

# Assignment1

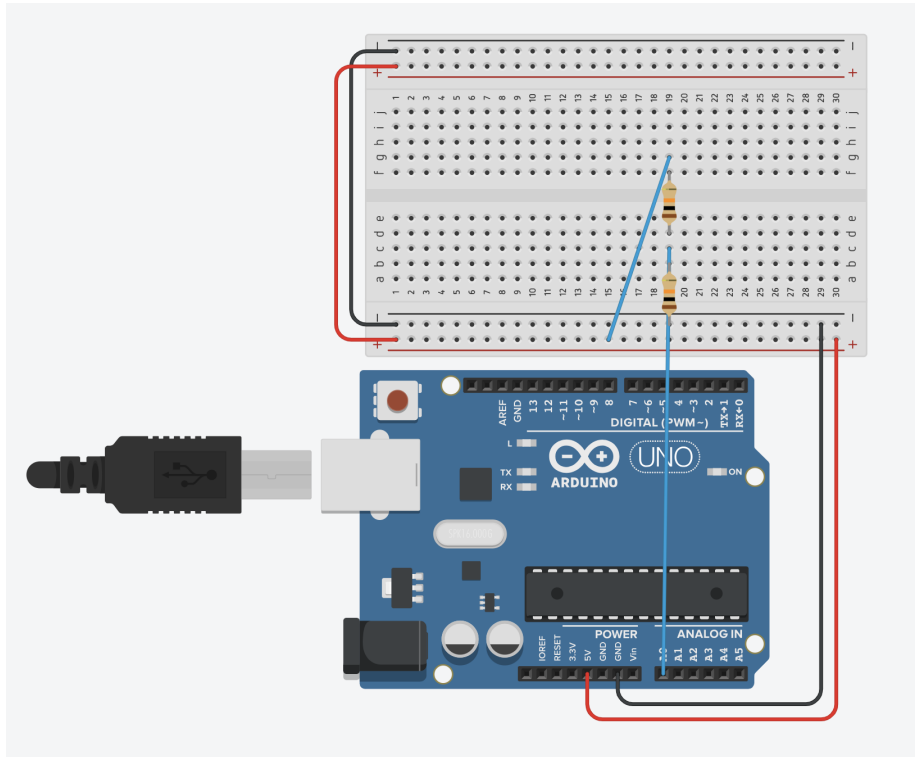
Arisht Daiya

December 2025

## 1 Task A

### 1.1 Circuit

We connect the A0 pin of the arduino to the junction of the 2 resistors to take the appropriate reading.



## 1.2 Code

```
// C++ code
//
void setup()
{
    //standard baud
    Serial.begin(9600);
}

void loop()
{
    int value = analogRead(A0);

    //formula for conversion of reading
    float v=value*(5.0/1023.0);

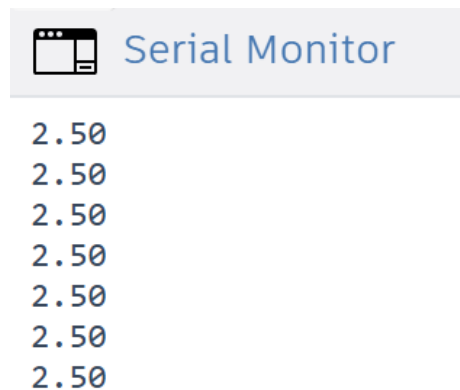
    //prints a new line at end
    Serial.println(v);

    delay(1000);
}
```

## 1.3 Results

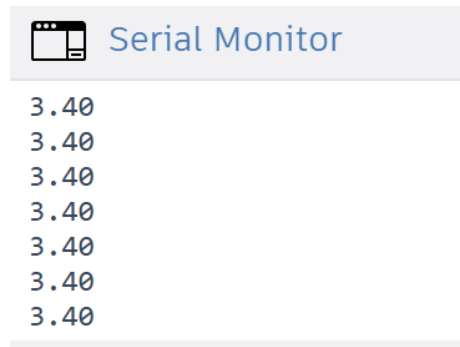
### 1.3.1 10k-10k

We got the theoretically accurate result of 2.50V.



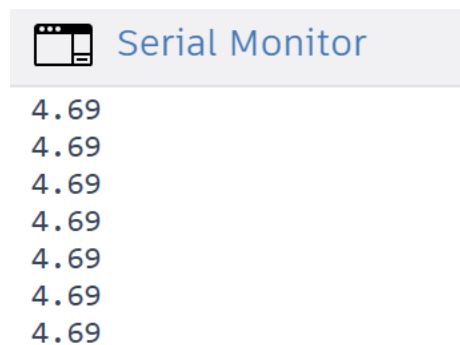
### 1.3.2 10k-4.7k

We were only able to compute the result upto 2 decimal places which was equal to the theoretical value.



### 1.3.3 15k-1k

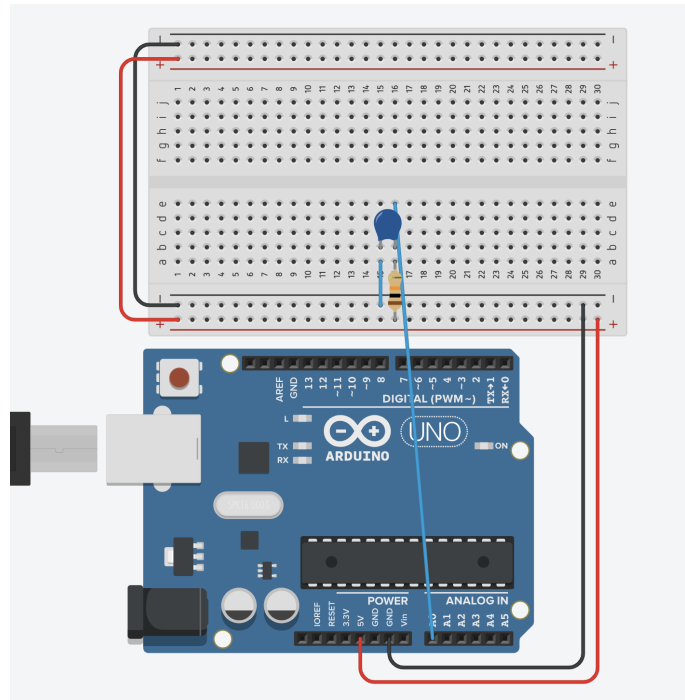
The result got rounded up and was different than the theoretical value(4.6875). This is because the arduino can only take readings in one of 1024 states.



## 2 Task B

### 2.1 Circuit

In an RC circuit, the capacitor charges from 0 to V(voltage of source) according to the formula  $V_c = V_s \cdot (1 - e^{-\frac{t}{RC}})$ . When the time t is equal to RC, the capacitor charges to 63 percent of its max value. For the circuit we connect the capacitor to ground and resistor to 5V and connect the A0 pin to their junction. We calculate when the voltage reaches 63 percent of its maximum possible value.



## 2.2 Code

```

3 void setup()
4 {
5     //standard baud
6     Serial.begin(9600);
7 }
8
9 void loop() {
10    delay(1000);
11
12    float threshold = 5.0 * 0.63;
13    float v = 0;
14
15    //Setting the midpoint voltage to 0 to discharge the capacitor
16    pinMode(A0, OUTPUT);
17    digitalWrite(A0, LOW);
18    delay(500);
19
20    //Setting the A0 pin back to input mode
21    pinMode(A0, INPUT);
22
23    //Start time
24    long st = micros();
25
26    //Loop continues until the condition is reached
27    while (v < threshold) {
28        int value = analogRead(A0);
29        v = value * (5.0 / 1023.0);
30    }
31
32    //End time
33    long et = micros();
34
35    //The time taken
36    Serial.println(et - st);
37
38    delay(1000);

```

We first calculate the threshold and set the initial voltage to 0. Then we set the midpoint voltage to 0 and give the capacitor some time to charge. Then we calculate the start time, apply the condition( $v_i 0.63 \cdot 5$ ) and then calculate the end time. Finally, we get the time taken.

### 3 Results

Serial No.	R Value ( $\Omega$ )	C Value (F)	Expected Value (us)	Measured Value (us)
1	10k	10uF	100000	98980
2	10k	1uF	10000	9948

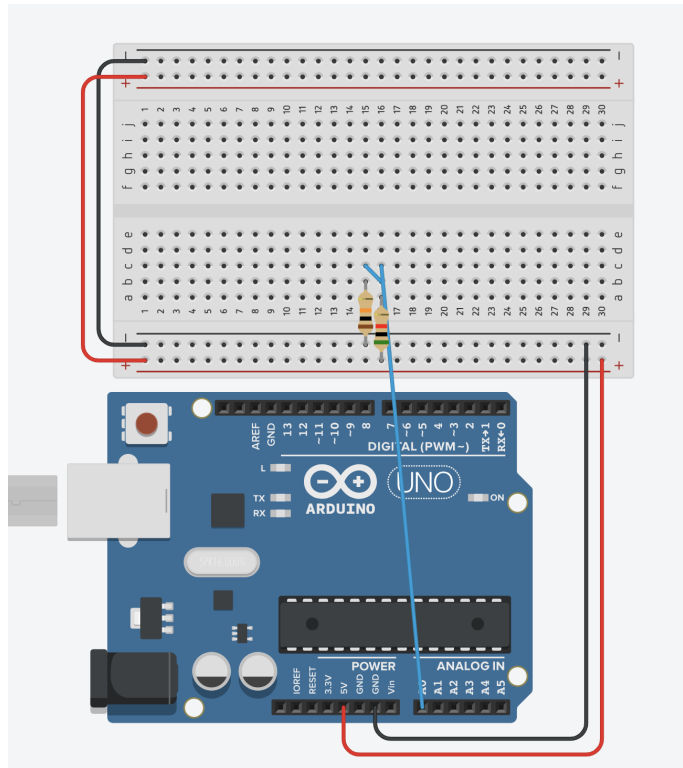
Table 1: Comparison of Expected vs Measured RC Time Constants

The error could be because of a variety of reasons like the resistance tolerance, capacitor tolerance, the ADC resolution, micors() measuring time every 4 microseconds.

### 4 Task C

#### 4.1 Circuit

To measure resistance value, we take  $R1 = 5k\Omega$ ,  $V_s = 5V$  and use the formula  $V_m = V_s \cdot \frac{R2}{R1+R2}$ . Using this, we get  $R2 = R1 \cdot \frac{V_m}{V_s - V_m}$ , where  $R1$  is known resistance,  $R2$  is unknown resistance,  $V_s$  is source voltage and  $V_m$  is measured voltage at the midpoint. For the circuit, we connect  $R2$  to ground,  $R1$  to 5V and connect pin A0 to their junction.



## 4.2 Code

```
// C++ code
//
void setup()
{
    //standard baud
    Serial.begin(9600);
}

void loop() {

    //R1 = 5k ohm
    float r1 = 5000;

    //getting v
    int value = analogRead(A0);
    float v = value * (5.0 / 1023.0);

    //formula for r2
    float r2 = r1*(v/(5.0-v));

    //printing r2
    Serial.println(r2);

    delay(1000);
}
```

We first set R1 to 5k  $\Omega$  and calculate the voltage at the junction. Then we get the value of R2 using the formula derived earlier.

### 4.3 Results

Serial No.	R1 Value (ohm)	Expected R2 Value (ohm)	Measured R2 Value (ohm)
1	5000	5000	4990.23
2	5000	1000	996.48
3	5000	10000	10000.00

Table 2: Comparison of Expected vs Measured RC Time Constants

The error could be due to a variety of reason which include arduino ADC quantization of the voltage(arduino can only measure in one of 1023 states). Other reasons include tolerance, internal impedance of arduino, rounding errors of float numbers, etc.