NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

- **GROUP-3**

- **TOPIC:- ELECTRONIC DATA ACQUISITION FROM A POTENTIOMETER RANGING FROM 10 OHM TO 1 MEGA OHM.**

- **GROUP MEMBERS :**

Sl. NO	NAME OF MEMBERS	ENROLLMENTNO
1	Manjesh Kumar	23UEI144
2	Nishant Kumar karn	23UEI133
3	Nikhil Singh Dhruwe	23UEI142
4	Aryan Giri	23UEI131
5	Anjali Patel	23UEI148
6	Pranjal Pal	23UEI215

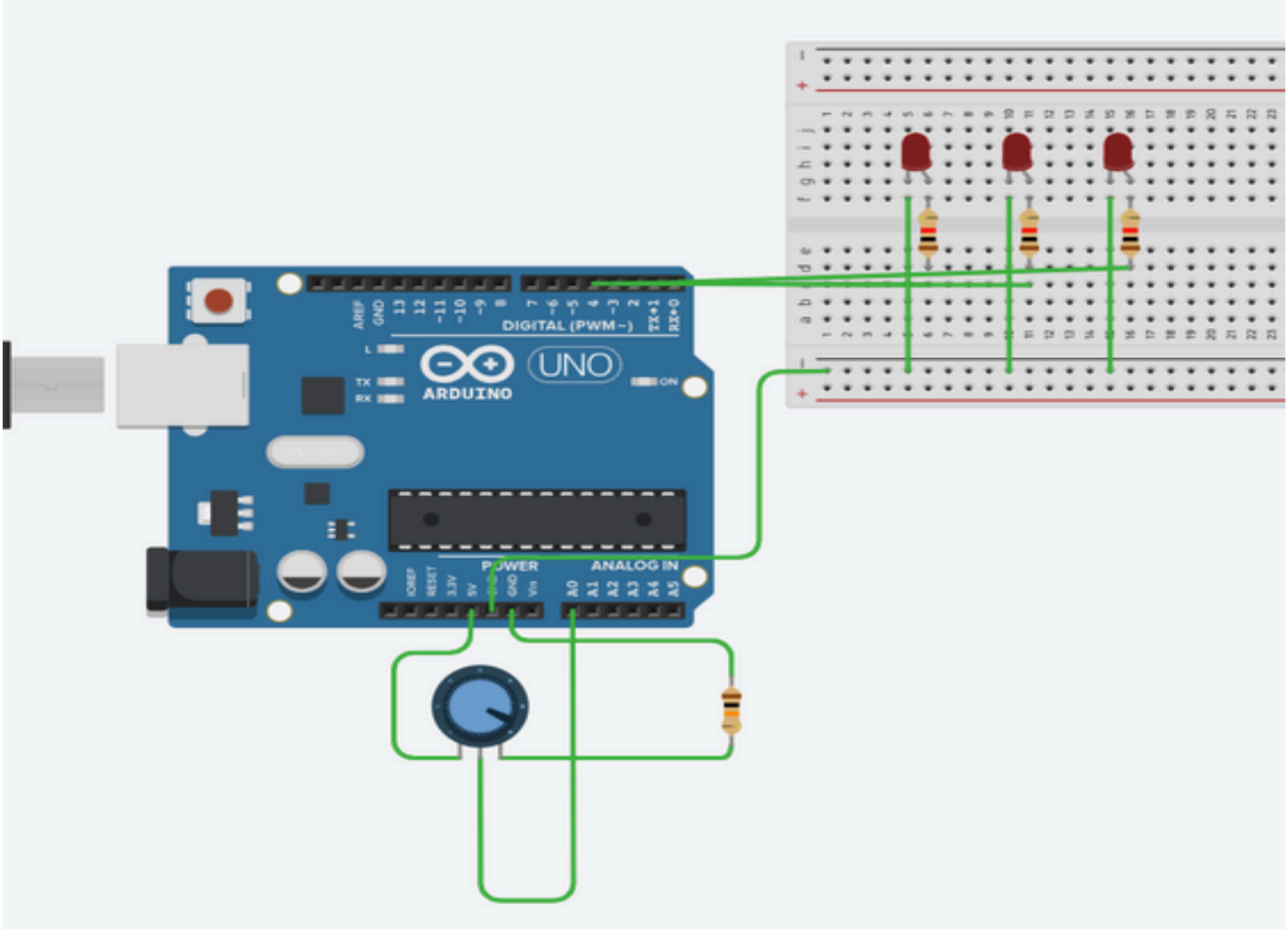
- **OBJECTIVE :**

The objective of this project is to measure the resistance of a potentiometer using an Arduino and display the results through LEDs and serial communication. By reading analog input from the potentiometer, calculating output voltage, and applying the voltage divider formula, the Arduino determines the potentiometer's resistance in ohms, kilo-ohms, or mega-ohms. Different LEDs indicate the resistance range, providing a visual, real-time representation of the potentiometer's resistance value.

- **Apparatus and quantity :**

1. Resistor (1PC)
2. Bread board (1PC)
3. Wire (as required)
4. Led(3PC)
5. Arduino Uno (1PC)
6. Potentiometer(1PC)

- **Circuit diagram :**



- **Working process :**

This project operates on the principle of a voltage divider circuit combined with analog-to-digital conversion (ADC). The core components include a potentiometer, a fixed resistor, an Arduino microcontroller, and an LCD display. By using a voltage divider, we can indirectly measure the resistance of the potentiometer by observing the changes in output voltage.

Step-by-Step Breakdown

1. **Voltage Divider Circuit**: In a voltage divider circuit consisting of a potentiometer in series with a fixed resistor, the voltage at the junction between the two components depends on the price of loresistance of the potentiometer. As the potentiometer is adjusted, its resistance changes, which in turn causes the voltage at the midpoint to vary proportionally.
2. **Reading Voltage with Arduino**: The Arduino's analog input pin is used to measure the midpoint voltage. It can detect voltages in the range from 0V to 5V (or the reference voltage used), converting this analog value into a digital reading. This process is carried out by the Analog-to-Digital Converter (ADC), which samples the continuous voltage range and converts it into discrete digital values. For example, with a 10-bit ADC, the voltage is represented by a value between 0 and 1023.
3. **Calculating the Resistance**: The Arduino uses the ADC reading to determine the potentiometer's resistance. Since the fixed resistor's value is known, we can use Ohm's Law and the voltage divider formula to solve for theunknown resistance:

$$V_{\text{out}} = V_{\text{in}} \times \frac{R_{\text{pot}}}{R_{\text{fixed}} + R_{\text{pot}}}$$

Rearranging this equation allows us to calculate R(pot) (the potentiometer'sresistance) based on Vout, Vin, and R(fixed).

4.**Displaying the Result**: Once the resistance of the potentiometer is calculated, the Arduino displays it on the LCD screen in units of ohms or kilo-ohms, providing real-time feedback to the user. This setup allows users tosee changes in resistance as they adjust the potentiometer.

- ## Hardware Setup:

Potentiometer (variable resistor):

The potentiometer is connected to the **analog input pin A0**.

The potentiometer's wiper (middle terminal) is used to read varying voltages corresponding to changes in resistance.

Fixed Resistor:

A fixed resistor (10kΩ) is used in a voltage divider configuration with the potentiometer to calculate the potentiometer's resistance.

LED's :

Three LED's are connected to digital pins 4, 10, and 11. These LED's are used to indicate the range of the resistance value (ohms, kilo-ohms, mega-ohms).

- ### Serial Communication:

The serial monitor displays the calculated resistance, allowing you to monitor the resistance value in real-time.

- ## Software Logic:

Initialization (setup()):

Serial.begin(9600): Initializes serial communication with a baud rate of 9600.

pinMode(): Configures the LED pins as outputs, so they can be controlled to light up based on the resistance values.

Reading the Potentiometer Value (loop()):

analogRead(analogPin): Reads the raw value from the potentiometer on **pin A0**. The value returned is between **0 and 1023** (10-bit ADC range).

Vout: Calculates the output voltage from the potentiometer based on the analog reading. This voltage is a fraction of the supply voltage (Vin = 5.0V).

Resistance Calculation: Using the voltage divider formula, the resistance of the potentiometer is calculated as:

$$V_{\text{out}} = V_{\text{in}} \times \frac{R_{\text{pot}}}{R_{\text{fixed}} + R_{\text{pot}}}$$

where:

R(fixed)is the fixed resistor (10kΩ).

V(out)is the output voltage from the potentiometer.

Vin is the input voltage (5V).

- **Displaying Resistance on the Serial Monitor:**

Based on the calculated resistance value (Rpot), the program:

Prints the resistance in **Ohms** if it is less than 1000Ω.

Prints the resistance in **Kilo-ohms** (kΩ) if it is between 1000Ω and 1,000,000Ω.

Prints the resistance in **Mega-ohms** (MΩ) if it exceeds 1,000,000Ω.

Controlling the LED's:

Depending on the range of the resistance (Rpot), the corresponding LED is turned on:

LED for Ohms: Lights up if the resistance is less than 1000Ω.

LED for Kilo-ohms (kΩ): Lights up if the resistance is between 1000Ω and 1,000,000Ω.

LED for Mega-ohms (MΩ): Lights up if the resistance exceeds 1,000,000Ω.

Other LEDs are turned off to ensure only the corresponding range LED is active.

- **Delay (delay(500)):**

Adds a half-second delay between readings to make the output stable and not update too rapidly in the Serial Monitor.

- ## Application

This principle demonstrates the concept of indirect measurement, where a quantity(resistance) is determined by observing related changes in voltage. It showcases basic electronics principles (Ohm’s Law, voltage dividers) and microcontroller interfacing (ADC, display output), making it an ideal introductory project for learningabout circuit analysis, Arduino programming, and display interfacing.