Assignment 1

Name: Parth Dhamija   Roll No: 240733

1 Task A: Voltage Divider Analysis and Measurement Module

1.1 Voltage Divider

A voltage divider is used to scale down a high input voltage to a lower value so it can be used by the arduino. The analog pins can only read values b/w 0-5V so if the input is higher than this range, we can't connect it directly. Hence a voltage divider is used

The output voltage of the voltage divider is given by:

$$V_{\text{out}} = V_{\text{in}} \times \frac{R_2}{R_1 + R_2} \tag{1}$$

The Arduino has a built-in Analog-to-Digital Converter (ADC) which converts an analog voltage into a digital value. The Arduino uses a 10-bit ADC, which maps voltages from 0 V to 5 V into digital values ranging from 0 to 1023(2 raised to power 10=1024).

The voltage corresponding to the ADC reading is calculated using:

$$V = \text{ADC value} \times \frac{5.0}{1023.0} \tag{2}$$

1.2 Circuit design and code

Figure 2 shows the voltage divider circuit(left) and code(right) used in Task A. The circuit was built and simulated using Tinkercad. The divided voltage is connected to analog pin A0 of the Arduino. This specific configuration had a 4.7 and 10kilo-ohm resistors respectively

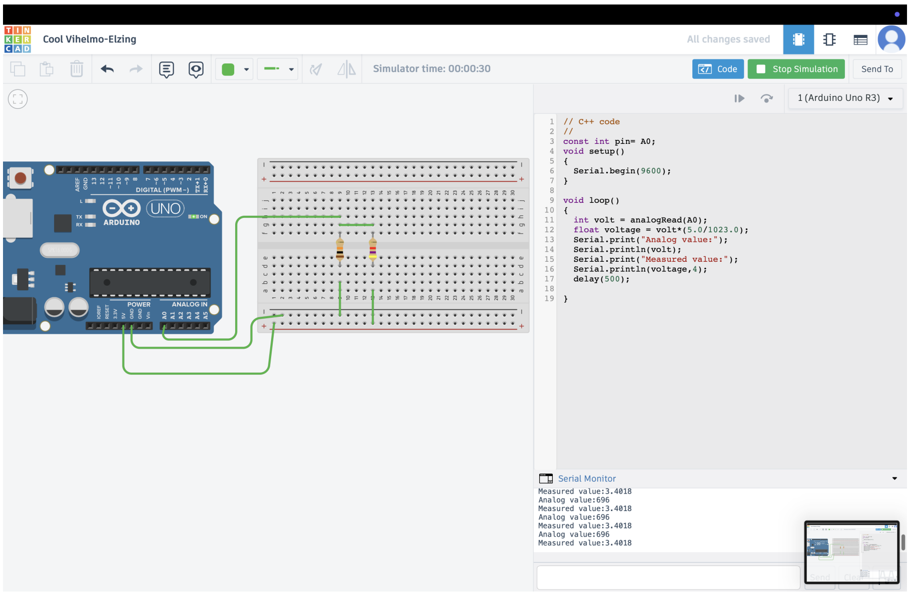

Figure 1: Voltage divider circuit and code for Task A

1.3 Comparison of Results

Vin (V)	R_1 (kΩ)	R_2 (kΩ)	Theoretical V_{out} (V)	Measured V_{out} (V)
5.0	10	10	2.5000	2.4976
5.0	4.7	10	3.4013	3.4018
5.0	1	15	4.6875	4.6872

Table 1: Comparison of theoretical and measured output voltages

1.4 Observations

The measured value is slightly different from the theoretical value due to the limited resolution of the Arduino ADC. The ADC divides the 0–5 V range into discrete steps, so rounding occurs during analog-to-digital conversion. When this rounded value is converted back to voltage, a small error appears. The input voltage was kept within 0–5 V since this is the valid measurement range of the Arduino analog pin.

2 Task B: Capacitance Measurement Using RC Time Constant

2.1 Task

The objective of this task is to measure the value of an unknown capacitor using an RC charging circuit. The Arduino is used to measure the time taken by the capacitor to charge up to 63% of the supply voltage and then calculate the capacitance using the RC time constant formula.

2.2 RC Time Constant and Significance of 63%

The time constant is defined as the product of resistance and capacitance connected in series:

$$\tau = R \times C \tag{3}$$

It represents how fast a capacitor charges or discharges in an RC circuit. When a capacitor is charged through a resistor from a DC voltage source, the voltage across the capacitor increases in a exponential manner

After one time constant(delta.t=R.C), the capacitor voltage reaches approximately 63% of its final value. This point is important because it provides a fixed reference to measure time and calculate the capacitance accurately, independent of the input voltage.

2.3 Circuit and code

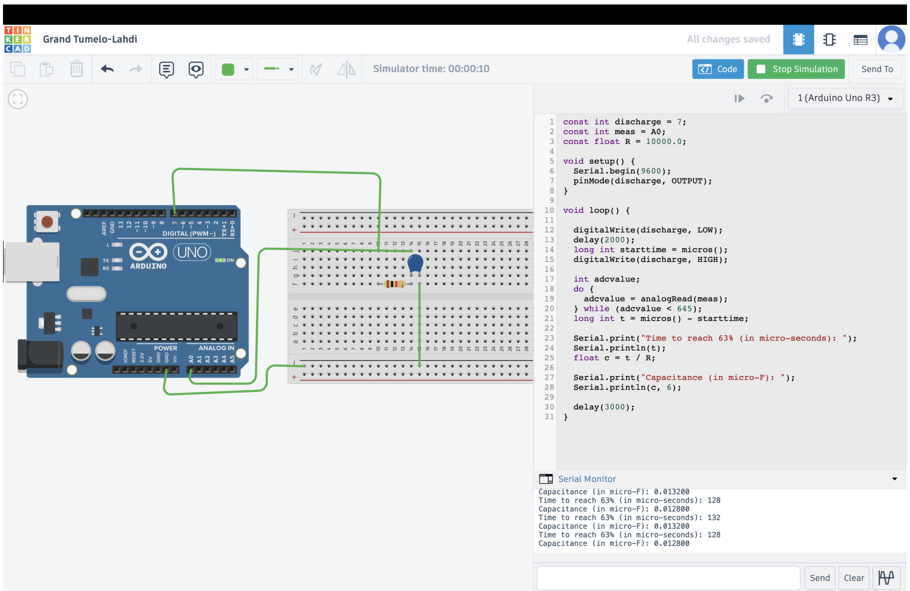

Figure 2: RC series circuit and code for Task B

2.4 Methodology

Firstly, the capacitor is fully discharged by setting the digital pin LOW for a short duration. After this, the digital pin is set HIGH to begin charging. The Arduino continuously monitors the capacitor voltage using the analog pin.

The time at which charging starts is recorded using the `micros()` function. When the capacitor voltage reaches 63% of the supply voltage, the time is recorded again. The difference between these two times gives the charging time corresponding to one time constant.

The capacitance is calculated using:

$$C = \frac{t_{63}}{R} \tag{4}$$

2.5 Measured vs Theoretical Values

Resistance (kΩ)	Capacitance (Theoretical) (μF)	t_{63} Expected (ms)	t_{63} Measured (ms)	Capacitance Measured (μF)
10	1	10000	10028	1.0028
10	100	1000000	999300	99.9300
10	50	500000	499752	49.9752

Table 2: Comparison of expected and measured t_{63} and capacitance values

2.6 Error

Difference(not significant) was observed between measured and expected values. These errors mainly occur due to tolerance in resistor and capacitor values. The limited resolution of the Arduino ADC also affects the accuracy of detecting the exact 63% voltage level. Incomplete discharge of the capacitor and minor electrical noise can further introduce small variations in the measured time.

3 Task C: Beginner Ohmmeter Prototype

This task is an extension of the voltage divider circuit used in Task A. Instead of measuring the output voltage, the same voltage divider is used to calculate an unknown resistance using a known reference resistor.

For a voltage divider circuit, the output voltage is given by:

$$V_{out} = V_{in} \times \frac{R_x}{R_{ref} + R_x} \tag{5}$$

Rearranging the equation to find the unknown resistance:

$$R_x = R_{ref} \times \frac{V_{out}}{V_{in} - V_{out}} \tag{6}$$

The output voltage is calculated from the Arduino ADC reading using the same method as in Task A.

3.1 Comparison of Actual and Measured Values

Reference Resistance (Ω)	Actual R_x (Ω)	Measured R_x (Ω)
10000	10000	9980.47
10000	30000	29960.94
5000	4000	4005.28

Table 3: Comparison of actual and measured resistance values

3.2 Circuit and Code

The circuit used in this task is more or less the same voltage divider circuit as in Task A, with a known reference resistor and an unknown resistor. The voltage at the midpoint is measured using the Arduino analog pin to calculate the unknown resistance.

Figure 3 shows the circuit diagram used for the ohmmeter prototype and the Arduino code used to calculate the unknown resistance.

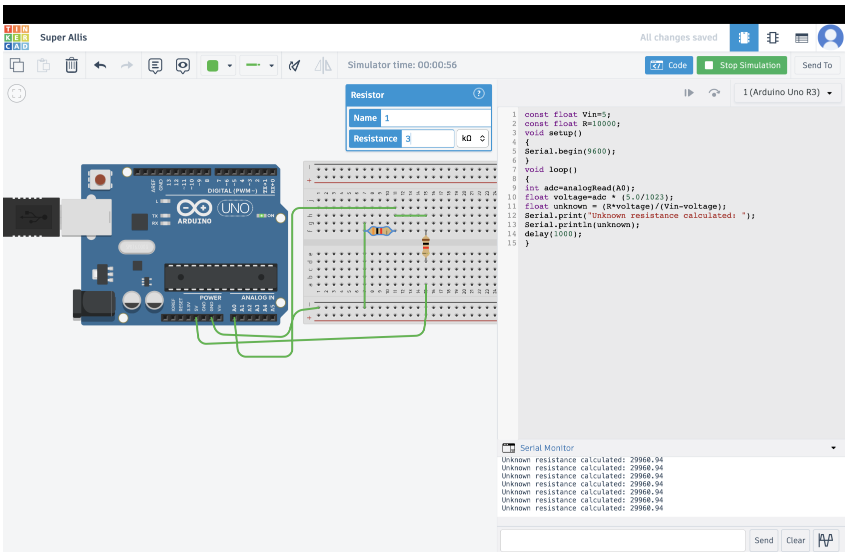

Figure 3: Voltage divider circuit and code for Task C

3.3 Measurement Uncertainty

The measured resistance values are close to the actual values, with small differences due to resistor tolerance, ADC resolution, and minor noise in the circuit. These error sources are similar to those observed in Task A.