

# Low power solar PV logger

Adapted from

<https://github.com/chickey/RS485-WiFi-EPEver>

Updated 24/12/23

Simplified GUI : removed MQTT (manual upload) : hardware documented

## The problem

A solar system provides lighting for a hut located in Scotland and used by various outdoor groups.

Battery storage capacity is being increased to 220aH 12 Volt Lithium (2400 Wh)

Because :-

- Winter solar irradiation is low.
- Occupancy has increased.
- Occupants may be using lights excessively.
- Verbal reporting of battery meter level by visitors is erratic.

## Solution

- Log user power, solar power and remaining battery capacity.
- GUI showing remaining battery capacity to non-technical users.
- ~~Graphic showing use against time.~~
- Data upload for analysis.

## Design Criteria

- Burden: the amount of power consumed by the logger itself must be minimised.
- Operation and UI must be straight-forward.

## Keeping it simple®

- Modify code written by Chickey... does most of what we need.
- ESP8266 in deep sleep and woken approximately every 30-60 Seconds using Timer-RESET
  - Read Epever parameters using MODBUS save to RTC RAM and shutdown
  - IF RTC time is midday then Epever LOAD OFF
  - IF load transition OFF- ON has occurred Wake Wi-fi for 5 mins
- Graph previous week/months load on GUI
- Display parameters on 'capture-portal', (No network available or req'd)
- Config password protected (hard coded)
- Download CSV log files using web GUI

<https://forum.arduino.cc/t/file-download-with-arduino-web-server/160140/9>

-Wi-Fi /GUI prompts timeout to disconnection?

- Whilst in web-gui update load/soar parameters and time every second
- 'interrupt' using microchip 12F683 for remote switch
- real time clock setting

### FRAM/EEPROM logging

32K bytes of FRAM

#### Daily Record

Parameter	Value	range	Comment	Bits	Cuml.Bytes
Peak Solar power	watts	0-120		7	1
Solar power (day)	w/h	<b>0-1024</b>		10	2
Peak Load power	watts	0-64	Not strictly needed	6	
Load power (day)	w/h	0-254		8	3
Battery Volts	V	10.7-15		8	4
Load ON	BIN	True/False		1	5
Battery Full	BIN	True/False	>=14.2 V + no current for 10 min?	1	
Daily Data Valid?	BIN	True/False	ESP running for 24 hours		
FLAG TBA	BIN	True/False			
FLAG TBA	BIN	True/False			
Min Temperature	(1)degC	-20 - +40		6(8)	6
Max Temperature	(1)degC	-20 - +40		6(8)	7

**8 Bytes of daily data, makes address mapping simple? 4000 days of data...**

**Even just using 4 K of internal EEPROM – 512 days**

#### Hourly Record

Parameter	Value	range	Comment	Bits	Cuml.Bytes
Solar power	w/h	<b>0-100</b>		8	1
Load power	w/h	0-100		8	2
Battery Volts	V	10.7-15		8	3
Load ON	BIN	True/False		1	Mask Temp
Battery Full	BIN	True/False			
Min Temperature	(1)degC	-20 - +40		6	4

4 bytes/hour \* 24 \* 4

Just put the last four days into 512 Bytes of ESP RTC RAM? – maybe even squeeze in five?

## Accumulating data

Sample: Temperature, Solar and Load power every 40 seconds – nothing will change much

Every minute add instantaneous power to total e.g. 10 Watts Increment sample number

At end of day.. w/h = Cumulative / number of samples \* 24 100 watts for 12 hours = 1200, samples 1440 = 4 BYTES

Written as a bit map for efficient memory use. With a daily write -

## Links

<https://community.platformio.org/t/correct-settings-for-esp8266-d1-mini/30681>

## Program design

Andreas Spiess

## Using RTC memory

<https://www.youtube.com/watch?v=r-hEOL007nw&t=64s>

ESP deep sleep

<https://www.youtube.com/watch?v=r75MrWIVlw4>

## Writing to EEPROM

[ESP8266 have 4K bytes of EEPROM](#)

<https://circuits4you.com/2016/12/16/esp8266-internal-EEPROM-arduino/>

[\(512 Bytes of NVRAM\)](#)

<https://www.aranacorp.com/en/using-the-EEPROM-with-the-esp8266/#:~:text=The%20EEPROM%20of%20the%20ESP8266%20has%20a%20size%20of%204kB.>

[Once daily recording parameters efficiently as packed BCD and scaled values requires 15 bytes](#)

[Programing write address as](#)

<https://www.arduino.cc/reference/en/libraries/osfs/>

## Hardware MH-ET-LIVE

[https://riot-os.readthedocs.io/generated/group/group\\_boards\\_esp32\\_mh-et-live-minikit.html](https://riot-os.readthedocs.io/generated/group/group_boards_esp32_mh-et-live-minikit.html)

[https://doc.riot-os.org/group\\_boards\\_esp32\\_mh-et-live-minikit.html](https://doc.riot-os.org/group_boards_esp32_mh-et-live-minikit.html)