ASSIGNMENT 1

SMART MULTIMETER USING MICROCONTROLLER SYSTEMS

BY ISHWIN KUMAR
BTECH(ELECTRICAL ENGINEERING)
ROLL NO: 240476

INTRODUCTION

Through this assignment, we use Arduino to find resistance, capacitance and voltages in simple RC circuits. Through a series of tasks, we determined capacitance and resistance using simple voltage divider and time constant equations.

Main objectives of this assignment are-

1) To verify the voltage divider rule experimentally using different resistor combinations.

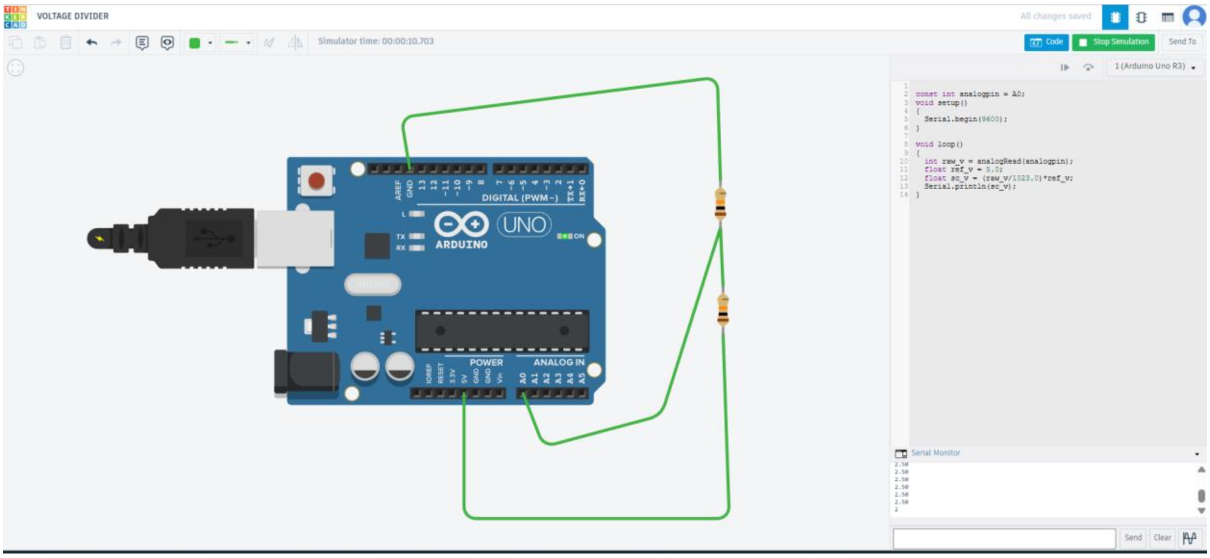
2) To understand Arduino's Analog-to-Digital Converter.

3) To determine time constant of an RC circuit.

4) To calculate unknown resistance using voltage divider.

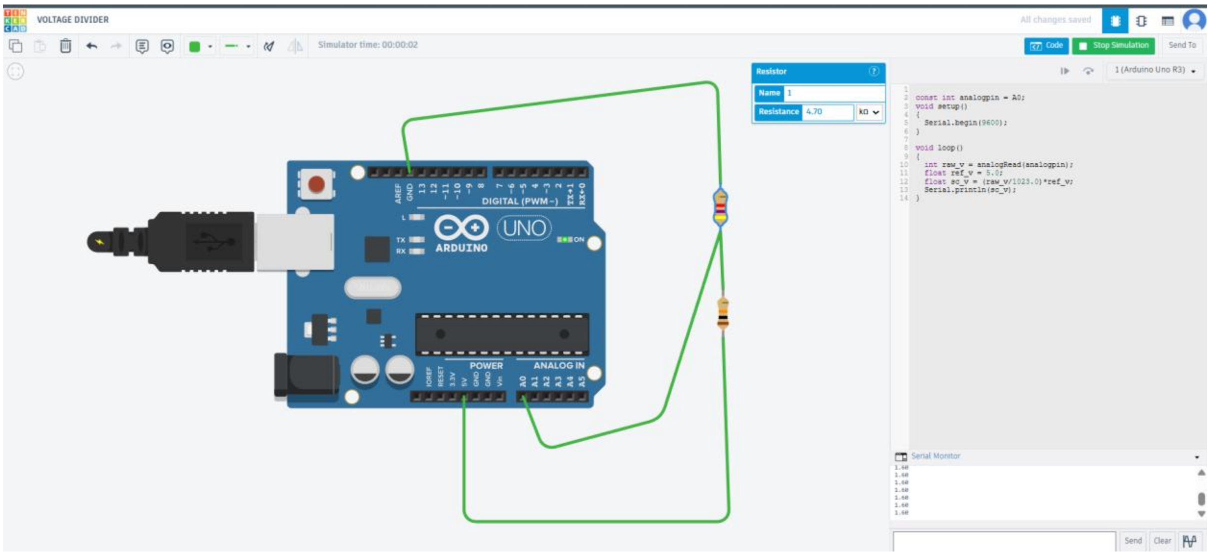
(TASK A)

1) 10kΩ–10kΩ:



Experimental voltage measurer	Theoretical voltage measurer = Vin × (R2 / (R1 + R2))
2.5V	2.5V

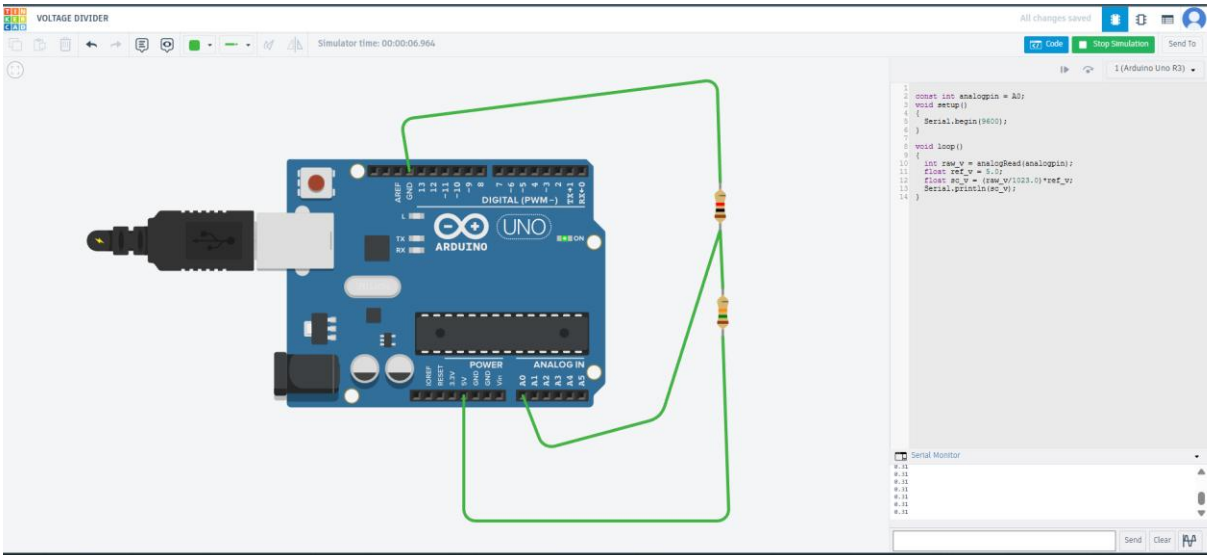
2) 10kΩ–4.7kΩ:



Experimental voltage measurer	Theoretical voltage measurer = Vin × (R2 / (R1 + R2))
1.60 V	= 5.0 (4.7/14.7) = 1.598 V

3) 1kΩ–15kΩ:

Experimental voltage measurer	Theoretical voltage measurer = Vin × (R2 / (R1 + R2))
0.31 V	= 5.0 (1/16) = 0.3125 V



Arduino uses a 10-bit ADC, having ADC range: 0–1023 but we need Voltage range: 0–5V

Conversion Formula-

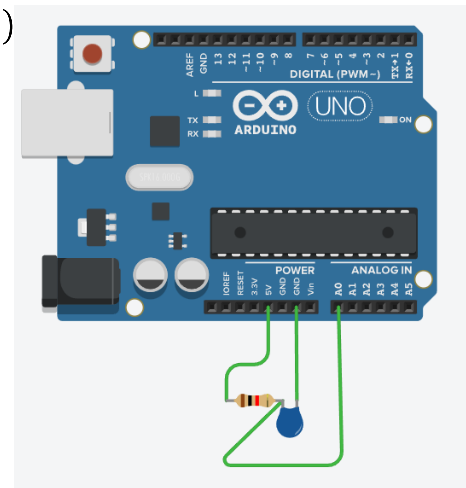
$$Voltage = ADC\_value \times \frac{5.0}{1023.0}$$

Measurement Error

- Minor differences between theoretical and measured values occur due to:
- ADC resolution limits (±1–2 counts)

(TASK B)

$$V(t) = V_{cc}\left(1 - e^{-t/RC}\right)$$

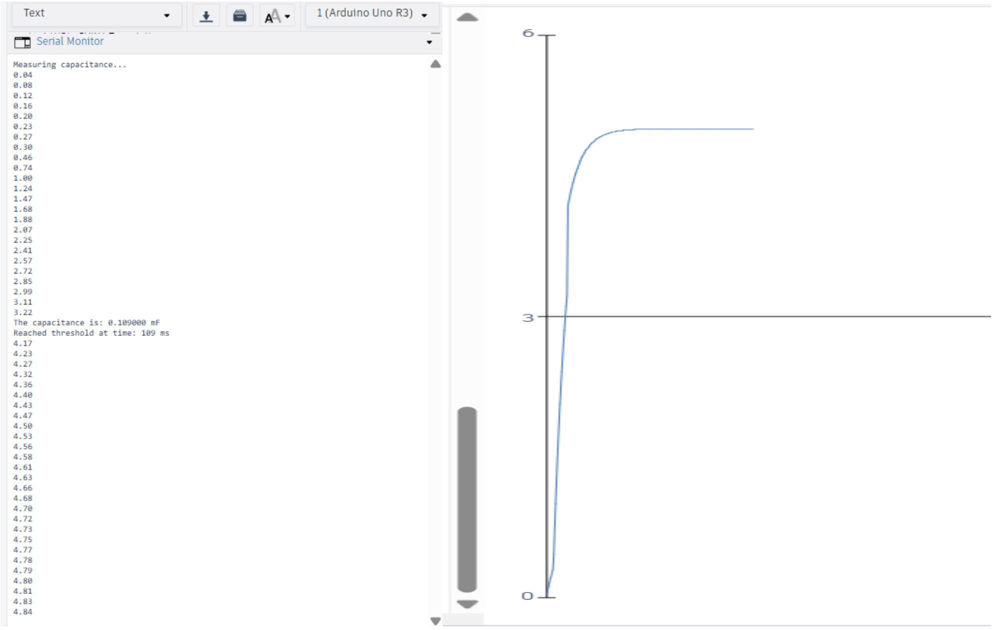


At $t = RC$ $$V(t) = 0.63 \times V_{cc}$$

Why 63% Is Significant

- At one time constant **τ = RC**, the capacitor reaches 63% of its final voltage
- This gives a **direct and simple way** to calculate capacitance:

$$C = \frac{t}{R}$$



OUTPUT:

The output presents the voltage of capacitor in its charging phase at each milli second (the least count of Arduino time). Just as the voltage across capacitor crosses 63% of the Arduino provided input voltage, the elapsed time which is nothing but the time constant is shown on the screen. Using that, the capacitance is calculated.

CODE:

```
const int analo = A0;
float source = 5.0;

unsigned long starttime = 0;
bool measuring = false;
unsigned long time_elapsed;

float R = 1000.0;  // 1k resistor

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

  measuring = true;
  starttime = millis();

  Serial.println("Measuring capacitance...");
}

void loop() {

  int raw = analogRead(analo);
  float A0_v = (raw / 1023.0) * source;

  Serial.println(A0_v);  // show the voltage rising

  float threshold = 0.63 * source;   // 63% of 5V = 3.15 V

  if (measuring && A0_v >= threshold) {

    time_elapsed = millis() - starttime;
    measuring = false;

    float cap = (float)time_elapsed / R;   // C = t / R

    Serial.print("The capacitance is: ");
    Serial.print(cap, 6);
    Serial.println(" mF");

    Serial.print("Reached threshold at time: ");
    Serial.print(time_elapsed);
    Serial.println(" ms");
  }
}
```

Errors:

1) ADC Resolution: Arduino ADC resolution ≈ **4.88 mV so** Threshold crossing may be detected slightly early or late

2) Noise: Can be caused by Analog pin fluctuations

3) Timing Resolution: micros () resolution is ~4 μs on Arduino Uno and so it limits precision for small capacitance values.

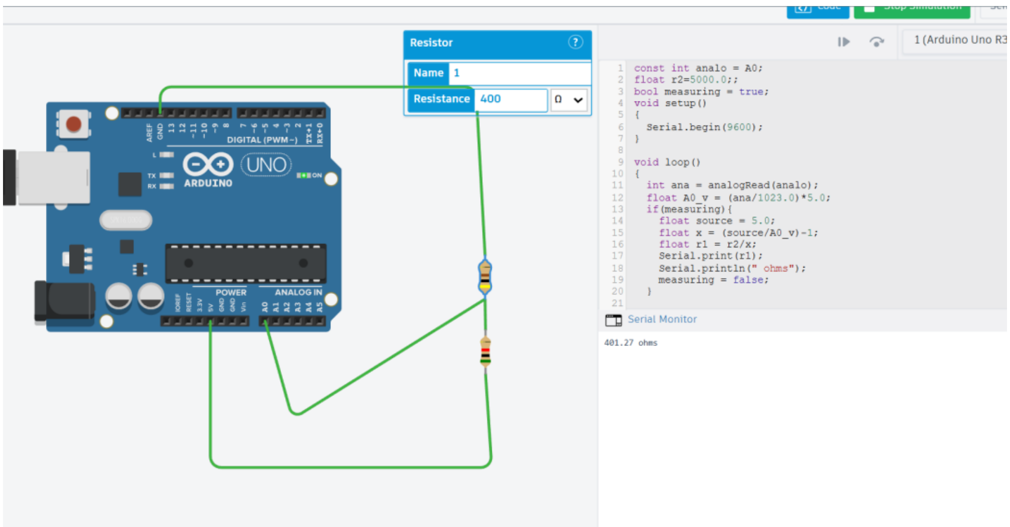
(TASK C)

Voltage Divider Equation:

$$V_{out} = V_{source} \times \frac{R_x}{R_{ref} + R_x}$$

Rearranging to Find Unknown Resistance:

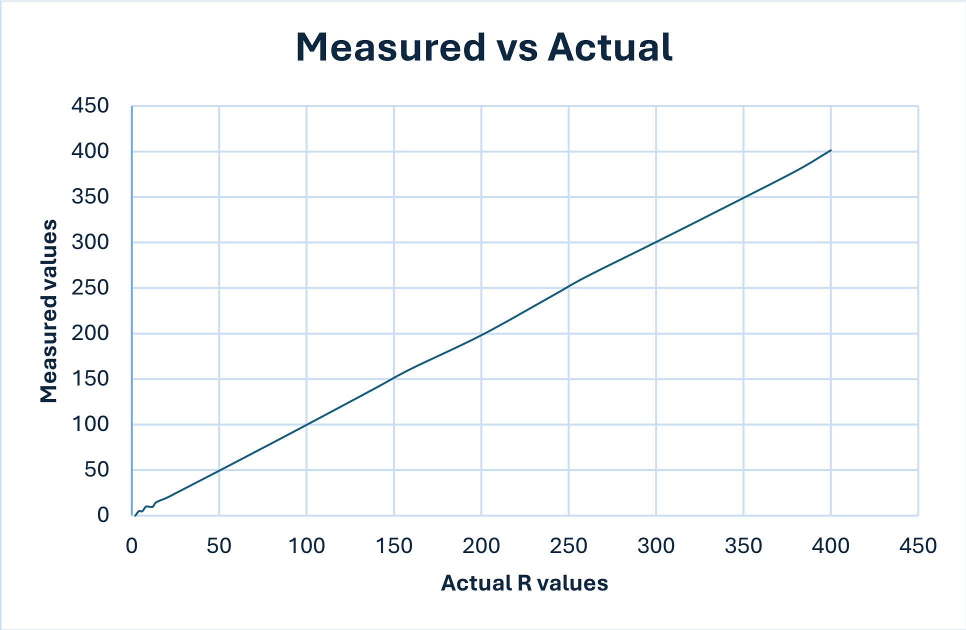
$$R_x = R_{ref} \times \frac{V_{out}}{V_{source} - V_{out}}$$



OUTPUT: Using the simple Voltage divider formula, we calculate the logic for unknown resistance. (The serial monitor gives the value of the unknown resistance).

The known resistance in this case is 5kΩ.

Actual Value of unknown R (in Ω)	Experimental Value of unknown R (in Ω)
400	401.27
21	19.63
45	44.38



ERROR & WHY NON-IDEALITY?

In the graph, in the **mid-high range** of resistor values, the graph is approximately linear and the fluctuations that happen are most likely caused by variations in **Reference voltage which is not exactly 5.000 V**

The Arduino assumes $V_{ref} = 5.0V$ but in reality, USB power fluctuates (≈ 4.8–5.1 V).

In the region where R is very small, the graph is not at all linear. This is due to the following reason-

When $R_x \ll R_{ref}$: $V_{out} \approx V_{in} \frac{R_x}{R_{ref}}$. As a result, the output voltage measured is very small at the Analog pin. These voltages correspond to very few ADC counts:

$$ADC = \frac{V_{out}}{5} \times 1023$$

Example:

- 0.05 V ≈ 10 counts
- 0.10 V ≈ 20 counts

Due to this, **even a single ADC count error can make a huge resistance error.**