Lab Report : Arduino 1

ECS 330 : Experiment 4

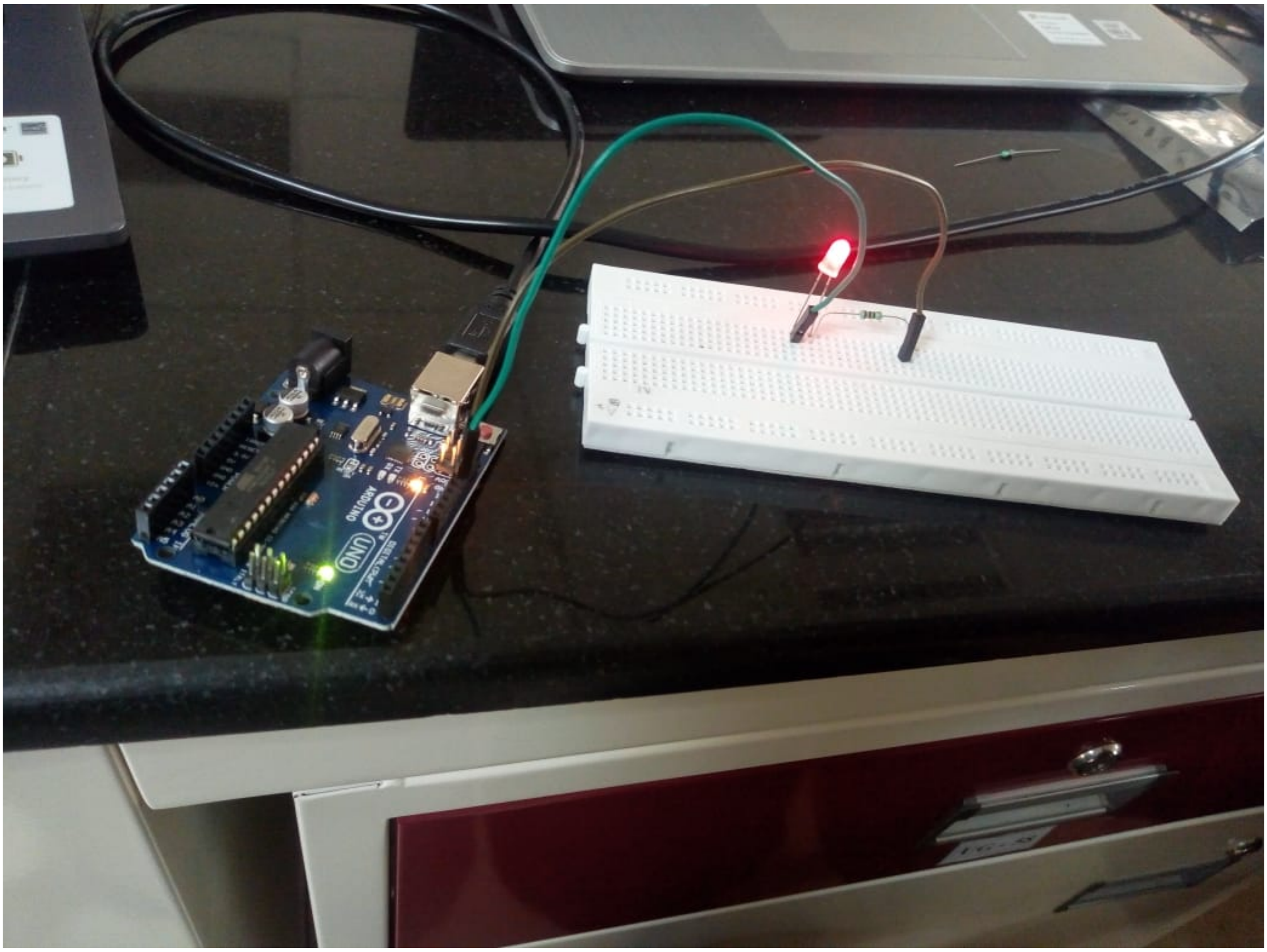
Name: Ajay Choudhury **Roll No:** 18018 **Date:** 28th March 2021

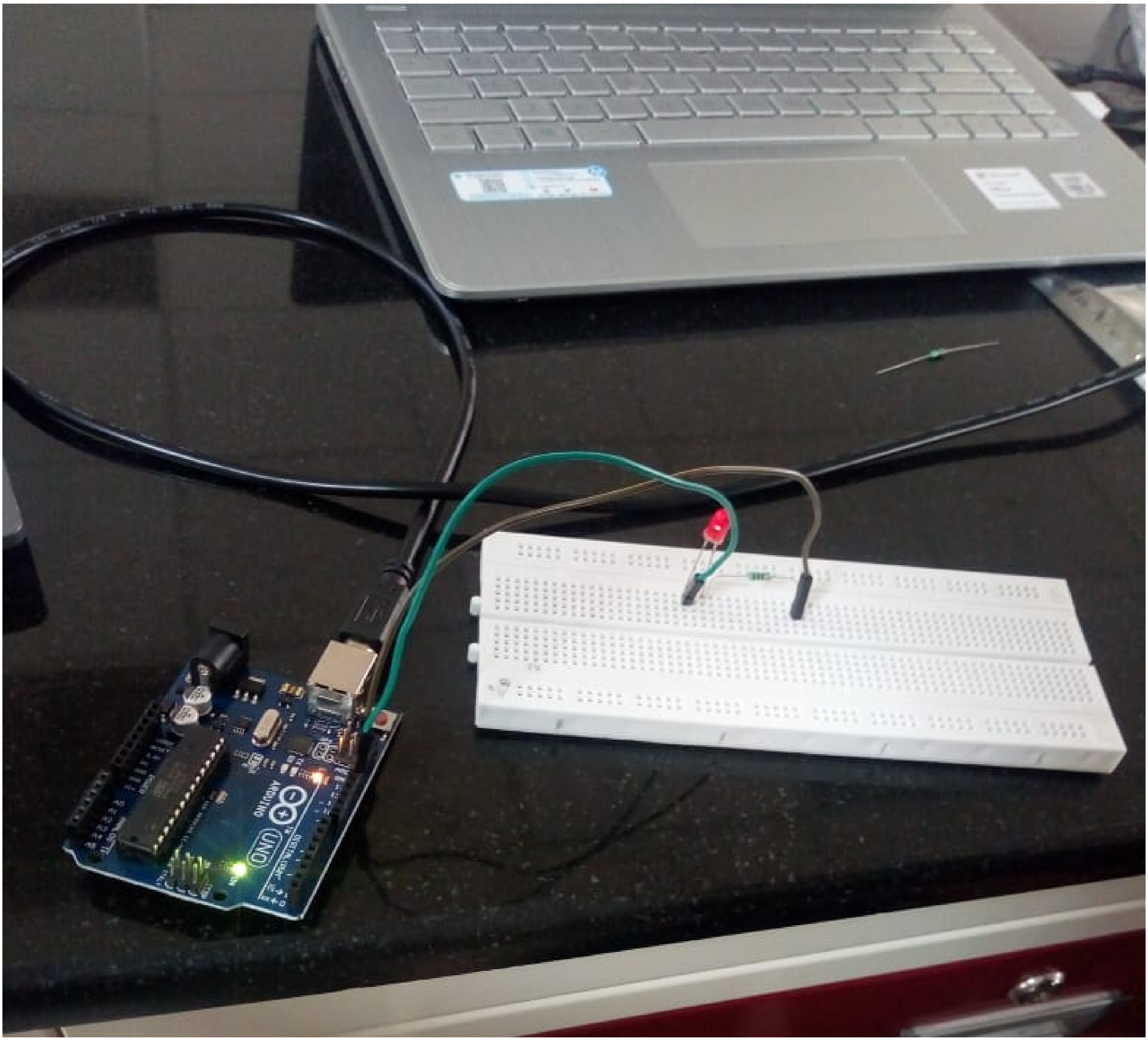
Title of the Experiment: Using Arduino to perform certain given tasks

Brief Description: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output - activating a motor, turning on an LED, publishing something online. One can tell its board what to do by sending a set of instructions to the microcontroller on the Board. Here we experiment with LED and potentiometer with the help of Arduino.

Assembly and Circuits:

Tasks 1 - 5: The circuit of Arduino board for tasks 1 to 5 is:



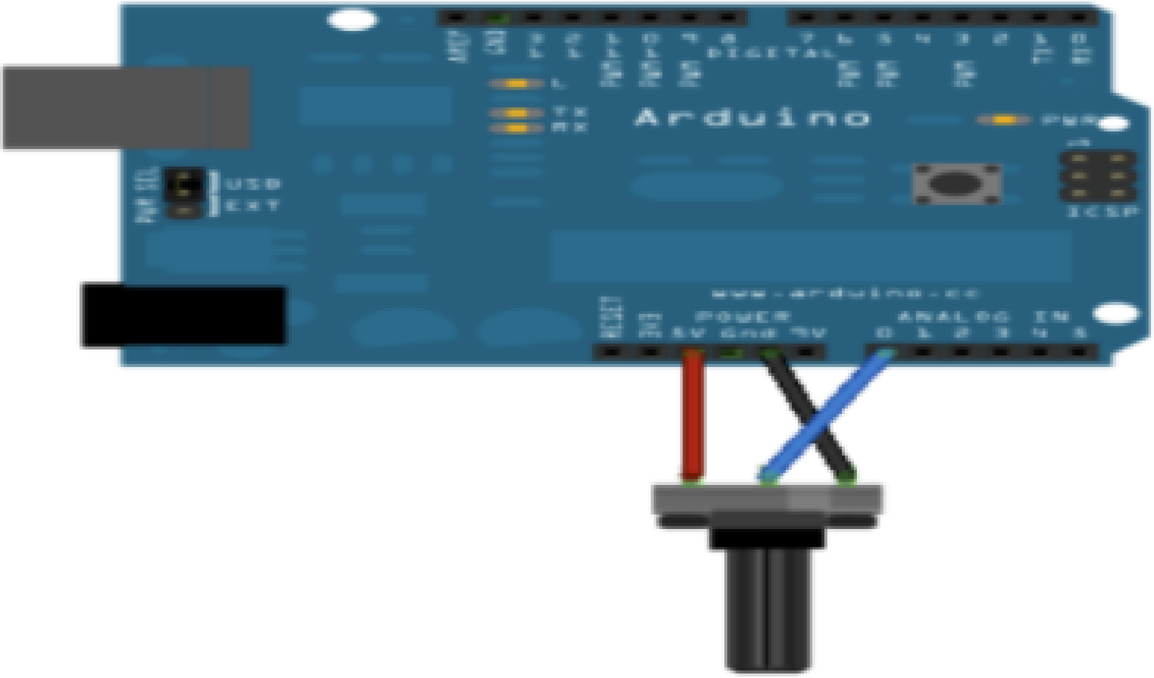


- The codes for these tasks are attached in the codes section.

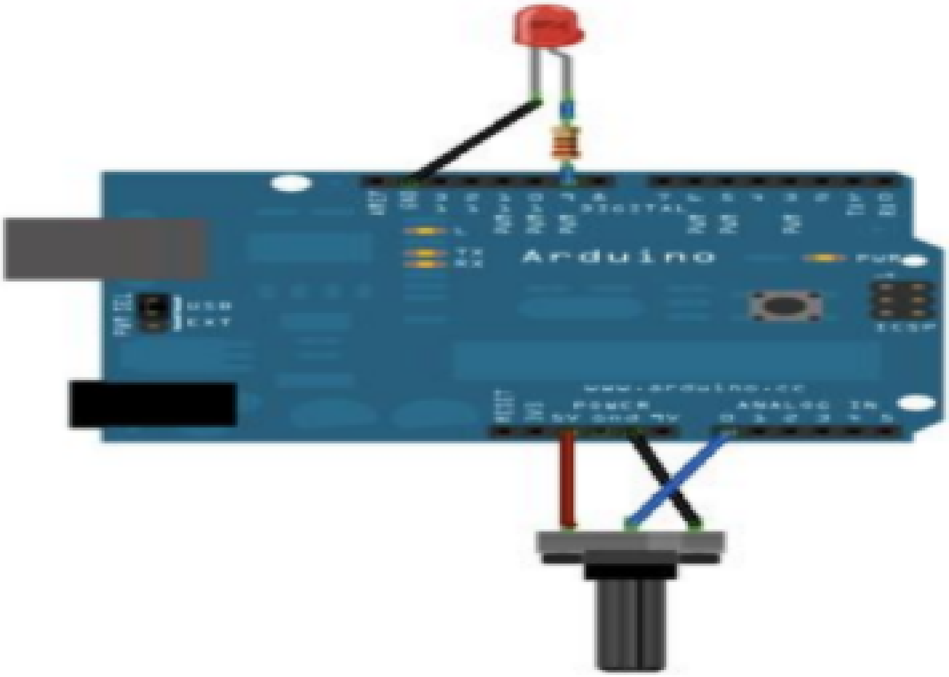
Link to the video of the circuit:

[Video of the Arduino assembly for changing intensity of LED’s brightness.]

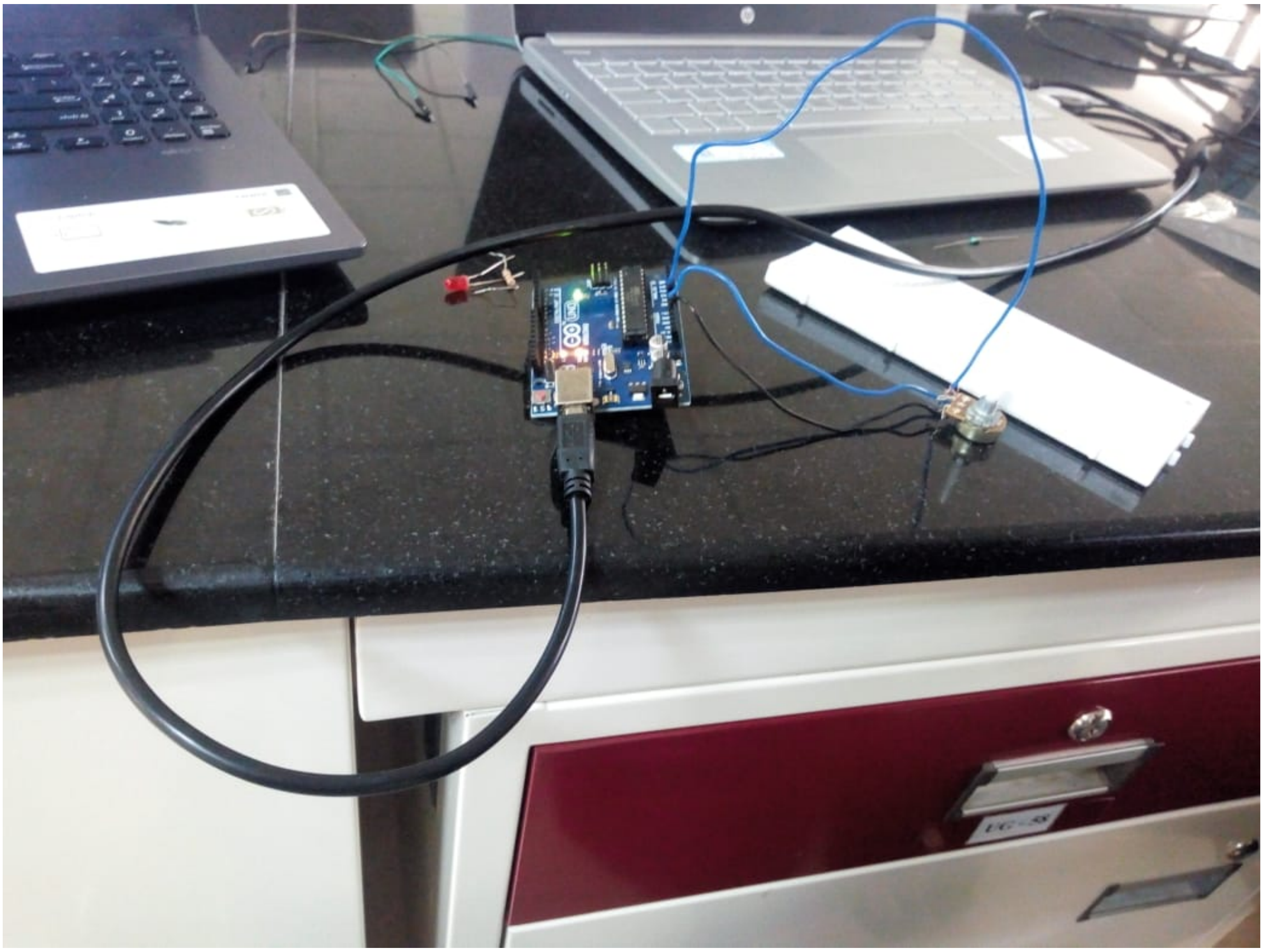
Task 6: Connect the 10K pot and vary the resistance.



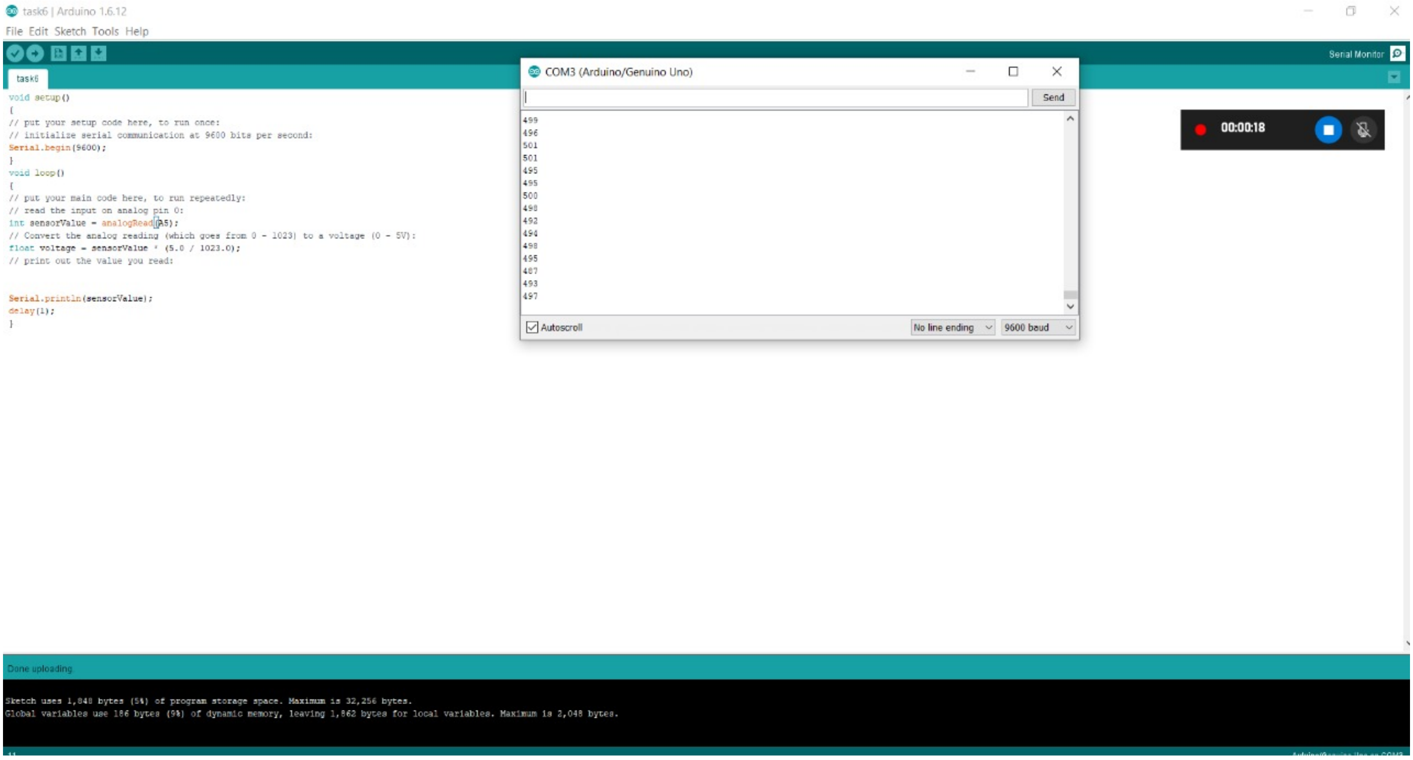
Task 7: With the help of pot regulate the brighten of external connected LED.



Following is the image of assembly of the potentiometer experiment:



The readings of the serial monitor are as follows:

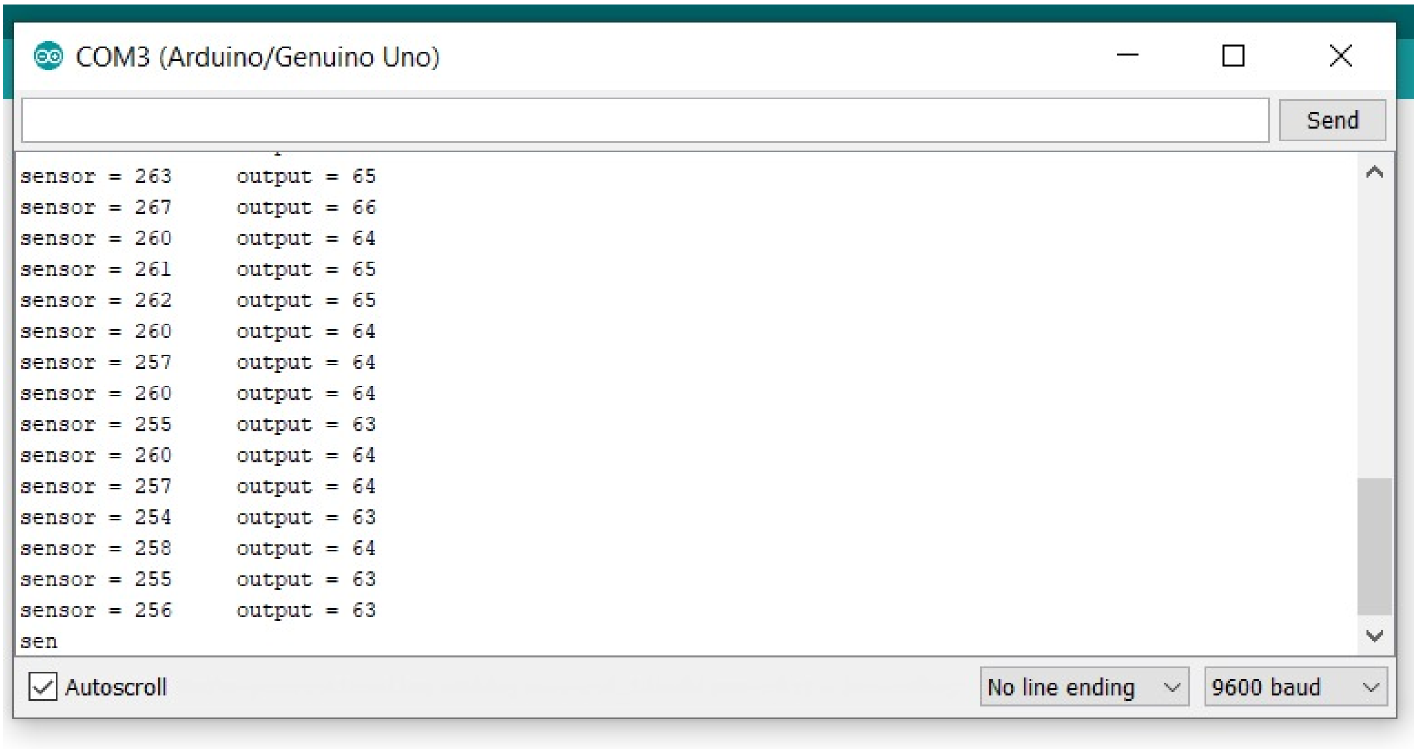


```
void setup()
{
// put your setup code here, to run once:
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
}
void loop()
{
// put your main code here, to run repeatedly:
// read the input on analog pin 0:
int sensorValue = analogRead(A5);
// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
float voltage = sensorValue * (5.0 / 1023.0);
// print out the value you read:


Serial.println(sensorValue);
delay(1);
}
```

```
499
496
501
501
495
495
500
498
492
494
498
495
487
493
497
```

```
Sketch uses 1,848 bytes (5%) of program storage space. Maximum is 32,256 bytes.
Global variables use 186 bytes (9%) of dynamic memory, leaving 1,862 bytes for local variables. Maximum is 2,048 bytes.
```



```
sensor = 263    output = 65
sensor = 267    output = 66
sensor = 260    output = 64
sensor = 261    output = 65
sensor = 262    output = 65
sensor = 260    output = 64
sensor = 257    output = 64
sensor = 260    output = 64
sensor = 255    output = 63
sensor = 260    output = 64
sensor = 257    output = 64
sensor = 254    output = 63
sensor = 258    output = 64
sensor = 255    output = 63
sensor = 256    output = 63
sen
```

No line ending   9600 baud

Codes:

Task 1: Blink internal LED

```
int LEDPin=LED_BUILTIN;
int waitTimeOn=200;         // time in milliseconds
int waitTimeOff=200;

void setup()
{ // put your setup code here, to run once:
pinMode(LEDPin,OUTPUT);
}

void loop()
{ // put your main code here, to run repeatedly:
digitalWrite(LEDPin,HIGH); // This supplies 5 volts to the LED anode.
delay(waitTimeOn); // Time delay between on and off
digitalWrite(LEDPin,LOW); // That takes the LED_BUILTIN pin back to 0 volts
delay(waitTimeOff); // Time delay between on and off
}
```

Task 2: Blink internal LED with 10 ms delay

```
int LEDPin=LED_BUILTIN;
int waitTimeOn=10; // time in milli sec
int waitTimeOff=10;
```

```
void setup()
{ // put your setup code here, to run once:
pinMode(LEDPin,OUTPUT);
}

void loop()
{ // put your main code here, to run repeatedly:
digitalWrite(LEDPin,HIGH); // This supplies 5 volts to the LED anode.
delay(waitTimeOn); // Time delay between on and off
digitalWrite(LEDPin,LOW); // That takes the LED_BUILTIN pin back to 0 volts
delay(waitTimeOff); // Time delay between on and off
}
```

Task 3: Blink external LED with 200 ms delay

```
nt led = 9; // the PWM pin the LED is attached to
int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by

void setup()
{
pinMode(led, OUTPUT); // declare pin 9 to be an output:
}

void loop()
{
analogWrite(led, brightness); // set the brightness of pin 9
brightness = brightness + fadeAmount; // change the brightness for next time through
the
loop:
// reverse the direction of the fading at the ends of the fade:
if (brightness <= 0 || brightness >= 255)
{
fadeAmount = -fadeAmount;
}
delay(30); // wait for 30 milliseconds to see the dimming effect
}
```

Task 4: Blink external LED with 200 ms delay in on to off and 20 ms delay in off to on

```
int LEDPin=13;
 int waitTimeOn=200;// time in millisec
int waitTimeOff=20;// time in millisec

void setup()
{
```

```
EECS Dept. IISER-B
pinMode(LEDPin,OUTPUT);
}
void loop()
{
digitalWrite(LEDPin,HIGH);
delay(waitTimeOn);
digitalWrite(LEDPin,LOW);
delay(waitTimeOff);
}
```

Task 5: Change the intensity of LED

```
int led = 9; // the PWM pin the LED is attached to
int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by
void setup()
{
pinMode(led, OUTPUT); // declare pin 9 to be an output:
}
void loop()
{
analogWrite(led, brightness); // set the brightness of pin 9
brightness = brightness + fadeAmount; // change the brightness for next time through
the
loop:
// reverse the direction of the fading at the ends of the fade:
if (brightness <= 0 || brightness >= 255)
{
fadeAmount = -fadeAmount;
}
delay(30); // wait for 30 milliseconds to see the dimming effect
}
```

Task 6: Connect the 10K pot and vary the resistance.

```
void setup()
{
// put your setup code here, to run once:
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
}
void loop()
{
// put your main code here, to run repeatedly:
```

```
// read the input on analog pin 0:
int sensorValue = analogRead(A0);
// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
float voltage = sensorValue * (5.0 / 1023.0);
// print out the value you read:
Serial.println(sensorValue);
delay(1);
}
```

Task 7: With the help of pot regulate the brighten of external connected LED.

```
const int analogInPin = A0; // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int sensorValue = 0; // value read from the pot
int outputValue = 0; // value output to the PWM (analog out)

void setup()
{
// initialize serial communications at 9600 bps:
Serial.begin(9600);
}

void loop()
{
// read the analog in value:
sensorValue = analogRead(analogInPin);
// map it to the range of the analog out:
outputValue = map(sensorValue, 0, 1023, 0, 255);
// change the analog out value:
analogWrite(analogOutPin, outputValue);
// print the results to the Serial Monitor:
float voltage = sensorValue * (5.0 / 1023.0);
Serial.print("sensor = ");
Serial.print(sensorValue);
Serial.print("\t output = ");
Serial.println(outputValue);
// wait 2 milliseconds before the next loop for the analog-to-digital
delay(2);
}
```

Discussion:

Thus we see that we can control the LED’s brightness with a potentiometer and vary the resistance with the potentiometer along with the help of an arduino.