Battery Monitoring and Safety Alert System

Submitted by: Sumit Ghule

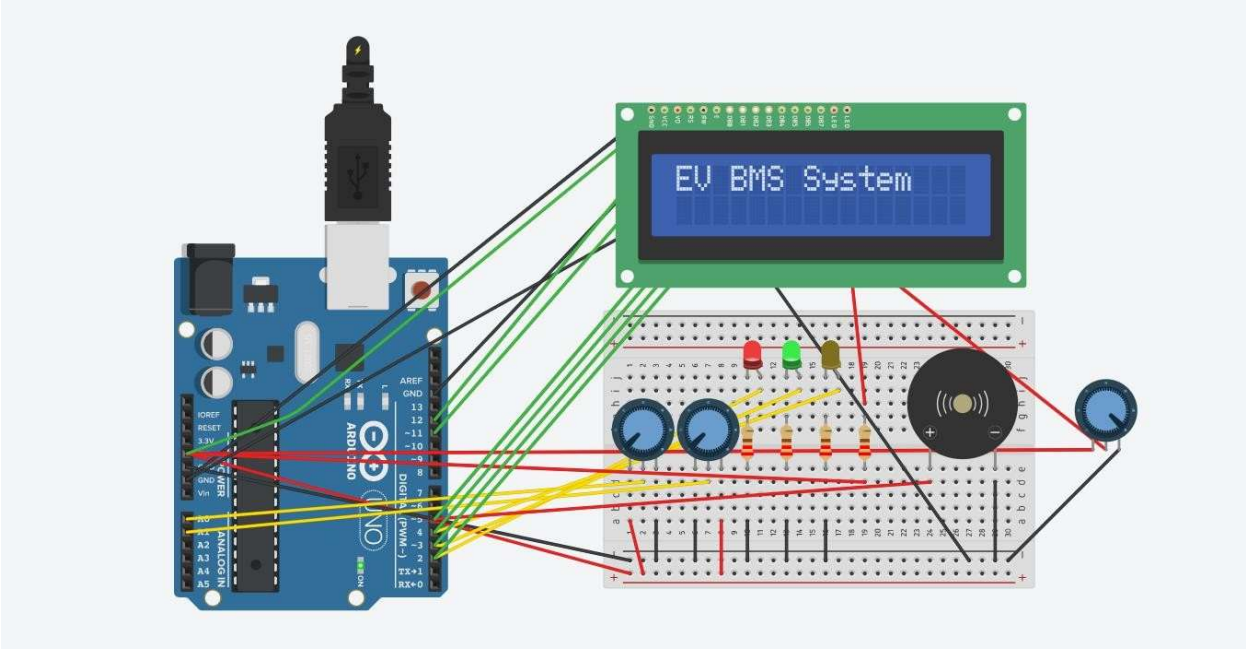
Objective

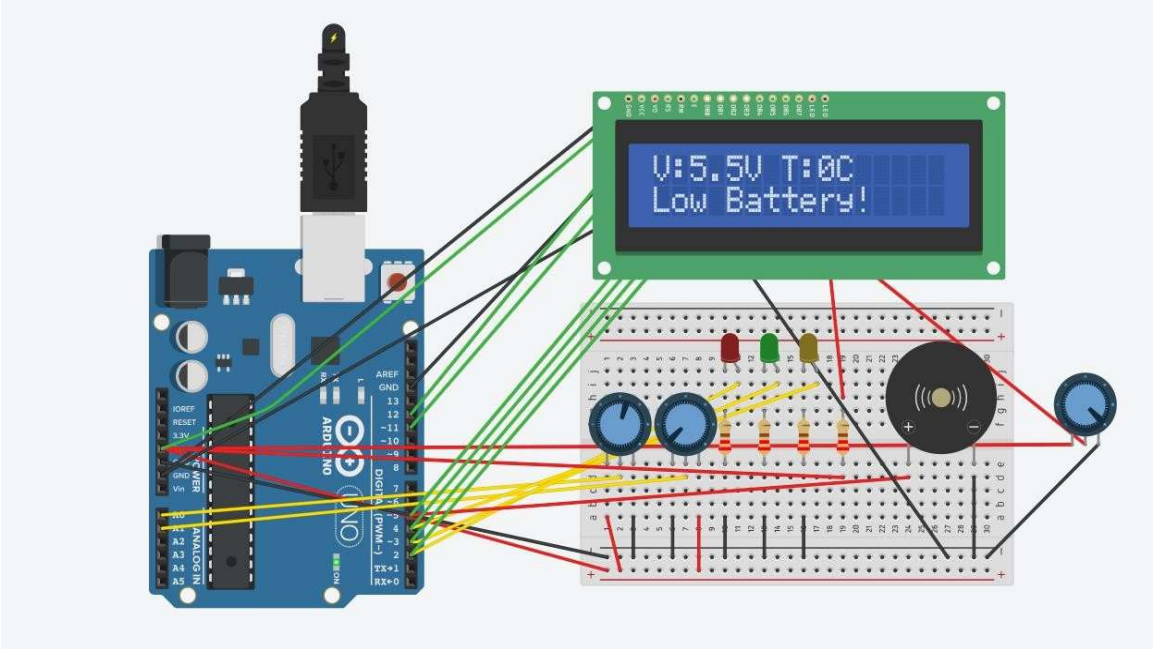
The objective of this project is to design and simulate a battery monitoring system that continuously tracks battery voltage and temperature, and provides visual and audible alerts when unsafe conditions are detected. The system ensures early warning for low voltage and overheating to protect the battery and connected electronics.

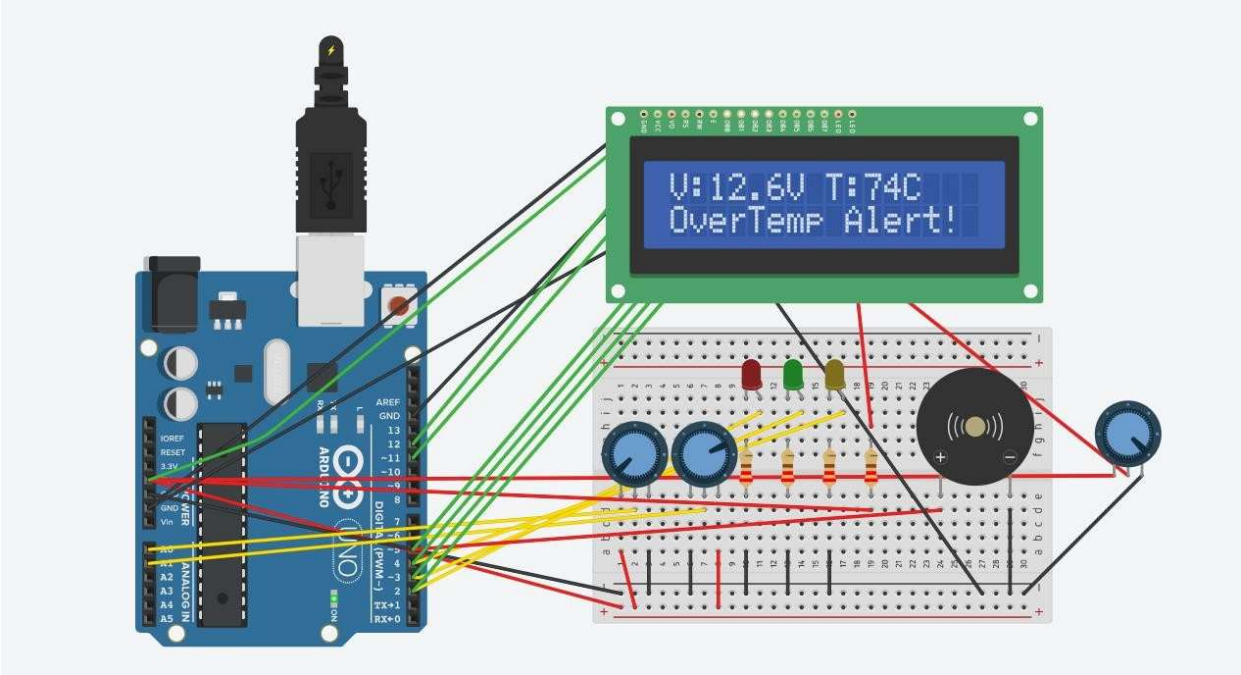
Components Required

Sr. No.	Component	Quantity
1	Arduino UNO R3	1
2	Potentiometer	2
3	Red LED	1
4	Green LED	1
5	Yellow LED	1
6	Resistors (220Ω)	3
7	Buzzer	1
8	Breadboard	1
9	Jumper Wires	As req.
10	LCD 16x2 (I2C)	Optional

OUTPUT Screenshots:







Working Principle

Two potentiometers simulate battery voltage and temperature readings. Analog inputs from the potentiometers are scaled in software to represent real-world values. The Arduino processes these readings and applies logic:

- Temperature > 45°C → Yellow LED + buzzer (Overheat)
- Battery voltage < 11.0V → Red LED + buzzer (Low battery)
- 11.0V ≤ voltage ≤ 12.6V → Green LED (Normal)
- Voltage > 12.6V → Green LED (Charging/High)

Alerts are displayed on the LCD and/or Serial Monitor.

Arduino Program:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int pinVinSense = A0;
const int pinTempSense = A1;
const int pinRedLED = 8;
const int pinGreenLED = 9;
const int pinYellowLED = 10;
const int pinBuzzer = 11;

const float maxSimVolt = 15.0;

void setup() {
  pinMode(pinRedLED, OUTPUT);
  pinMode(pinGreenLED, OUTPUT);
  pinMode(pinYellowLED, OUTPUT);
  pinMode(pinBuzzer, OUTPUT);

  Serial.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print("BMS Simulator");
  delay(1000);
  lcd.clear();
}

void loop() {
  float batteryVolt = (analogRead(pinVinSense) / 1023.0) * maxSimVolt;
  float tempC = (analogRead(pinTempSense) / 1023.0) * 100.0;

  bool overheat = (tempC > 45.0);
  bool lowBattery = (batteryVolt < 11.0);
  bool batteryOk = (batteryVolt >= 11.0 && batteryVolt <= 12.6);

  digitalWrite(pinRedLED, LOW);
  digitalWrite(pinGreenLED, LOW);
  digitalWrite(pinYellowLED, LOW);
  noTone(pinBuzzer);

  if (overheat) {
    digitalWrite(pinYellowLED, HIGH);
    tone(pinBuzzer, 1000);
    showLCD("Overheat!", batteryVolt, tempC);
  }
  else if (lowBattery) {
    digitalWrite(pinRedLED, HIGH);
    tone(pinBuzzer, 800);
```

```
    showLCD("Low Batt!", batteryVolt, tempC);
  }
  else if (batteryOk) {
    digitalWrite(pinGreenLED, HIGH);
    showLCD("Battery OK", batteryVolt, tempC);
  }
  else {
    digitalWrite(pinGreenLED, HIGH);
    showLCD("Battery High", batteryVolt, tempC);
  }

  delay(1000);
}

void showLCD(const char* status, float v, float t) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(status);
  lcd.setCursor(0, 1);
  lcd.print("V:");
  lcd.print(v, 1);
  lcd.print(" T:");
  lcd.print(t, 1);
}
```

Conclusion:

This project successfully simulates a battery monitoring system with temperature and voltage alerts. It demonstrates integration of sensors, Arduino programming, and visual/audible indicators. The system can be expanded with actual sensors and used in electric vehicle battery management.

Future Enhancements:

- Integration with real Li-ion cells
- Adding data logging to SD card
- Wireless alerts via Bluetooth/Wi-Fi
- Cell balancing and SOC estimation

References:

Arduino Documentation: https://www.arduino.cc/
Tinkercad Circuits: https://www.tinkercad.com/dashboard