PCB WORKSHOP

1. Simulation Results for LED Blinking Using 555 Timer

a. Detailed Description of the Simulation Results

Circuit Overview:

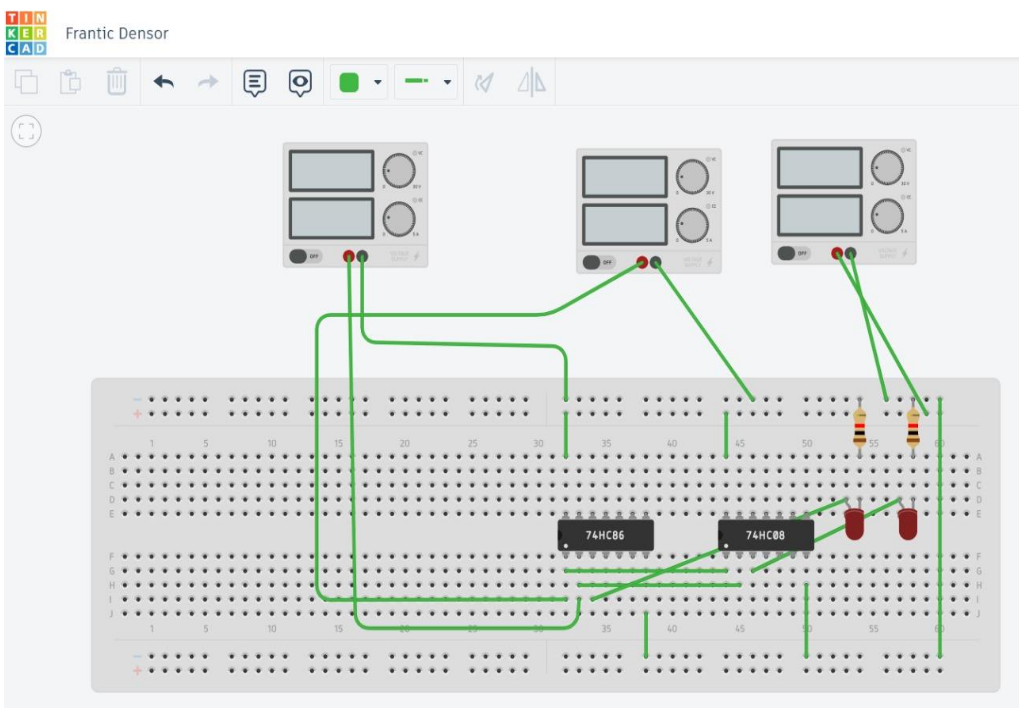
- **555 Timer Configuration:** The 555 timer is configured in astable mode to generate a continuous square wave.
- **Components:**
  - **Resistors (R1 and R2):** These set the timing interval.
  - **Capacitor (C1):** This charges and discharges to create the timing cycle.
  - **LED:** Connected to the output of the 555 timer through a current-limiting resistor.

Operation:

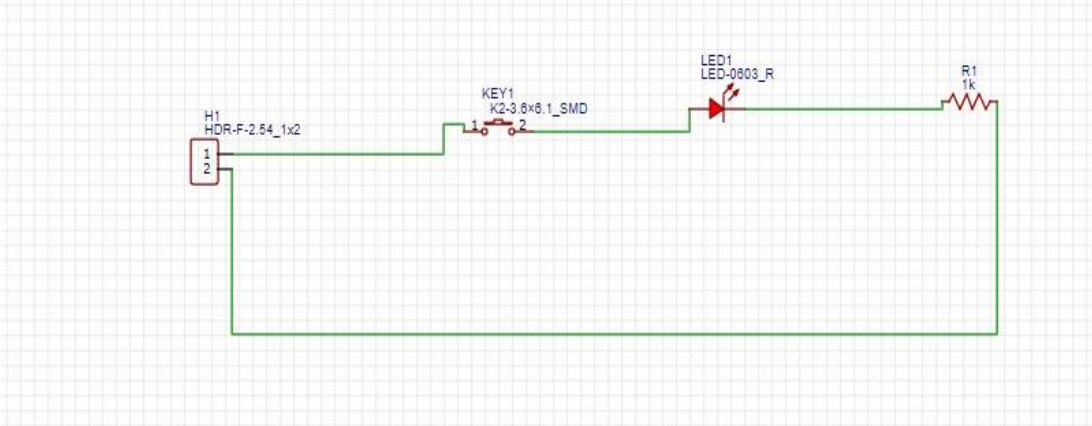
- The 555 timer in astable mode continuously oscillates between high and low states, creating a square wave output.
- The frequency of the oscillation is determined by R1, R2, and C1.
- The LED blinks on and off as the output of the 555 timer toggles between high and low states.

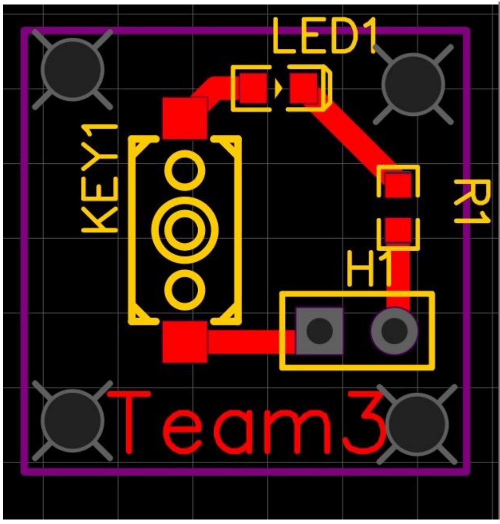
b. Screenshots or Diagrams

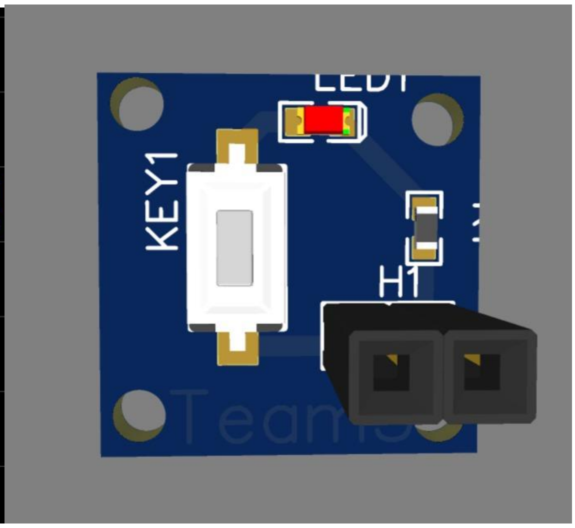
1.Circuit Schematic in Tinkercad:



IN EASYEDA







i. Input/Output Waveforms

Output Waveform (Square Wave):

- The 555 timer output (pin 3) produces a square wave.
- This square wave causes the LED to turn on and off at regular intervals.

ii. Frequency Response Plots

Frequency Response:

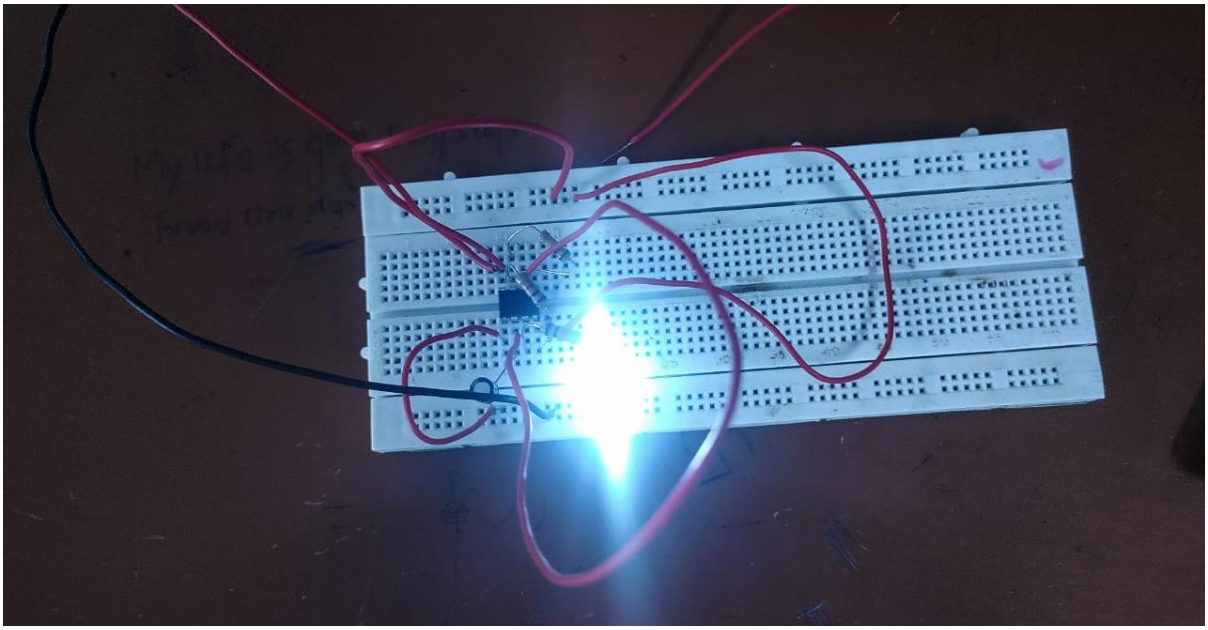
- Since the 555 timer in astable mode generates a fixed frequency determined by the external resistors and capacitor, the frequency response plot typically shows a peak at this oscillation frequency.

iii. Transient Responses

Transient Response:

- The transient response shows how the voltage across the capacitor (C1) charges and discharges over time, and how this affects the output voltage (Vout) of the 555 timer.
- The capacitor charges through R1 and R2, and discharges through R2 only

HARDWARE RESULTS:



2. Simulation Results for Inverting Summing Amplifier (Analog Circuit)

a. Detailed Description of the Simulation Results

Circuit Overview:

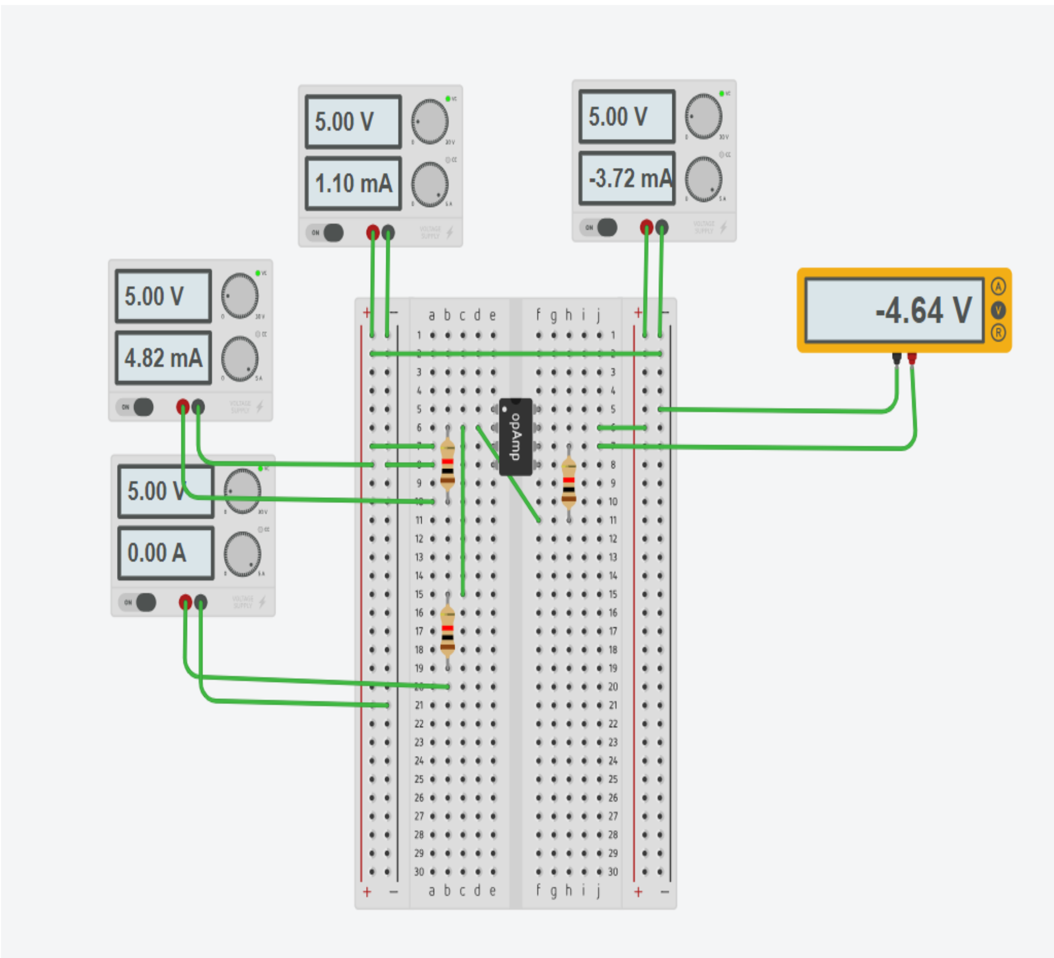
- **Inverting Summing Amplifier:** The circuit sums multiple input voltages, inverts the result, and amplifies it.
- **Operational Amplifier (Op-Amp):** Typically, an op-amp like the LM741 is used.
- **Resistors:** Input resistors

  Operation:

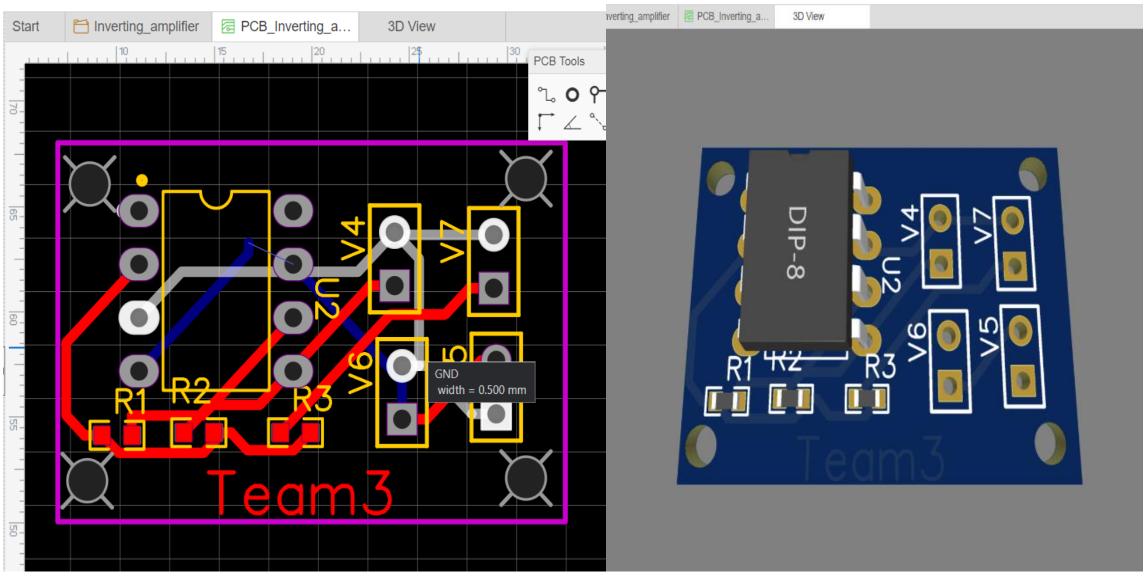
- The inverting summing amplifier adds multiple input voltages Vin1 ,Vin2 ,...,Vin after inverting them.

b. Screenshots or Diagrams

1. In Tinker Cad:



2. In EASYEDA:



i. Input/Output Waveforms

Input and Output Waveforms:

- Inputs Vin1, Vin2, and Vin3 *are* sinusoidal signals with different amplitudes and frequencies.
- The output waveform $Vout$ is the inverted sum of the inputs.

ii. Frequency Response Plots
Frequency Response:

- The frequency response plot shows the gain of the amplifier over a range of frequencies.
- For an ideal op-amp, the gain should be constant up to a certain frequency (bandwidth) beyond which it decreases.

iii. Transient Responses
Transient Response:

- The transient response shows how the output voltage changes over time in response to the input signals.
- It illustrates the inverting and summing action of the amplifier

3. Simulation Results for Blinking LED Using Arduino(Digital circuit)

a. Detailed Description of the Simulation Results

Circuit Overview:

- **Arduino Board:** Arduino Uno is used.
- **LED:** Connected to one of the digital pins of the Arduino (e.g., pin 13).
- **Current-Limiting Resistor:** Usually a 220Ω resistor is used in series with the LED to limit the current and prevent damage to the LED.

Operation:

- The Arduino is programmed to blink the LED on and off at regular intervals.
- The code toggles the digital pin connected to the LED between HIGH (on) and LOW (off) states, creating the blinking effect.

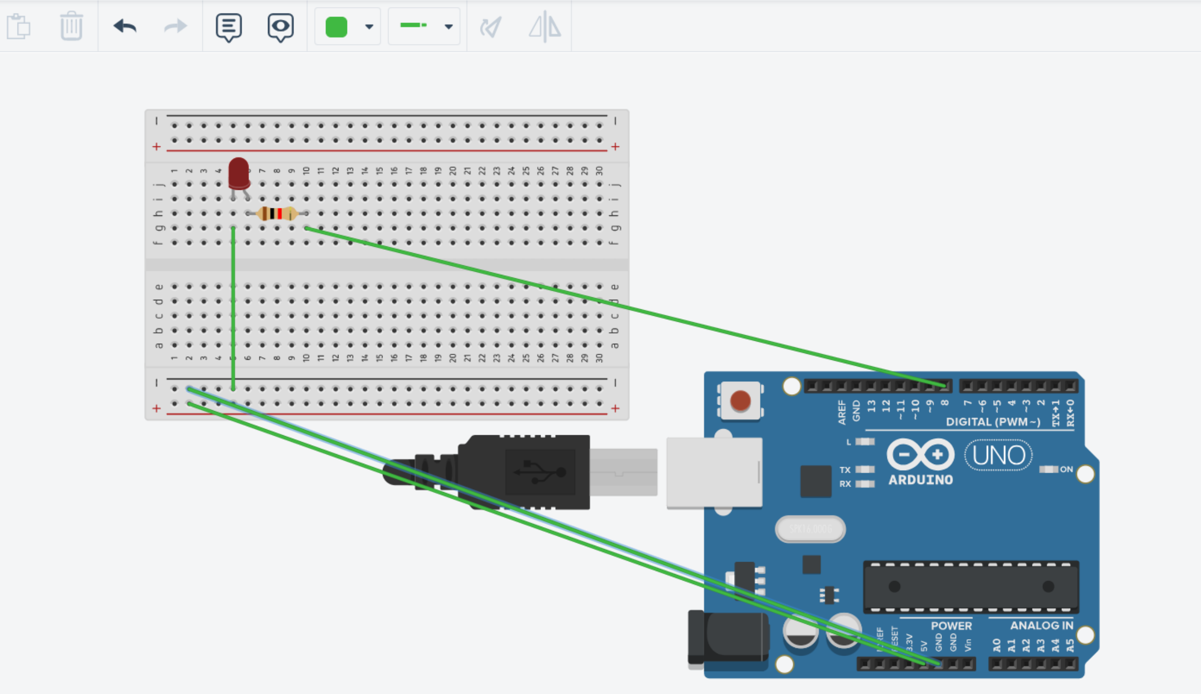
Code:

```
int ledPin=8; //definition digital 8 pins as pin to control the LED
void setup()
{
  pinMode(ledPin,OUTPUT);    //Set the digital 8 port mode, OUTPUT: Output mode
}
void loop()
{
  digitalWrite(ledPin,HIGH); //HIGH is set to about 5V PIN8
  delay(1000);               //Set the delay time, 1000 = 1S
  digitalWrite(ledPin,LOW);  //LOW is set to about 5V PIN8
  delay(1000);               //Set the delay time, 1000 = 1S
}
```

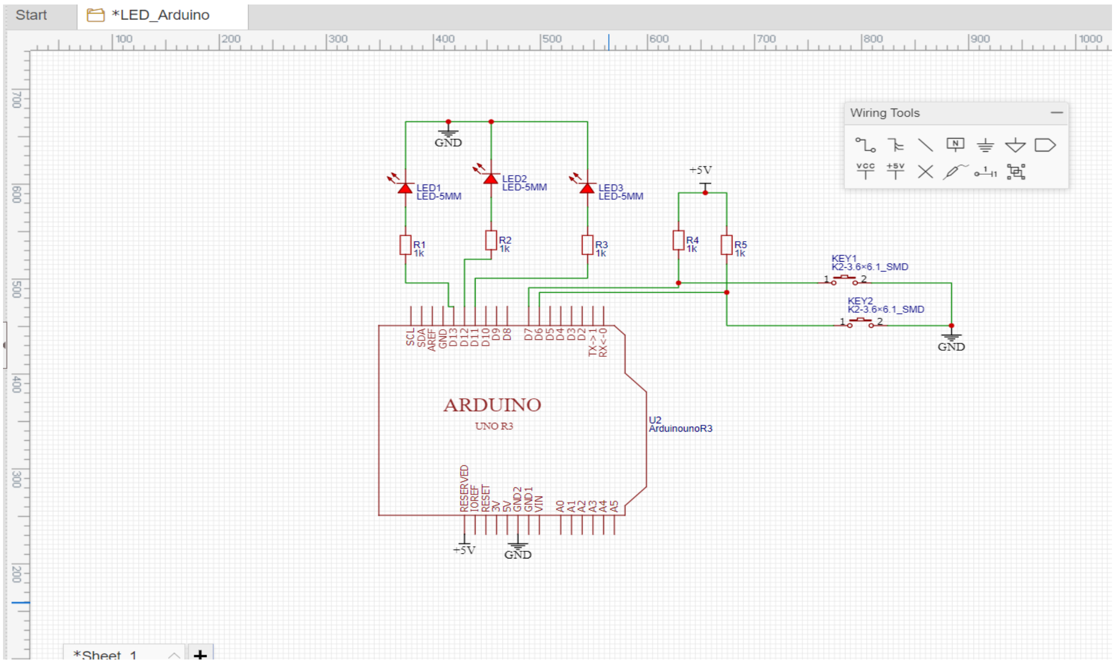
b. Screenshots or Diagrams

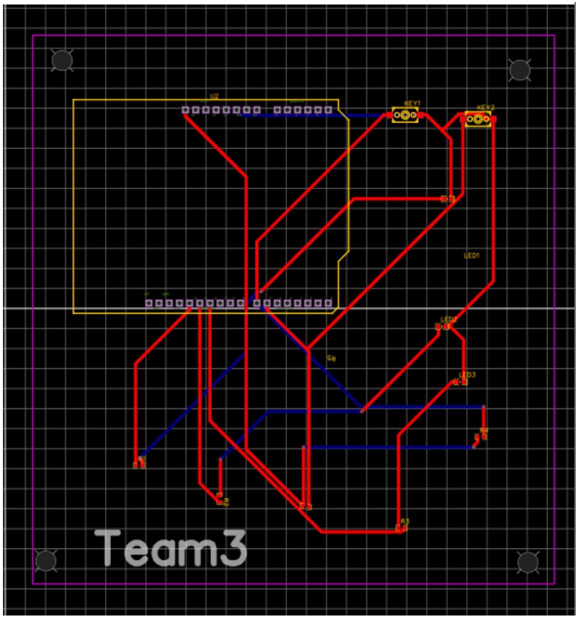
1. In Tinker Cad:

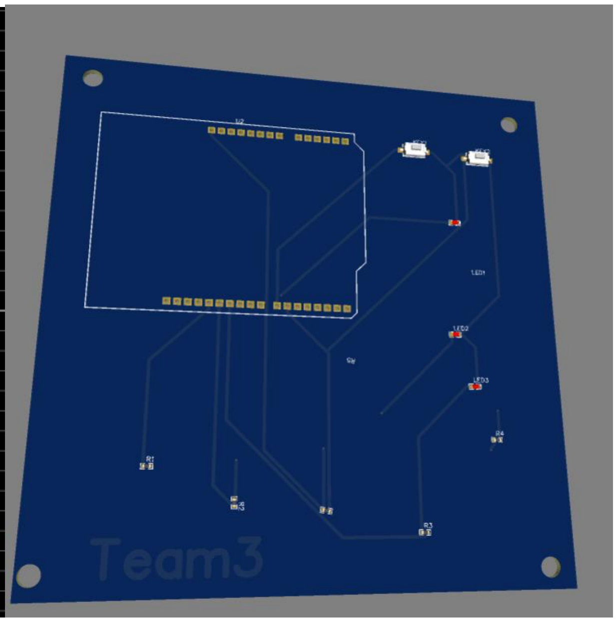
LED blink with arduino



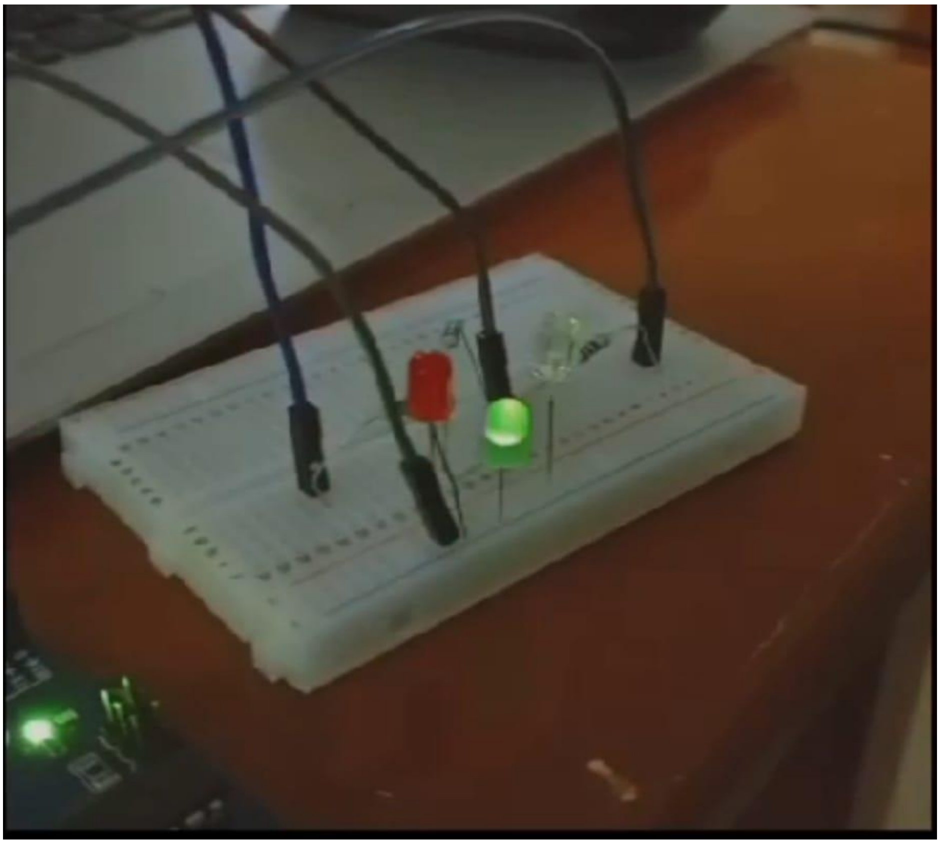
2. In EASYEDA:







Hardware Results:



Steps to Generate and Export Gerber Files from EasyEDA

1. **Log into EasyEDA:**
    - Open your web browser and go to EasyEDA.
    - Log into your account using your credentials.
2. **Open Your Project:**
    - Once logged in, navigate to your project by clicking on "My Projects" from the dashboard or the main menu.
    - Select the project containing the PCB design you want to export.
3. **Open the PCB Design:**
    - Within your project, click on the PCB design to open it in the editor.
4. **Check Your Design:**
    - Before generating the Gerber files, ensure your PCB design is complete and free of errors. Use the Design Rule Check (DRC) feature in EasyEDA to identify and fix any potential issues.
        - Go to **`Design > Design Rule Check (DRC)`**, and run the check. Address any errors or warnings that are flagged.
5. **Generate Gerber Files:**
    - Once your design is verified, you can proceed to generate the Gerber files.
    - Go to **`Fabrication > PCB Fabrication File (Gerber)`**.
        - This will open the "Gerber Output" dialog.
6. **Configure Gerber Output Settings:**
    - In the "Gerber Output" dialog, you can configure various settings such as:
        - **Output Format**: Ensure it is set to RS-274X.
        - **Layers to Export**: By default, EasyEDA will select all necessary layers (Top Layer, Bottom Layer, Solder Mask, Silkscreen, etc.).
        - **Other Options**: You can choose options like "Use Protel File Pad/Line Width" or set a specific board outline if needed.
    - After configuring, click the "Generate" button.
7. **Download Gerber Files:**
    - Once the generation process is complete, EasyEDA will create a zip file containing all the necessary Gerber files.
    - You will be prompted to download this zip file. Click on the "Download" button to save the file to your local computer.