Assignment 1

Shivam Bansal (240976)

December 13, 2025

1 Task A - Voltage Divider Analysis

The objective of this task is to understand how a voltage divider works and how a digital multimeter measures voltage internally.

1.1 Purpose of a voltage divider in measurement systems

A voltage divider is a linear circuit block that produces an output that is a fraction of its input voltage. It is used to scale down higher voltages to a safe and measurable level for electronic circuits and ADCs.

In measurement systems, it allows accurate voltage sensing without damaging internal components while maintaining proportionality to the original input voltage.

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

1.2 ADC reading-to-voltage conversion formula

- Arduino Uno has a 10-bit ADC
- ADC range: $0 - 1023$
- Reference voltage: 5V

Conversion Formula:

$$V_{measured} = \frac{ADC\ Value\ \times 5.0}{1023}$$

R_1 - R_2	**Theoretical Value (Volts)**	**ADC Value**	**Tinkercad Value (Volts)**
10 kΩ - 10 kΩ	2.5	511	2.5
4.7 kΩ - 10 kΩ	3.4	696	3.4
1 kΩ - 15 kΩ	4.69	959	4.69

Table 1: Comparison of Theoretical and Tinkercad Voltage Values

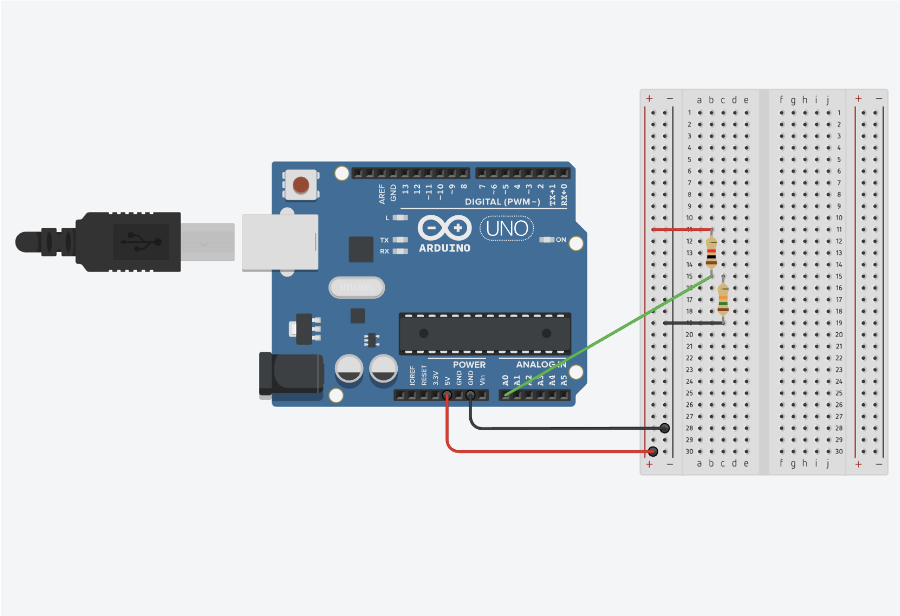

Figure 1: Voltage Divider Circuit

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int adcValue = analogRead(A0);
  float voltage = adcValue * (5.0 / 1023.0);

  Serial.print("ADC Value: ");
  Serial.print(adcValue);
  Serial.print(" | Voltage: ");
  Serial.println(voltage);

  delay(1000);
}
```

Serial Monitor

```
ADC Value: 511 | Voltage: 2.50
ADC Value: 511 | Voltage: 2.50
ADC Value: 696 | Voltage: 3.40
ADC Value: 696 | Voltage: 3.40
ADC Value: 696 | Voltage: 3.40
ADC Value: 959 | Voltage: 4.69
ADC Value: 959 | Voltage: 4.69
```

Figure 2: Code snippet and output

1.3 Observations

Minor deviations between theoretical and measured voltages were observed due to ADC quantization and rounding errors. Small fluctuations in ADC readings were noticed, which are normal in digital measurements. Overall, the measured values closely matched the theoretical calculations.

2 Task B - Time Constant Analysis

The objective of this task is to measure an unknown capacitance by observing the RC charging time using an Arduino Uno.

2.1 RC Time Constant τ

The time constant is defined as:

$$\tau = R \times C$$

The capacitor voltage during charging is:

$$V(t) = V_{max}(1 - e^{-\frac{t}{\tau}})$$

At time t = τ

$$V(t) = 0.632 \times V_{max}$$

Hence, measuring the time to reach 63% of 5V (3.15V) allows us to calculate capacitance.

2.2 Expected time vs Measured time

	Theoretical C (μF)	**Theoretical time ($\tau = RC$)**	**Measured Time (t) (μs)**	**C = $\frac{t}{R}$ (μF)**
R = 10 kΩ	100 μF	1000000	1004532	100.45
	10 μF	100000	100488	10.05

2.3 Potential error sources

Potential errors arise due to component tolerances of resistors and capacitors, limited 10-bit ADC resolution causing voltage quantization, electrical noise near the threshold voltage, timing inaccuracies of the micros() function, and non-ideal capacitor behavior such as leakage.

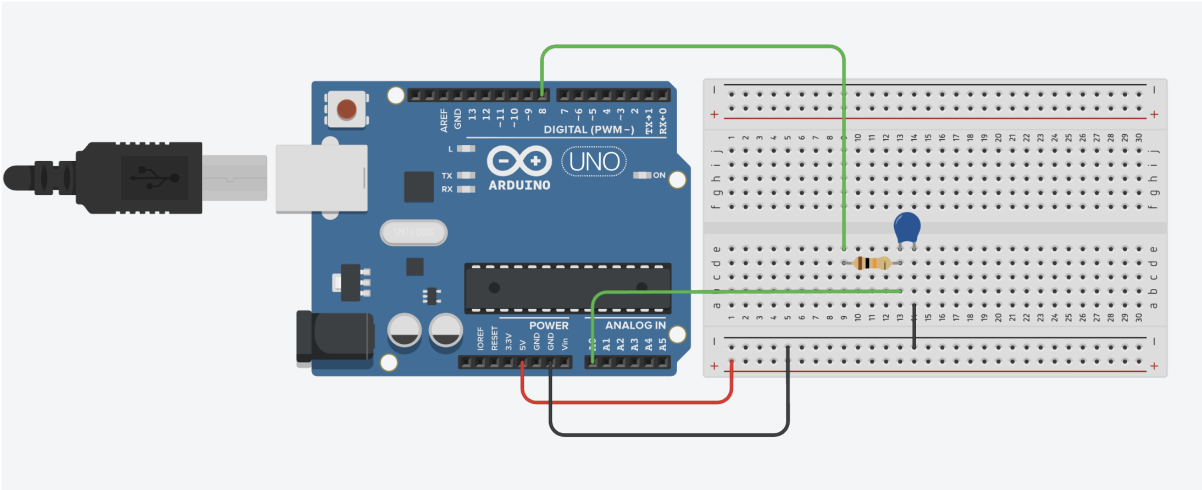

Figure 3: Circuit

```
const int chargePin = 8;
const int analogPin = A0;
const int resistorValue = 10000; // 10kΩ

void setup() {
  Serial.begin(9600);
  pinMode(chargePin, OUTPUT);
}

void loop() {
  // Discharge capacitor
 digitalWrite(chargePin, LOW);
  delay(5000);

  // Start charging
  pinMode(chargePin, OUTPUT);
  digitalWrite(chargePin, HIGH);

  unsigned long startTime = micros();
  int adcValue = 0;
  // Wait until capacitor reaches ~63% of 5V
  while (adcValue < 647) { // 63% of 1023 ≈ 647
    adcValue = analogRead(analogPin);
  }

  unsigned long elapsedTime = micros() - startTime;

  // Calculate capacitance
  float capacitance = (float)elapsedTime / resistorValue;

  Serial.print("Time (microseconds): ");
  Serial.print(elapsedTime);
  Serial.print(" | Capacitance (uF): ");
  Serial.println(capacitance);

  delay(100);
}
```

Serial Monitor

```
Time (microseconds): 1004532 | Capacitance (uF): 100.45
Time (microseconds): 100488 | Capacitance (uF): 10.05
```

Figure 4: Code Snippet and output

3 Task C - Beginner Ohmmeter Prototype

The objective of this task is to design a basic ohmmeter using an Arduino Uno by applying the voltage divider principle to calculate an unknown resistance.

3.1 Step-by-step calculations

Resistance cannot be measured directly. Instead:

- A known reference resistor is used
- A known voltage is applied
- The resulting voltage drop is measured
- The unknown resistance is calculated mathematically

Voltage divider relation:

$$V_{out} = V_{in} \cdot \frac{R_{unknown}}{R_{ref} + R_{unknown}}$$

Rearranging to calculate unknown resistance:

$$R_{unknown} = R_{ref} \cdot \frac{V_{out}}{V_{in} - V_{out}}$$

3.2 Comparison of actual vs measured values

$R_{unknown}$ Actual	V_{out}	ADC Value	$R_{unknown}$ Calculated
10 Ω	0.01	2	9.79 Ω
5 kΩ	2.5	511	4.99 kΩ
100 kΩ	4.76	974	99.39 kΩ

Table 2: Actual vs. Measured values

3.3 Reflection on measurement uncertainty

The measured resistance values showed small deviations from the actual resistor values due to component tolerances, ADC resolution limits, and voltage measurement noise. The uncertainty increased when the unknown resistance was much higher or lower than the reference resistor, reducing measurement sensitivity.

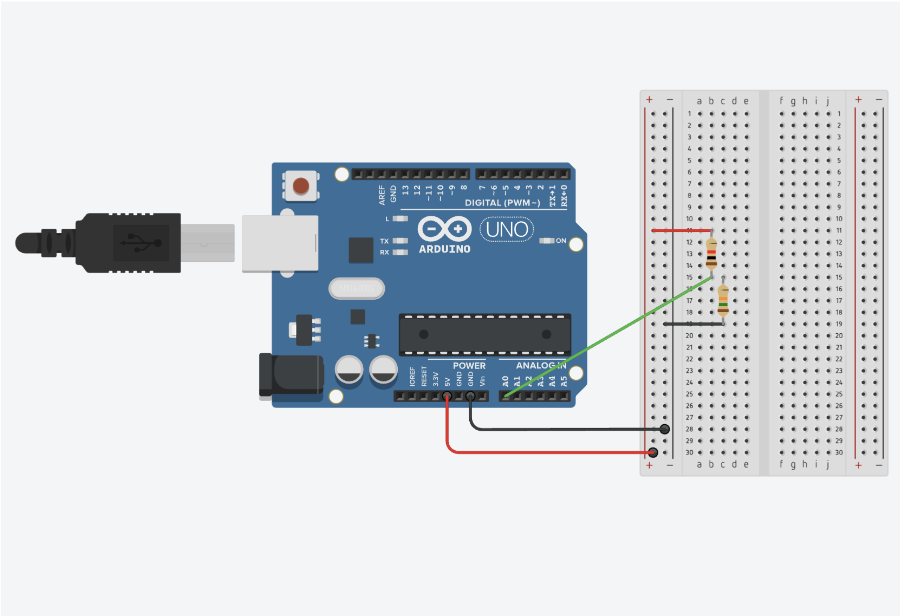

Figure 5: Circuit

```
const int analogPin = A0;
const float referenceVoltage = 5.0;
const float referenceResistor = 5000.0; // 5kΩ

void setup() {
  Serial.begin(9600);
}

void loop() {
  int adcValue = analogRead(analogPin);
  float vOut = (adcValue * referenceVoltage) / 1023.0;

  float unknownResistor = referenceResistor * (vOut / (referenceVoltage - vOut));

  Serial.print("ADC Value: ");
  Serial.print(adcValue);
  Serial.print(" | Vout: ");
  Serial.print(vOut);
  Serial.print(" V | Unknown Resistance: ");
  Serial.print(unknownResistor);
  Serial.println(" ohms");

  delay(3000);
}
```

Serial Monitor

```
ADC Value: 2 | Vout: 0.01 V | Unknown Resistance: 9.79 ohms
ADC Value: 511 | Vout: 2.50 V | Unknown Resistance: 4990.23 ohms
ADC Value: 974 | Vout: 4.76 V | Unknown Resistance: 99387.86 ohms
```

Figure 6: Code