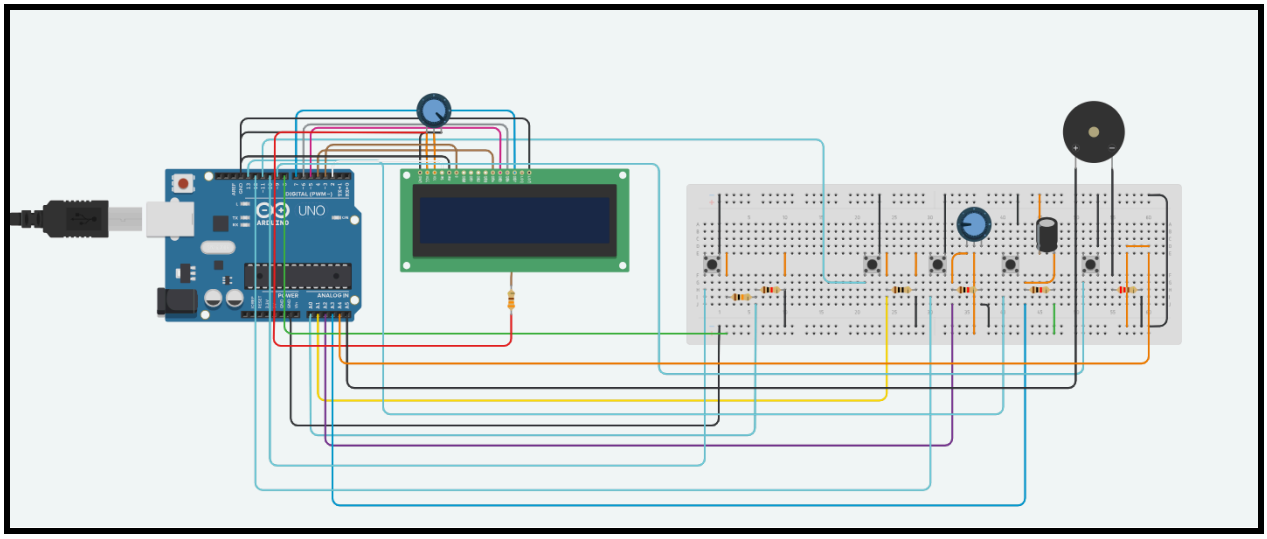
Measurements Report

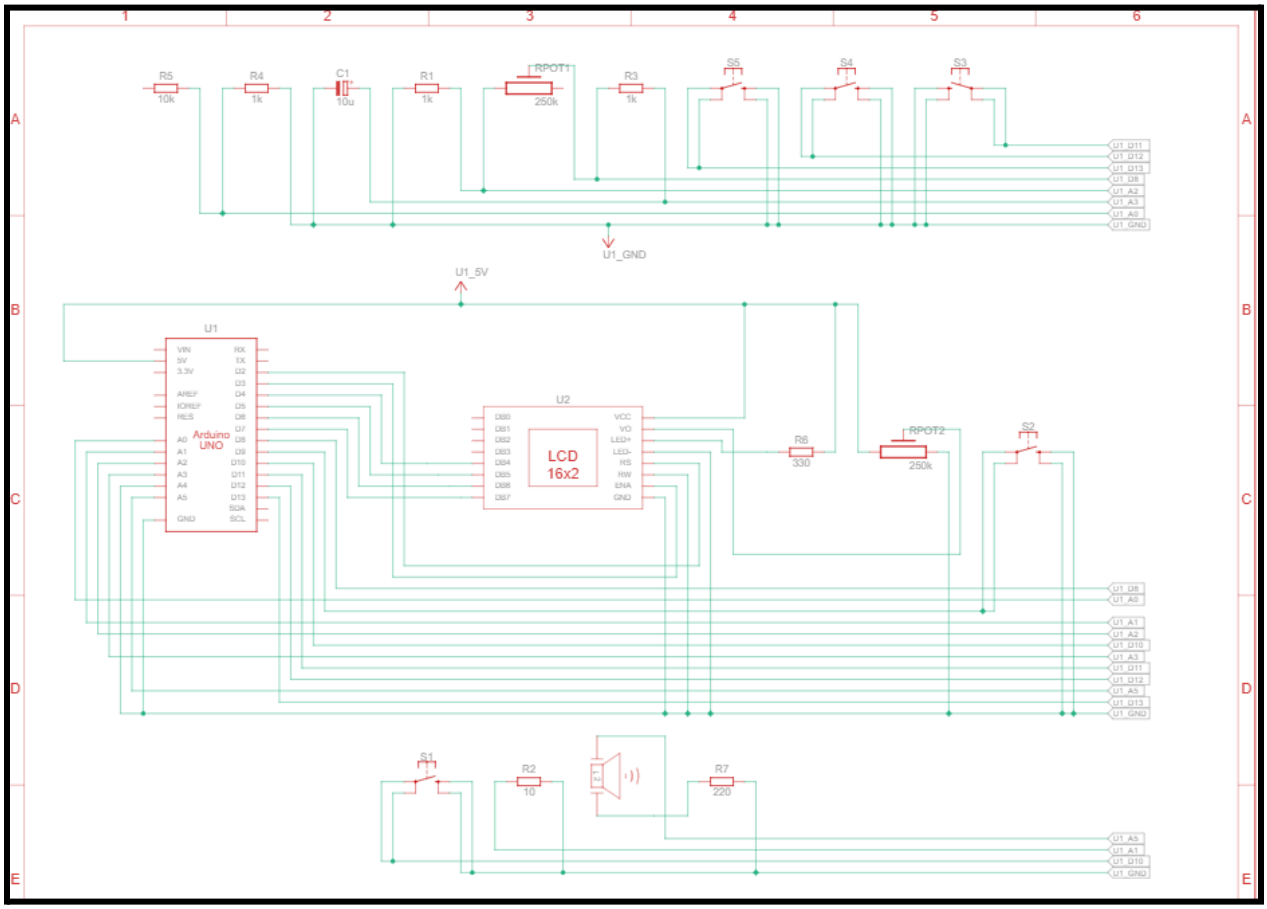
Avo Project

Students Data	
الاسم	**الرقم الجامعي**
1-عمر محمد حسن السيد	*23011933*
2-علي محمود علي	*23010594*
3-عبد الرحمن علي عبد الله	*23010531*
4-عمرو أحمد إبراهيم	*23010624*
5-عمر محمد محمود	*23010621*

Content

1.Overview


UNO


R5
10k
R4
1k
C1
10u
R1
1k
RPOT1
250k
R3
1k
S5
S4
S3
U1_GND
U1_5V
U1
Arduino
UNO
U2
LCD
16x2
R6
330
RPOT2
250k
S2
S1
R2
10
R7
220

-**A digital multimeter,** often abbreviated as **DMM** or sometimes called a digital Avometer, is a versatile electronic tool used by technicians, engineers, and hobbyists to measure various electrical properties accurately.

Here's a brief overview of how it works and what it can measure:

1. Voltage (Volts): One of the primary functions of a digital multimeter is measuring voltage. It can measure both direct current (DC) and alternating current (AC) voltage levels. This capability allows users to check the voltage of batteries, power supplies, circuits, and electrical outlets.

2. Current (Amperes): Another crucial function is measuring current. Like voltage, it can measure both DC and AC currents. By breaking and connecting the circuit in series, the multimeter can measure the flow of electrical current through the circuit, helping users assess the current drawn by various components or devices.

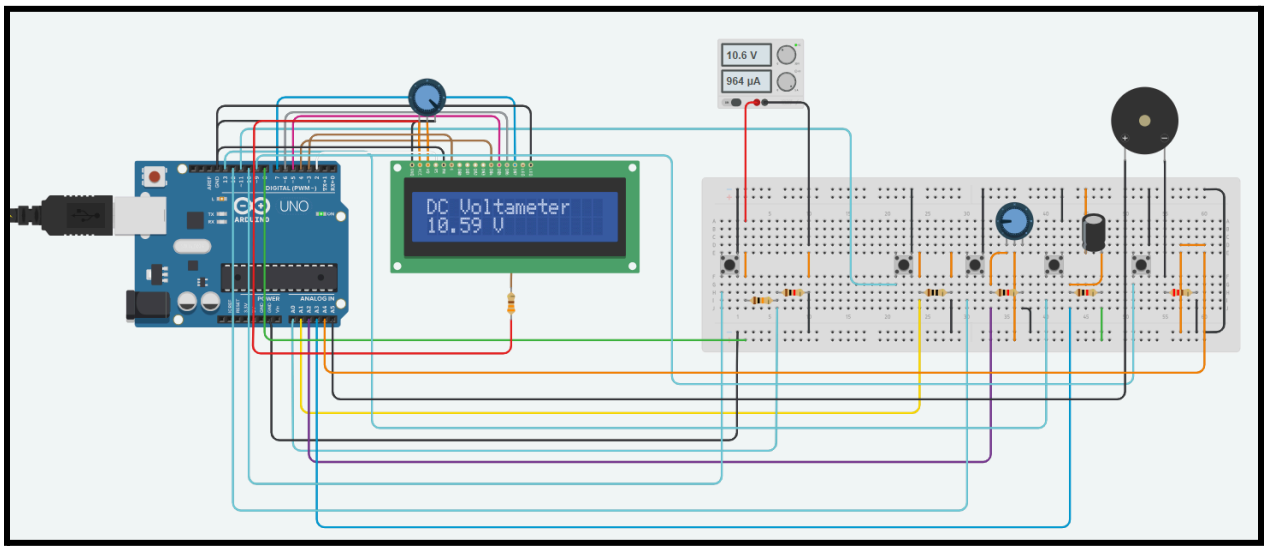
3. Resistance (Ohms): Digital multimeters can also measure resistance. This feature is useful for diagnosing electrical faults, identifying open or short circuits, and determining the continuity of wires and components. By connecting the multimeter leads across the component or circuit under test, it measures the resistance in ohms.

4.**Cap.meter(farad):** measures a capacitor's ability to store charge, expressed in **farads (F)**. It applies a known voltage or current and calculates capacitance based on the response. These meters are used for checking, diagnosing, and designing electronic circuits. Accurate measurement usually requires discharging the capacitor first.

5.Continuity test: A continuity test with buzzer checks if an electrical path is complete. Using a multimeter set to continuity mode, the device emits **a beep** (buzzer sound) when the circuit is complete, indicating very low resistance. It's a fast and easy way to find broken wires, bad connections, or faulty components.

2.Simulation

A)Voltmeter:



-A **voltmeter** is a device that can measure both **voltage** in the circuit. In this setup, an **Arduino** acts as a voltmeter by using its analog pins. To measure voltage, it reads the analog input (0–5V, scaled with a voltage divider). To measure current, it detects the voltage across a **series resistor**. By combining both measurements, the Arduino can function as a simple voltmeter.

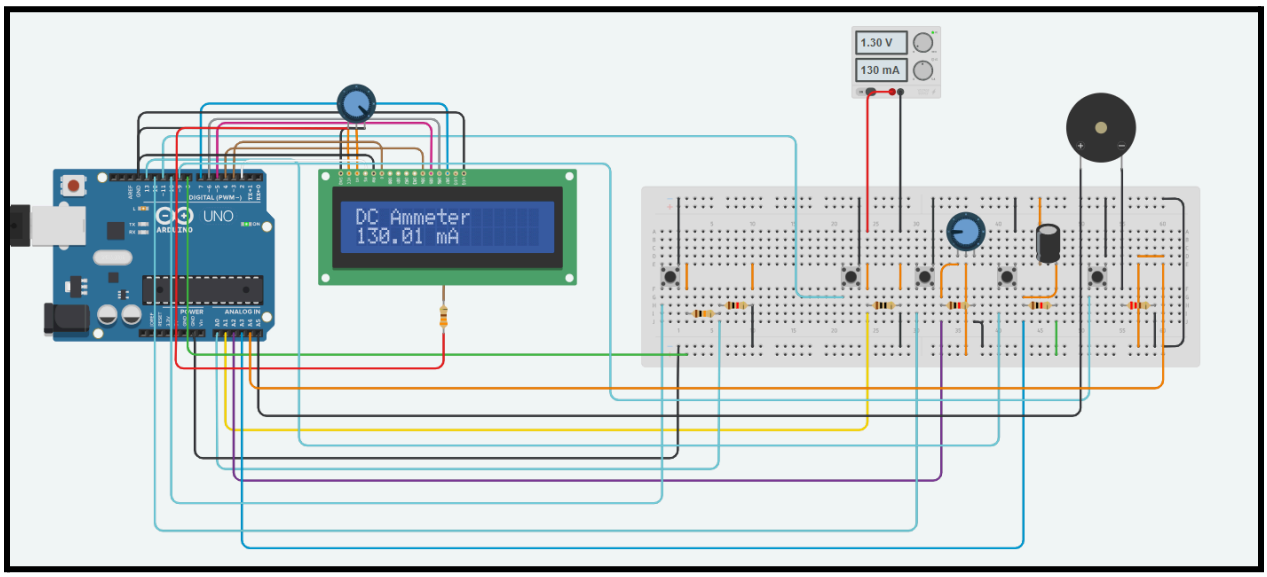
- **Here How the Voltmeter Works:**

1. **Reads voltage** from pin **A0**, which is connected to a **voltage divider**.

2. The voltage divider steps down the input voltage using two resistors:
   $Vin = Vout \times (R1 + R2)/R2$
   (where `R1 = 10kΩ`, `R2 = 1kΩ`)

3. `analogRead()` measures `V_out` (0–5V), and the code scales it to get the original voltage (`V_in`).

B)Ammeter:



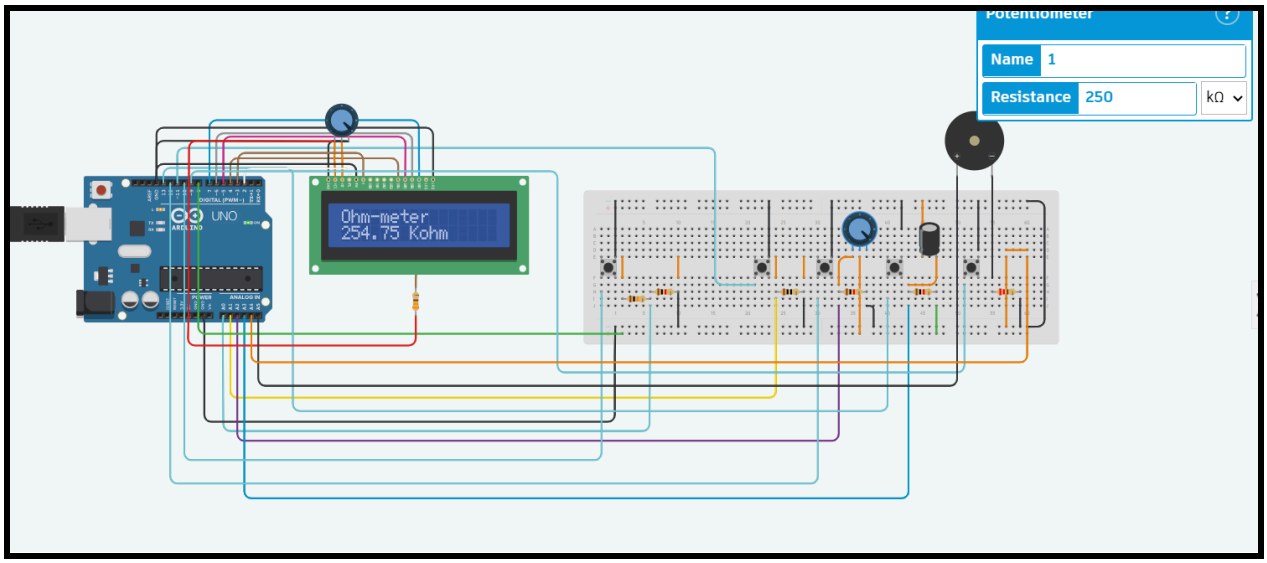
-An **ammeter** is a device used to measure the **electric current** flowing through a circuit. In this setup, the ammeter is connected to an **Arduino**. Since Arduino's analog pins measure voltage (0–5V), a **shunt resistor** (a low-resistance resistor) is placed in series with the circuit. The voltage drop across the shunt is measured and, using **Ohm's law (I = V/R)**, the current is calculated. By selecting the value of the shunt resistor carefully, we can control the range of current that can be accurately measured.

-Here How the Ammeter Works:

1. **Reads voltage** across a **shunt resistor (10Ω)** from analog pin **A1**.

2. Uses Ohm's Law:
   $I = V/R$
   So:
   Current=$(Vout/10)\times 1000$ mA
3. `analogRead()` measures the voltage drop (`V_out`) across the shunt resistor, then converts it to current.

C)Ohmmeter:



-An **ohmmeter** measures the **resistance** of a component. In this setup, an **Arduino** acts as an ohmmeter by applying a known voltage across the unknown resistor and measuring the resulting current or voltage drop. Using **Ohm's Law (R = V/I)**, the resistance is calculated. Often, a **voltage divider** is used where the unknown resistor and a known resistor are connected in series, and Arduino measures the voltage across one of them to find the unknown resistance.

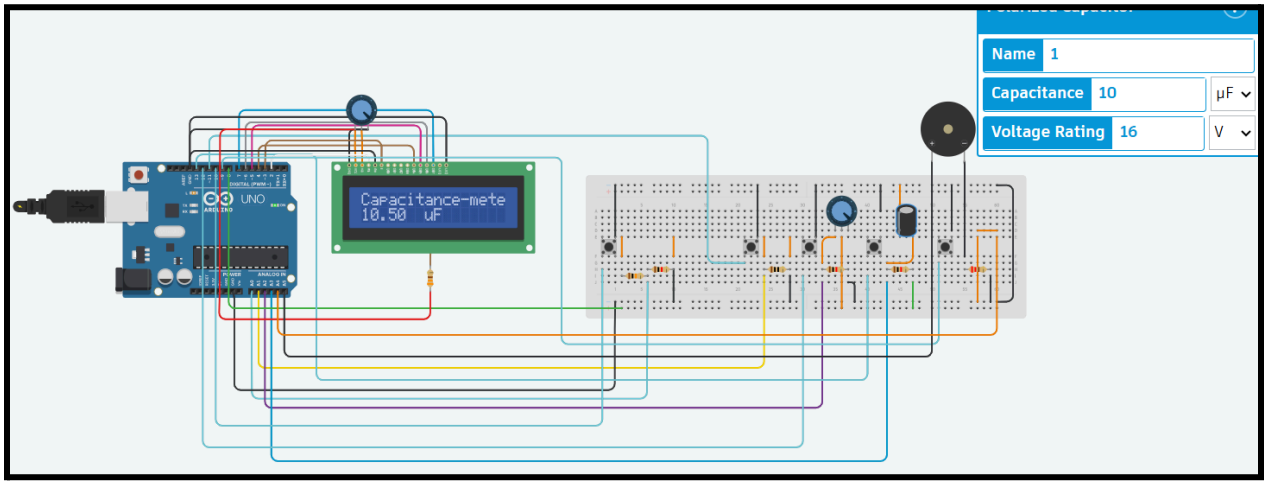
-Here How the Ohmmeter Works:

1. **Applies 5V** across a **known resistor (1kΩ)** in series with the **unknown resistor**.

2. Measures voltage drop across the **known resistor** at pin A2.

3. Calculates current:
   $I = Vknown/Rknown$
4. Uses Ohm's Law to find the unknown resistance:
   $Runknown = Vknoown/I = (5 - Vknown)/I$
5. Displays result in ohms or kilo-ohms.

D)Cap.meter:



-A **capacitance meter (cap meter)** measures the **capacitance** of a capacitor. In this setup, an **Arduino** can act as a cap meter by charging the capacitor through a resistor and measuring the **time constant** (how fast the voltage rises). Using the formula **t = rc,**$Vt = Vmax(1 - e^{-t/rc}$**),** the Arduino calculates the capacitance. This method depends on the known resistor value and the timing accuracy of the Arduino.

-Here How the Capacitance Meter Works:

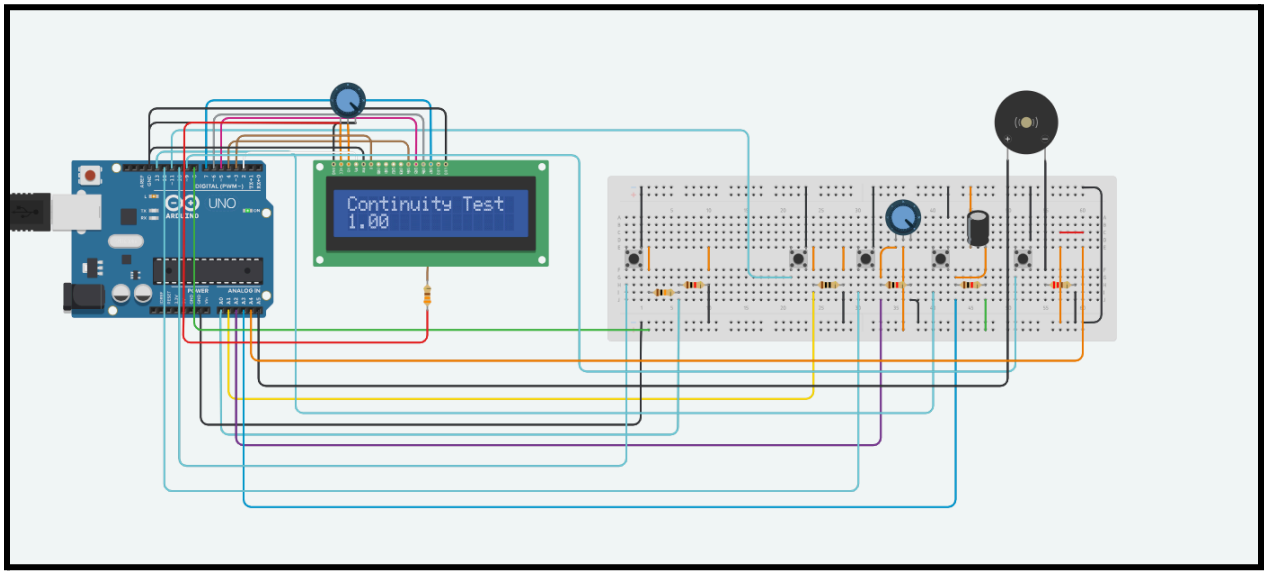
1. **Discharges** the capacitor (sets pin A3 LOW).

2. **Starts charging** the capacitor through a resistor when VCC is set HIGH.

3. Measures the **time** it takes for voltage at pin A3 to reach ~63.2% of 5V (≈3.16V, or ADC value 646).

4. Uses the RC time constant formula:
   $Vt = Vmax(1 - e^{-t/rc}$**)**
5. Solves for capacitance:
   $C = t/R$ (Here, R is assumed known **=1KΩ**)

E)Continuity test:



-The continuity test is a simple yet essential method used to check if there is a complete electrical connection between two points in a circuit. It works by connecting two probes to the points being tested; if the circuit is closed. This technique is commonly used to detect broken wires, faulty connections, or damaged circuit paths.When the circuit is closed, or a buzzer sounds to indicate a successful connection.

-Here How the Continuity Tester Works:

1. **Reads** pin `A4` (defined as `probe`) to detect electrical connection.

2. If the pin reads **LOW**, it means the circuit is closed (i.e., there's continuity).

3. Then it:

    - **Activates a buzzer** using `tone()` on pin `A5`

    - **Displays** “1” on the LCD
4. If the pin reads **HIGH** (open circuit), the buzzer is off and “0” is shown.

3.Code

```ino
#include <LiquidCrystal.h>

#define Voltage_Divder_Volt A0
#define ammeterVolt A1
#define readed  A2
#define capacitor  A3
#define probe A4
#define buzzer_sound A5


#define RS 2
#define EN 3
#define D4 4
#define D5 5
#define D6 6
#define D7 7

LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);


//Resistors
#define Shunt_Resistance 10
#define Series_Resistance  1
#define Ammeter_Resistance 0.01

int switch5 = 9;
int Switch1 = 10;
int Switch2 = 11;
int Switch3 = 12;
int Switch4 = 13;
int VCC = 8 ;
```

```ino
int Active_Device = 0 ;


int Pressed_Button() {
  for (int i = 9 ; i < 14 ; i++) {
    if (digitalRead(i) == LOW) {

      return i;
    }
  }
  return -1;
}

void setup() {
  Serial.begin(9600);
  pinMode(Voltage_Divder_Volt, INPUT);
  pinMode(ammeterVolt, INPUT);
  pinMode(readed, INPUT);
  pinMode(Switch1, INPUT_PULLUP);
  pinMode(Switch2, INPUT_PULLUP);
  pinMode(Switch3, INPUT_PULLUP);
  pinMode(Switch4, INPUT_PULLUP);
  pinMode(switch5, INPUT_PULLUP);
  pinMode(probe, INPUT_PULLUP);
  pinMode(buzzer_sound, OUTPUT);
  pinMode(VCC , OUTPUT) ;
  digitalWrite(VCC , LOW );
  lcd.begin(16, 2);


}
```

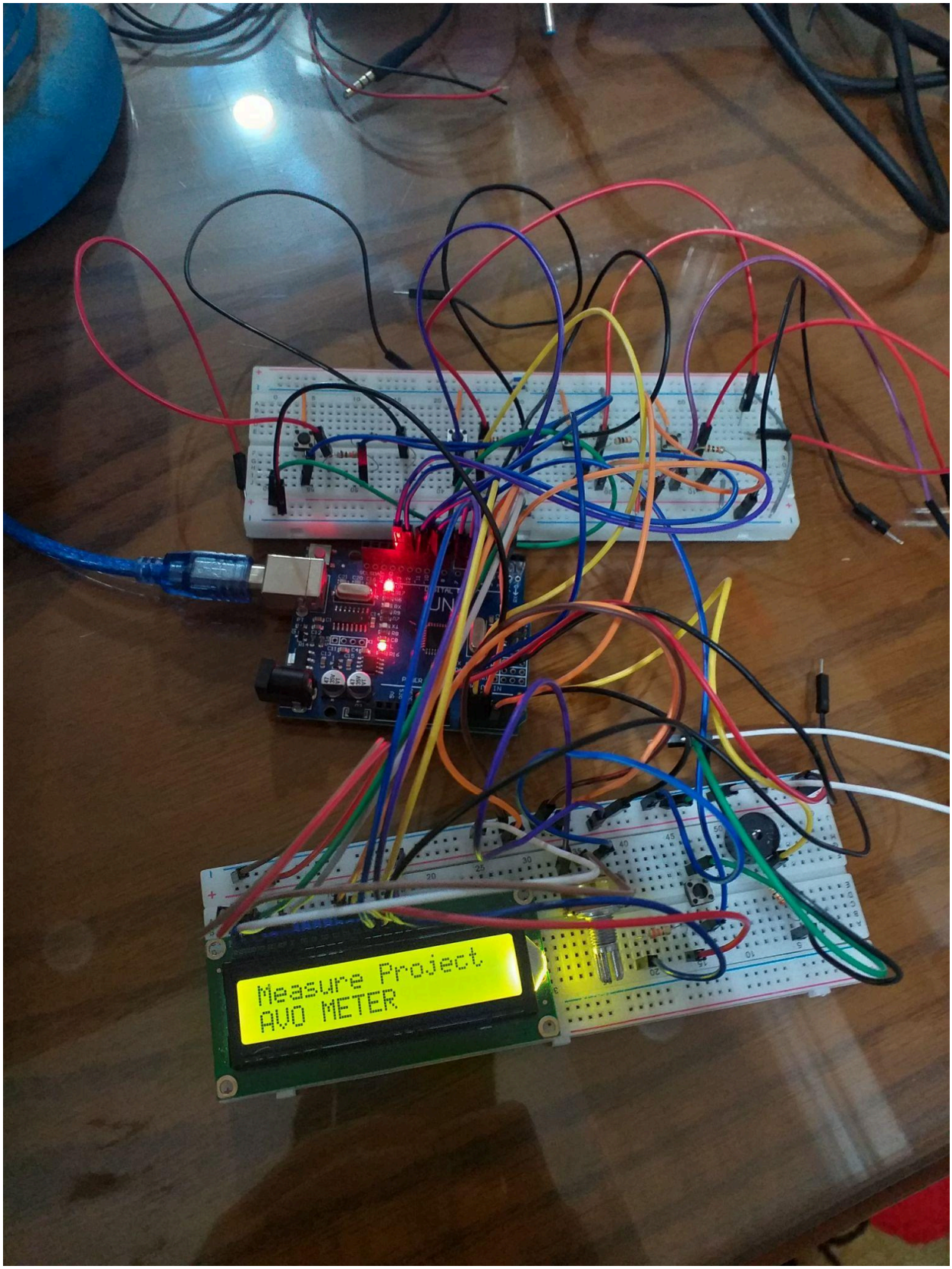
```ino
void Print_Reading(String Device , float Reading , String Unit , int Fraction_Num = 2 ) {
  lcd.clear();  // Clear previous output
  lcd.setCursor(0, 0);
  lcd.print(Device);

  lcd.setCursor(0, 1);
  lcd.print(Reading, Fraction_Num);  // Print float with 2 decimal places
  lcd.print(" ");
  lcd.print(Unit);
}

void WelcomeMessage(){
  lcd.clear();  // Clear previous output
  lcd.setCursor(0, 0);
  lcd.print("Measure Project");
  lcd.setCursor(0, 1);
  lcd.print("AVO METER");

}


void voltmeter() {
  float voltOut = (analogRead(Voltage_Divder_Volt) * 5.0)/1023;
  float voltIn = voltOut * (Shunt_Resistance + Series_Resistance) / Series_Resistance;
  Serial.print("The voltage = ");
  Serial.print(voltIn);
  Serial.println(" V");
  Print_Reading("DC Voltameter" , voltIn ,"V");
}
```

```
void ammeter() {
  int val = analogRead(ammeterVolt);
  float voltOut = (val * 5.0)/1023;
  float current = (voltOut / Ammeter_Resistance);
  Serial.print("The Current = ");
  Serial.print(current);
  Serial.println(" mA");
  Print_Reading("DC Ammeter" , current ,"mA");

}

void ohmmeter() {
  digitalWrite( VCC , HIGH);
  int val = analogRead(readed);
  float volt = (val * 5.0) / 1023.0;
  float volt_across_unknown = 5.0 - volt;
  float current = volt / 1000.0;
  float res = volt_across_unknown / current;
  Serial.print("resistance = ");
  Serial.print(res);
  Serial.println(" ohm");
  digitalWrite( VCC , LOW);
  if (res > 1000) {
    Print_Reading("Ohm-meter" , res/1000 ,"Kohm");
  }
  else {
    Print_Reading("Ohm-meter" , res ,"Ohm");
  }

}
```

```
void capacitance_meter() {
  float cap = 0;
  long start_time = 0;
  long time_interval = 0;

  pinMode(capacitor, OUTPUT);
  digitalWrite(capacitor, LOW);
  delay(500);

  pinMode(capacitor, INPUT);
  start_time = micros();
  digitalWrite(VCC, HIGH);
  while (analogRead(capacitor) < 646) {}

  time_interval = micros() - start_time;
  cap = (float)time_interval / 1000.0;

  Serial.print("capacitance = ");
  Serial.print(cap);
  Serial.println(" uF");
  digitalWrite(VCC, LOW);
  Print_Reading("Capacitance-meter" , cap ," uF");
  delay(200);

}


void buzzer()
{
  int state = 0;
  state = digitalRead(probe);
```

```
  if (state == LOW) {
    tone(buzzer_sound, 985);
    Print_Reading("Continuity Test" , 1 ," ");

  }
  else {noTone(buzzer_sound);Print_Reading("Continuity Test" , 0 ," ");}
  delay(100);
}



void loop() {

int newPressed = Pressed_Button();
if (newPressed != -1 && newPressed != Active_Device) {
  Active_Device = newPressed;
}



  switch (Active_Device) {
    case 9:
      Serial.println("Now buzzer");
      buzzer();
      break;

    case 10:
      noTone(buzzer_sound);
      Serial.println("Now Voltmeter");
      voltmeter();
      break;
```

```
189
190     case 11:
191       noTone(buzzer_sound);
192       Serial.println("Now Ammeter");
193       ammeter();
194       break;
195
196     case 12:
197       noTone(buzzer_sound);
198       Serial.println("Now Ohmmeter");
199       ohmmeter();
200       break;
201
202     case 13:
203       noTone(buzzer_sound);
204       Serial.println("Now Capacitance");
205       capacitance_meter();
206       break;
207
208     default:
209       WelcomeMessage();
210       noTone(buzzer_sound);
211       break;
212   }
213
214   delay(50) ;
215 }
216
```


Measure Project
AVO METER