Low power solar PV logger

Adapted from https://github.com/softwarecrash/EPEver2MQTT

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 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.
- UI must be straight-forward.

Keeping it simple

- Modify code written by TR … does most of what we need.
- ESP8266 in deep sleep and woken approximately every 20 Seconds by 12F683
  - WiFi – off
  - Read Epever parameters using MODBUS save to RTC RAM and shutdown
  - If Epever RTC is 'on the hour' :
    - AND IF solar or Load use data exists for the hour write to hourly EEPROM log.
    - If load ON Wake Wi-fi and check if user hotspot available.
    - If hotspot sync EEPROM data to MQTT Server.
      - Display progress on local web-portal, Wi-Fi /GUI remains on until hotspot disabled
  - If time is past midnight then log to daily EEPROM
  - If RTC time is midday then Epever LOAD OFF

EPROM logging

ESP8266 4K bytes of EEPROM 100,000 write cycles

Hourly Record

Hourly data	Last record, flag	Load on	hour	Date	Solar w(h)	Use w(h)	bits	hrly bytes
Parameter			00-24	31/10/23	<100	<100		
bits	1	1	6	24	7	7	48	6

240 records would occupy a block of 1440 bytes

Data only written if necessary - solar or load in use.

Writing and reading done by iterating through flag bit locations and searching for set bit

Distributes EEPROM wear which would occur with a single index register

Clear Flag, increment counter and write new record, if counter rolls over overwrite RRD.

Daily Record

Daily data	Flag Register	Date	Solar wh	Use wh	Batt%	Volts	Load ON	Load OFF	Daily Bytes
Parameter	BITS	31/10/23	520	120	80	12.0	08:00	12:00	
Bytes	1	3	2	2	1	1	2	2	14

120 Days 1600 Bytes RRD, with index register and null records

Web GUI 1

Epever Logger

Device Time:	19/11/2023	18:58
Use since full charge:	10/09/2023	14:00
Solar	120	Wh
Load	439	Wh
Burden	20	Wh
Remaining Capacity		70%

Live Data

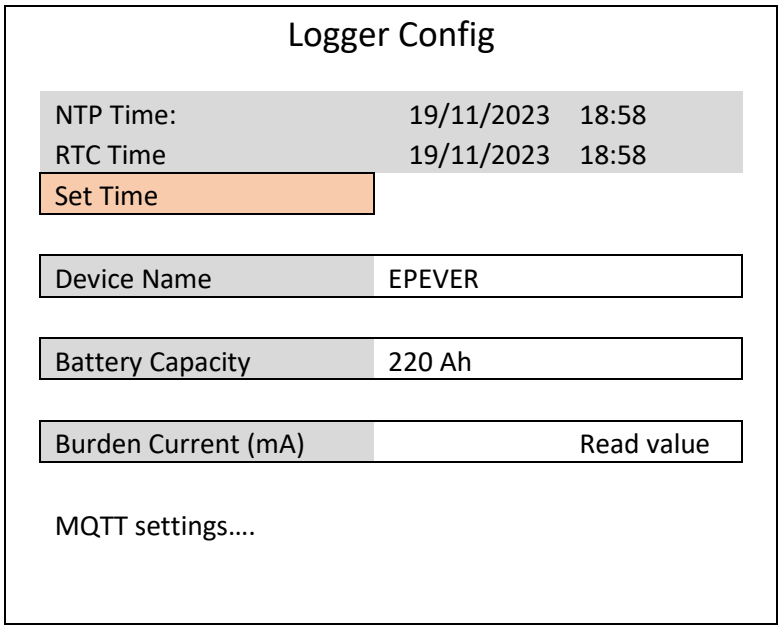
Solar	31.0V 0.5A 15.5W
Battery	12.5V 1.0A 12.5W 65%
Load	12.5V 0.24A 3 W
Free Memory	90%
Battery Type	Lithium

Read only Config Checks

Parameter	Live data	Value
Charge Voltage	14.2	14.2
Under Temperature	0	0
LVD	10.6	10.6

Config (password)

Web GUI 2



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=r75MrWIVIw4

ESP exact Time (swiss railway clock)

https://www.youtube.com/watch?v=BzFM3PWx1rg

https://www.youtube.com/watch?v=BzFM3PWx1rg

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://doc.riot-os.org/group__boards__esp32__mh-et-live-minikit.html

Board Configuration