Low power solar PV logger

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

Updated 26/11/23

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 'Softwarecrash' TR … does most of what we need.

- ESP8266 in deep sleep and woken approximately every 20 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

  - Display status on 'capture-portal', (No network available or req'd)

  - Download CSV files using web GUI

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

  Config interface password protected

  -Wi-Fi /GUI remains on if client remains connected?

- **If Epever RTC is ‘on the hour’**
  - Solar or Load use data exists for the hour? > write to hourly EEPROM log.

Reduces data, for example in mid-winter, hut unoccupied only 6 records logged

EPROM logging

ESP8266 4K bytes of EEPROM 100,000 write cycles

Hourly Record

Hourly data	Index	Load		<0C	hour	dd	mm	yy	Solar w(h)	Use w(h)	Battery V	hrly bits
Parameter					00-24				<100	0-24	10.0-15.0	
bits	1	1		1	5	5	4	5	7	5	6	40

Records	Bytes	Size	
400	5	2000	Bytes

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

Written as a bit map for efficient memory use – code complexity?

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 data

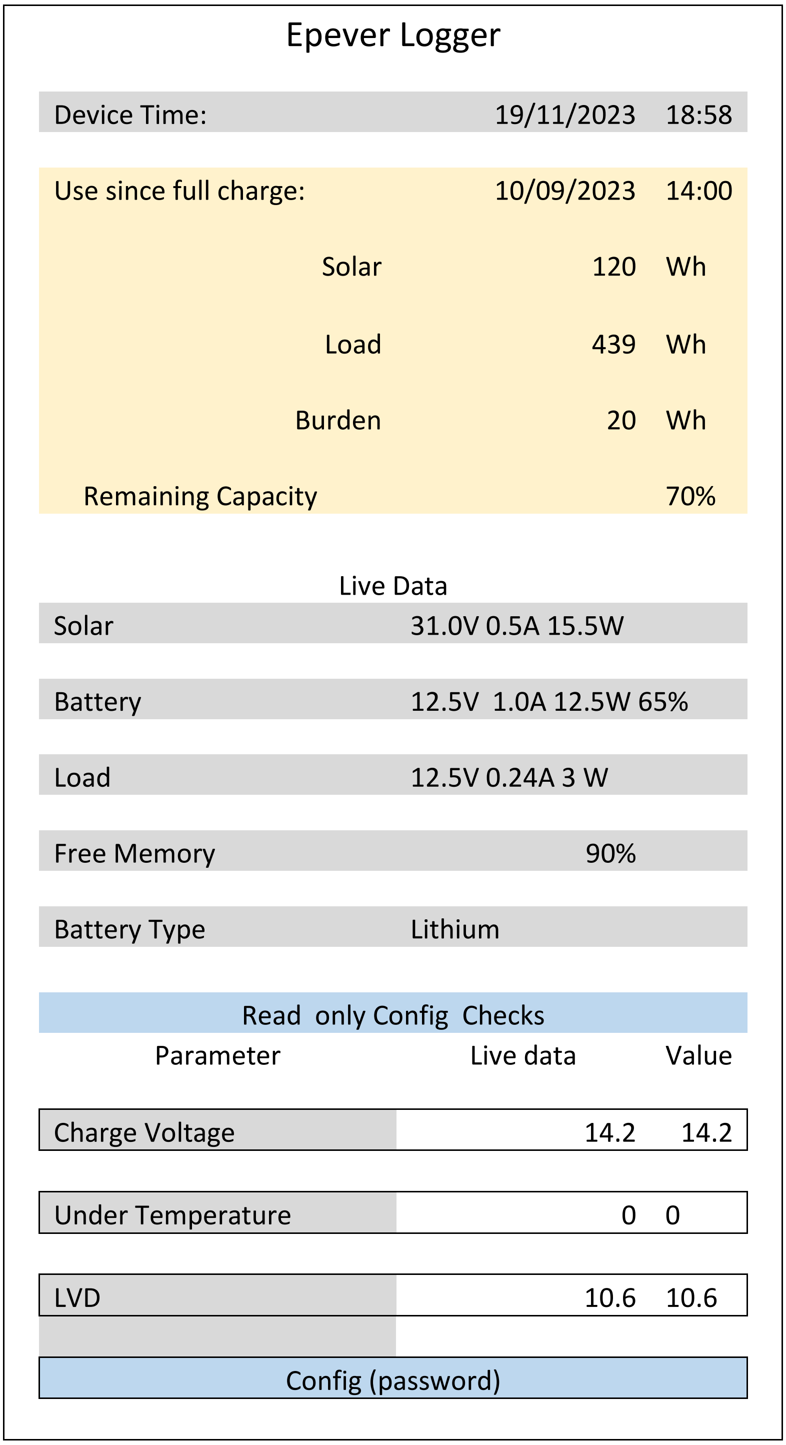
Again bit map for most efficient use of EEPROM

Daily data	flag	Date			Solar wh	Use wh	Battery V	Daily Bytes
Parameter	Batt Full	dd	mm	yy	<1024	<512	10.0-15.0	
Bits	1	5	4	5	10	9	6	40

days	bytes
240	1200

No need to iterate through index flag – just use index register location

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

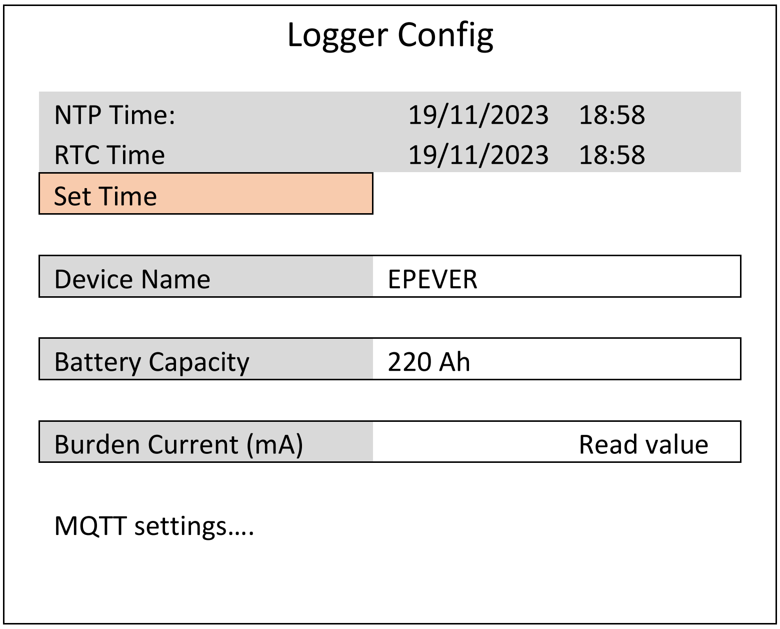
Solar	31.0V 0.5A 15.5W
Battery	12.5V 1.0A 12.5W 65%
Load	12.5V 0.24A 3 W
Free Memory	90%
Battery Type	Lithium

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



Logger Config

NTP Time:	19/11/2023	18:58
RTC Time	19/11/2023	18:58
Set Time		

Device Name	EPEVER
Battery Capacity	220 Ah
Burden Current (mA)	Read value

MQTT settings....

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