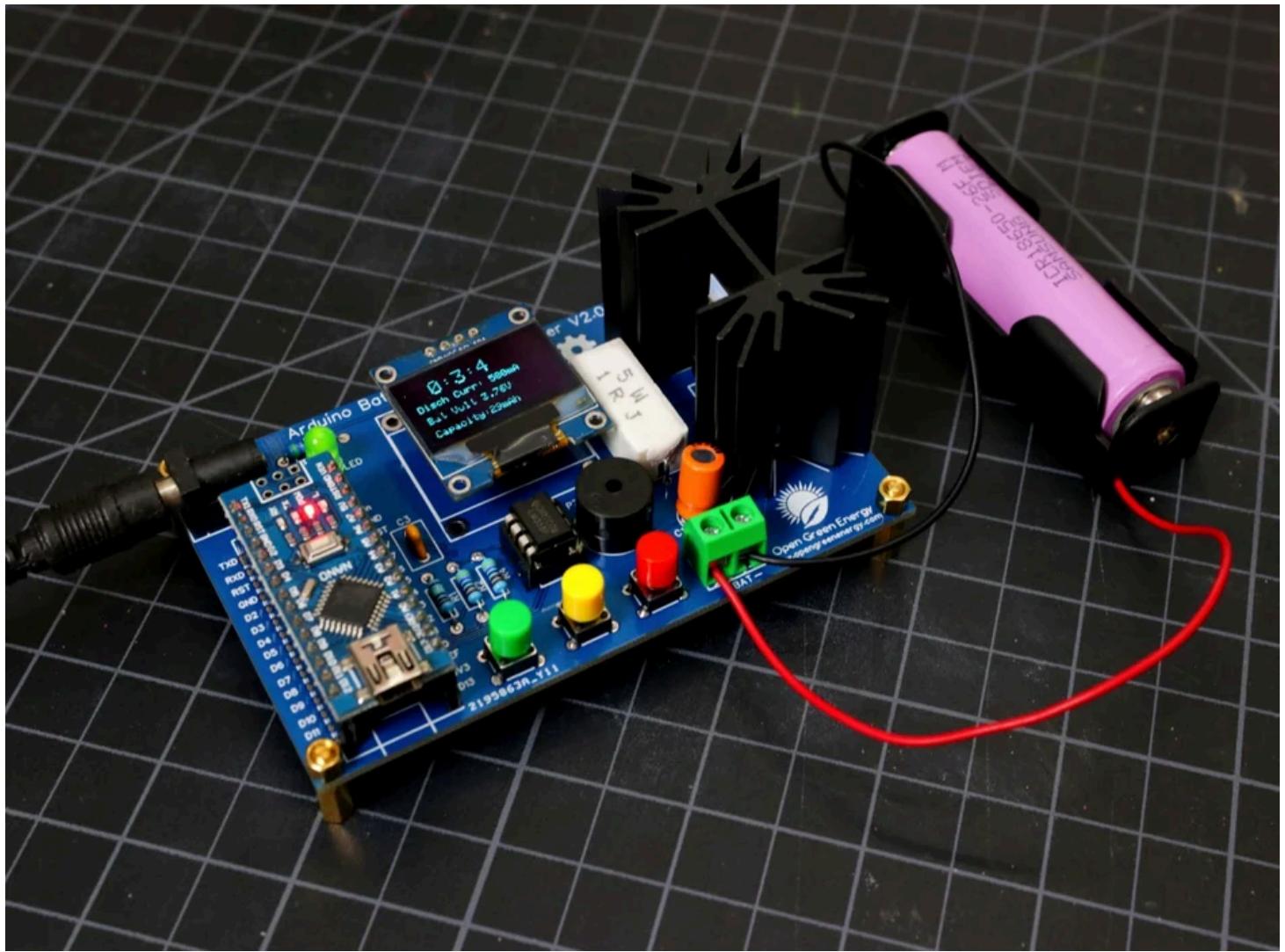


DIY Arduino Battery Capacity Tester - V2.0

By [opengreenenergy](#) in [CircuitsArduino](#)



Introduction: DIY Arduino Battery Capacity Tester - V2.0



Nowadays fake Lithium and NiMH batteries are everywhere and are sold by advertising with higher capacities than their true capacity. So it is really difficult to distinguish between a real and a fake battery. Similarly, it is difficult to know the capacity retained in the salvaged 18650 laptop batteries. So, a device is required to measure the true capacity of the batteries.

In the year 2016, I have written an Instructable on "[Arduino Capacity Tester - V1.0](#)" which was a very straightforward and simple device. The earlier version was based on Ohms Law. The battery to be tested is discharged through a fixed resistor, current and time duration is measured by Arduino, and capacity is calculated by multiplying both the readings (Discharge current and time).

The drawback of the earlier version was that during the testing, the battery voltage decreased, and the current also decreases which make the calculations complex and inaccurate. To overcome this, I have made the V2.0 which is designed in such a way that the current will remain constant throughout the discharging process. I made this device by inspiring the original design by [MyVanitar](#)

The main features of Capacity Tester V2.0 are :

1. Capable of measuring the capacity of AA / AAA NiMh / NiCd, 18650 Li-ion, Li-Polymer, and Li FePO4 battery. It is suitable for almost any kind of battery rated below 5V.
2. Users can set the discharge current by using pushbuttons.
3. OLED user Interface
4. The device can be used as an Electronic Load

Update on 14.10.2024

You can check out the updated version [here](#)

Update on 10.08.2022

You can order the fully assembled board (V2.1) from my [Tindie Store](#).

Support me On Patreon:

If you enjoy my work here on Instructables, consider joining my [Patreon](#), it will be a great help for me to make more interesting projects in the future.

Patreon Link: <https://www.patreon.com/opengreenenergy>

Update on 02.12.2019

Now you can order the PCB and components together in a kit from [PCBWay](#)

Disclaimer: Please note that you are working with Li-Ion battery which is highly explosive and dangerous. I cannot be held responsible for any loss of property, damage, or loss of life if it comes to that. This tutorial was written for those who have knowledge of rechargeable lithium-ion technology. Please do not attempt this if you are a novice. Stay Safe.

Supplies

Components Used

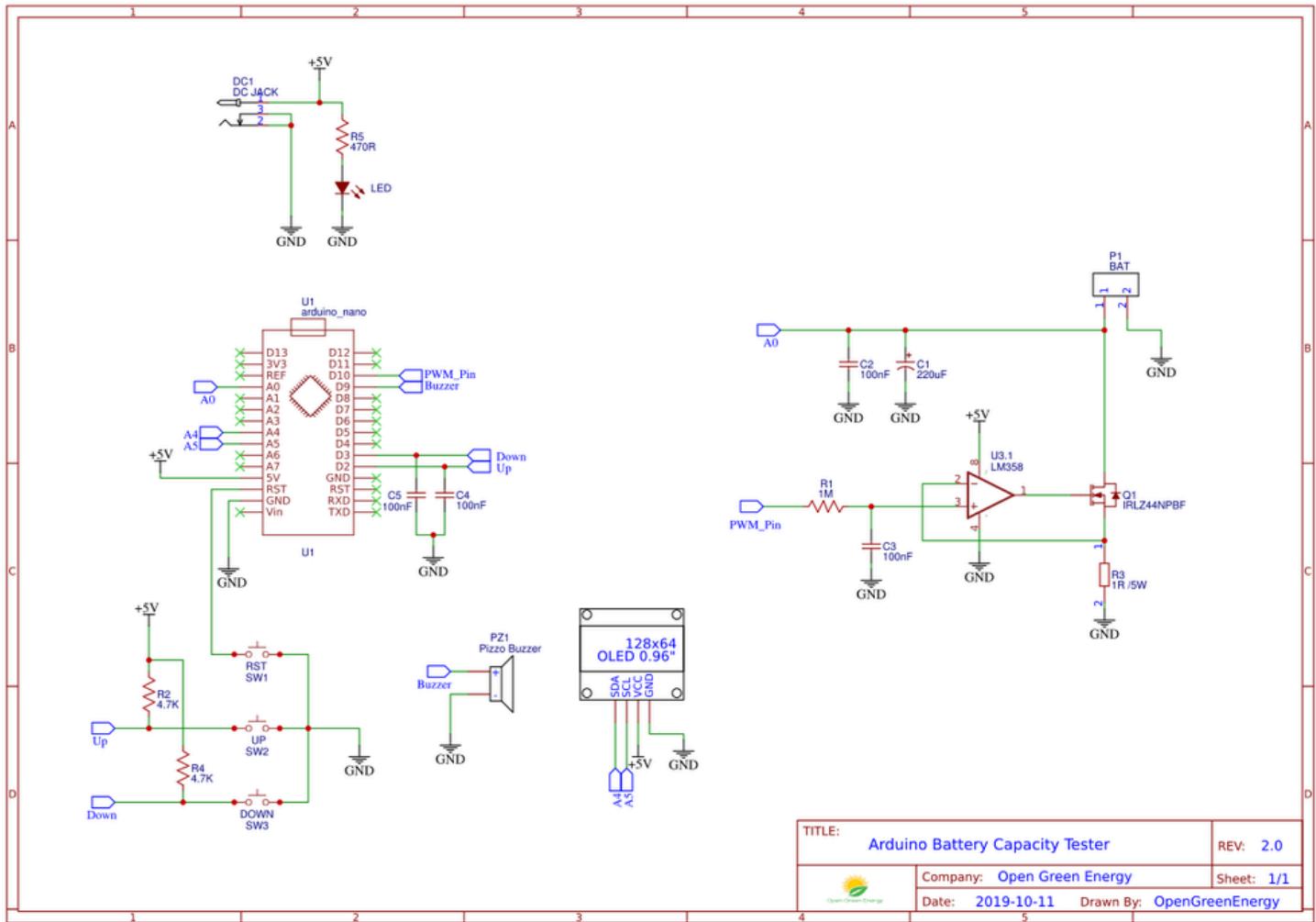
Now Order PCB and all the components to build this project in a kit from [PCBWay](#)

1. PCB: [PCBWay](#)
2. Arduino Nano: [Amazon](#) / [Banggood](#)
3. Opamp LM358: [Amazon](#) / [Banggood](#)
4. Voltage Reference (LM385BZ-1.2): ([Amazon](#))
5. 0.96" OLED display : [Amazon](#) / [Banggood](#)
6. Ceramic Resistor: [Amazon](#) / [Banggood](#)
7. Capacitor 100nF : [Amazon](#) / [Banggood](#)
8. Capacitor 220uF: [Amazon](#) / [Banggood](#)
9. Resistors 4.7K & 1M : [Amazon](#) / [Banggood](#)
10. Push Button: [Amazon](#) / [Banggood](#)
11. Push-Buttons Cap: [Aliexpress](#)
12. Screw Terminal: [Amazon](#) / [Banggood](#)
13. Prototype Board: [Amazon](#) / [Banggood](#)
14. PCB Stand-off: [Amazon](#) / [Banggood](#)
15. Heatshrink Tubing: [Amazon](#) / [Banggood](#)
16. Heatsink: [Aliexpress](#)

Tools Used

1. Soldering Iron: [Amazon](#) / [Banggood](#)
2. Clamp Meter: [Amazon](#) / [Banggood](#)
3. Multimeter: [Amazon](#) / [Banggood](#)
4. Hot Air Blower: [Amazon](#) / [Banggood](#)
5. Wire Cutter: [Amazon](#) / [Banggood](#)
6. Wire Stripper: [Amazon](#) / [Banggood](#)

Step 1: Schematic Diagram



The whole schematic is divided into to followings sections:

1. Power Supply Circuit
2. Constant Current Load Circuit
3. Battery Voltage Measurement Circuit
4. User Interface Circuit
5. Buzzer Circuit

1. Power Supply Circuit

The power supply circuit consists of a DC Jack (7-9V) and two filter capacitors C1 and C2. The power output (Vin) is connected to the Arduino pin Vin. Here I am using the Arduino on-board voltage regulator to step down the voltage to 5V.

2. Constant Current Load Circuit

The main component of this project is **LM358 dual amplifier**. LM358 is a dual amplifier in a single package. However, in this project, only one op-amp is required. The PWM output from Arduino **Pin D10** is connected to the first operational amplifier's non-inverted pin. The PWM is filtered through a low pass filter (

R1 and C3) to get an equivalent analog voltage. Whereas, the Inverting input is acting as feedback for the operational amplifier which is connected between the IRLZ44N MOSFET source pin and the 1R shunt resistor (R3).

This op-amp, **R3**, and the MOSFET build a constant current load circuit. When we set the voltage on the non-inverting input pin, the op-amp turns on the MOSFET and tries to get the same voltage across the R3. The Voltage appears due to the ohms law, as current flows through the resistor a voltage drop must appear. So now we can control the current through the load resistor (R4) by changing the PWM signal pulse width.

3. Battery Voltage Measurement Circuit

The battery voltage is measured by the Arduino analog input pin A0. Two capacitors C1 and C2 are used to filter out the noises coming from the constant current load circuit which can degrade the ADC conversion performance.

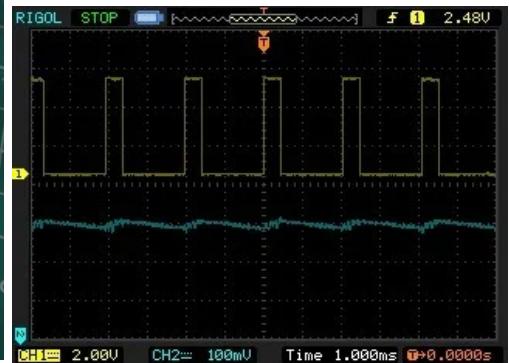
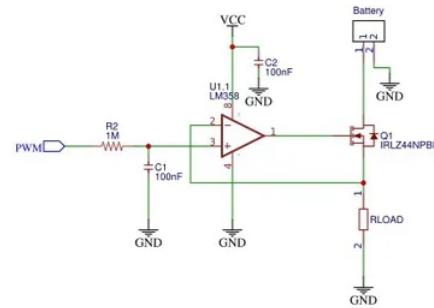
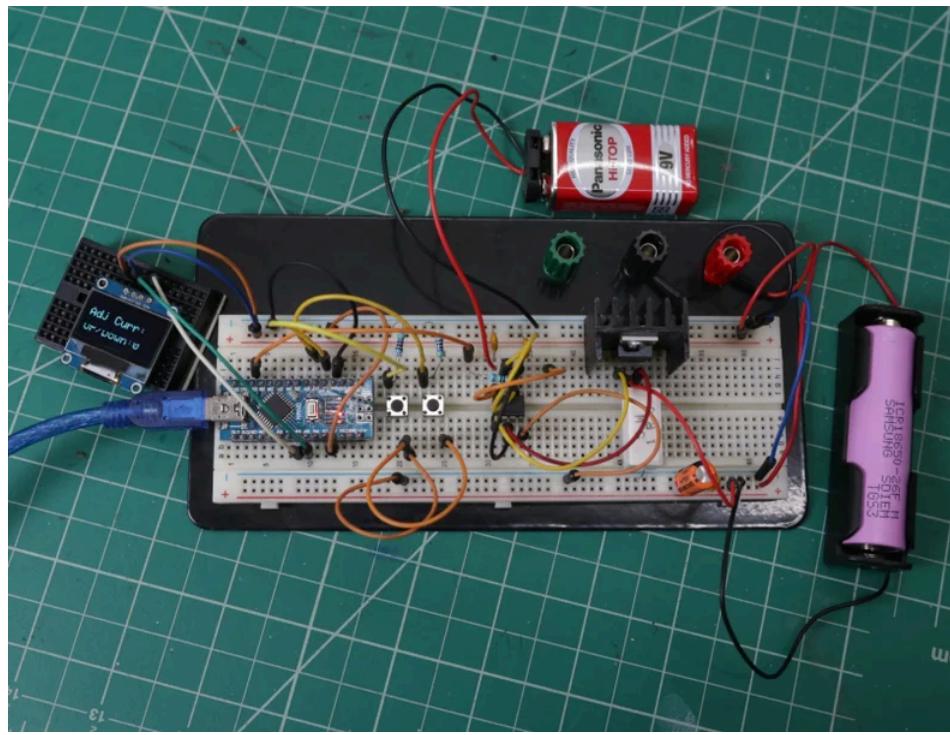
4. User Interface Circuit

The user interface circuit consists of two push buttons and a 0.96" I2C OLED display. The Up and Down push-button is to increase or decrease the PWM pulse width. R3 and R4 are pull-up resistors for the Up and Down push-buttons. C7 and C8 are used to debounce the push buttons. The third push-button (RST) is used for resetting the Arduino.

5. Buzzer Circuit

The buzzer circuit is used to alert the starting and end of the test. A 5V buzzer is hooked to Arduino digital pin D9.

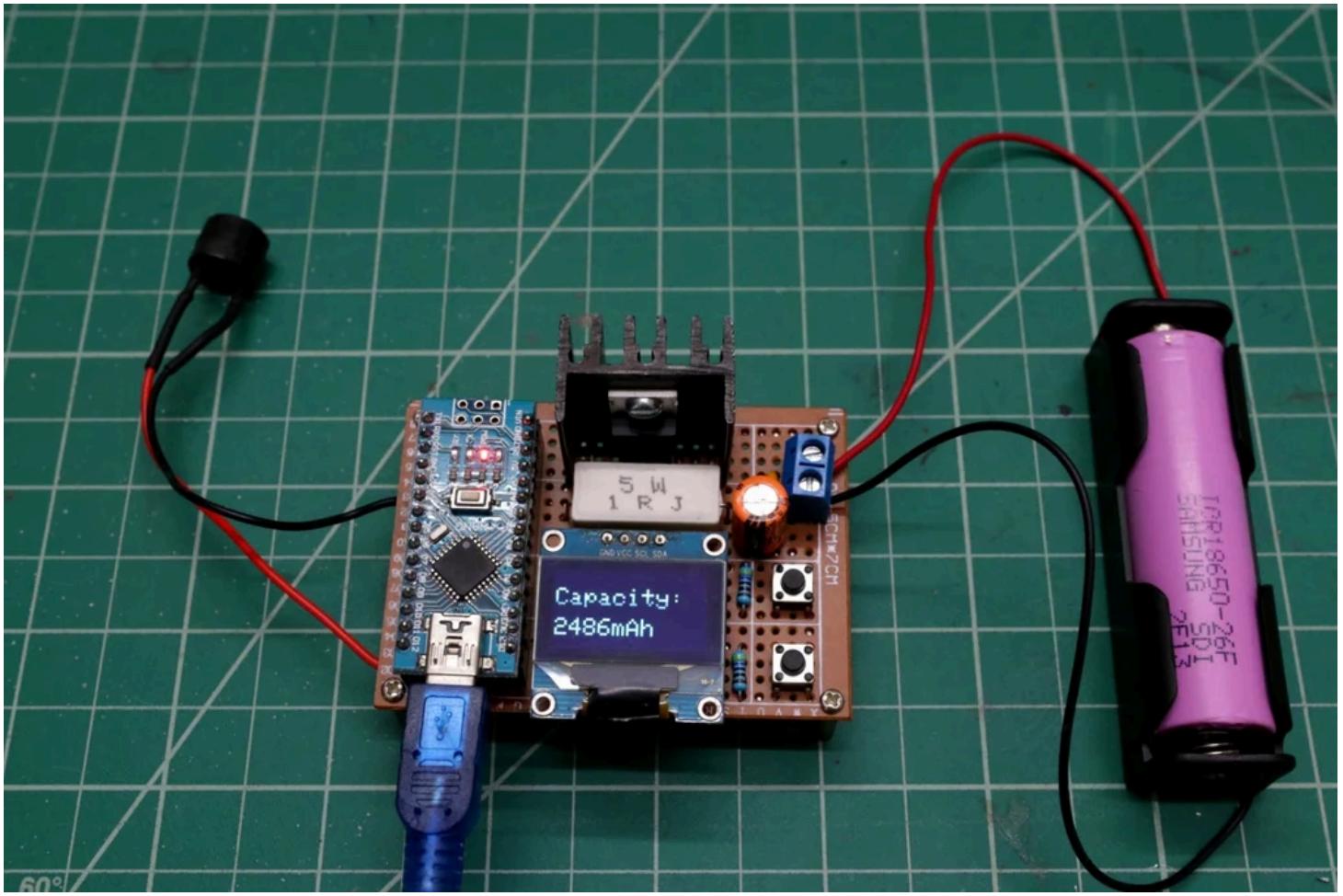
Step 2: How Does It Work?



The theory is based on the voltage comparison of the inverting (pin-2) and the non-inverting (pin-3) inputs of the OpAmp, configured as a unity amplifier. When you set the voltage applied to the non-inverting input by adjusting the PWM signal, the output of the opamp opens the gate of MOSFET. As the MOSFET turn-on, the current runs through R1, it creates a voltage drop, which provides negative feedback to OpAmp. It controls the MOSFET in such a way that the voltages at its inverting and non-inverting inputs are equal. So, the current through the load resistor is proportional to the voltage at the non-inverting input of the OpAmp.

The PWM signal from the Arduino is filtered by using a low pass filter circuit (R2 and C1). To test the PWM signal and filter circuit performance, I hooked up my DSO ch-1 at the input and ch-2 at the output of the filter circuit. The output waveform is shown above.

Step 3: Capacity Measurement



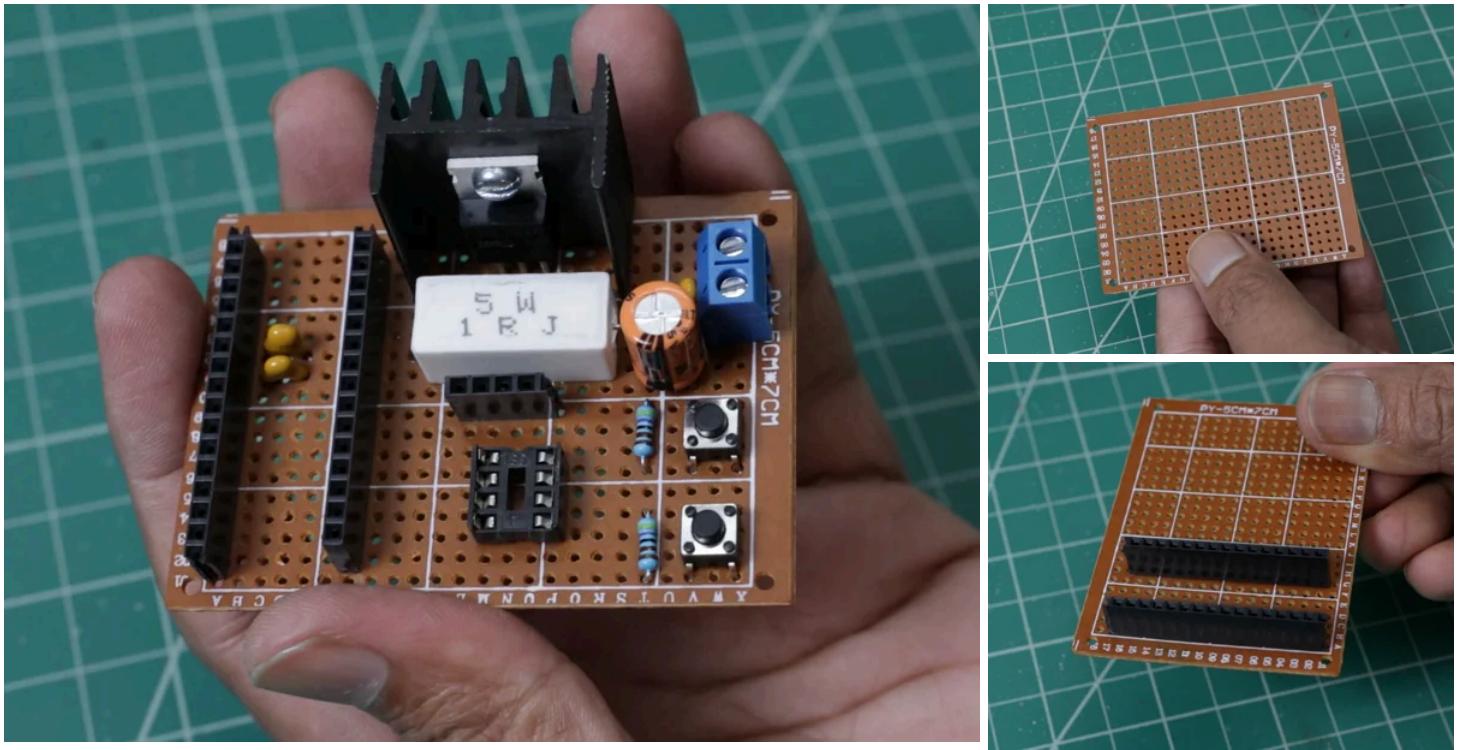
Here Battery is discharged to its low-level threshold voltage (3.2V).

Battery Capacity (mAh) = Current (I) in mA x Time (T) in Hours

From the above equation it is clear that to calculate battery capacity (mAh), we have to know the current in mA and time in Hour. The designed circuit is a constant current load circuit, so the discharge current remains constant throughout the testing period.

The discharge current can be adjusted by pressing the Up and Down Button. The time duration is measured by using a timer in the Arduino code.

Step 4: Making the Circuit



In the previous steps, I have explained the function of each of the components in the circuit. Before jump to make the final board, test the circuit on a breadboard first. If the circuit works perfectly on the breadboard, then move to solder the components on the prototype board.

I used the 7cm X 5cm prototype board.

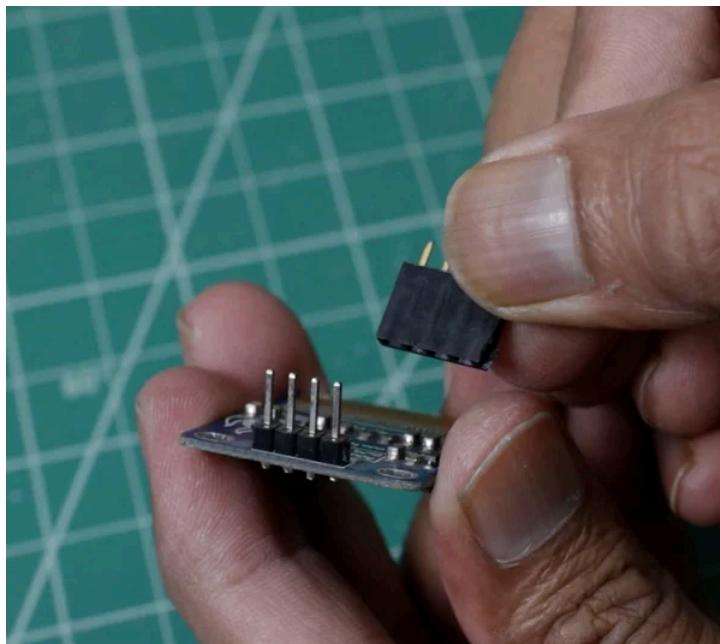
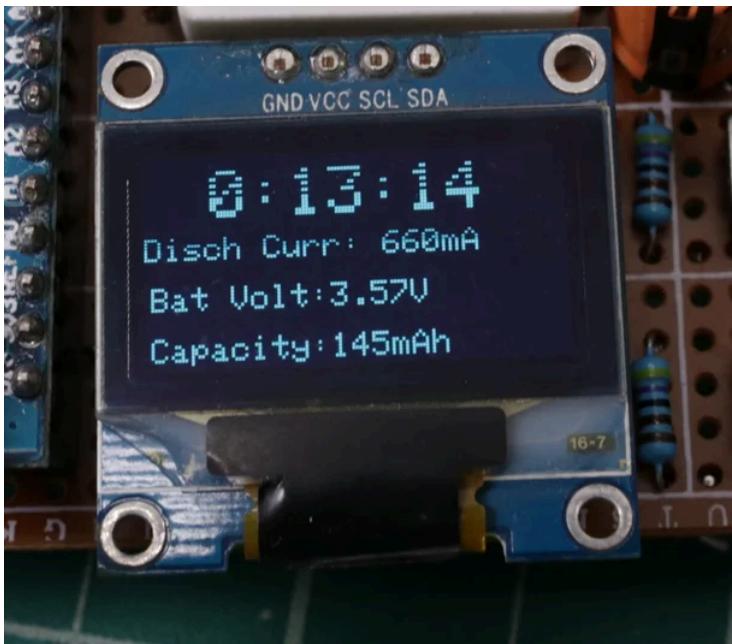
Mounting the Nano: First cut two rows of female header pin with 15 pins in each. I used a diagonal nipper to cut the headers. Then solder the header pins. Be sure the distance between the two rails fits the Arduino nano.

Mounting OLED Display: Cut a female header with 4pins. Then solder it as shown in the picture.

Mounting the terminals and components: Solder the remaining components as shown in pictures.

Wiring: Make the wiring as per the schematic. I used colored wires to make the wiring so that I can identify them easily.

Step 5: OLED Display



To display the Battery Voltage, discharge current and capacity, I used a 0.96" OLED display. It has 128x64 resolution and uses an I2C bus to communicate with the Arduino. Two pins SCL (A5), SDA (A4) in Arduino Uno are used for communication.

I am using the Adafruit_SSD1306 library to display the parameters.

First, you have to download the [Adafruit_SSD1306](#). Then installed it.

The connections should be as follows

Arduino --> OLED

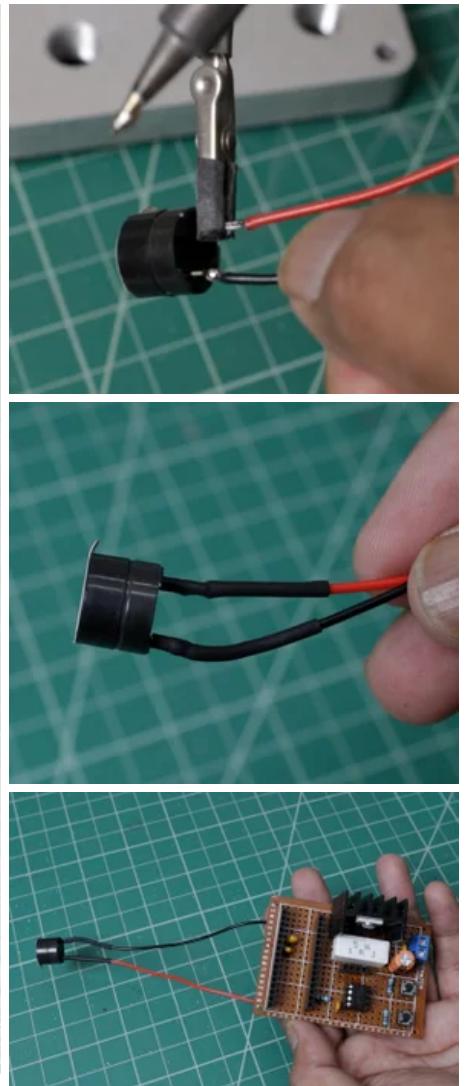
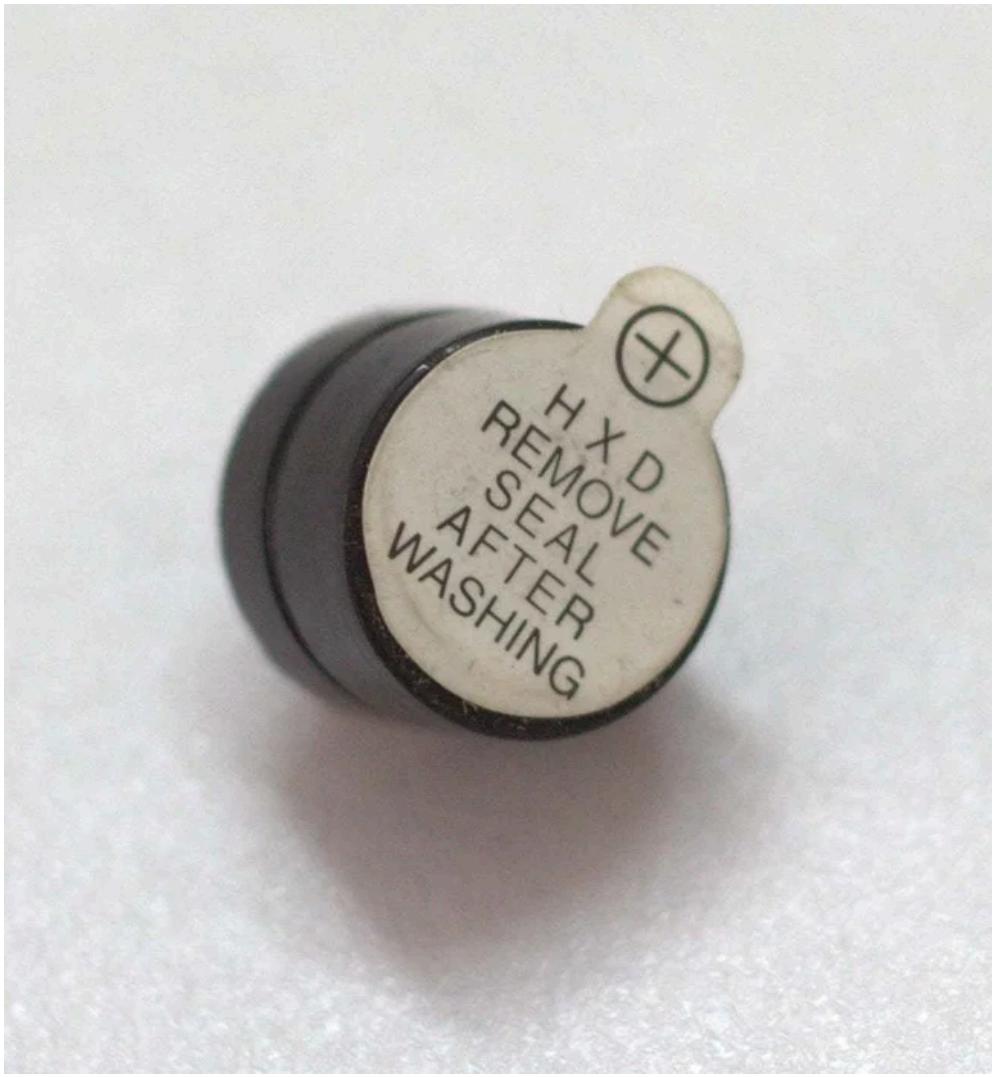
5V --->VCC

GND -->GND

A4----> SDA

A5----> SCL

Step 6: Buzzer for Warning



To provide alerts during the start and completion of the test, a piezo buzzer is used. The buzzer has two terminals, the longer one is positive and the shorter leg is negative. The sticker on the new buzzer has also "+" marked to indicate the positive terminal.

As the prototype board doesn't have enough room to place the buzzer, I have connected the buzzer to the main circuit board by using two wires. To insulate the bare connection, I have used heat shrink tubing.

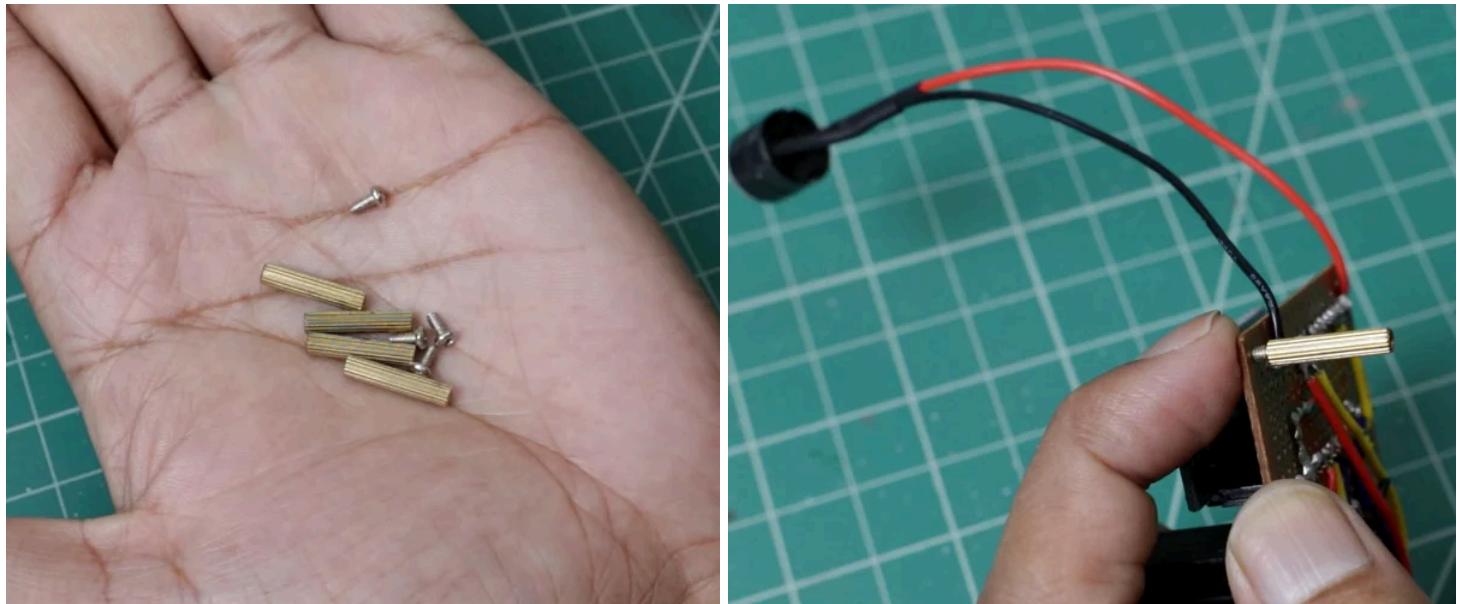
The connections should be as follows

Arduino --> Buzzer

D9--> Positive terminal

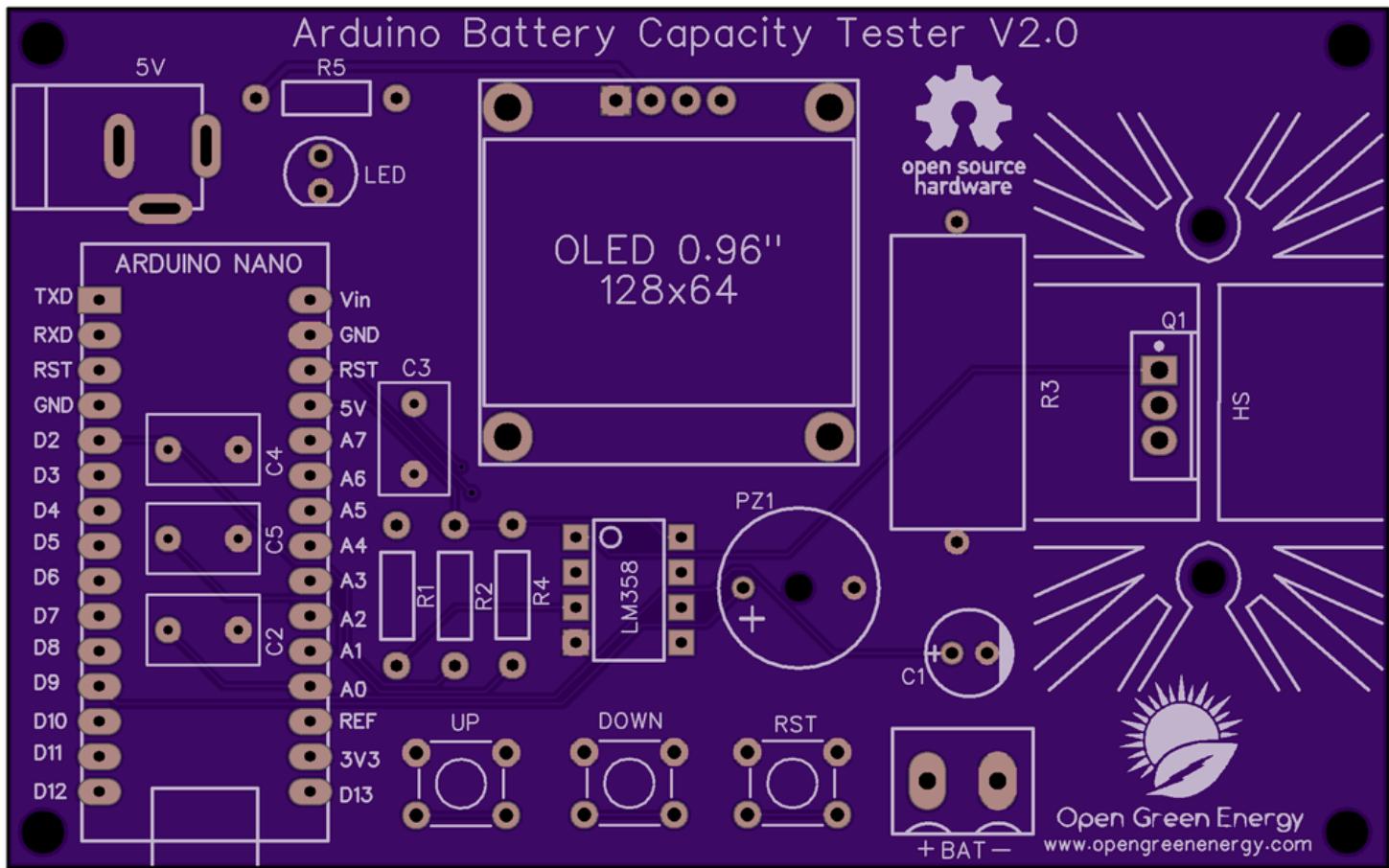
GND--> Negative terminal

Step 7: Mounting the Standoffs



After soldering and wiring, mount the standoffs at 4 corners. It will provide sufficient clearance to the soldering joints and wires from the ground.

Step 8: PCB Design



I have drawn the schematic by using EasyEDA online software after that switched to the PCB layout.

All of the components you added in the schematic should be there, stacked on top of each other, ready to be placed and routed. Drag the components by grabbing on its pads. Then place it inside the rectangular borderline.

Arrange all the components in such a way that the board occupies minimum space. Smaller the board size, the cheaper will be the PCB manufacturing cost. It will be useful if this board has some mounting holes on it so that it can be mounted in an enclosure.

Now you have to route. Routing is the most fun part of this entire process. It's like solving a puzzle! Using the tracking tool we need to connect all the components. You can use both the top and the bottom layer for avoiding overlap between two different tracks and making the tracks shorter.

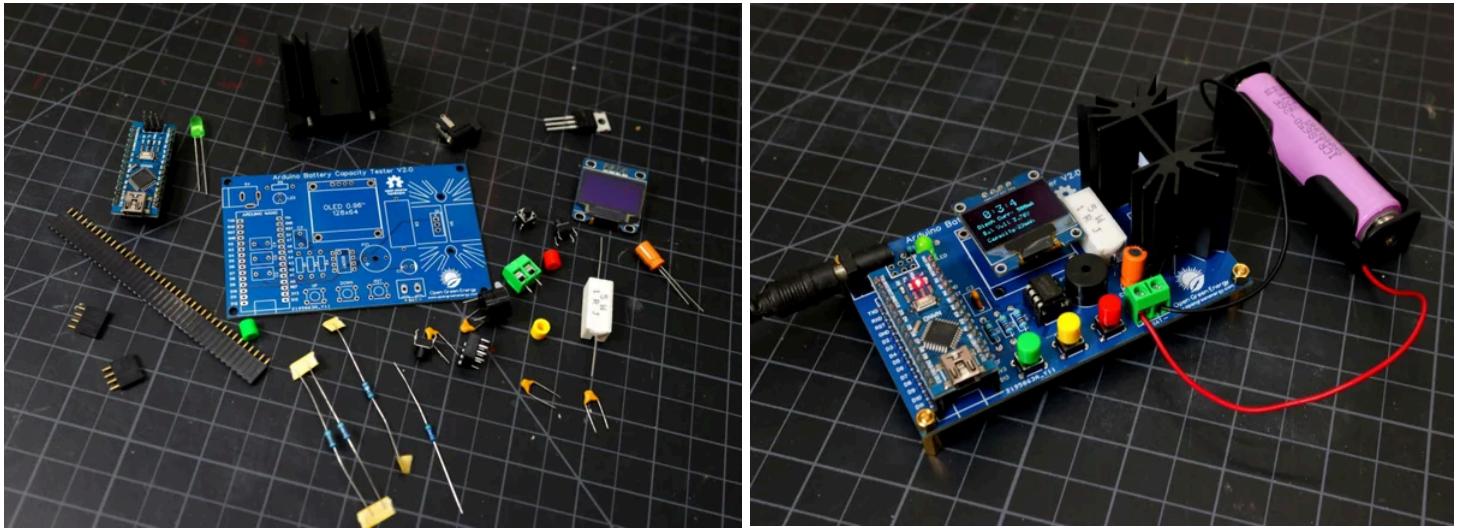
You can use the Silk layer to add text to the board. Also, we are able to insert an image file, so I add an image of my website logo to be printed on the board. In the end, using the copper area tool, we need to create the ground area of the PCB.

You can order it from [PCBWay](#).

Sign up PCBWay now to get a US \$5 coupon. That means your first order is free of cost only you have to pay the shipping charges.

When you place an order, I will get 10% donation from PCBWay for a contribution to my work. Your little help may encourage me to do more awesome work in the future. Thank you for your cooperation.

Step 9: Assemble the PCB

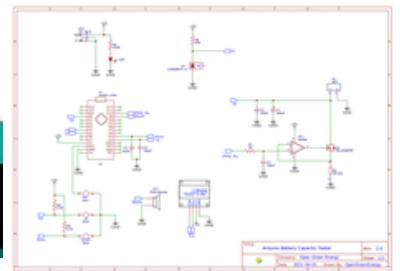
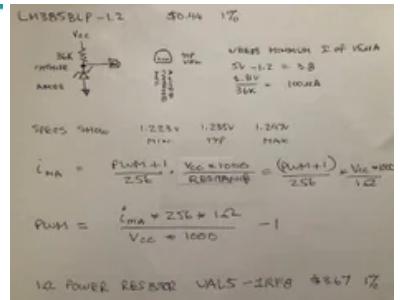


For Soldering, you will need a decent Soldering Iron, Solder, Nipper and a multimeter. It is good practice to solder the components according to their height. Solder the lesser height components first.

You can follow the following steps to solder the components :

1. Push the component legs through their holes, and turn the PCB on its back.
2. Hold the tip of the soldering iron to the junction of the pad and the leg of the component.
3. Feed solder into the joint so that it flows all around the lead and covers the pad. Once it has flowed all around, move the tip away.

Step 10: Software & Libraries



First, download the attached Arduino Code. Then download the following libraries and install them.

Libraries:

Download and install the following libraries:

1. JC_Button: https://github.com/JChristensen/JC_Button
 2. Adafruit SSD1306: https://github.com/JChristensen/JC_Button

In the code, you have to change the following two things.

1. Current Arrays values: This can be done by connecting a multimeter in series with the battery. Press the up button and measure the current, the current values are the elements of the array.
 2. Vcc: You use a multimeter to measure the voltage at the Arduino 5V pin. In my case it is 4.96V.

Updated on 20.11.2019

You can change the **Low_BAT_Level** value in the code as per the battery chemistry. It is better to take a little margin over the cut-off voltage stated below.

Here is the discharge rates and cutoff voltages for various Lithium-Ion Battery chemistries:

1. Lithium Cobalt Oxide: Cut-off Voltage = 2.5V at 1C discharge rate
2. Lithium Manganese Oxide: Cut-off Voltage = 2.5V at 1C discharge rate
3. Lithium Iron Phosphate: Cut-off Voltage = 2.5V at 1C discharge rate
4. Lithium Titanate: Cut-off Voltage = 1.8V at 1C discharge rate
5. Lithium Nickel Manganese Cobalt Oxide: Cut-off Voltage = 2.5V at 1C discharge rate
6. Lithium Nickel Cobalt Aluminum Oxide: Cut-off Voltage = 3.0V at 1C discharge rate

Updated on 01.04.2020

jcgrabo, suggested some changes to the original design to improve the precision. The changes are listed below:

1. Add a precision reference ([LM385BLP-1.2](#)) and connected it to A1. During setup, read its value which is known to be 1.215 volts, and then calculate Vcc thereby eliminating the need to measure Vcc.
2. Replace the 1 ohm 5% resistor with a 1 ohm 1% power resistor thereby reducing errors that depend on the value of the resistance.
3. Rather than using a fixed set of PWM values for each current step (in increments of 5) create an array of desired current values that used to compute the necessary PWM values to achieve those current values as close as possible. He followed that by calculating the actual current values that will be achieved with the computed PWM values.

By considering the above changes, he revised the code and shared it in the comment section. The revised code is attached below.

Thank you so much jcgrabo for your valuable contribution to my project. I hope this improvement will be helpful for many more users.