mask>, <Do DHCP>;
#i, s, 6, <IP Address>, <Gateway>, <Name Server1>, <Name Server2>, <Net Mask>, <Do DHCP>;

Definitions:

1. <IP Address> - IP address in octet format.
2. <Gateway> - IP address in octet format.
3. <Name Server1> - IP address in octet format.
4. <Name Server2> - IP address in octet format.
5. <Net Mask> - Mask as IP address in octet format.
6. <Do DHCP> - Boolean - character '0' = False, character '1' = True

=====

C. Query Extended Network/Internet Configuration:

#I, E, 0;

#i, e, 5, <Post Host>, <Post Port><Post File>, <User Agent>, <Post Interval>;

<Post Host> - URL/IP Address as String, MAX LENGTH 40
<Post Port> - Unsigned 16-bit Integer
<Post File> - URI as String, MAX LENGTH 40
<User Agent> - URL/IP Address as String, DEFAULT: 'WattsUp.NET'
<Post Interval> - Time in seconds as 32-bit signed integer

=====

D. Set Extended Network/Internet Configuration:

#I, X, 5, <Post Host>, <Post Port>, <Post Address>, <Post File>, <User Agent>, <Post Interval>;

#i, x, 5, <Post Host>, <Post Port>, <Post Address>, <Post File>, <User Agent>, <Post Interval>;

<Post Host> - URL/IP Address as String, MAX LENGTH 40
<Post Port> - Unsigned 16-bit Integer
<Post File> - URI as String, MAX LENGTH 40
<User Agent> - HTTP Client name as String, MAX LENGTH 40
<Post Interval> - Time in seconds as 32-bit signed integer

=====

F. Read current Network/Internet configuration from NV-Memory.

#I, R, 0;

#i, r, 0;

=====

F. Write current Network/Internet configuration to NV-Memory.

#I, W, 0;

#i, w, 0;

G. Reset Factory Defaults.

#I, Z, 0;

#i, z, 0;

H. Soft reset the meter

#V, W, 0;

III. Web Server POST Data Format

WattsUp.Net makes a TCP connection to a host (assumed HTTP server) at default port 80, and uses HTTP POST to send a parameter list in standard URL encoding, as shown in the following example.

EXAMPLE POST FORMAT

POST /remote/netlog.php HTTP/1.0
Host: meters.wattsup.net
User-Agent: WattsUp.NET
Content-Length: 59
Content-Type: application/x-www-form-urlencoded
id=1&w=0&v=1199&a=381&wh=0&pcy=0&frq=599&va=458&rnc=0&sr=20

File for POST, Host, and User-Agent are configured using the #I,X command as described above.

The following 3 fields are always sent with a POST.

Table with 3 columns: Description, variable name, and explanation. Rows include Machine ID (id), Relay Status (rnc), and Send Rate (sr).

Depending on fields selected for logging, any or all of the following will also be posted.

Table with 3 columns: Description, variable name, and explanation. Rows include Watts (w), Volts (v), and Amps (a).

CommunicationsProtocol 090824.txt

Watt Hours	wh	
Max Watts	wmx	
Max Volts	vmx	
Max Amps	amx	
Min Watts	wmi	
Min Volts	vmi	
Min Amps	ami	
Power Factor	pf	
Power Cycle	pcy	
Frequency	frq	-- Line frequency scaled by 10
Volt Amps	va	

After each post, the host returns the relay status in brackets:
[0]

0 = closed (load on)
1 = open (load off)

The interval can also be changed by returning a new value (sec) after !:

[0!4] = relay closed, change interval to 4 seconds

The .Net meter and the SC20 can both output data as fast as once per second. Please keep the following in mind: Update the interval only when a change is required - not in every reply. Also, there should not be a space between the '0' and the '!' .

If the interval is truly being updated to a new value, the device will report out the USB:

"net_update_rate_from_server(2) net_config_write_to_nv(), CRC8 pre-write verify OKAY."

If there is no update to the interval it will report:

"No change; interval already ==2"

So, to reiterate please follow these guidelines,

- No spaces in the reply from server.
- The update "!2" (or "!15", etc.) is optional, not necessary to send every time.
- Normally only reply with "[0]" or "[1]".