JACOBS UNIVERSITY

CH-220-B
Introduction to Robotics and Intelligent Systems Lab:

Lab 1

Experimented and Prepared By:

Suraj Giri
s.giri@jacobs-ujniversity.de

Aryans Rathi
a.rathi@jacobs-university.de

Riley Edwin Sexton
r.sexton@jacobs-university.de

Instructor's Name:

Dr. Fangning Hu
F.Hu@jacobs-university.de

INTRODUCTION AND THEORY

1. The Arduino Board and IDE:

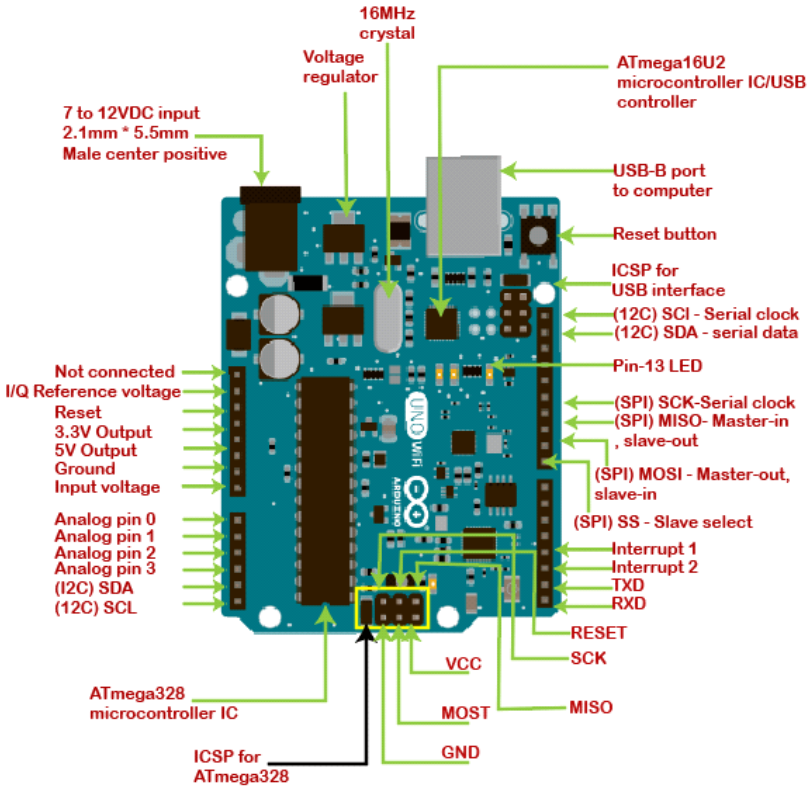

Fig 1.1: Arduino Board

The Arduino is a microcontroller board. It consists of 14 Digital I/O Pins, 6 PWM output pins and 6 Analog Input Pins. Analog Pins can have a maximum voltage of 5V which is the maximum circuit of the whole Arduino Board) while the Digital Pins are given power through the commands from the computer via Arduino IDE.The Arduino board works only if any external power supply such as a coputer or a battery isconnected to it via the USB portal in it. There area many types of Arduinos available in the market but the board that we are going to use for our experiments is Arduino UNO. We can also use TXD and RXD pins for Bluetooth connection with a bluetooth controller like a phone or any other digital device. For using bluetooth, a abluetooth module is used where the cable needs to be connected to the transmission pin of the module, while the cable should be connected to the receiving port of the arduino. This is a form of serial communication.

The IDE used for arduino is a free and open source software and it is generally known as the Arduino IDE. Following is a screenshot of the General Arduino IDE.

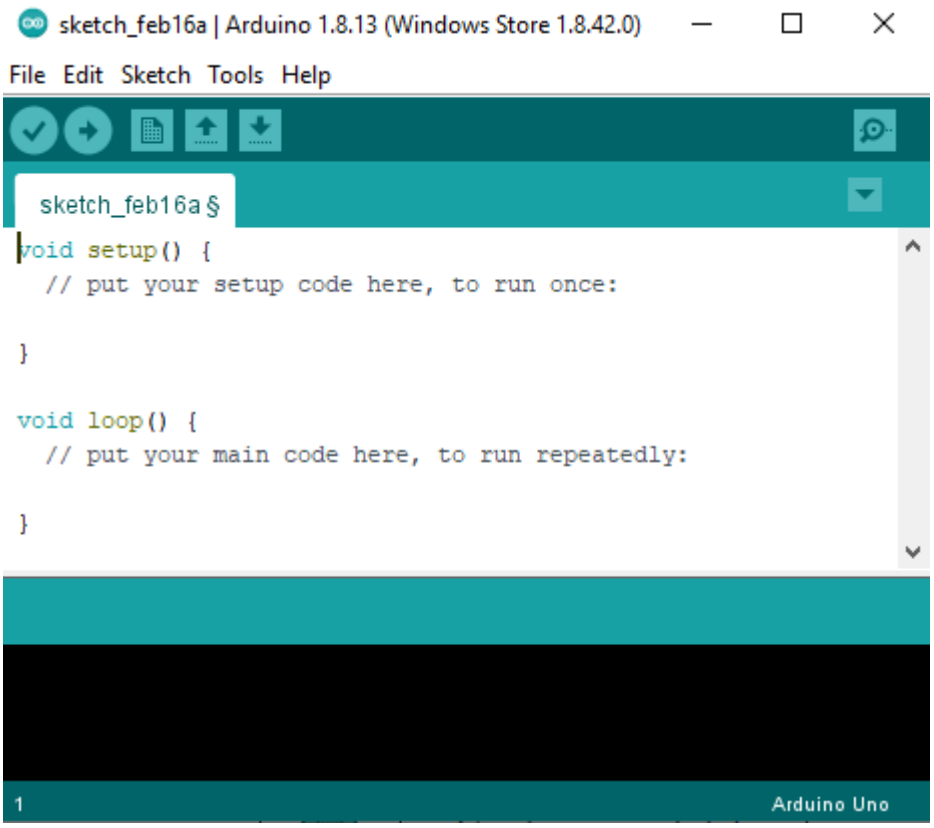

Fig 1.2 : Arduino IDE

Generally, codes for Arduino are split into two parts: setup() and loop().Usually, in the setup() function we configure the pins of Arduino or start communi-cation protocols and the loop() function is run repeatedly run until the Arduino is switched off.

2. BreadBoard and Jumper Wires:

A BreadBoard is a solderless device which is very useful for creating temporary prototypes of circuit designs. And Jumper wires are the small wires that are used to connect the two points of a breadboard. To perform our experiments, we will also use the jumper wires to connect our Arduino and the whole circuit that we will make on a breadboard.

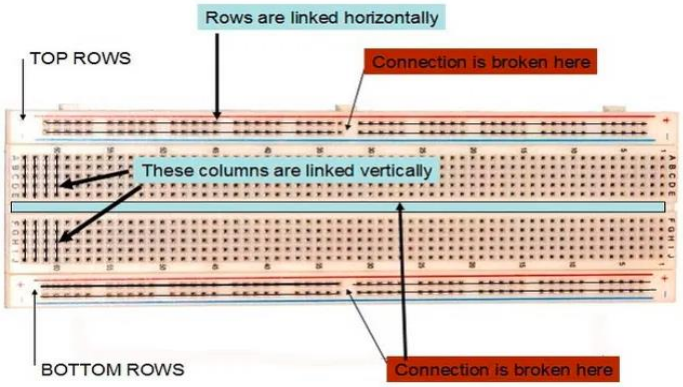

Fig 2.1: BreadBoard

Fig 2.2: Jumper WIres

3. LED and Resistors:

An LED (Light Emitting Diode) is a two-lead semiconductor light source which emits light when a suitable voltage is applied to the leads. An LED lights up when the electrons emitted due to the voltage/current interacts with the semiconductor materials. The longer lead is the positive terminal and the shorter one is the negative. As we have conducted the experiment in the simulation, the bend lead is the positive and the straight one is the negative.

With certain resistors, an electrical component that implements electrical resistance, we can modify the amount of current flowing through the circuit

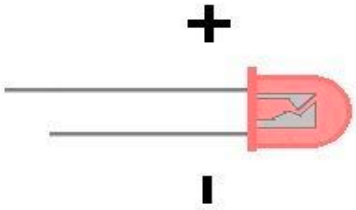

Fig 3.1: LED with its positive and negative terminals

Fig 3.2: Resistor

According to Ohm's Law, voltage (V) across the resistor is the product of the current and the resistance: $V_R = I \cdot R$

$V_{Forward}$ is the forward voltage of the LED and V_{source} is the output from the Arduino (typically it is 5V).

4. Multimeter:

Multimeter is a device that can measure current, resistance and voltage in different ranges. It can be used to measure the continuity and also to troubleshoot various circuit problems.

Fig 4.1: Multimeter

5. **Potentiometer:**

A potentiometer is a resistor that has three terminals in which the resistance is manually Varied so that we could ourself direct the amount of current passing though our circuit.

6. **A Simple Arduino Code**

```
//the setup function runs once when the reset or power is pressed
void setup () {
pinMode (13 , OUTPUT );    // Initialize pin 13 for output (digital – write).
}

//the loop funciton runs overa nd over again forever
void loop() {                      // Called repeatedly
digitalWrite (13 , HIGH );     // Set pin 13 to 5V
delay (1000) ;                     //wait for 1 second = 1000 millisecond
digitalWrite (13 , LOW);       // Set pin 13 to 0V
delay (1000) ;                     // Wait 1 sec = 1000 millisecond .
}
```

Code Description:

Firstly, pin 13 is set up as the output pin in void setup() that will make the LED to turn on or off. In the void loop(), the LED connected to pin 13 will light up for 1 second as pin 13's voltage is turned to 5V and after 1 second it will turn off as the voltage is turned to 0V. Again afer 1 second the LED will light up and again after another 1 second it will go off.

EXPERIMENTATION AND EVALUATION

Task 1.1

The whle experiment is conducted on the simulation. Firstly the code written in the part 6 of the introduction was written into the Arduino IDE that was available in the simulation we had. After that the code was uploaded to the Arduino noard. After the code was run, it lit up the LED that was in the Arduino board. This showed that our code and Arduino were working properly. This was the working example for working with the Arduino.

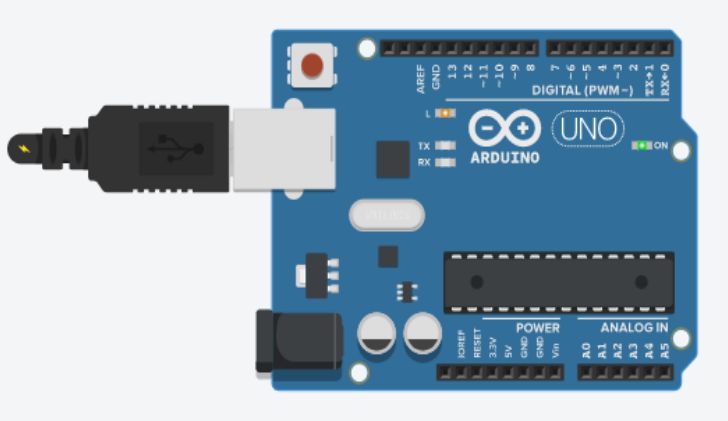

Fig T1.1: Arduino Conncetion and Arduino is working properly.

Task 1.2

After disconnecting the Arduino, we chose the Red coloured LED and the resistor of 200 Ω. Then we connected the LED and the resistor in the breadboard according to the circuit provided. After finalizing the circuit in the breadboard and the Arduino, we implemented the same code that we used for task 1.1 and uploaded it to the Arduino, which made the Red LED to blink cery 1 second.

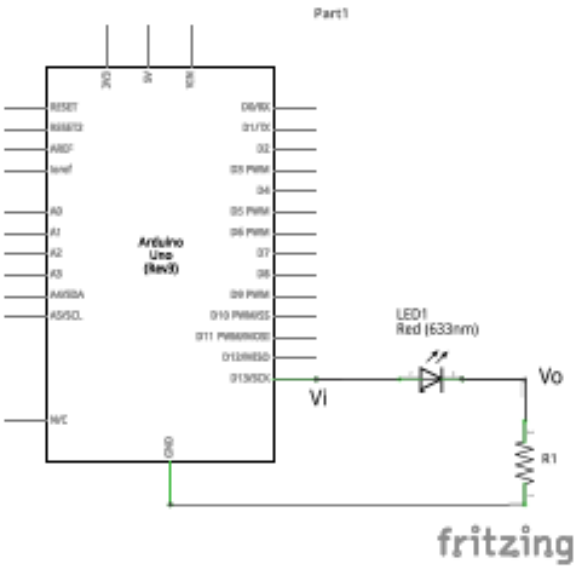

Fig T1.2.1: Arduino Circuit

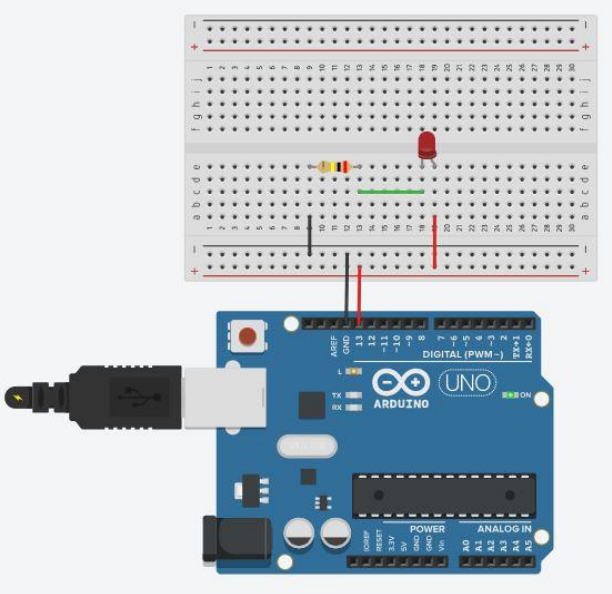

Fig T1.2.2: Blinking LED in the Arduino

Task 1.3

Since we were were in the group that was assigned to do the experiments using simulation, we calculated the resistance of consecutive resistors by ourselves providing the equivalent value to the resistor simulation. As it was the simulation, it showed us the exact values of corresponding resistances.

	R1	R2	R3	R4	R5	R6
INPUT	1850 mΩ	1700 mΩ	100 Ω	120 Ω	17 MΩ	2 MΩ
Multimeter Reading	1.85 Ω	1.7 Ω	100 Ω	120 Ω	17.0 MΩ	2 MΩ

Fig T1.3: Multimeter measuring

Task 1.4

For this task, we created a new circuit in our simulation. We created a circuit that resembled the circuit that was given in the lab manual. Once the circuit was set up, we connected that circuit to a multimeter that was set to the Voltage setting to provide the voltage output across the Blue LED. When we decrease the resistance of the resistor, the voltage across the LED increases because the current that would pass though the Resistor would increase but the resistance of the bulb is constant. This is also proved by V=I.R.

Since we have used the simulation, the voltage through the bulb for a consequent resistor was always the same. For this calculation we used the resistance of 150 Ohm.

Reading	Voltage V
1	2.02
2	2.02
3	2.02
4	2.02
5	2.02

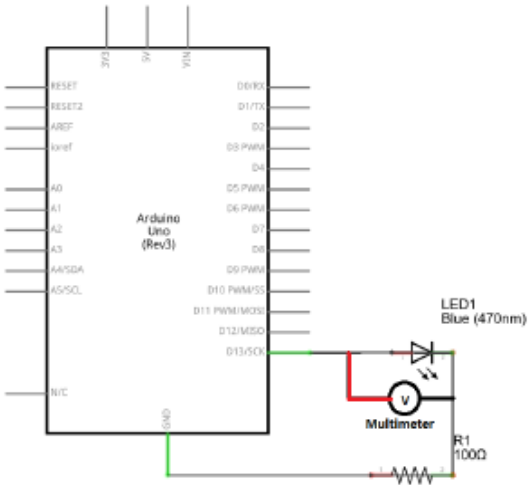

Fig T1.4.1: Circuit Diagram for using

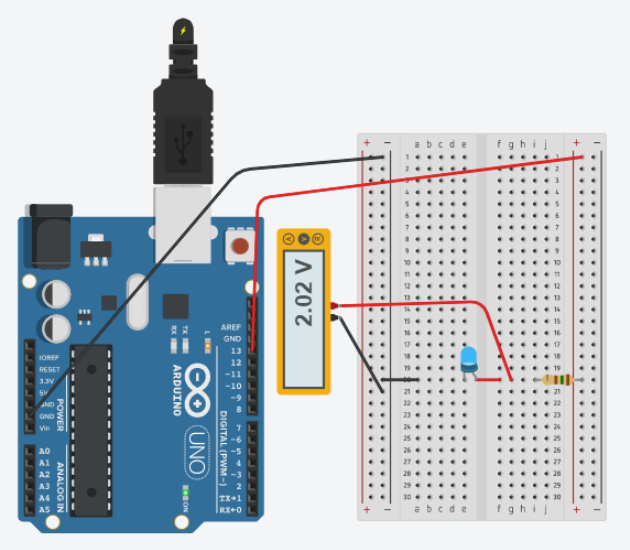

Fig T1.4.2: Calculation of the Voltage

Task 1.5

Since we were told to do our experiment on the simulation, we were unable to carry out the continuity test. We tried to carry out the continuity test but instead of the beep sound that was told we could expect, we got the voltage of 0.00 V across our multimeter simulation.

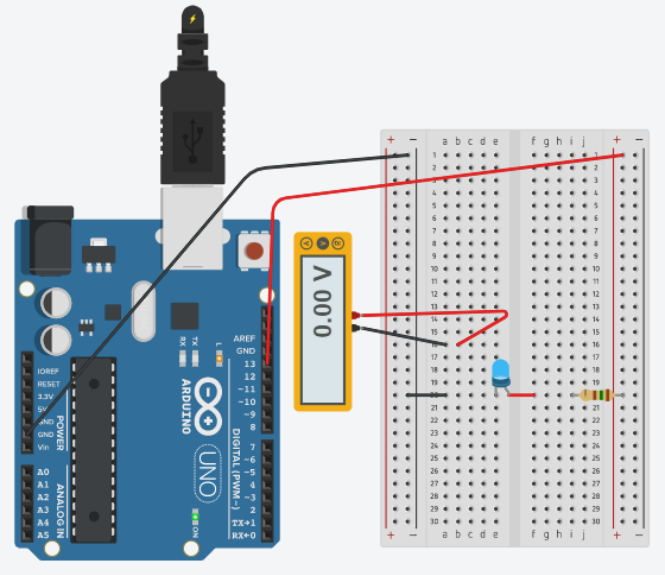

Fig T1.5: Continuity

Task 1.6

For this task, we set up the circuit as we were told in the Lab manual. The diagram of the circuit is drawn below too. For this task, we used a resistor of 400 Ω. Also, we used a button to start and stop the flow of current through the circuit. Finally the code that was given in the lab manual (copied below) was uploaded to the Arduino. Pressing the button lighted up the LED that was in the Arduino Board.

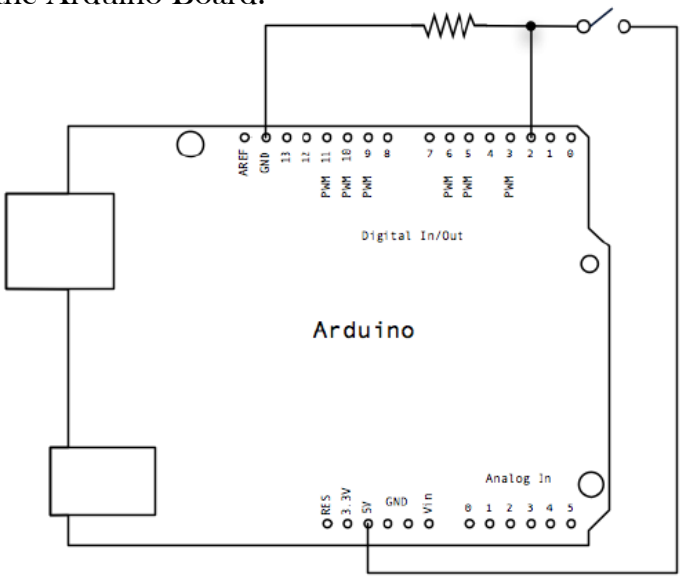

Fig T1.6.1: Circuit Diagram for connecting a swicht to Arduino

Code: Code for turning on the bulb when the switch was pressed and off when the switch was released :

```
/* Button
Turns on and off a light emitting diode (LED) connected to digital
pin 13, when pressing a pushbutton attached to pin 2. */
// constants won 't change . They 're used here to
// set pin numbers :
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin
// variables will change :
int buttonState = 0; // variable for reading the pushbutton status
void setup () {
        // initialize the LED pin as an output :
        pinMode (ledPin , OUTPUT );
        // initialize the pushbutton pin as an input :
        pinMode ( buttonPin , INPUT );
}
void loop (){
        // read the state of the pushbutton value :
        buttonState = digitalRead ( buttonPin );
        // check if the pushbutton is pressed .
        // if it is , the buttonState is HIGH :
        if ( buttonState == HIGH ) {
                // turn LED on:
                digitalWrite (ledPin , HIGH );
        }
        else {
                // turn LED off:
                digitalWrite (ledPin , LOW );
        }
}
```

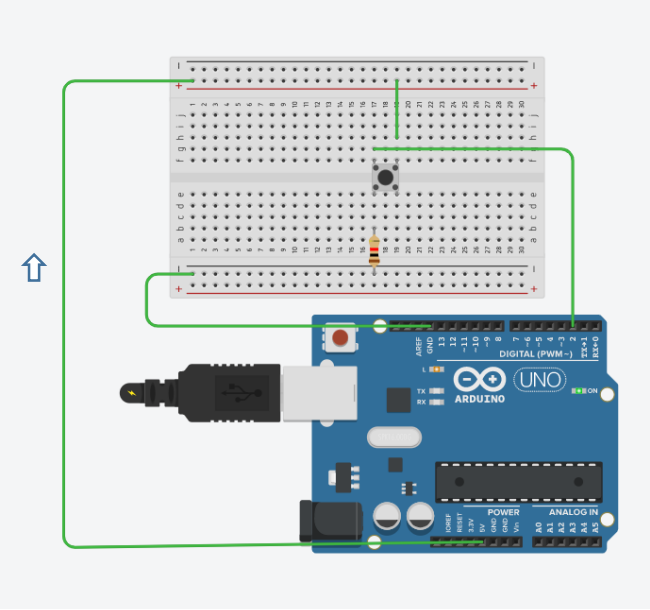

Fig T1.6.2: When the button was pressed, i.e. when it was **HIGH**, the LED lit up.

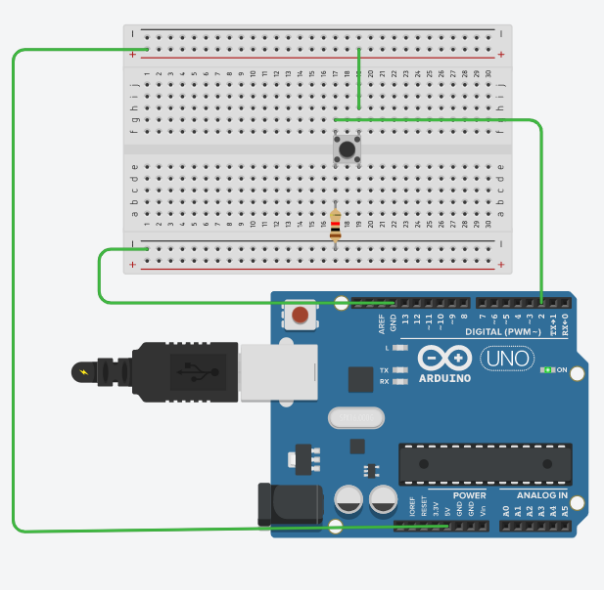

Fig T1.6.3: When the button was not pressed, i.e. when it was **LOW**, the LED turned off.

Task 1.7

For this task, we used the simulation and provided different resistances to the potentiometer. For the consecutive resistance we entered, we got 1 more Ω in the Multimeter. This means that the resistance of the potentiometer is 1 Ω.

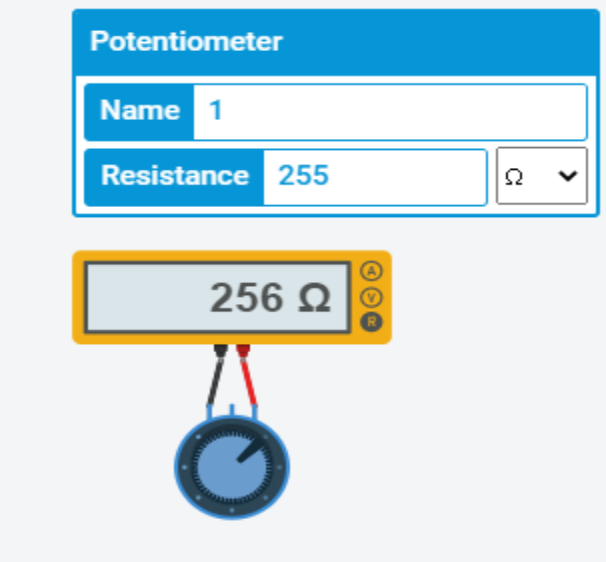

Fig T1.7: Measuring the resistance of a Potentiometer using

Task 1.8

For this task, the potentiometer was connected to the Arduino as shown in the figure by connecting it to the Breadboard and then connecting the Arduino to the BreadBoard. The wiper of the Potentiometer was connected to the Analog pin 0 of the Arduino. Terminal 1

was connected to the GND of the Arduino while the Terminal 2 was connected to the pin with 5V output.

We fed the following code to out Arduino IDE:

```
/* ReadAnalogVoltage
Reads an analog input on pin 0, converts it to voltage ,
and prints the result to the serial monitor .
*/
// the setup routine runs once when you press reset :
void setup () {
   // initialize serial communication at 9600 bits per second :
   Serial.begin (9600) ;
}
   // the loop routine runs over and over again forever :
void loop () {
   // read the input on analog pin 0:
   int sensorValue = analogRead (A0);
   // Convert the analog reading ( from 0 - 1023) to a voltage (0 - 5V):
   float voltage = sensorValue * (5.0 / 1023.0) ;
   // print out the value you read :
   Serial.println ( voltage );
}
```

We then changed the amount of resistance by turning the shaft of the Potentiometer and observed the result on the serial monitor. We observed the following outcome:

The code initialized the serial communication to 9600 bits per second and looped those values while we provided the varying voltages from 0V to 5V by turning the shaft of the potentiometer using the mathematical formula that we have fed to our code. The mathematical formula was ((5.0/1023.0)*sensorValue).
On the serial monitor we observed the corresponding varying voltages between 0V and 5V when we turned the shaft of the Potentiometer.

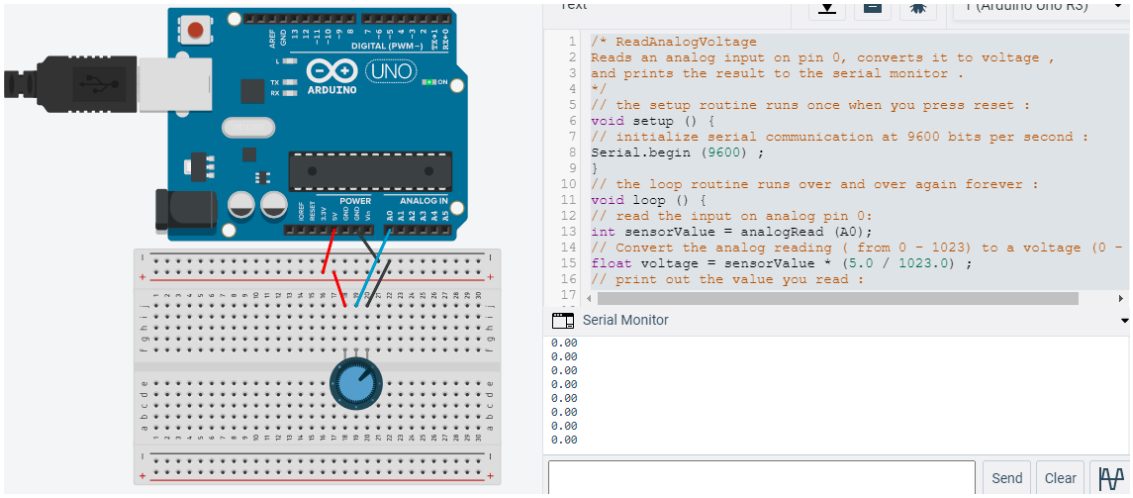

Fig T1.8.1: Potentiometer Connected to the Arduino and output on the Serial Monitor

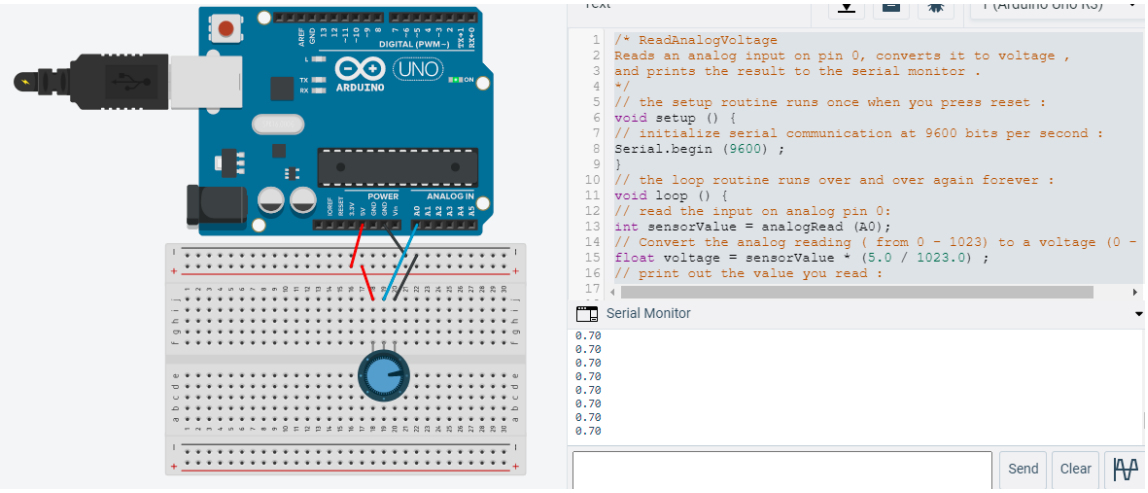

Fig T1.8.2: Potentiometer Connected to the Arduino and output on the Serial Monitor

Conclusions:

Using the codes and a resistor of specific resistance, we firstly made a circuit in which a light blinked. Similarly we also leaarned to measure the resistance of a resistor using the multimeter and then to calculate the coltage across the specified path using the same multimeter. The voltage measured across the specific path was the forward Voltage. Since we used the simulation, we were unable to measure the faul tof the equipment multimeter but we learned the way to do so. We hope we can do it in person when we are provided with the real equipment.

Then we also connected a switch to our Arduino and then uploaded the corresponding code which would turn the LED on only when the switch was turned on. Finally we measured the resistance of a Potentiometer and then uploaded a Code to the Arduino IDE to measure the varying voltage (which was between 0V and 5V) by turning the shaft of the potentiometer.

References:

- Introduction to RIS Lab Manual.
- Introduction RIS Lab Slides by Ph.D. Dr Fangning HU
- Images in the introduction section were taken from Google and all the other Images used in the experiment section were the screenshots of the simulation that we performed.
- Codes were copied from the Lab Manual.