**7" WIRELESS WATERPROOF REMOTE DISPLAY FOR OPENPLOTTER
(PART #2)**

HOW TO PROTECT RPI DISPLAY FROM ACCIDENTAL POWER OFF
The problem is that, as all Pi's, the unit starts on every power-on, but it is needed to make a software shutdown BEFORE power-off. Not doing this way, there is a risk to corrupt the SD and being unable to re-start.

So is no good to switch-off "navigation instruments" to stop the display. (And to make a software shutdown in the display requires to exit from Chromium, return to Desktop Menu, etc.) Not everybody wants to do that, and my co-skipper has the good costume to use ONLY the main battery switch when closing the boat.

A good idea from Techstyle in OPEN MARINR FORUM led me to develop a circuit to automatically shutdown de Pi by detecting a switch-off of supply. (But I use a different approach, because I did not want to pass new cables to navpod)

This was, really, a completely new project.

THE UPS PROJECT
If you want to have an automatic shutdown whenever the equipment is switched-off, you need:
1) An alternative supply to "maintain" the Pi during the software shutdown process.
2) A falling voltage detector in the supply.
3) A GPIO signal to the pi, to force shutdown

 1.- For the first issue, I installed a small bank of super-capacitors (about 7,50 euros in Aliexpress) This is much more simple that tradicional Li-Po batteries, longer life, and also cheaper. Do not need circuits to control charge or discharge. (The bank has his electronic to balance capacities and max. voltage between capacitors. Those banks are actually used as an alternative to batteries in many toys and, in bigger sizes, in emergency start boosters for car engines).
So; with a super-capacitor bank connected in parallel whit the battery 12V supply, you get an UPS for the Pi.
I made some calculations to forecast the capacity needed in this project and determined that 1,6 Farad and a pack of six SC (16V limit) was enough. The bank I bought is exactly this (even if vendor claims 2 farad; really is 1,67Farad)
As the SuperCap. is connected in parallel, in seconds is charged to the battery voltage (12 - 13 V)

This is a simple way to secure a Pi from accidental power-off, and only need to buy enough capacity for the time you want to maintain it alive.
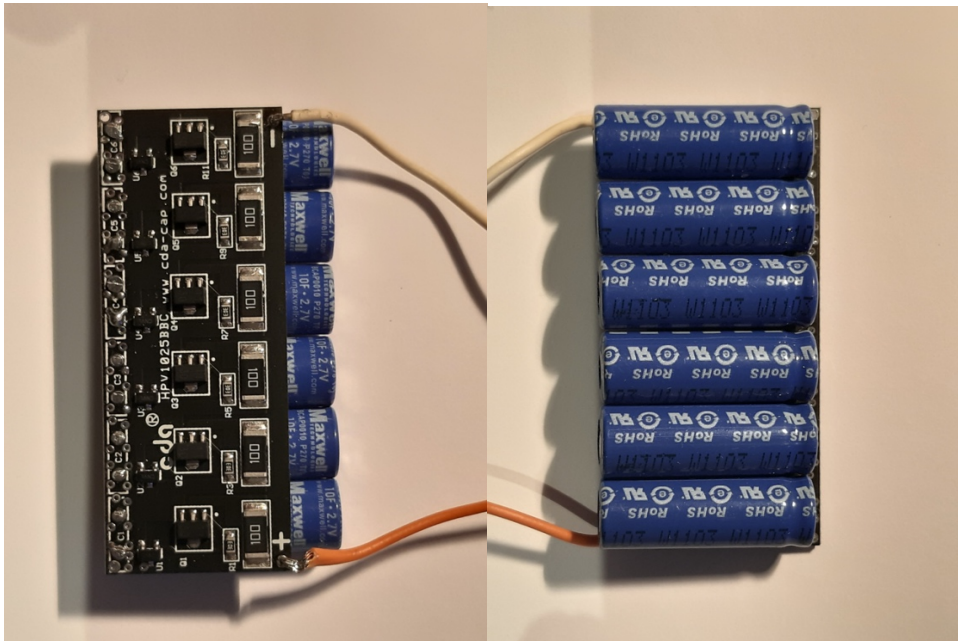
2.- For supply voltage detection, I used a simple IC (Operational Amplifier ref. CA3130) that compares the supply voltage from battery input with a reference voltage generated by a Zener diode that is fed from de SC. When input voltage falls below reference; the IC output goes high and activate a reed-relay (R1). The normally open contact of R1 closes and trigger the third circuit.

3.- This part works in the following way:
A electrolytic condenser is charged any time the normal supply is on.
When R1 detects power-off, the condenser discharges trough a voltage reducer (Ref.7805) that reduces to 5V and activates a reed-relay (R2). When voltage drops to aprox. 7V, there is not enough current to maintain the relay, and R2 is disengaged again.

The contact of R2 connects GPIO3 (pin 5) to GND (pin 6) during about 1 second, to force a software shutdown.
Only is needed to add the line:

```
dtoverlay=gpio-shutdown
```

In the file :   `/boot/config.txt`

If this is done when is shut, (but with supply), the pi starts again.
This is a standard feature of Rpi's

In parallel with this contact in R2, a pushbutton can make the same effect manually.

So;
The Pi starts normally when power is switched on.
Every time that the equipment is switched off; the circuit provides a soft shutdown (or if there is any other supply failure)

With the manual pushbutton in the navpod, the system can be shutdown and re-started without using the supply switch.
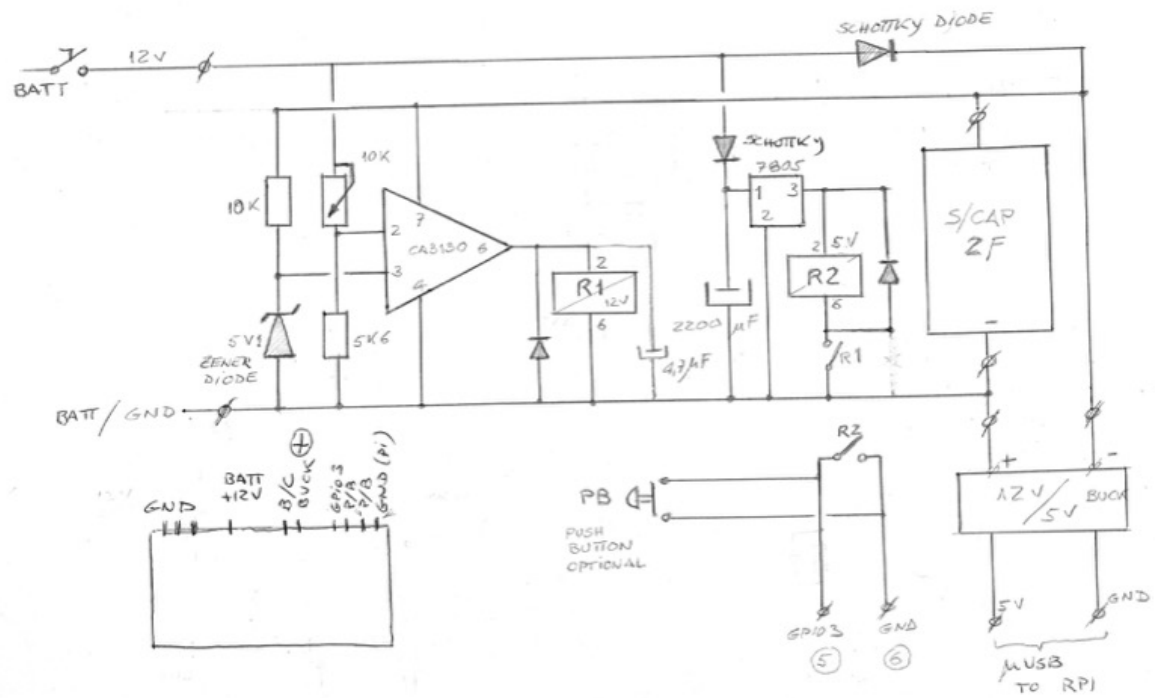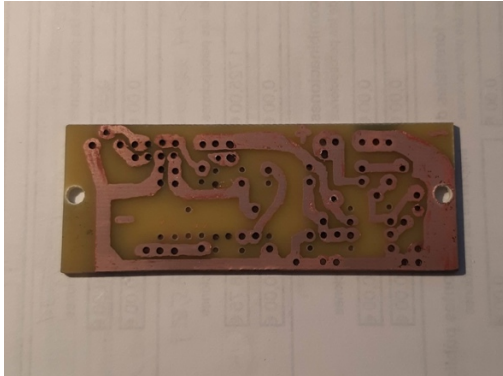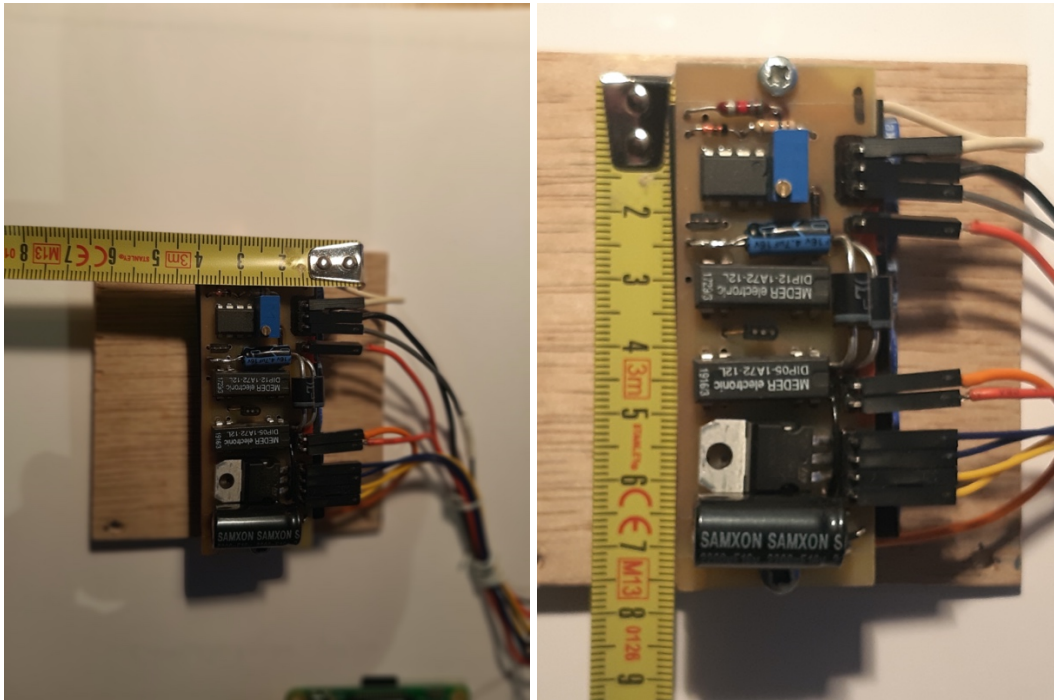
All material needed for this arrangement costs a few euros, and can be placed in a small PCB (that can be much more reduced using SMD format of IC's, but a can't solder pins so close!). Except the reed-relays, all other components were recycled from an ancient battery park monitor I made years ago and now has been substituted by a INA3221 through I2C to SignalK in Openplotter project.
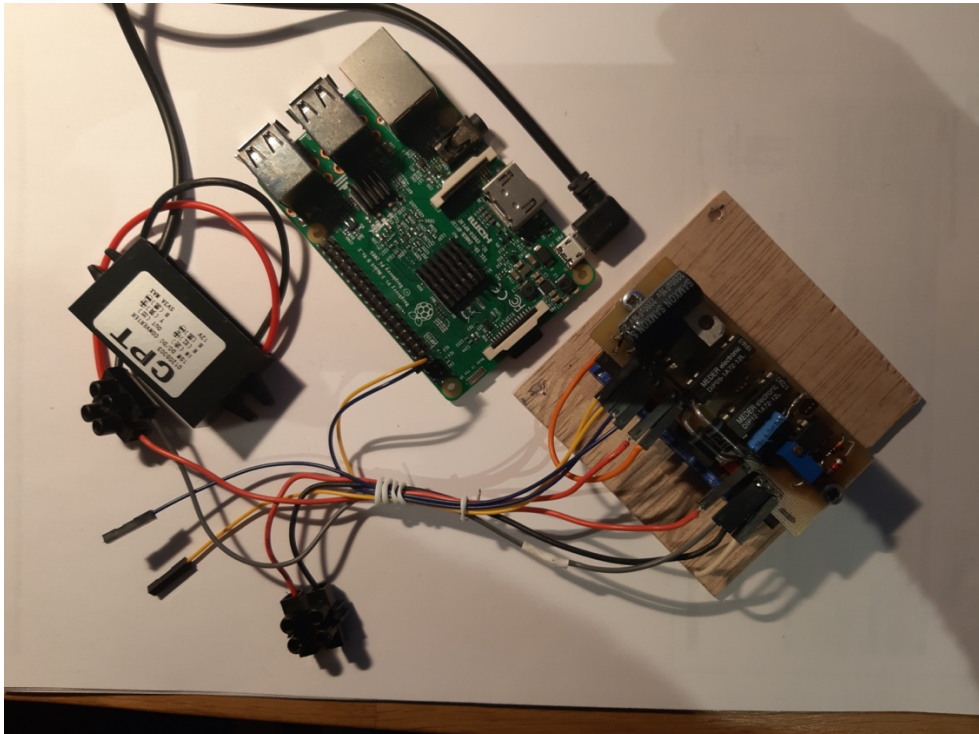
(Testing the project with another Pi)