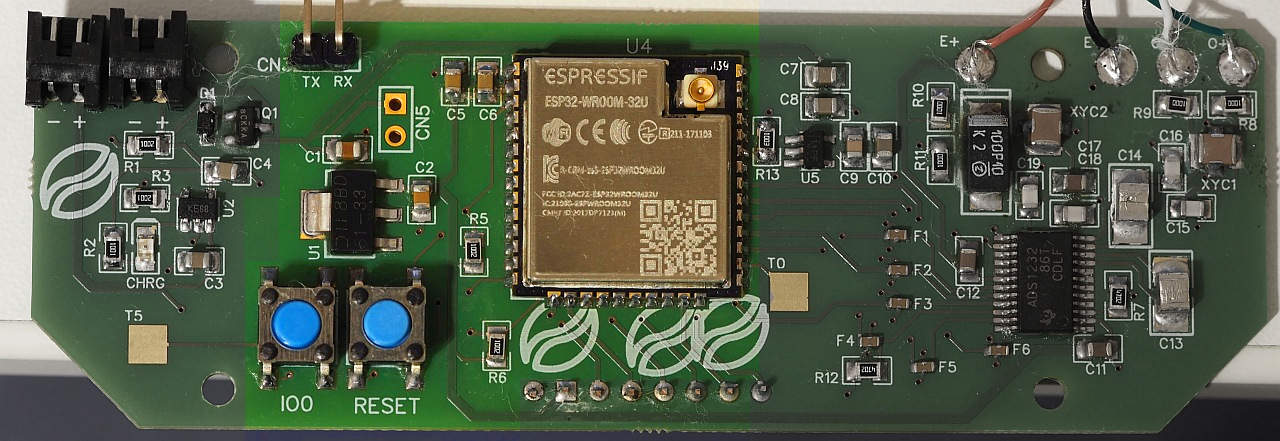
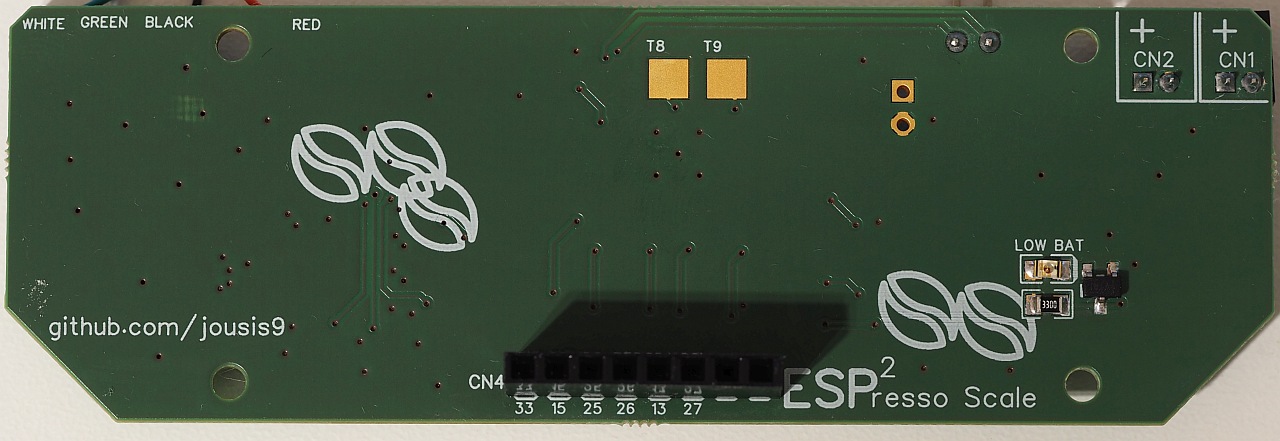
**ESP2resso Scale PRO**

**PCB Assembly Guide (v1)**

**v2 coming soon**





With a little care, you can easily hand solder all the components. No need for hot air.

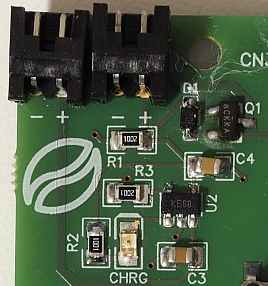
If you want to minimize the cost, all components except the ADC can be replaced with cheaper alternatives. Try to avoid very cheap components in the analog section (RC filtering) or you might lose some accuracy/stability.

Why “espresso” scale?  
High resolution, high refresh rate needed during an espresso extraction. Apart from that, it is a normal scale.

Why “PRO”?

You can use any SPI (or I2C) screen (even TFT with backlight pin), slightly better RC filtering, speed/gain control of the ADC from the software, better separation of analog/digital tracks, power on/off of the second LDO.

**Section 1 – USB / Battery Input & charging**

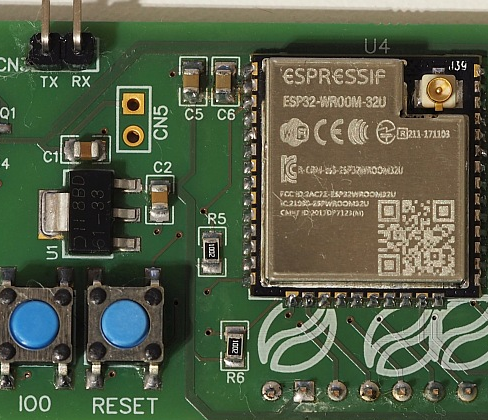


[Battery charging circuit design: Sparkfun ESP32 Thing](https://cdn.sparkfun.com/assets/learn_tutorials/5/0/7/esp32-thing-schematic.pdf)

If you don’t care about battery / battery charging you can connect DC+5V to CN5 and do not solder any of the parts shown above.

Please see the schematics for more info.

**Section 2 – LDO & MCU**



**LDO**

I have selected an 1A LDO with relatively low quiescent current, but you can replace it with an even better 500mA if you don’t intend to use Wifi. Remember that, when the esp is in deep sleep mode, the LDO will probably drain more power than the mcu.

Consider also changing package to SOT-23-5. In that case, consider 500mA TLV75533PDBVR.

CN5 is optional header, you can use it to give power to the LDO without having to connect the Section 1 part. Very handy for testing.

**MCU**

You can use any esp32 wroom with the same footprint (ESP32, ESP32D, ESP32U).

Why module (wroom) and not barebones chip?  
Easier to solder, readily available, reasonable size.

**Buttons**

Warning: these buttons are taller than all the other components. New revision has the same buttons as Lunar Eclipse version. Check BOM.

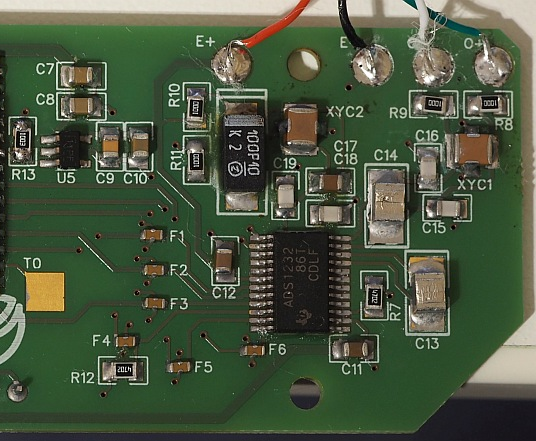
**Programming**

The only way to program your MCU for the 1st time is through the serial connector on the bottom. There is no ftdi chip on the PCB. Afterwards, if you wish, you can setup an OTA method.

**Touch Input**

The exposed 2 solder pads on the top/bottom are connected to touch pins T0,T5,T8,T9. Solder a small cable and terminate it in a copper tape behind the plastic case (or any other way you might think of).

**Section 3 – ADC / Load Cell**



**LDO**

U5 is the local LDO for the analog +3.3V voltage used for the load cell excitation. Use high quality components on everything related to the analog circuit.

The selected LDO has its enable pin connected to the MCU (IO21). Remember to select it as output and send HIGH when you wake up and LOW when going to sleep.

If you choose another LDO, make sure that the output voltage is very stable. Every miniscule variation can lead to false readings or excessive creeping. Also, the higher the voltage, the better the stability/accuracy. If you are designing your own high precision scale that does not need to be battery operated, use 5V.

**ADC**

Compared to the HX711 used in almost all the cheap (and not) designs, ADS1232 from TI is very accurate and stable. Although much more expensive, for this type of scale it is an expense you will have to make. Even when heating it up it could manage 0.1g accuracy (has internal temp sensor). Kudos to TI for their awesome documentation and white papers. (notice: This project is in no way affiliated with TI)

Digital +3.3V comes from the main LDO, analog +3.3V from the local LDO. There is only one ground for both signals (analog/digital) but I tried to avoid any digital signal return path through the analog part of the ADC. Unfortunately, due to the size constraints and the large number of tracks needed for the LED segment controller, digital and analog tracks are relatively close together. That said, I tested the same design with separate analog ground (join under the ADC) and couldn’t find measurable difference.

**Notice: ADS1232 does not communicate using standard SPI. Please see my library.**

**RC filtering**

If you have experience in RC filtering, please redo the calculations and find better values for R/C. I tried my best 😊  
Otherwise, please use high quality caps here or you will lose accuracy and possibly suffer from more creeping than you see on the demo videos.

**Why X2Y and not simple feed throughs (shown on ADS1232REF)?**

I wanted to test them, and I have found measurable difference in the +/-0.01 region which is of no importance to a amateur scale project like this. So, use feed through If you like or even omit them (you will lose some stability).

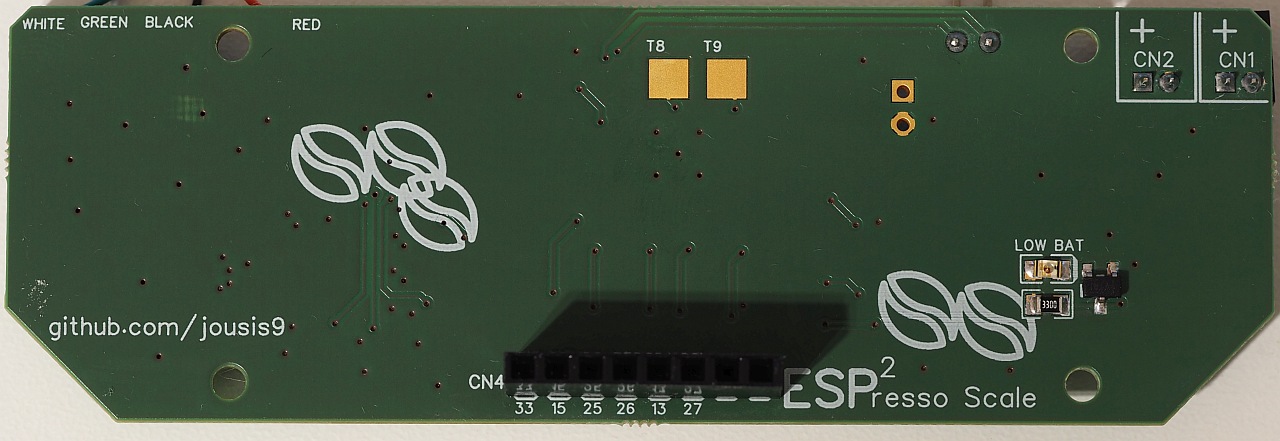
**Why those tiny feed throughs?**

I cannot measure/see any difference with/without them, but TI knows best (they have them on the reference design). Run a jumper cable across if you don’t have them or even edit the pcb and remove them completely (if you can’t ask me – I can make a new revision).

**Load cell pads**

Some cells have the white/green cables inverted. If you read negative values, swap red/white position.

**Bottom Layer**



**Warning: Silk screen has the wrong pin numbers. Please see below.**

U3 is a simple voltage supervisor circuit that will light LED2 when LDO output is <2.99 volts\*

CN4 is for OLED/TFT screens and I have labeled the pins for your convenience.  
As is, you can connect the usual Chinese OLED modules (be careful of VCC/GND pins) and some TFT modules.

**PINS**

From left to right as you see on the picture above:

1. **Screen Backlight control (TFT) – IO25**
2. **CS – IO15 (HSPI SS)**
3. **DC – IO26**
4. **RES – IO27**
5. **SDA – IO13 (HSPI MOSI)**
6. **SCL – IO14 (HSPI SCK)**
7. VCC
8. GND

Tip:

In order to mount your display as close to the PCB as possible without soldering it directly to it, use 90 degree headers in a “Z” configuration.

\*still testing various batteries and voltage supervisors, if you have any feedback, please share.