

DIY 60W PD Power Bank With Arduino Real Time Battery Info Display



by tobychui

Recently, I was planning to get a 60W PD power bank for my laptop so that I don't need to worry about my laptop running out of battery during lectures. However, all the 60W power bank on the market are bulk and expensive. That is why I decided to make my own.

While I was designing the power bank, I thought to myself, why can't I make one with a real time battery display? That is why I started this project: A 60W PD power bank with real time battery information display!

Supplies:

To make this power bank, you will need some general parts that you might have already lying around in your home workshop.

- 10K resistors x 2
- 10K variable resistors x 2
- 5.1V zener diode x 2
- Arduino nano
- prototyping boards
- 3D printer
- M2x5 screws x 4
- soldering tools
- buck converter (16.8V -> 5V)
- ACS712 (5A)
- power switch

However, you will also need to get some unique parts, which includes

- UART HMI display (The one I am using is 2.2 inch)
- 4S (14.8v) Lithium Ion Battery (I am using 4 x Panasonic 3200mAH cells)
- 60W IP2716 based power boost & charging module



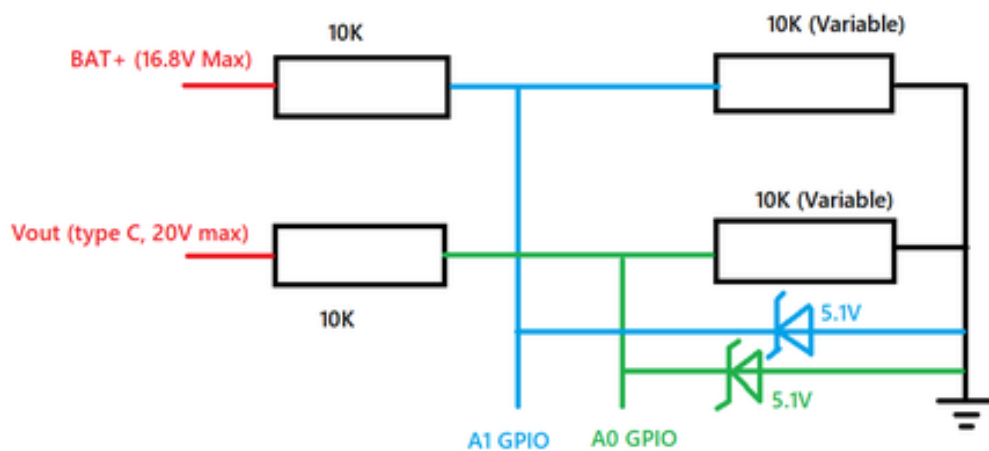
60W PD Power Bank w/ Info Display

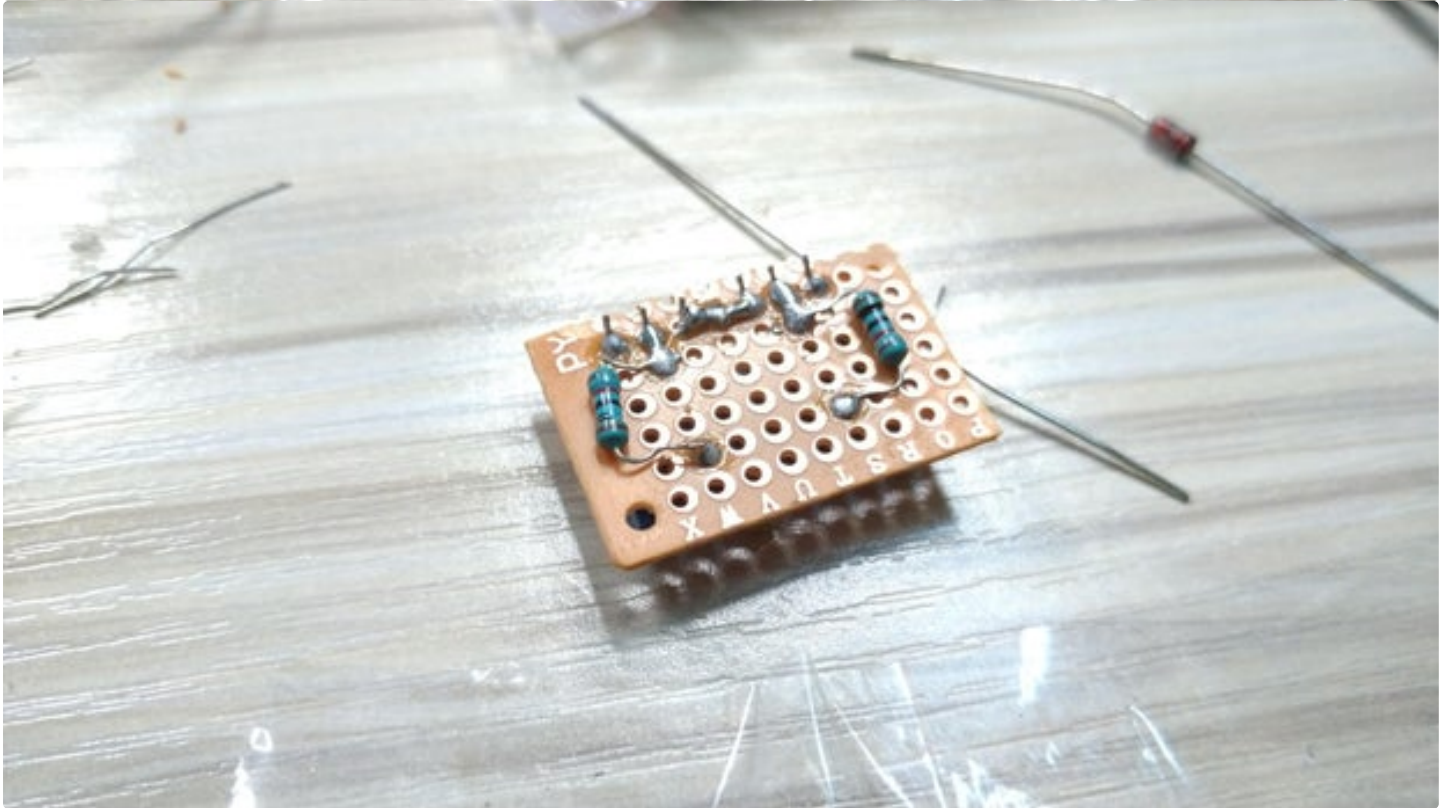
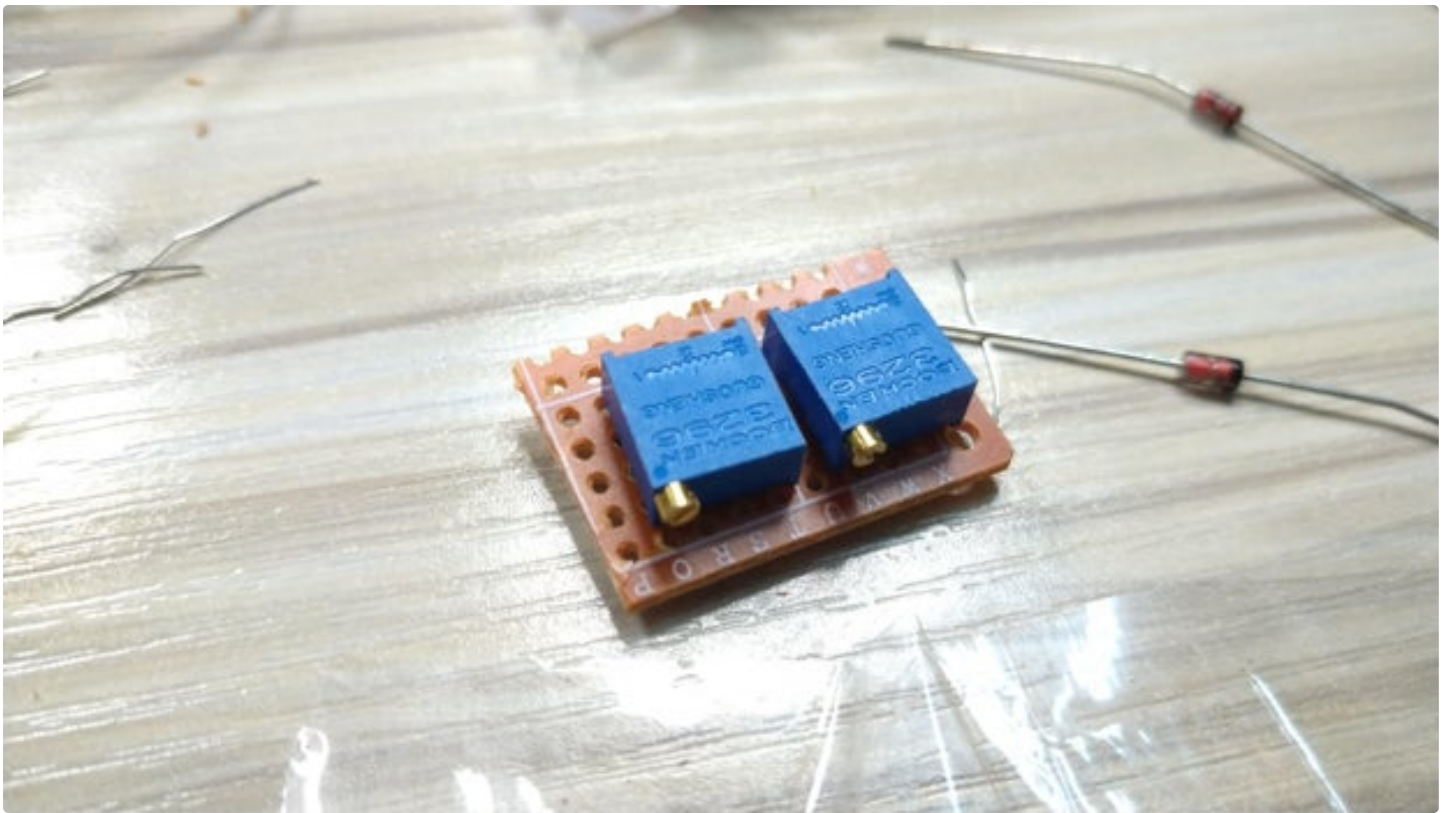


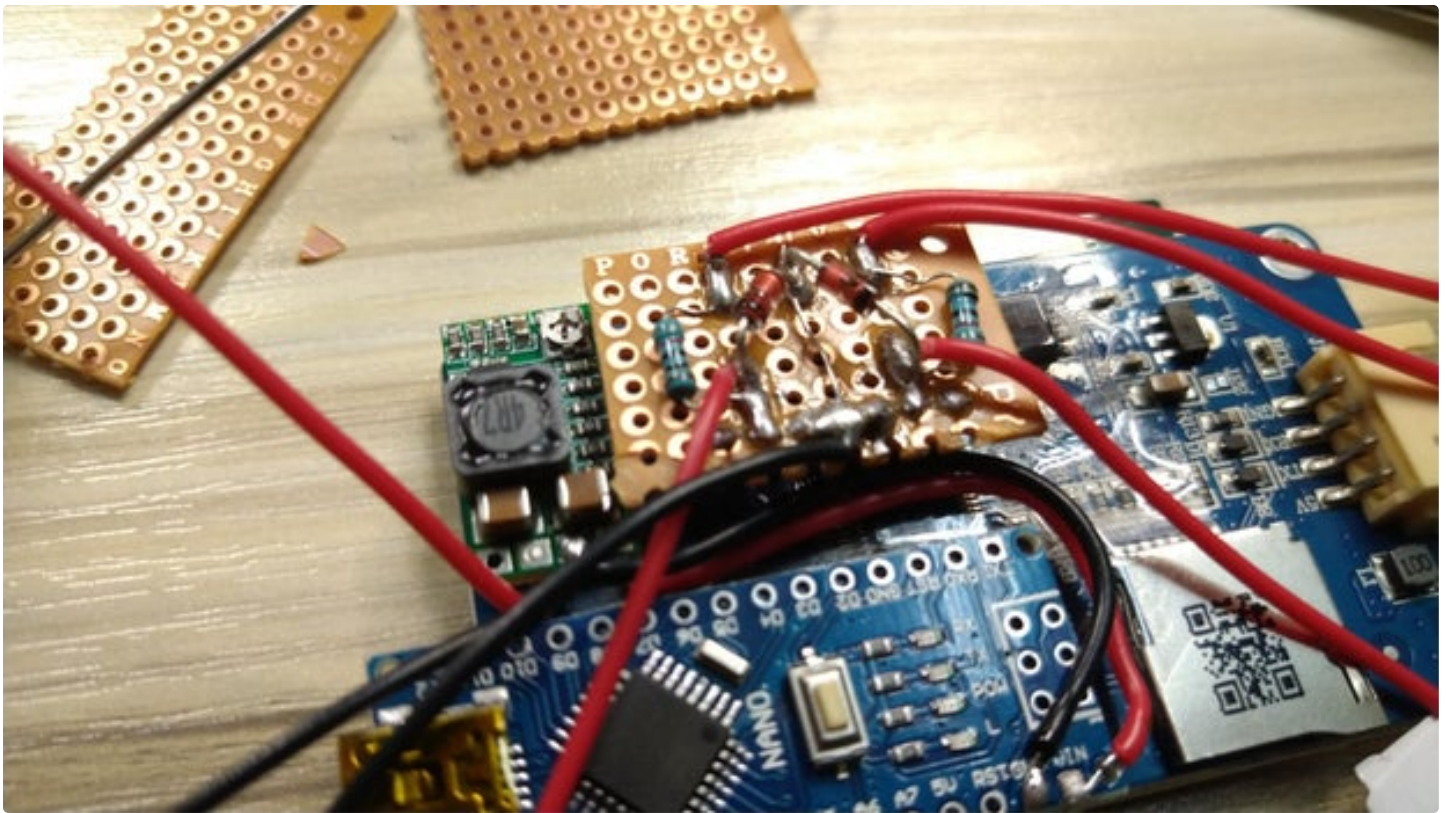


Step 1: Voltage Divider & Buck Converter

First, you need to make the voltage divider for the Arduino to read the voltage from the battery and the type C output port. The diagram is attached in the images above. Following the configuration in the photos, you can make it really small and it should be able to fit in a corner next to the green buck converter which convert the 16.8v battery voltage into 5V for the Arduino.

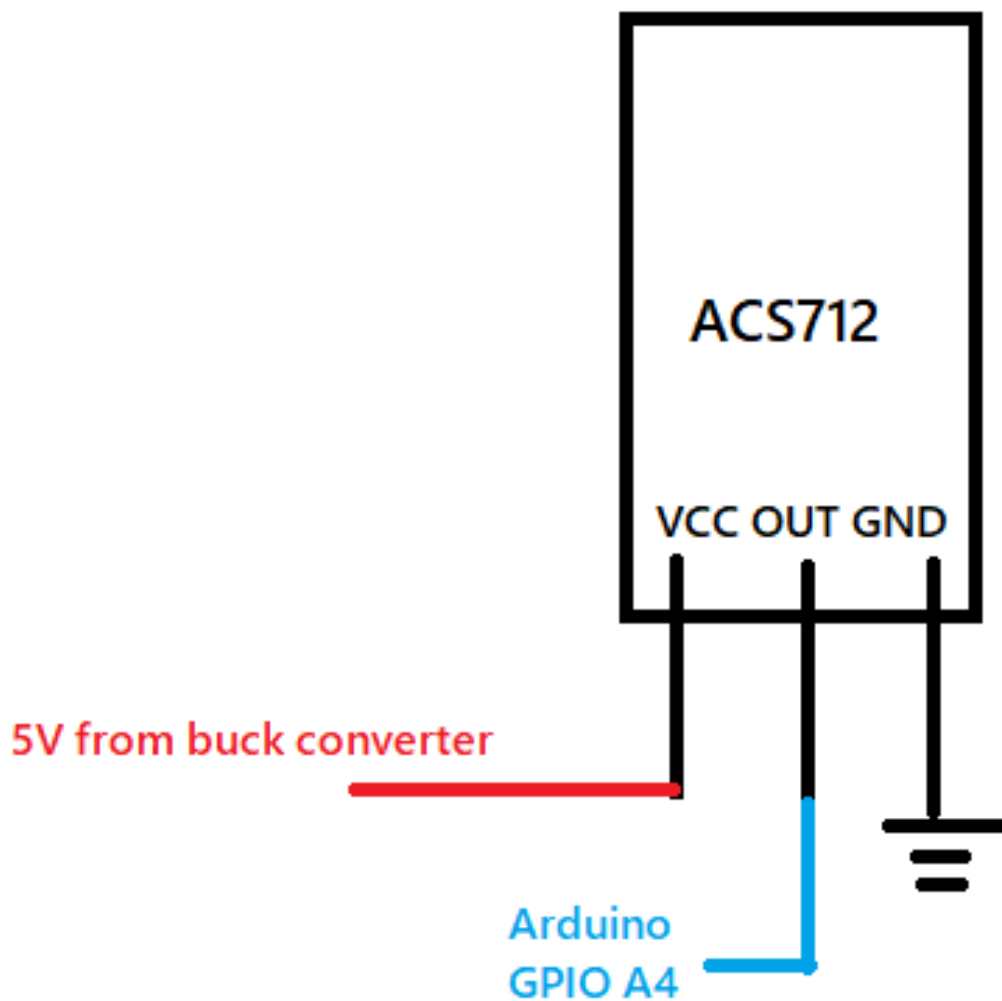






Step 2: ACS712

Next, connect the ACS712 sensor as described in the picture. Notes that you would want to get the **5A version of the sensor** instead of the 10A or 20A versions. After finish, glue this to the back of the HMI display

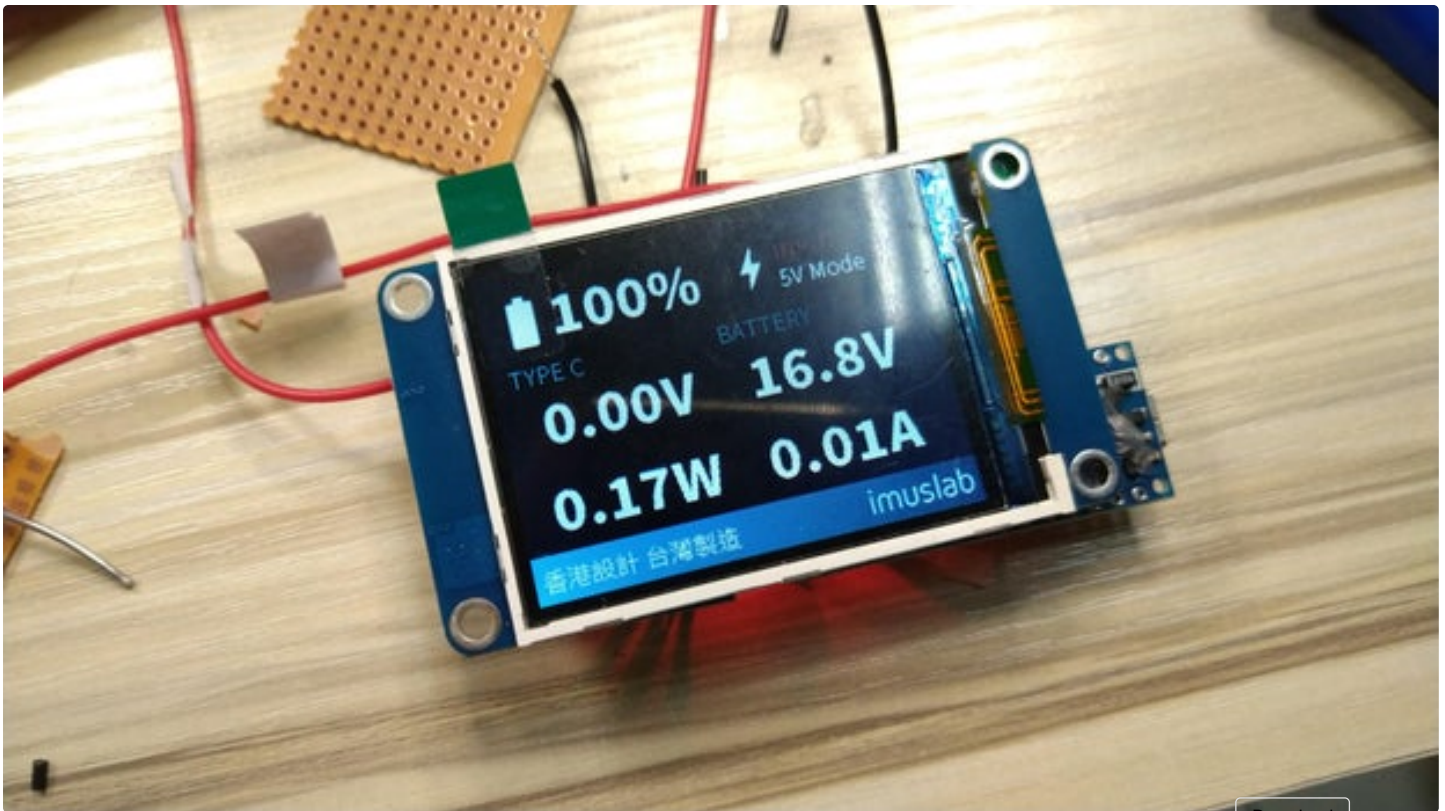


Step 3: Program the UART HMI Display

Then, you can download the UART HMI Studio tool from the HMI display supplier and program it. The one I am using was from a manufacturer named "Taojingchi" and they have offer me the interface editor where I can download from [here](#). But notes that your HMI displat might be from another manufacturer and they should have their own software for editing the UI.

After finished editing the code and uploaded the interface to the UART HMI display, unplug the display from your PC and connect it to the Arduino nano's pin 2 and 3 for its Software Serial.

You can find the zip file for the interface code attached here as a pdf file. Download the pdf file and rename its file extension to .zip, unzip it to get the source code for my interface.

[Download](#)

<https://www.instructables.com/ORIG/F6C/E9BJ/KZ19ZD9S/F6CE9BJKZ19ZD9S.pdf>

Step 4: Program the Arduino Nano

Next, edit the code based on your configurations and upload the following code to the Arduino. Fields that you might need to change includes

- `currentCalibrationOffset` (The offset current for your current sensor)
- `debugMode` (Set to true if you want to test the code without the HMI display)
- `HMISerial.begin(9600)` (Change this line if your HMI display doesn't runs on 9600 baudrate)

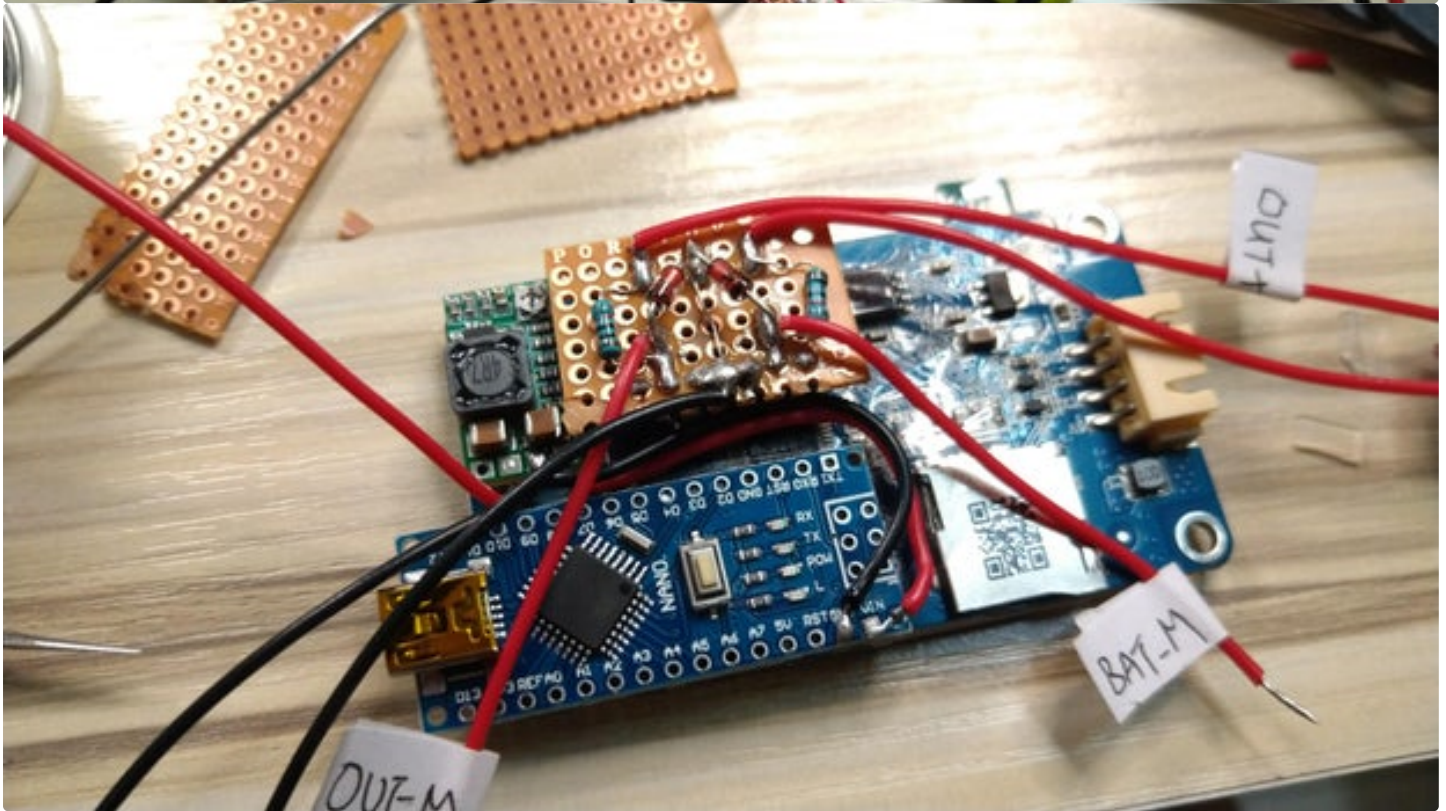
[Download](#)

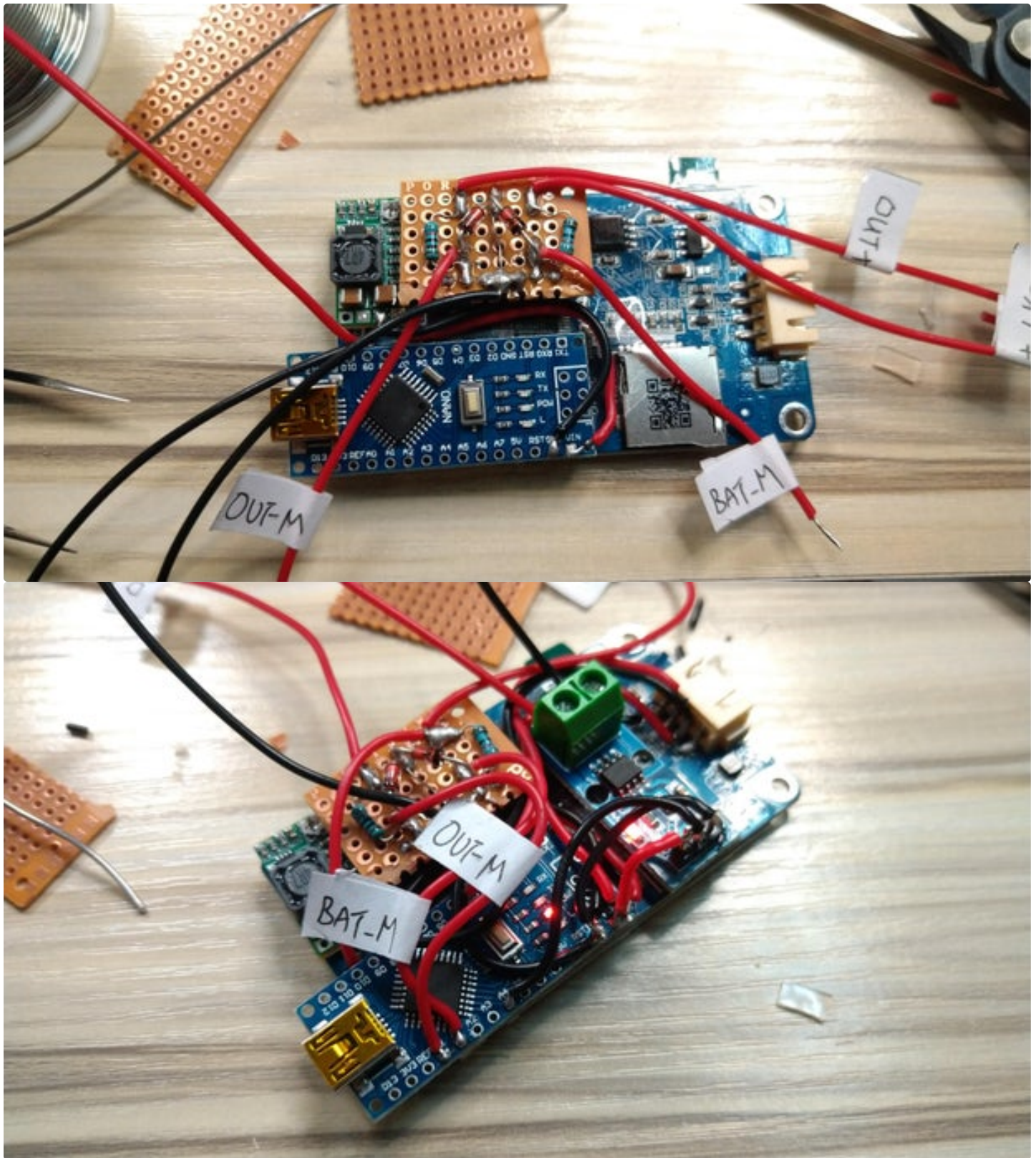
<https://www.instructables.com/ORIG/FJP/559O/KZ19ZGVX/FJP559OKZ19ZGVX.ino>

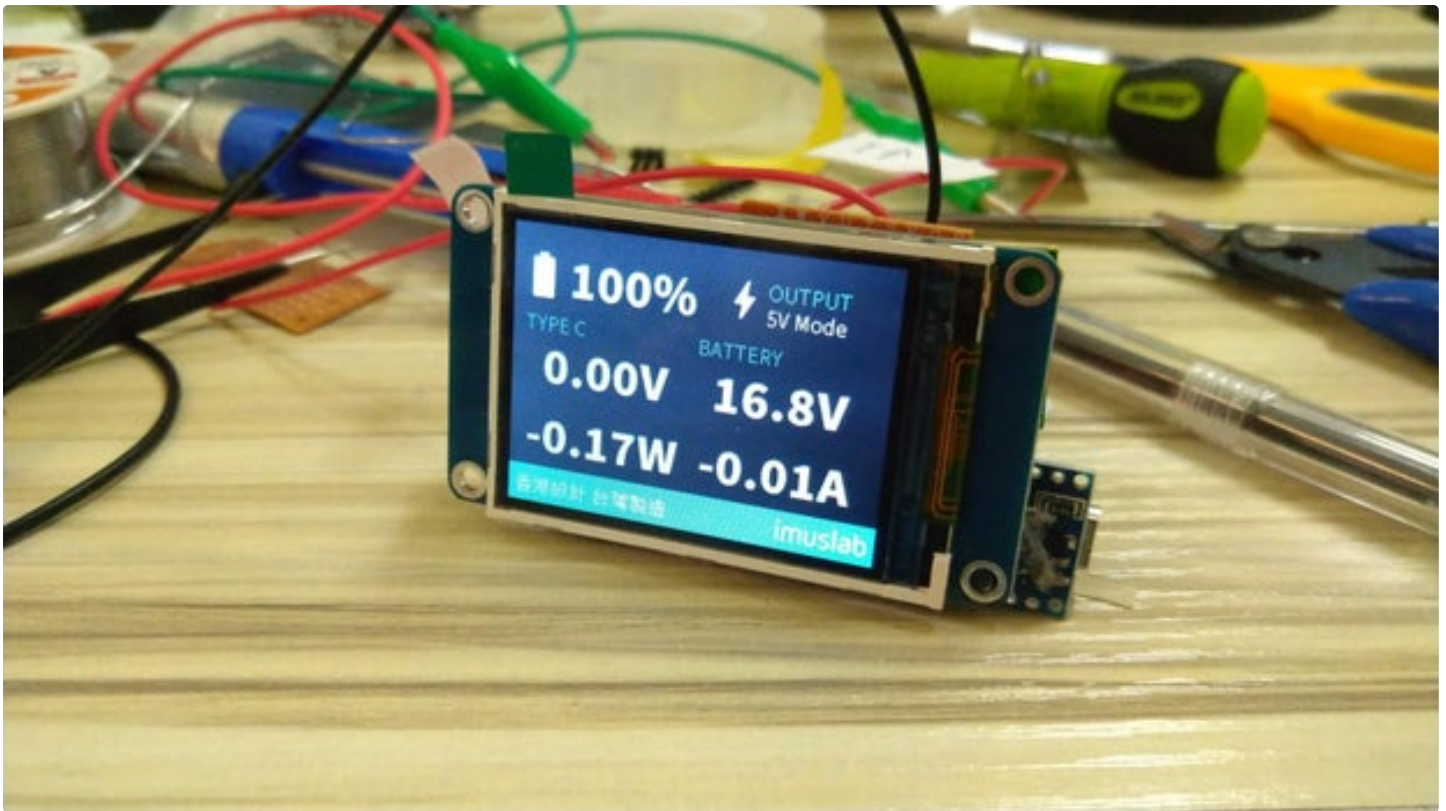
Step 5: Finalizing the Information Display

After all the wiring and program are uploaded, glue everything together and supply a voltage on 16.8V on the battery+ and a 20V on the out+ pins. You should be able to see the information displaying correctly. If not, try adjust the variable resistor so that the voltage on the screen matches your lab bench power supply output. For the current testing, you can observe the difference between your power supply and the displayed information and adjust the offset value in Arduino code manually.

Notes: You should get the power for the buck converter from the BAT+ wire instead of the current sensing wire that runs through the ACS712.

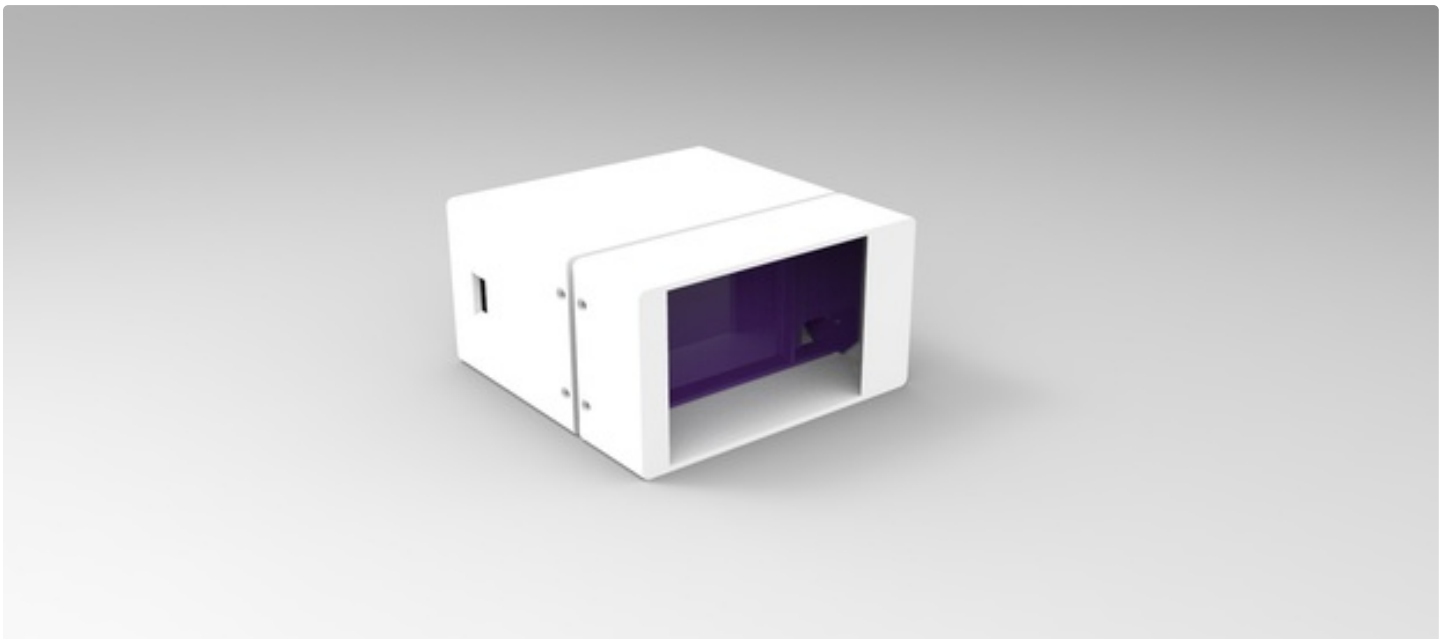


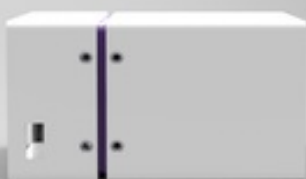
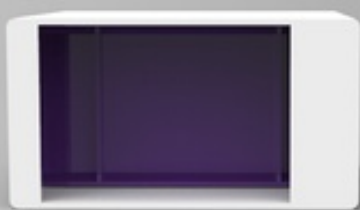


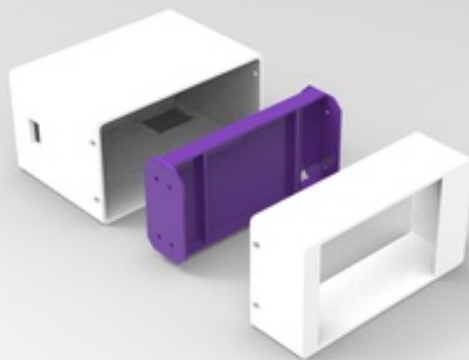


Step 6: 3D Printing Case

Then, you can start 3D printing the case. You can print all of them in single color but don't you think adding a purple layer in between makes it feel really amazing?







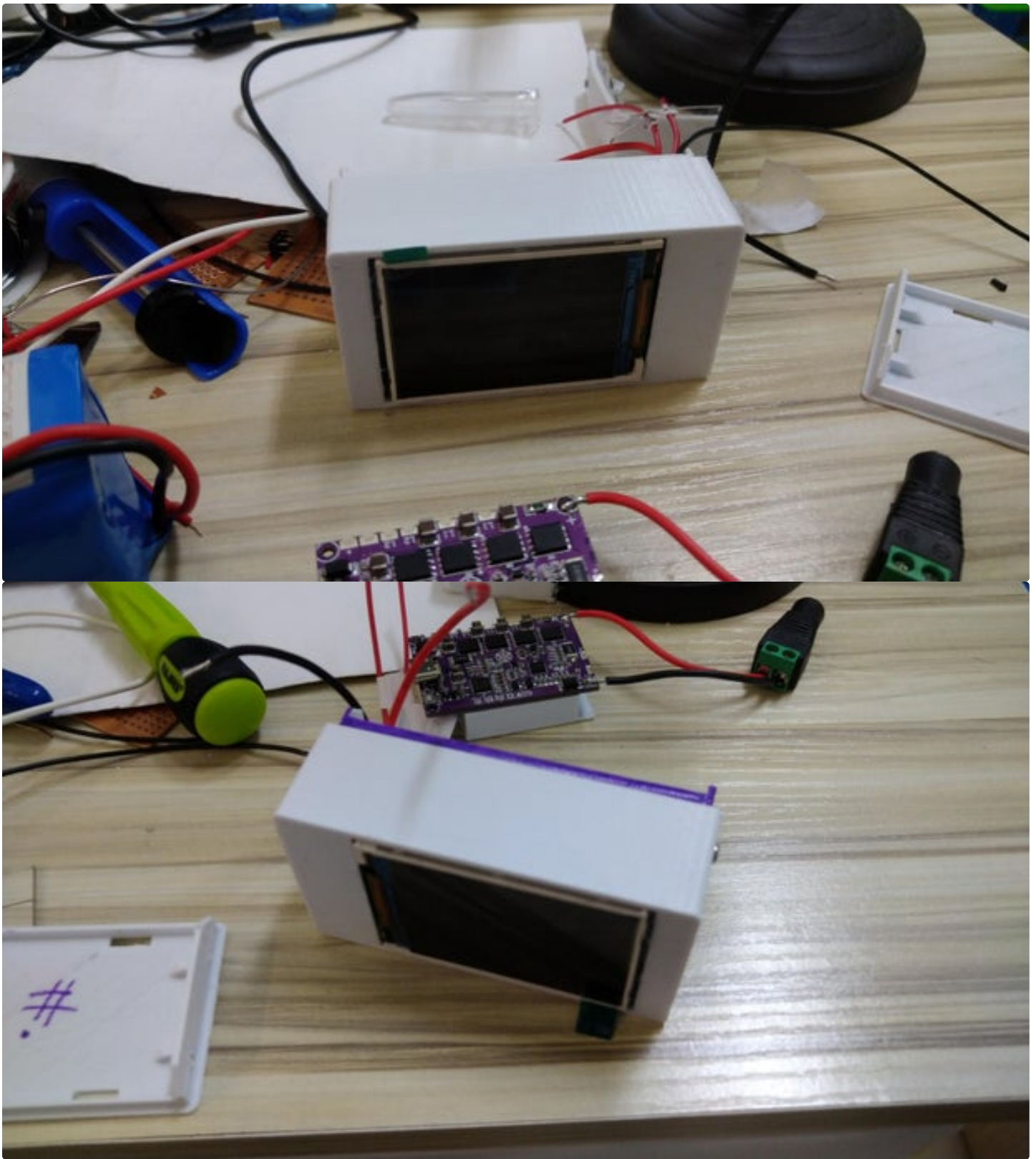
	https://www.instructables.com/ORIG/FHK/Z350/KZ19ZMOV/FHKZ350KZ19ZMOV.ipt	View in 3D	Download
	https://www.instructables.com/ORIG/FN7/W6F4/KZ19ZMOW/FN7W6F4KZ19ZMOW.stl	View in 3D	Download
	https://www.instructables.com/ORIG/FA3/TAAH/KZ19ZMOX/FA3TAAHKZ19ZMOX.ipt	View in 3D	Download
	https://www.instructables.com/ORIG/FTR/A6ND/KZ19ZMOY/FTRA6NDKZ19ZMOY.stl	View in 3D	Download
	https://www.instructables.com/ORIG/FAF/NID5/KZ19ZMOZ/FAFNID5KZ19ZMOZ.ipt	View in 3D	Download
	https://www.instructables.com/ORIG/FA1/O08Y/KZ19ZMP0/FA1O08YKZ19ZMP0.stl	View in 3D	Download
	https://www.instructables.com/ORIG/FEW/HWY0/KZ19ZWWE/FEWHWY0KZ19ZWWE.svg		Download

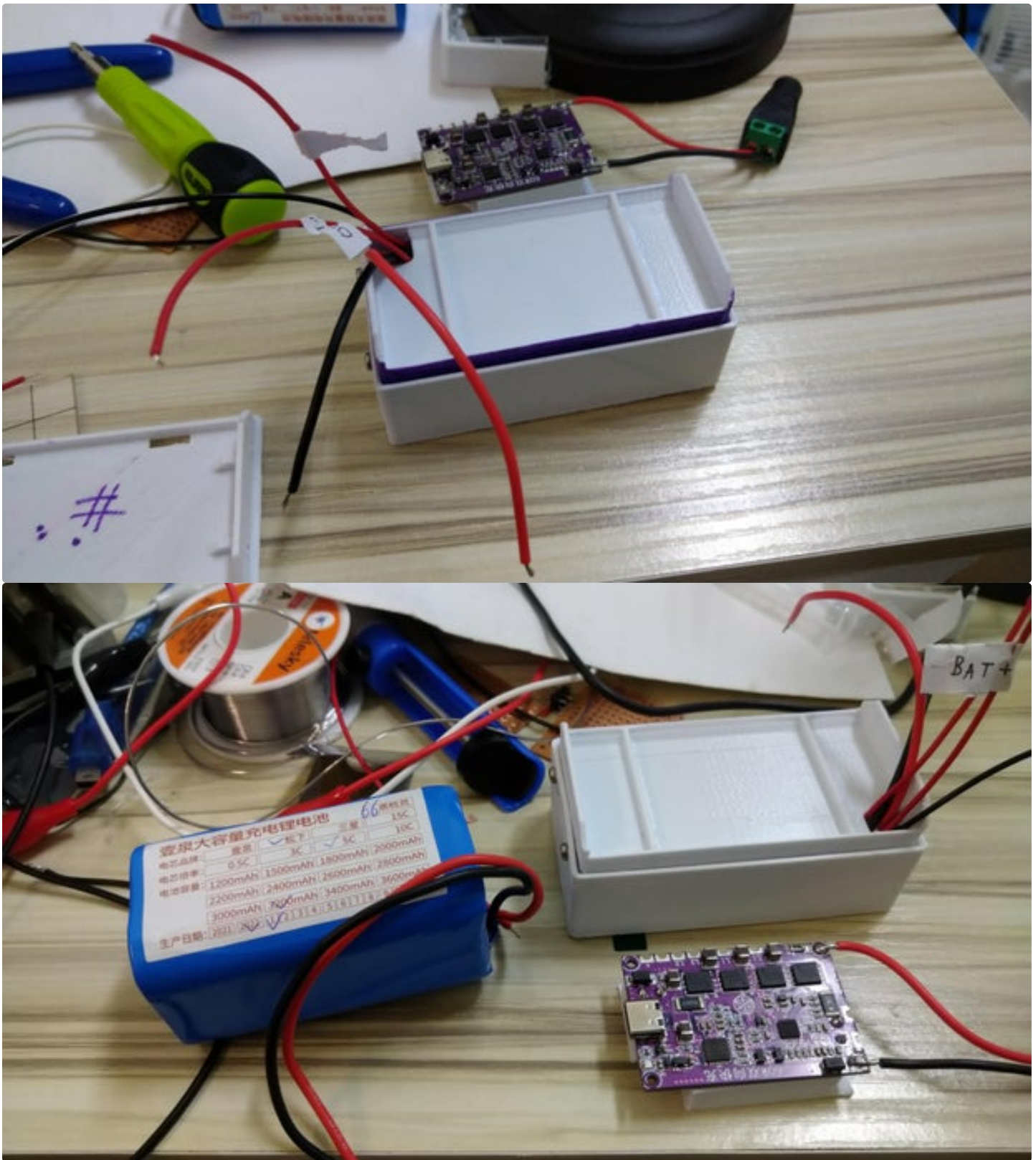
Step 7: Seal Up the Display & Arduino

After the printing is done, you can use the screen cover and the black plate to seal up the screen unit with the Arduino and current sensor.

After this stage, you should only got a few wires exposed, they should be

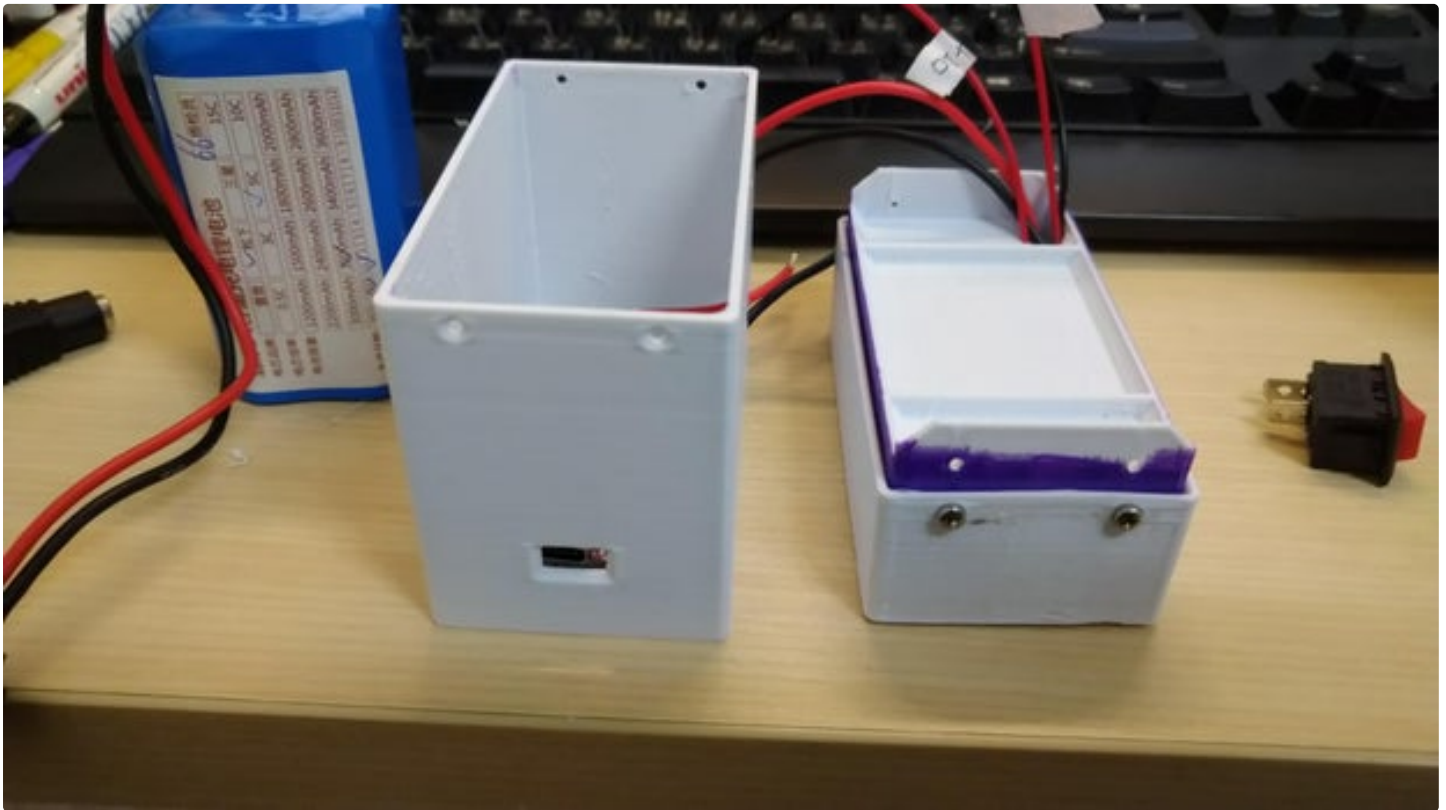
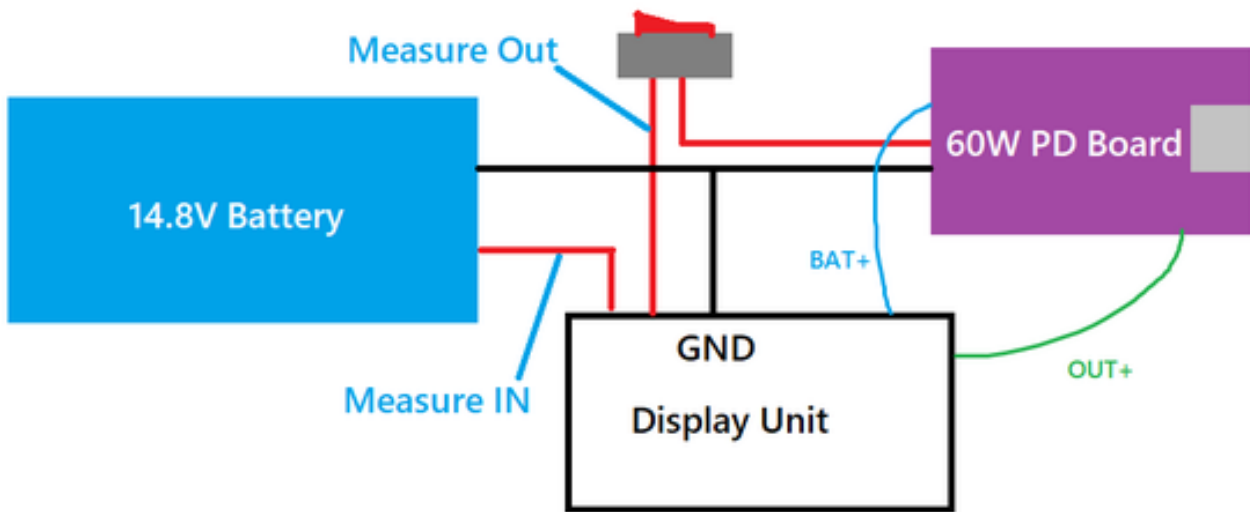
- BAT+ (Battery + terminal, also use as powering the buck converter)
- OUT+ (Type C output port)
- GND
- Current Measure Input
- Current Measure Output

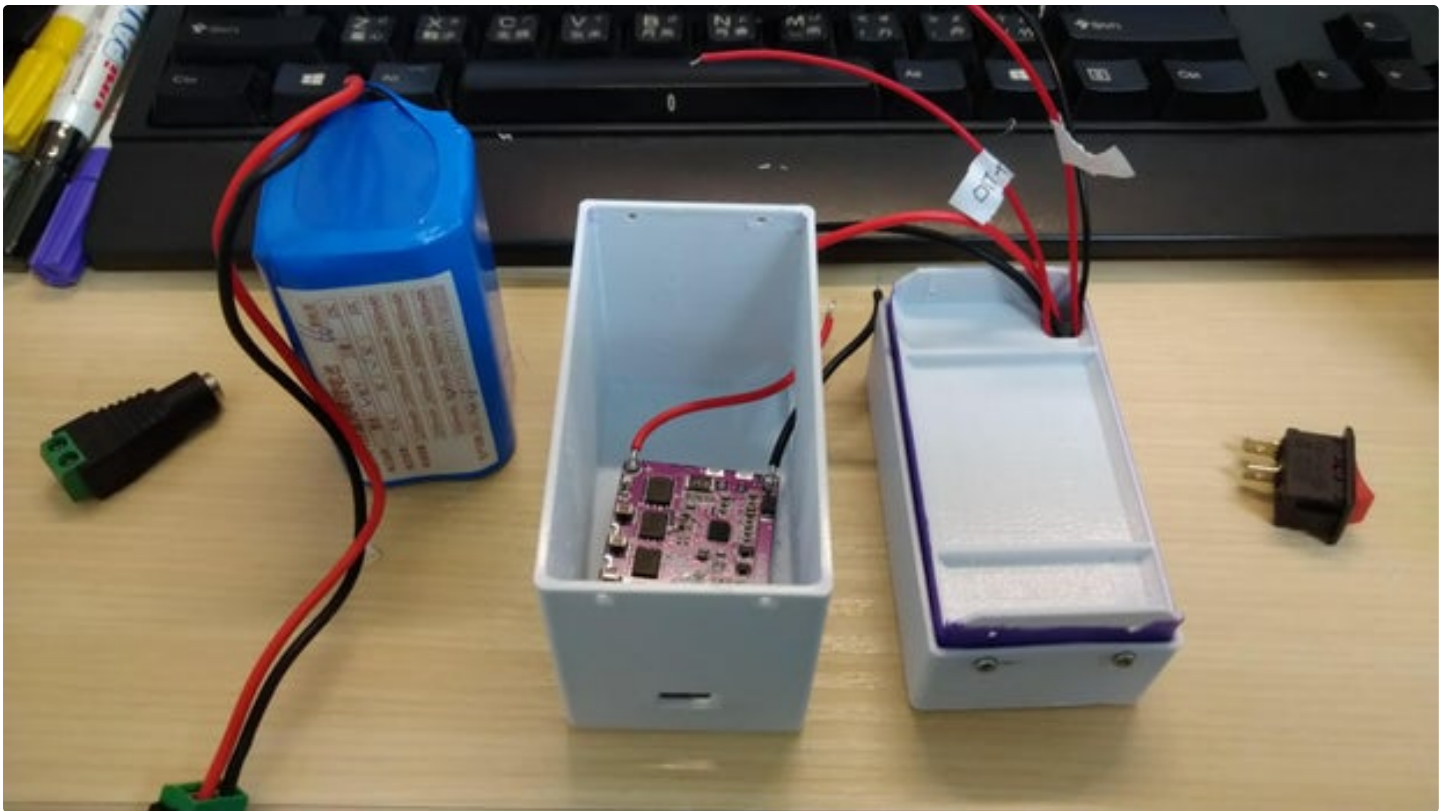




Step 8: Connect the Battery and PD Power Board

After that, connect the battery to the power switch, PD power board and the information display as follows.





Step 9: Done!

Lastly, seal it up and turn it on, your unique 60W PD power bank is now ready to use!





