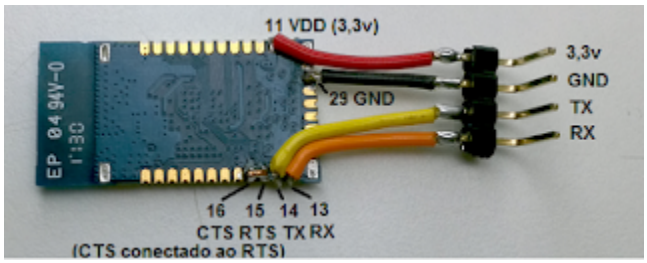
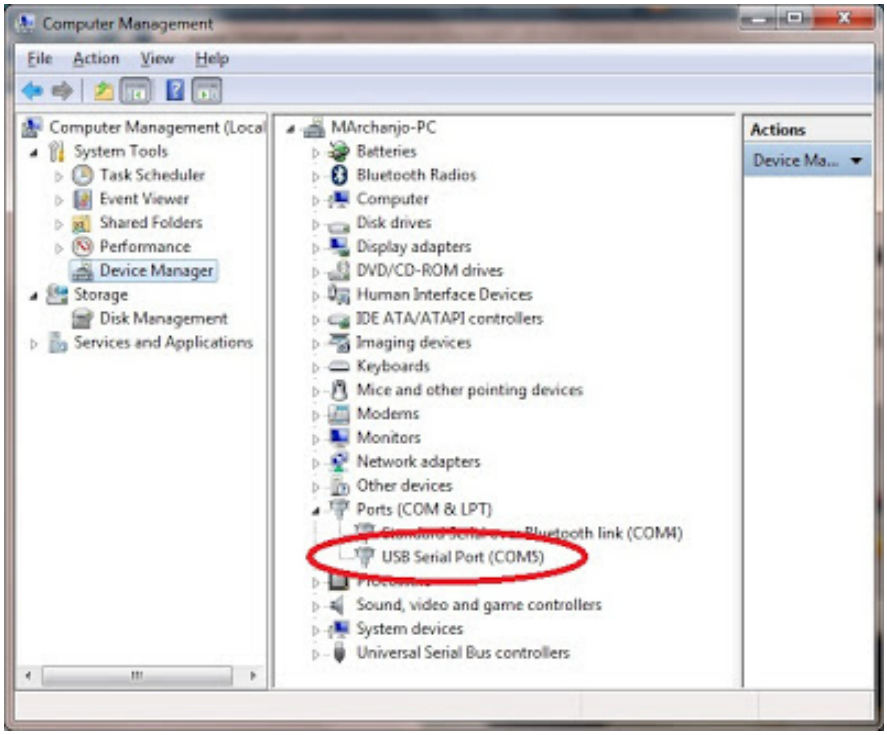

# Whame assembly steps

1. In case the Arduino Nano provides the pins without soldering, first solder all.
2. Then do the same to the MPU6050 in case it is needed.
3. Now solder cables and pins to the RN42 bluetooth module to configure it.



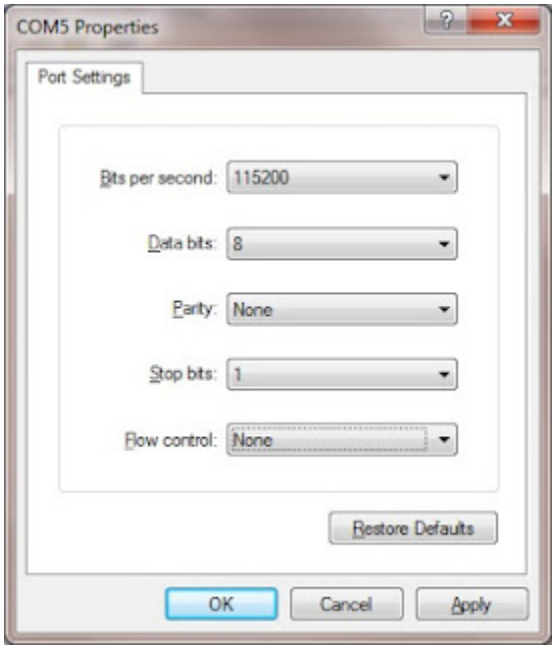


4. Now connect it to the FTDI cable and connect the FTDI USB part to a computer.
5. Start any serial port monitor program like Hyperterminal or any other.
6. Look for the COM Nº port in the computer.

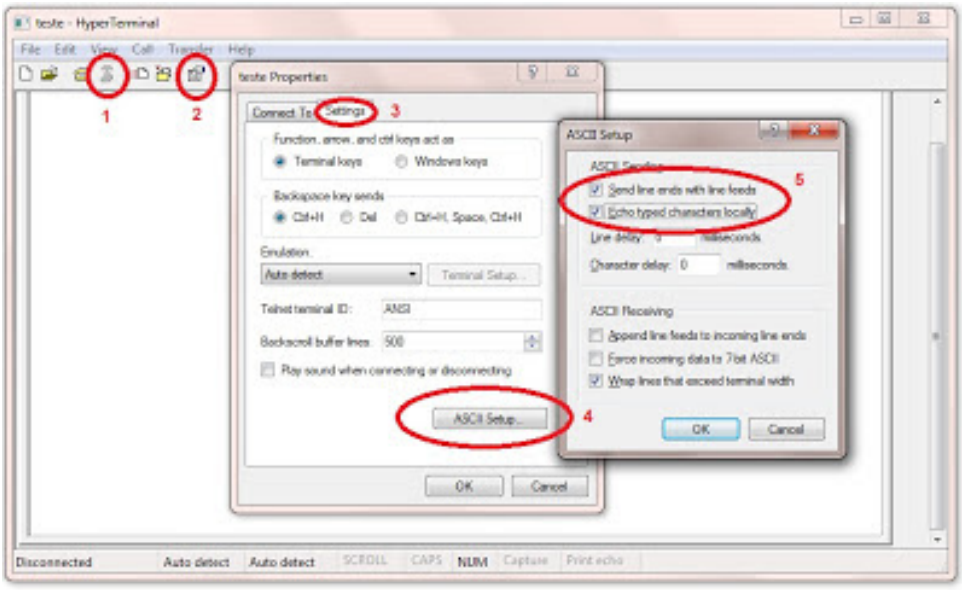


7. Now configure Hyperterminal or your serial port monitor program.


8. Parameters: **Baudrate**=115200, **Bits**=8, **Parity**=None, **Stop bits**=1, **Flow control**=None



9. Now configure the terminal:
    a. Disconnect
    b. Properties
    c. Settings
    d. ASCII Setup
    e. Set both indicated options:
        i. Send line ends with line feeds
        ii. Echo typed characters locally




10. Now you can configure the bluetooth module as a HID Mouse, complete list of commands to configure this module are in: [Advanced User Manual Roving Networks Bluetooth™ Product User Manual]

11. With the following commands we configure it as we want for Wahme purpose:

```
$$$                 will respond CMD, now you are in command mode
ST,255                  will respond AOK, disable timer for been able of
sending commands more than 60 seconds
D                   shows basic module information
E                   shows advanced module information
SF, 1                restore values to default
R,1                  reboot the bluetooth module
$$$                 enter again in command mode
ST, 255             disable timmer
S-, Whame-Mouse   rename device to Whame-Mouse-xxxx, the last 4 digits are
the MAC address of the module and are automatically added.
SH,0220             set the device as a mouse
S~,6                turn on HID mode
SM, 4                   Auto-Connect DTR Mode, allow auto reconnect the last
connected address when receive a C from command mode
R,1          reboot the bluetooth module
```

12. Now the bluetooth module is configured as a HID Mouse, you can check it by performing a search with a phone and see the name has changed to Whame-Mouse and is detected as a bluetooth mouse.

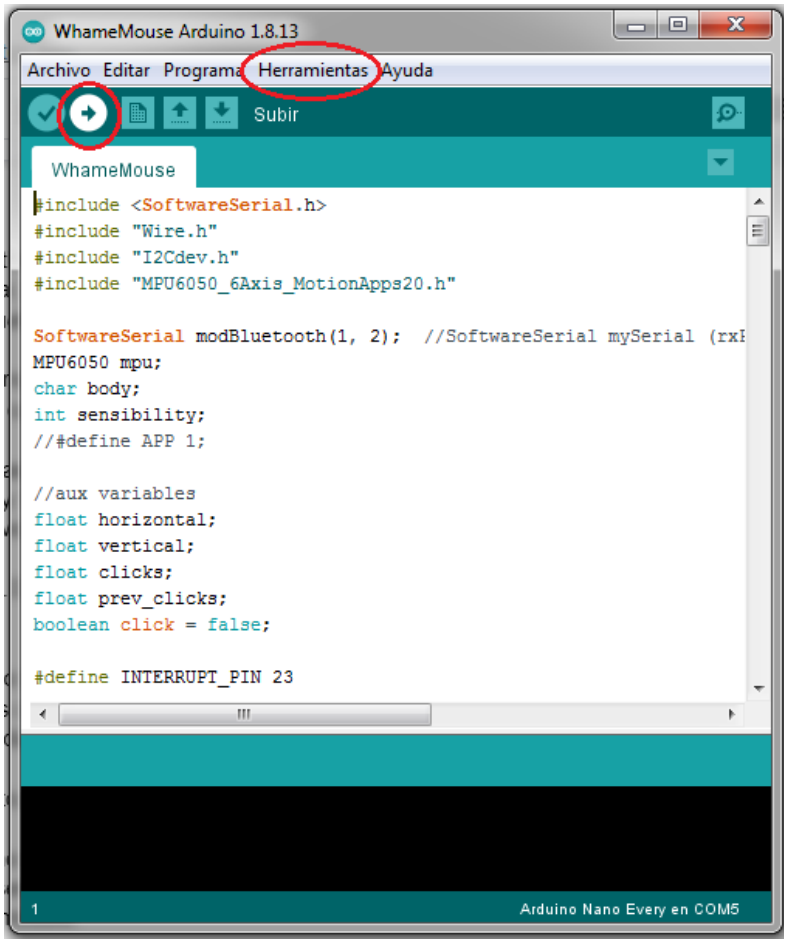
13. Next step is to program the Arduino Nano, for that you need to install in your computer the Arduino IDE, can download from this link https://www.arduino.cc/en/software

14. Once the IDE was installed, download the files into the “Arduino SW” folder in Whame GitHub and copy the libraries to your Arduino→Libraries folder, usually in Documents folder for Windows users.

15. Open *Whame_Head_1.0* or *Whame_Arm_1.0* depending on where you're gonna wear it, and connect the Arduino Nano to your computer using a micro USB cable.

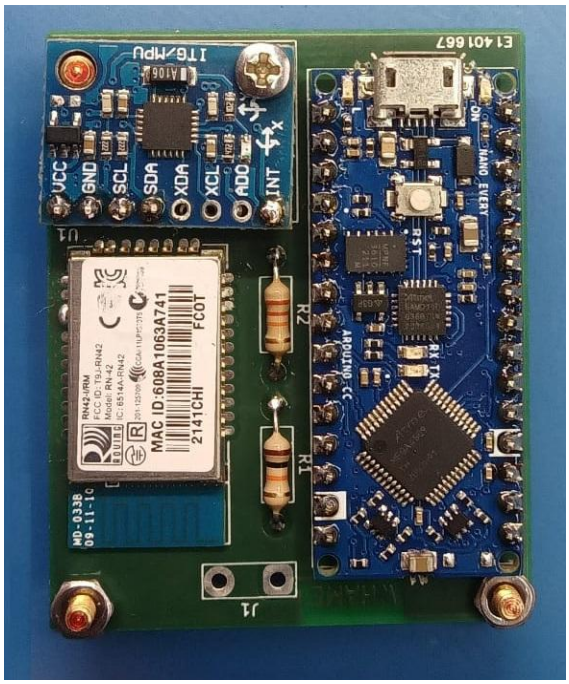
16. Set Arduino Nano Every in Tools → Board. Set the port COM X also in Tools, is the port where Arduino is connected, can look for it on Computer Management, like we did for the bluetooth module detection at the beginning.

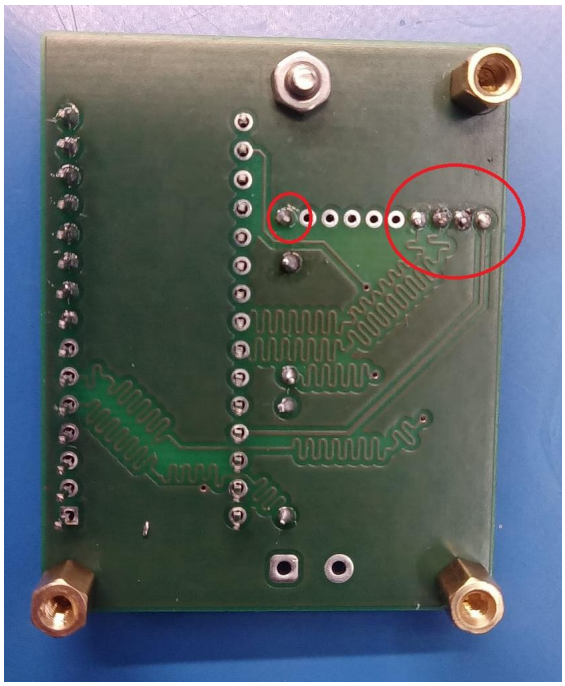
17. Click on Upload to program the Arduino.



18. Once the Arduino is programmed and the bluetooth module has been configured, the PCB can be assembled. But before doing it, we need to desolder the cables from the RN42 bluetooth module.

19. Solder the Arduino, the MPU and the bluetooth module as is shown in the figure, notice not all the pins need to be solder, recommendation is to solder at least the four GND points of the bluetooth module for a good signal performance, and due a fail in the PCB pads design for the MPU they are too close each other so solder only the four used pads, thats Vcc, GND, SDA and SCL.

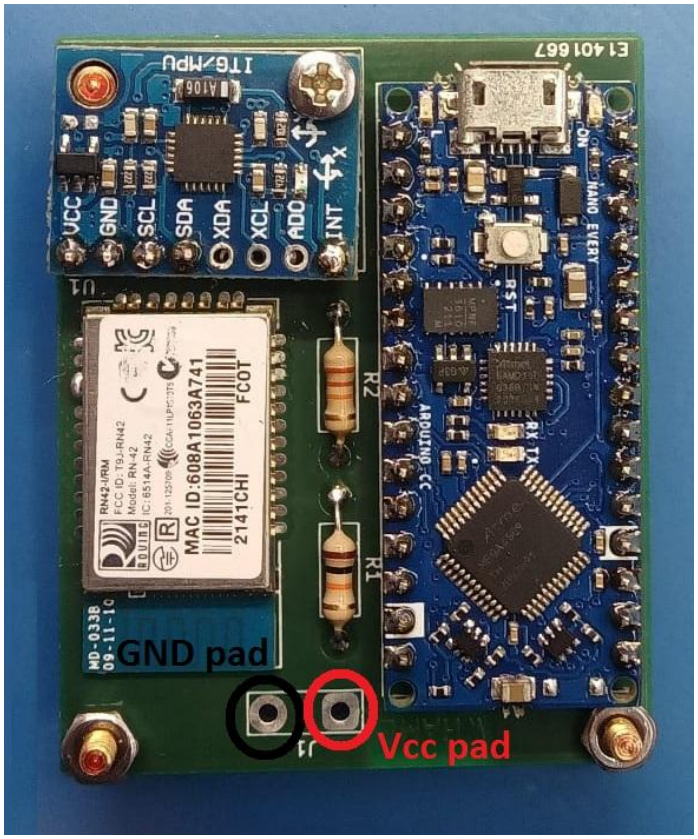


20. Cut the remaining part of the pins.

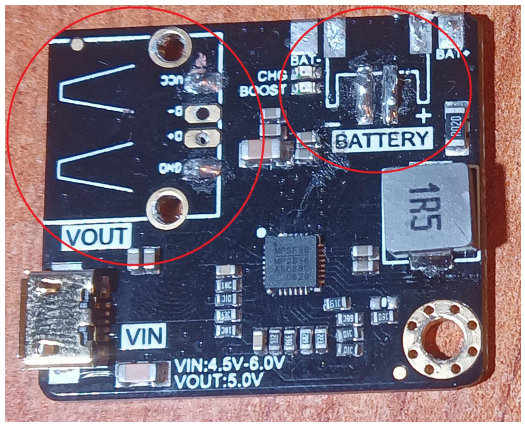
21. Notice the MPU needs a 5 mm Male-Female spacer on the screw part to remain fixed and stable, it is not critical that it's completely horizontal, some degrees are not a problem because it will be corrected by the calibration once it is assembled into the box.

22. Cut and strip 10 cm of AWG21 red cable and another 10 cm of black cable. Solder the red cable to the Vcc pad and the black cable to the GND pad.



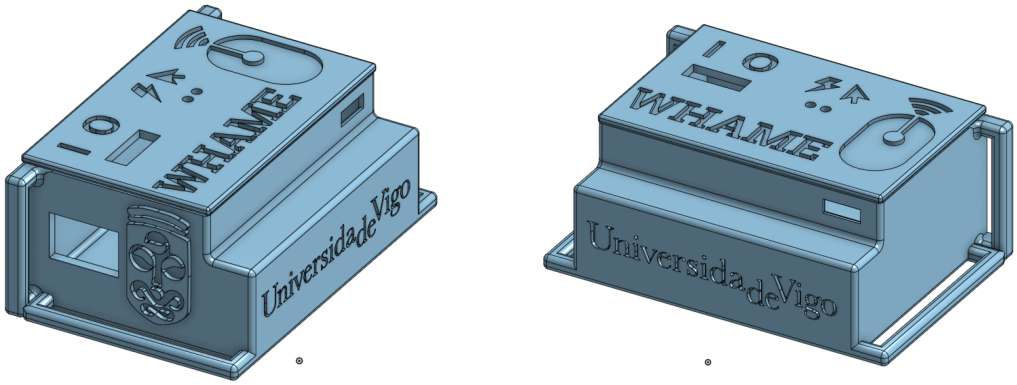

23. Now the PCB is finished. Let's go to the power booster part.

24. The first is to desolder the USB female from the *MP2636 power booster and charger*.

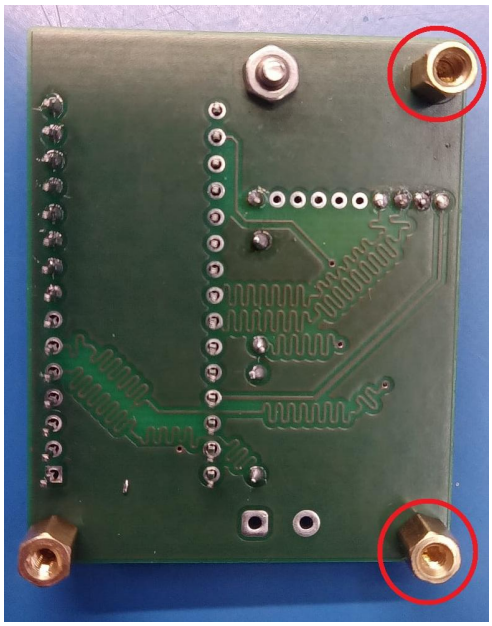

25. In my case I've also desoldered the battery connector to solder the battery cable directly cause it was the first prototype, but is not necessary due the battery connector fits if the power booster connector.

26. Now is time to build the box, the stl files for print are in the 3D Print Box folder, if you have a 3D printer recommendation is to print with a 0.4 mm extruder and at 0.2mm Z resolution maximum. Also let 0.4mm Z distance for the supports because the internal ones are hard to remove.



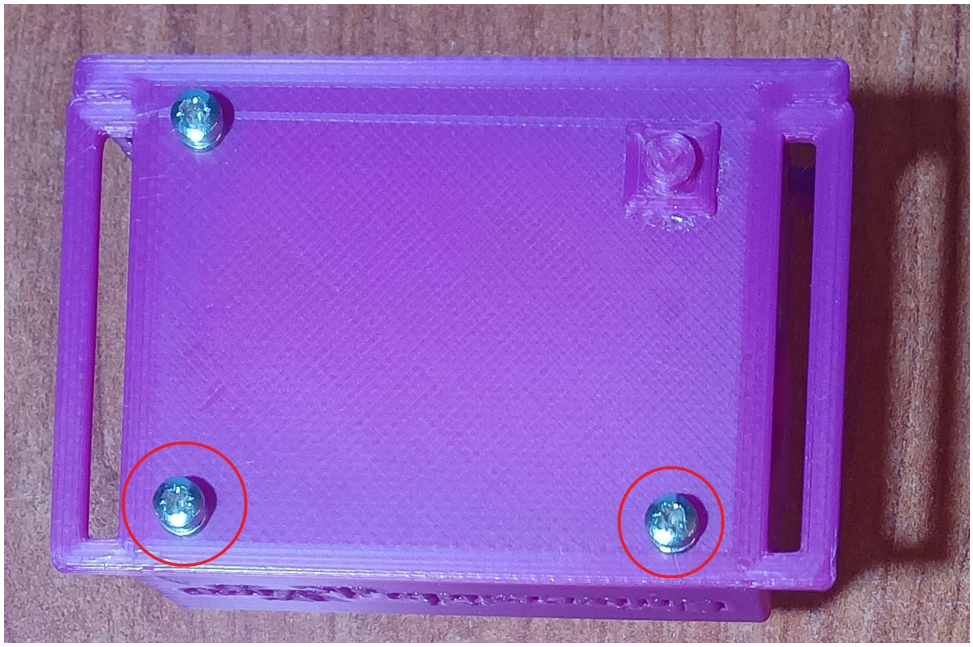

27. The battery goes first, double sided tape to fix the battery to the box floor, will be under the PCB, so we need to use 2.5x8 mm Female-Female spacers in the PCB bottom. Take care the cable stays out passing through the border when you put the PCB over it.

28. Assembly the two spacers to the PCB as shown in the following figure, use 2.5x6 mm screws and pressure washers from the PCB top. Don't assemble the third spacer yet.



29. The third spacer must be assembled first in the box using a 2.5x6mm screw and a pressure washer from the bottom of the box.

30. Then fit in and fix the PCB using 2.5x6mm Female-Female screws and pressure washers from the bottom of the box.



31. Now the PCB is fixed, adjust a 2.5x6mm Male-Female spacer on the free spacer at the corner of the Arduino Nano.

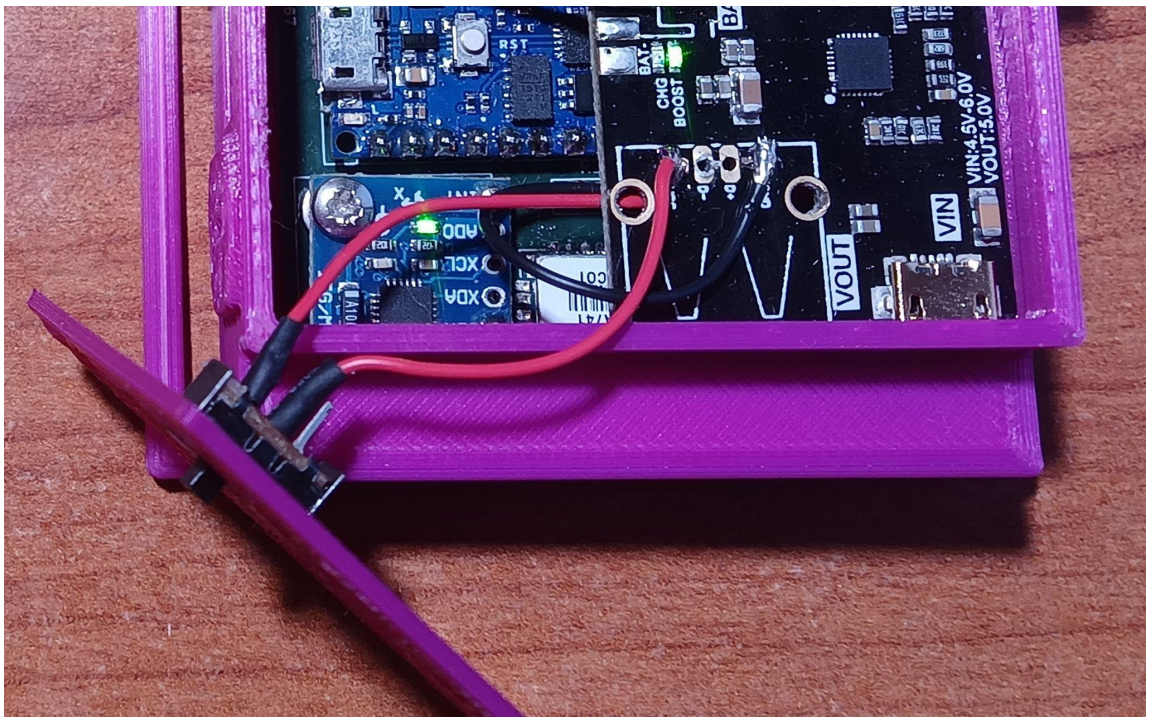
32. Carefully fit the power booster and adjust it to the corner spacer using a 3x6mm screw and a pressure washer. Take care the micro USB input fits the hole of the box.

33. Fit the switch in the hole of the top cover of the box, if it is not fixed enough, apply hot silicone to the bottom side.

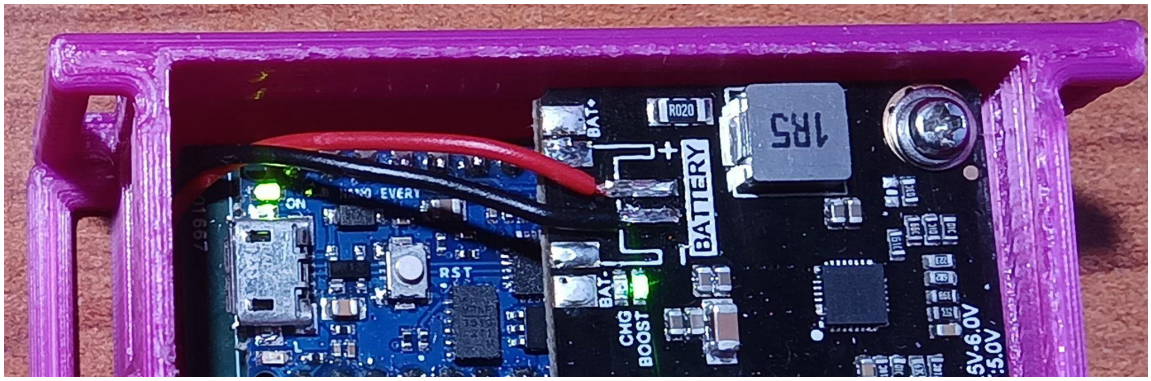

34. Cut, strip and solder the black cable from the PCB to the GND pad of the unsoldered USB on the power booster.

35. Cut the red cable, with some extra length to reach the cover, and strip it and solder it to the left pin of the switch. Cut the remaining part of the pins.

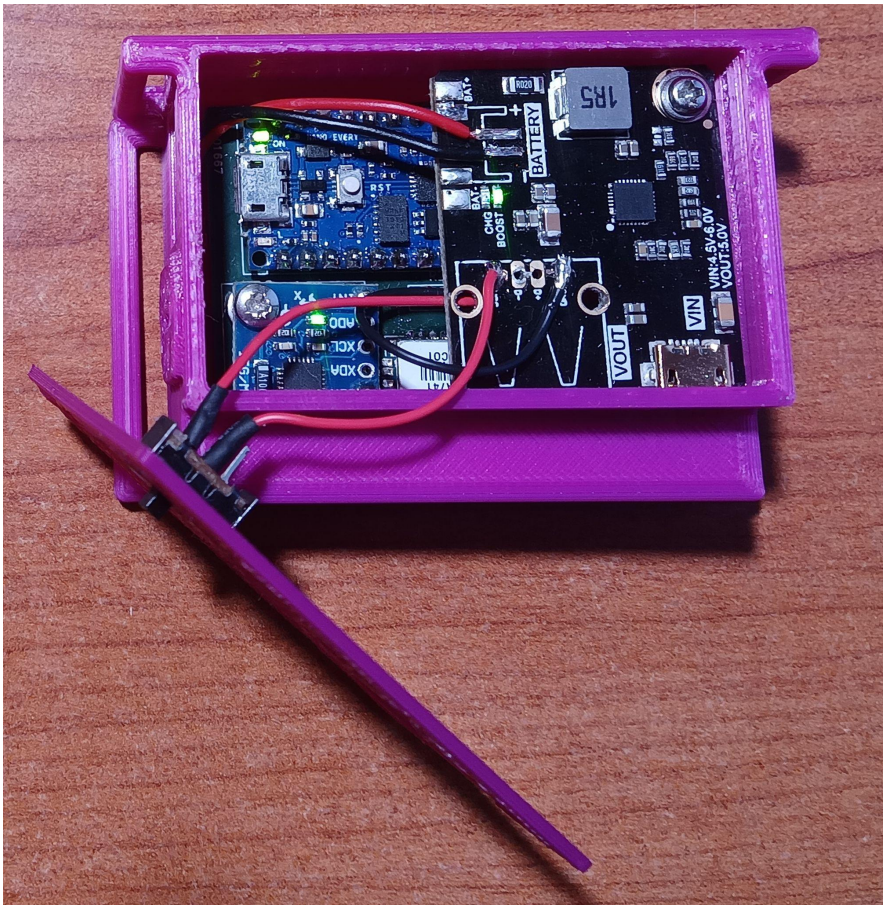
36. Cut another red cable and strip it and solder it to the middle pin of the switch, and the other side to the Vcc pad of the unsoldered USB on the power booster



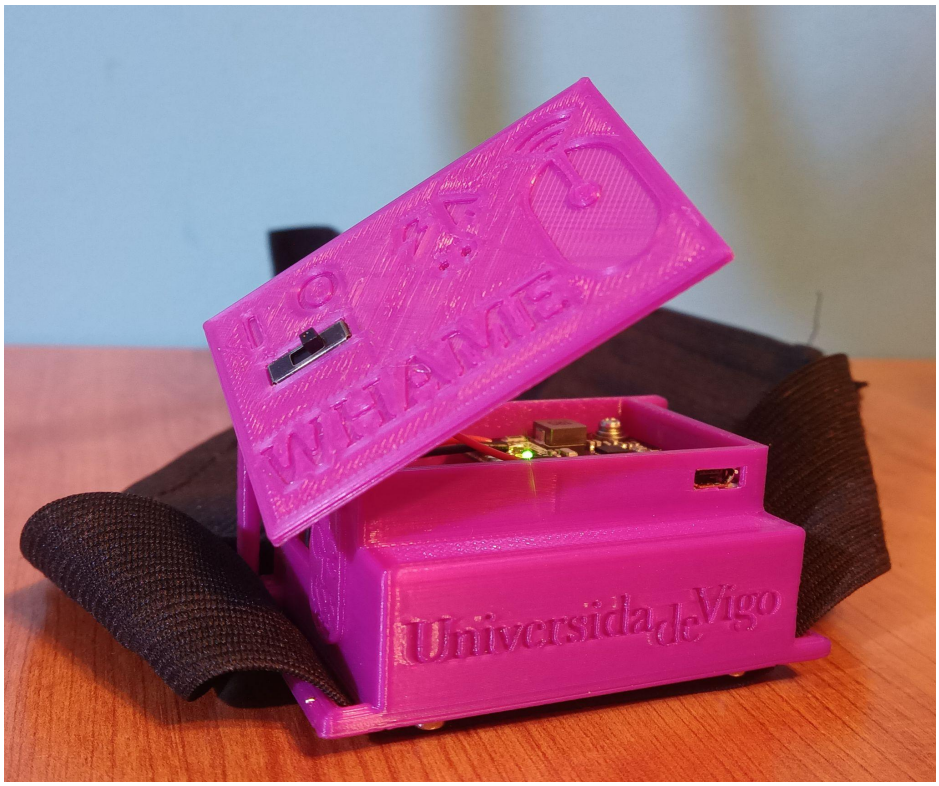
37. Connect the battery to the power booster, in case you desoldered the connector of the power booster as I did you need to cut, strip and solder the cables of the battery and solder them to the pins on the power booster. There are positive and negative voltage indications in the power booster, red always to V+ and black always to V- or GND.

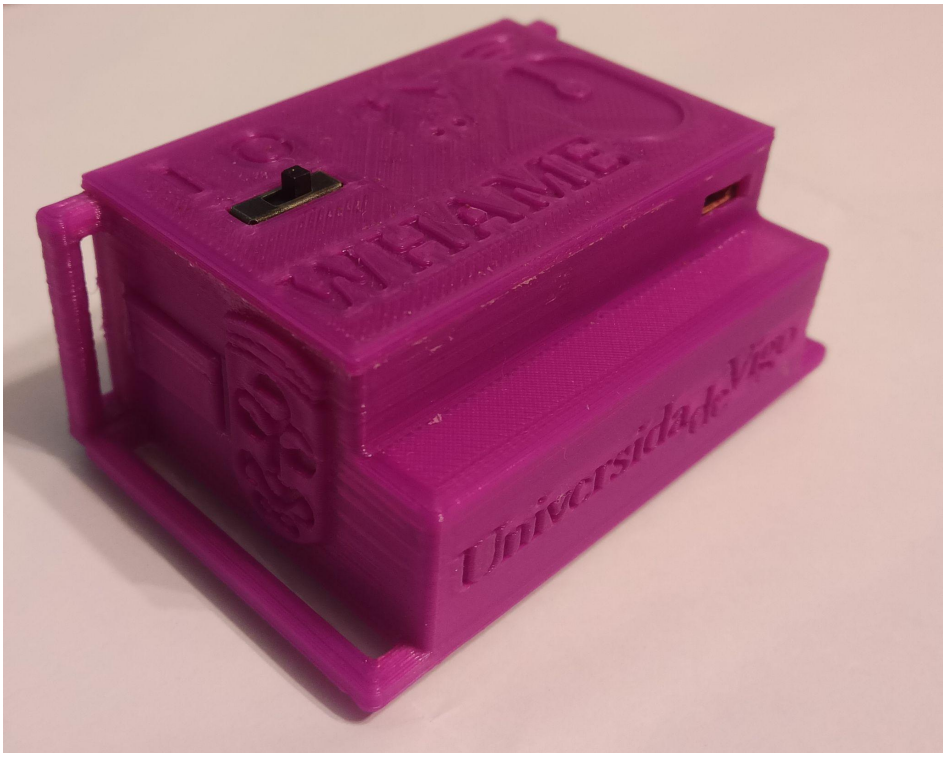



38. Now everything is fixed and connected, before closing the box we can check the system powers on by turning the switch. Arduino leds should power on, if not, review and repeat the process.



39. If the system is working we can close the box, glue the top cover using flexible plastic glue. Glues containing Acétone works well if the box is printed on PLA.



40. The system is ready to use, now you can choose where to put a ribbon depending on what part you are going to wear it, the box position must always be the same, the top cover must be oriented to the sky, the X and Y axis can be recalibrated and corrected via software, but not the Z axis, so that's why the box has two options to fix it.