VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

CSE 3112 – SENSORS AND ACTUATORS

LABORATORY MANUAL

WINTER SEMESTER 2021 - 2022

Submitted By

Name: Arkadeep Banerjee

Reg.No.: 19BPS1129

CONTENTS

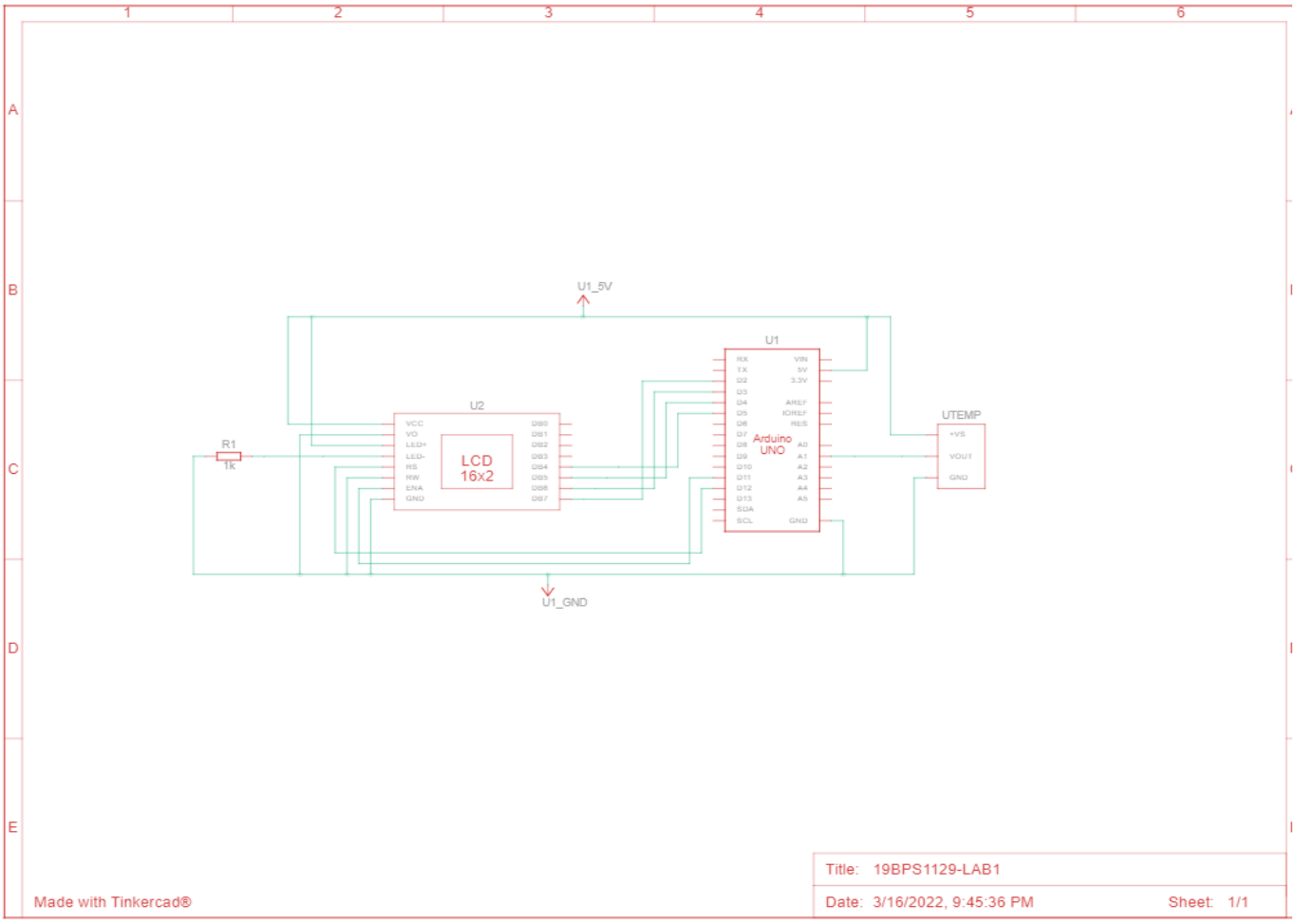
Exp.1. Interfacing analog temperature sensor and A/D conversion of temperature data

Aim: To Interface analog temperature sensor and A/D conversion of temperature data

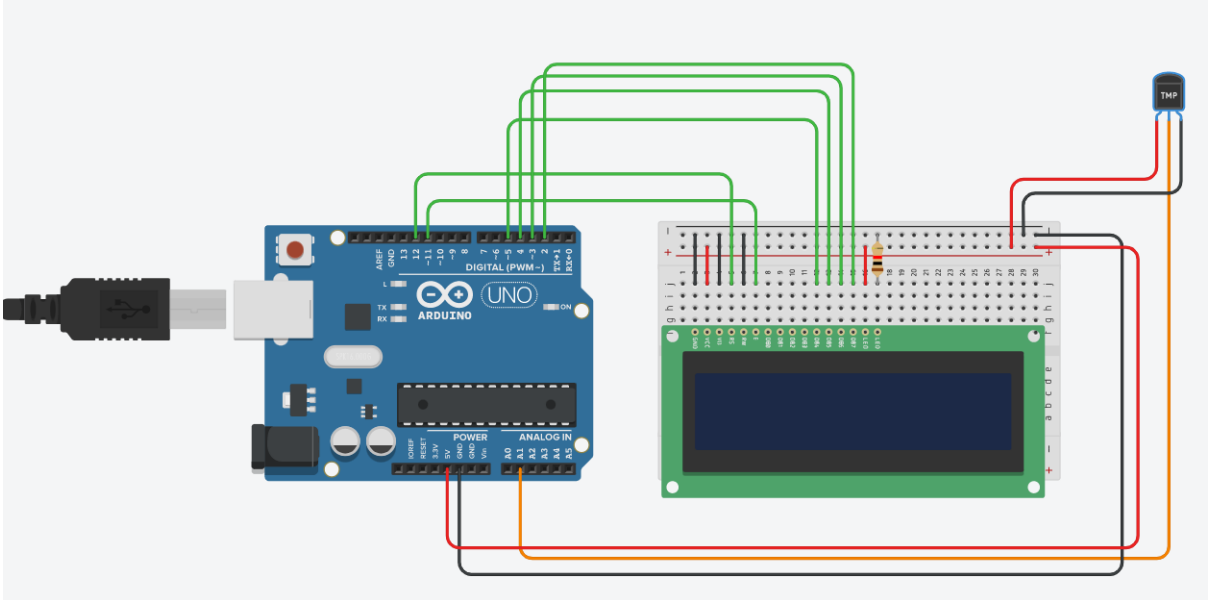
Component Used:

1. Arduino UNO
2. Jumper wires
3. Breadboard
4. Temperature sensor
5. LED
6. Resistor

Schematic diagram:



Electric Wiring Diagram

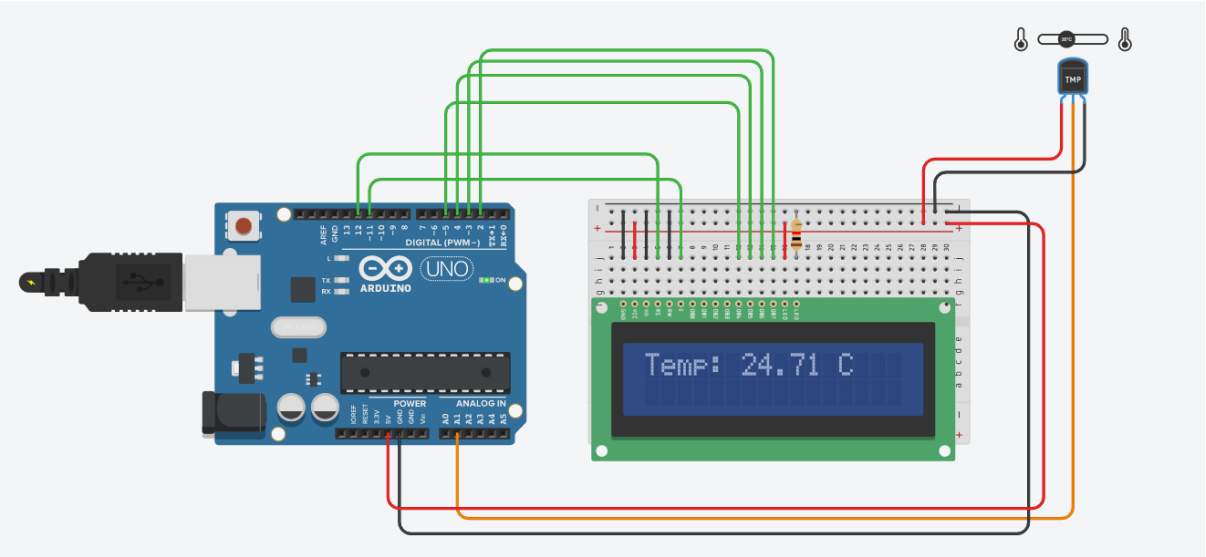


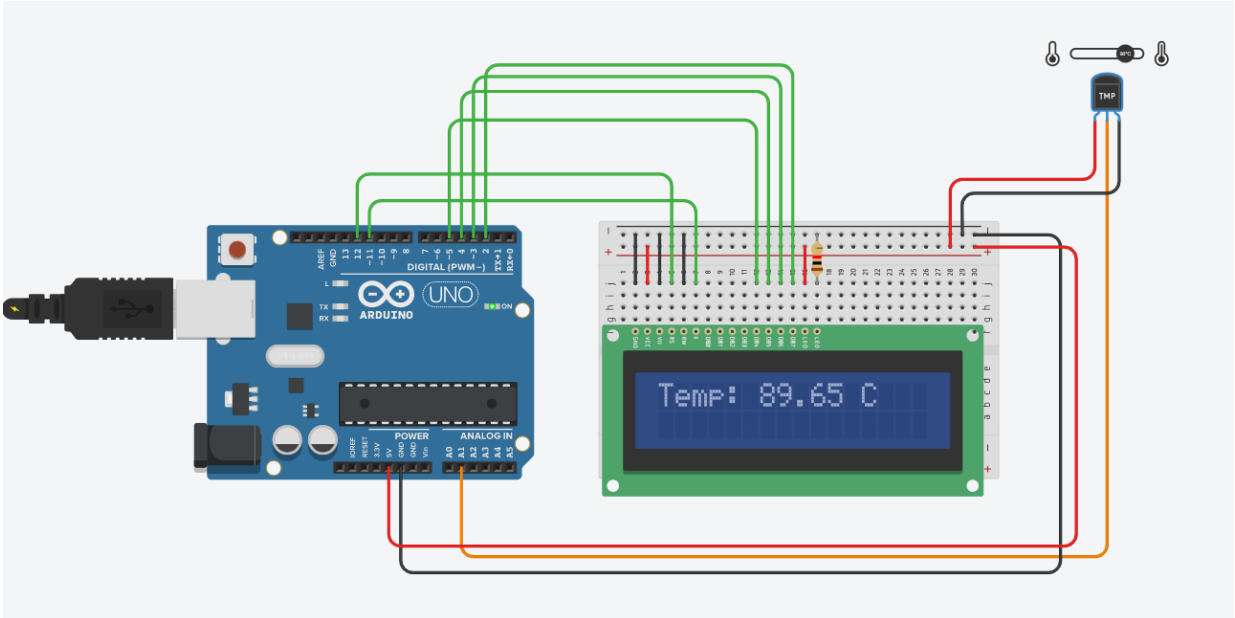
Program:

```
#include<LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
float celsius;
int temp = A1;
void setup(){
pinMode(temp,INPUT);
}
void loop(){
celsius = analogRead(temp)*0.004882814;
celsius = (celsius - 0.5) * 100.0;
lcd.setCursor(0,1);
```

```
lcd.print("Temp: ");
lcd.print(celsius);
lcd.print(" C");
delay(1000);
lcd.clear();
}
```

Snapshot of the Output:





Inference

From the given experiment the temperature is calculated and as we can see for increasing the temperature the reading changes and thereby it is displayed

Result:

Different temperatures are recorded and displayed.

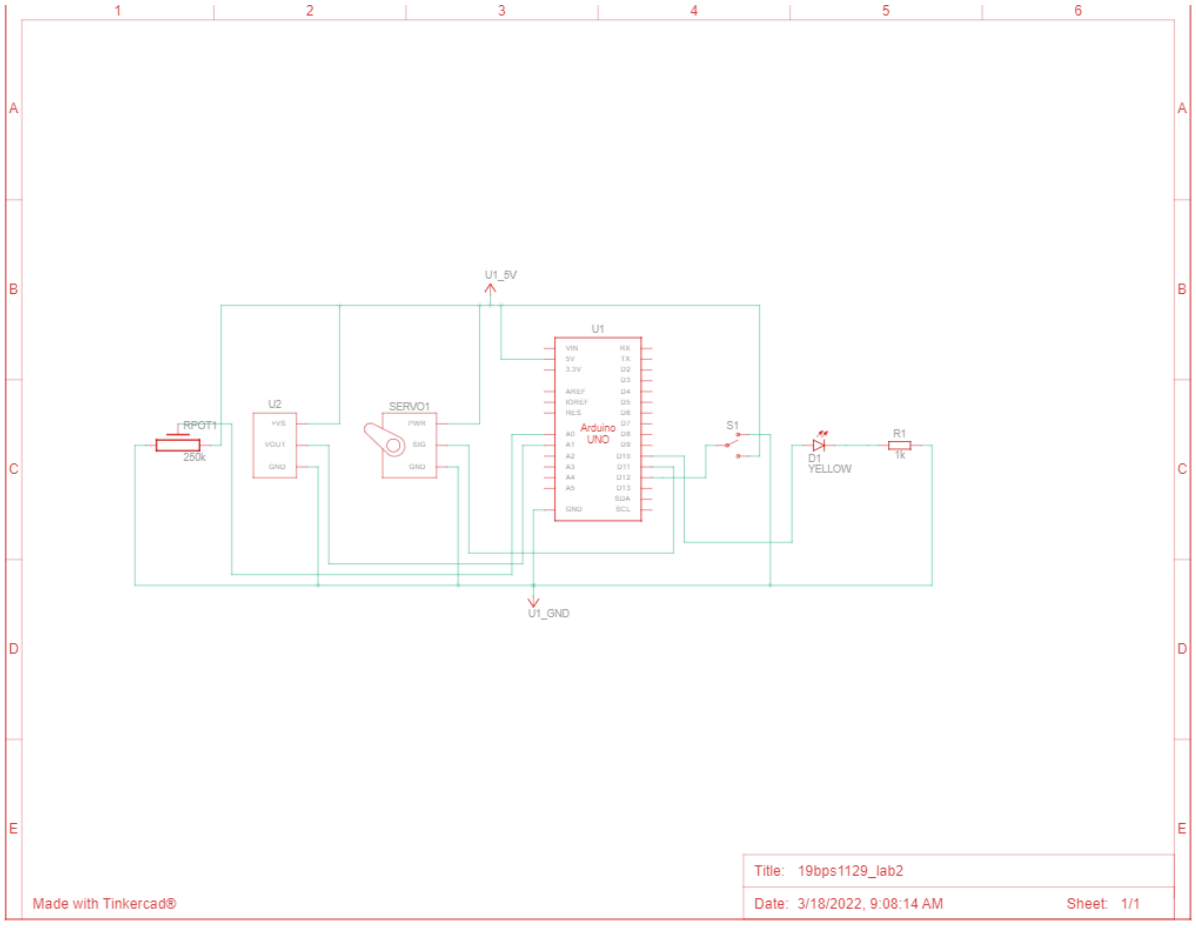
Exp.2. Interfacing digital temperature sensor and thermal actuator

Aim: To interface digital temperature sensor and thermal actuator

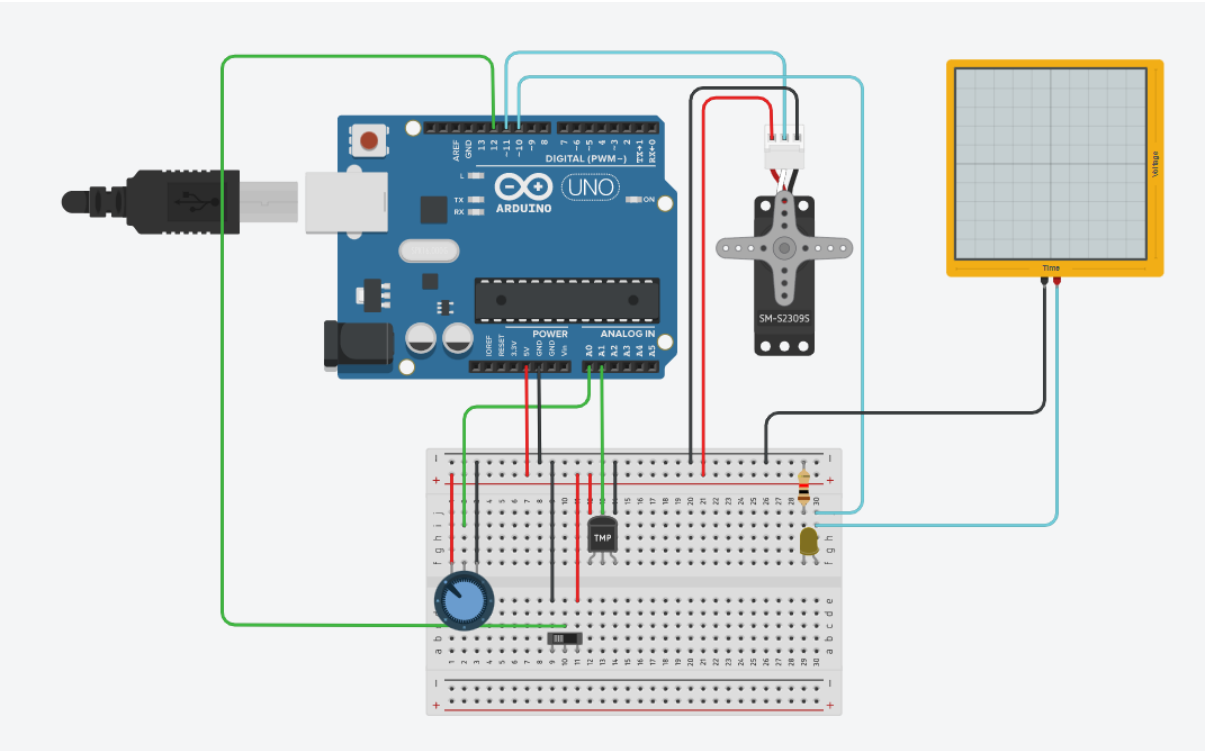
Component Used:

1. Arduino UNO
2. Micro Servo motor
3. Oscilloscope
4. Temperature sensor
5. Potentiometer
6. Sideswitch
7. LED
8. Resistor
9. Breadboard
10. Jumper Wires

Schematic diagram:



Electric Wiring Diagram

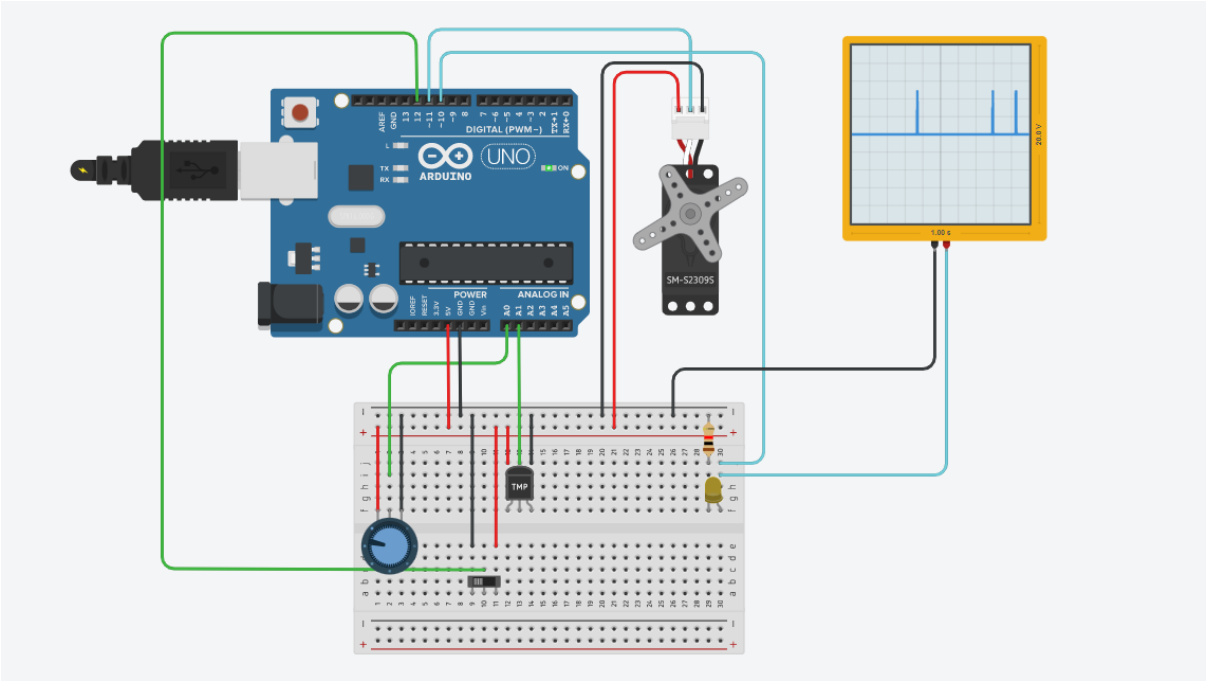


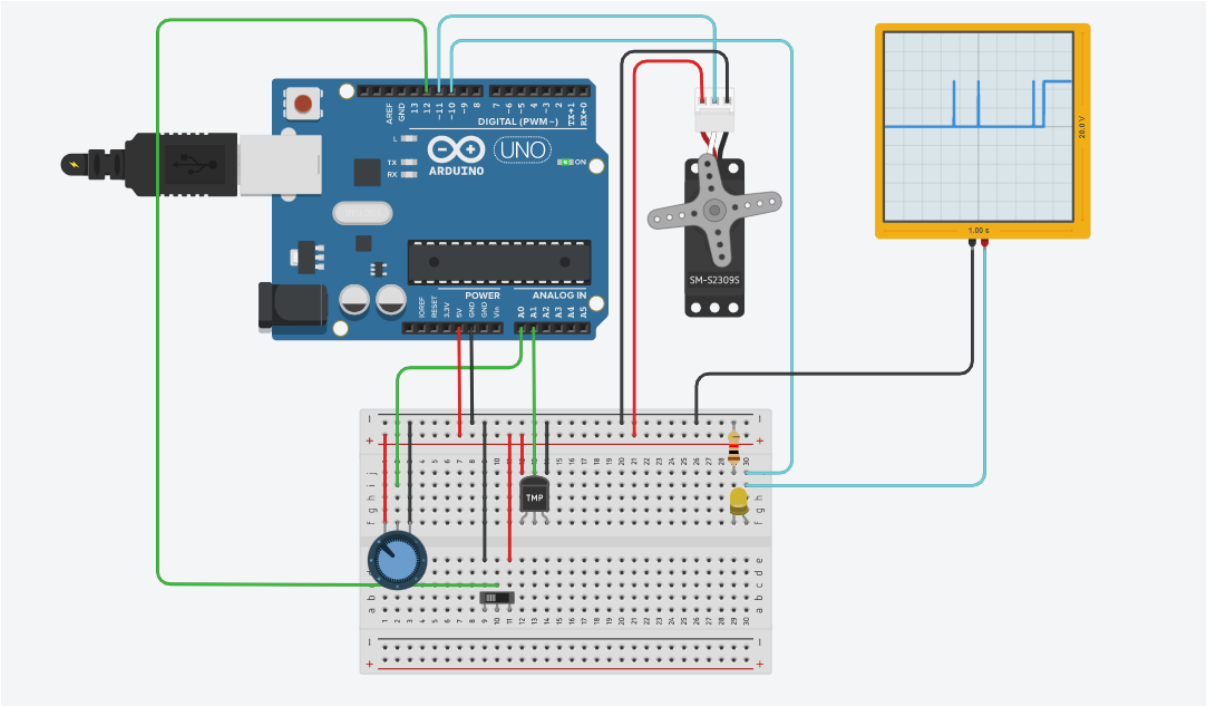
Program:

```
# include <Servo.h>


int reading = 0;

int duty;

int angle;


Servo servo_11;


void setup()

{

pinMode(12, INPUT);

pinMode(A0, INPUT);
```

```
pinMode(10, OUTPUT);
servo_11.attach(11);
pinMode(A1, INPUT);
}

void loop()
{
if (digitalRead(12) == 0)
{ reading = analogRead(A0);
duty= map(reading,0,1023,0,255);
analogWrite(10, duty);
angle= map(reading,0,1023,0,180);
servo_11.write(angle);
}
else
{ reading = analogRead(A1);
duty= map(reading,20,359,0,255);
analogWrite(10, duty);
angle= map(reading,20,359,0,180);
servo_11.write(angle);
}
delay(100);
}
```

Snapshot of the Output:





Inference

We can see that on rotating the potentiometer the resultant graph on the oscilloscope changes and so does the glowing of LED

Result:

Digital temperature sensor and thermal actuator are interfaced.

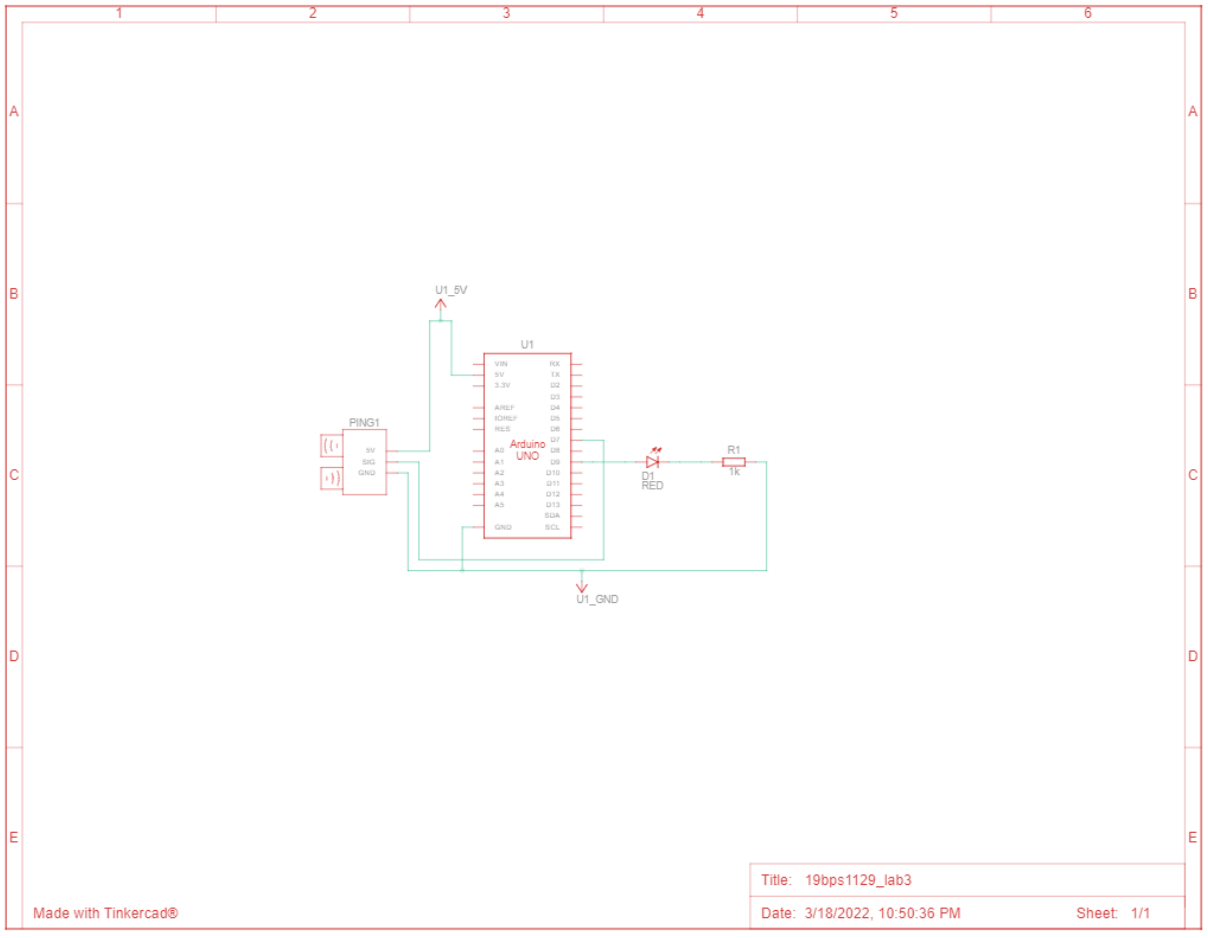
Exp.3. Interfacing optical sensor and A/D conversion of sensor output

Aim: To interface optical sensor and A/D conversion of sensor output

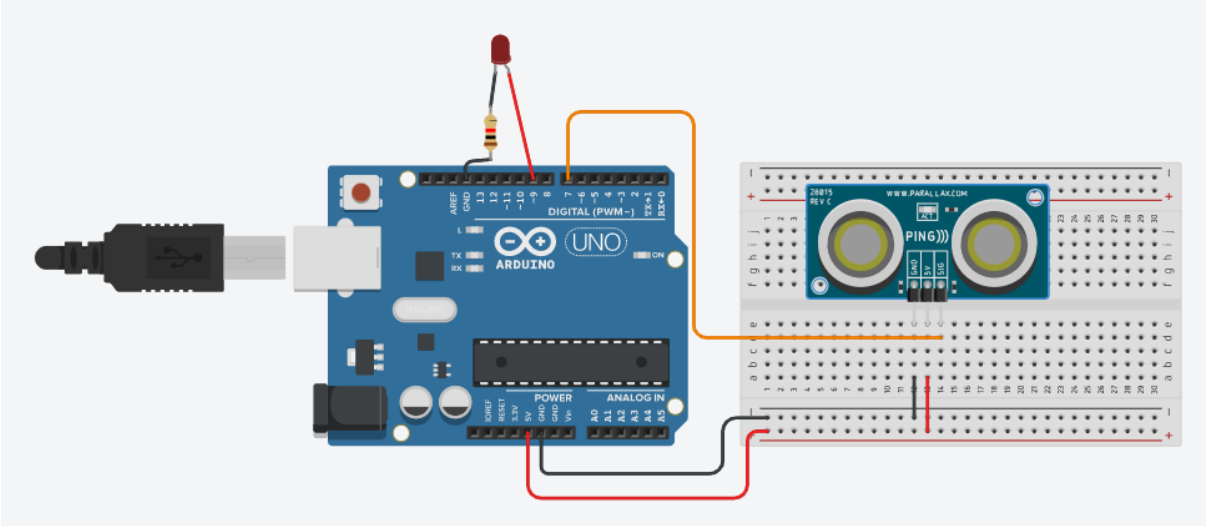
Component Used:

1. Arduino UNO
2. Ultrasonic distance sensor
3. LED
4. Resistor
5. Jumper wires
6. Breadboard

Schematic diagram:



Electric Wiring Diagram



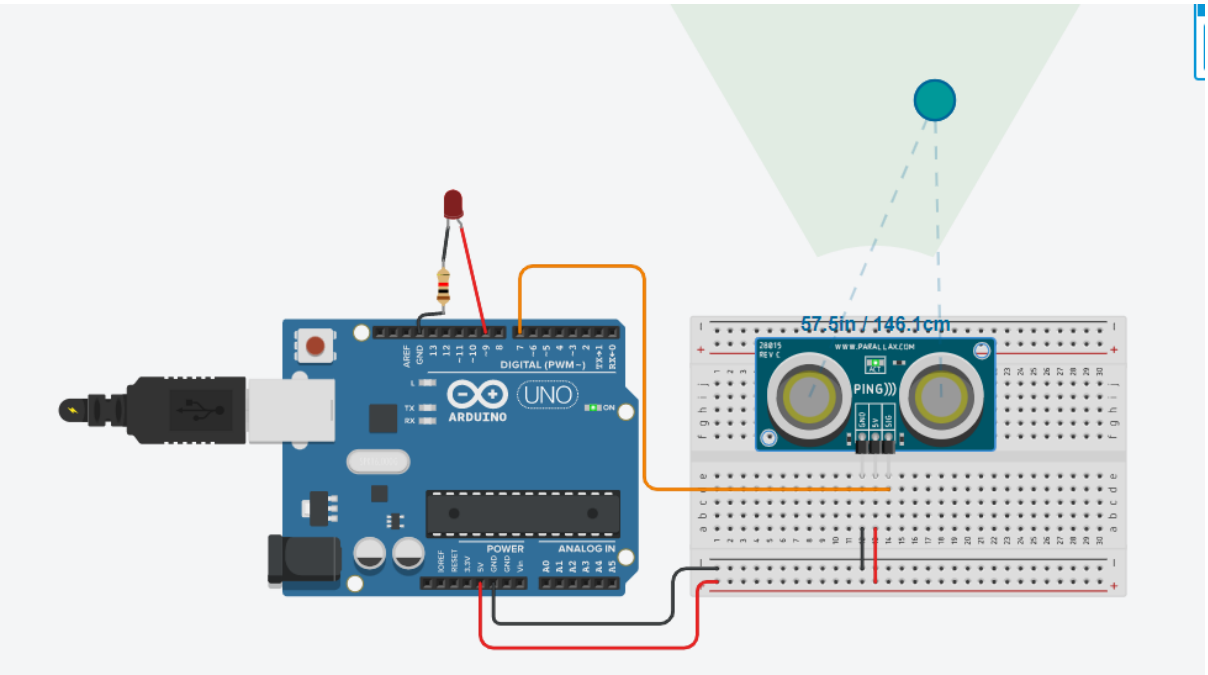
Program:

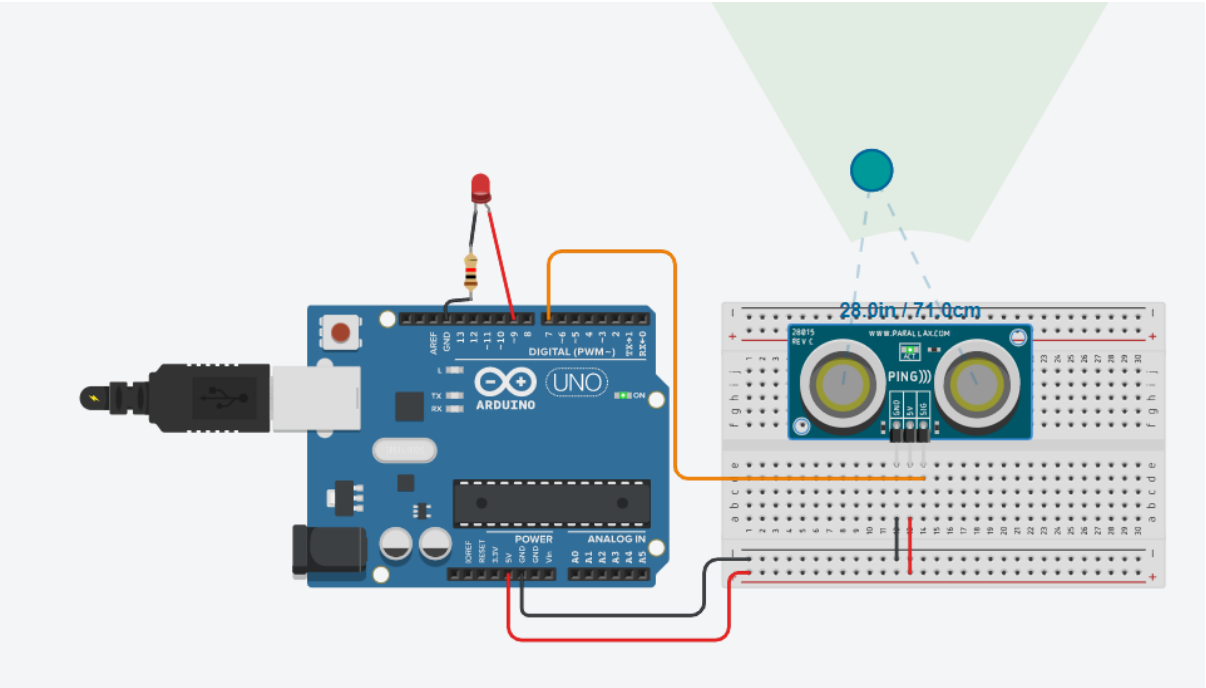
```
const int pingPin = 7;

const int ledPin = 9;


void setup() {

// initialize serial communication:

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

//pinMode();

}


void loop() {

// establish variables for duration of the ping,

// and the distance result in inches and centimeters:

long duration, cm;
```

```
 // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

 // The same pin is used to read the signal from the PING))): a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

 // convert the time into a distance
cm = microsecondsToCentimeters(duration);

 // Print the distance
Serial.print("Distance: ");
Serial.print(cm);
Serial.print("cm");
Serial.println();

 // Turn on the LED if the object is too close:
if(cm < 100) {
```

```
digitalWrite(ledPin, HIGH);
}
else {
digitalWrite(ledPin, LOW);
}
delay(100);
}

long microsecondsToCentimeters(long microseconds) {
// The speed of sound is 340 m/s or 29 microseconds per centimeter.
// The ping travels out and back, so to find the distance of the
// object we take half of the distance travelled.
return microseconds / 29 / 2;
}
```

Snapshot of the Output:





Inference

From the above experiment it is observed that for a certain distance the led does not glow and only when the distance crosses a certain value, it glows.

Result:

Therefore optical sensor interfacing and A/D conversion of sensor output is shown.

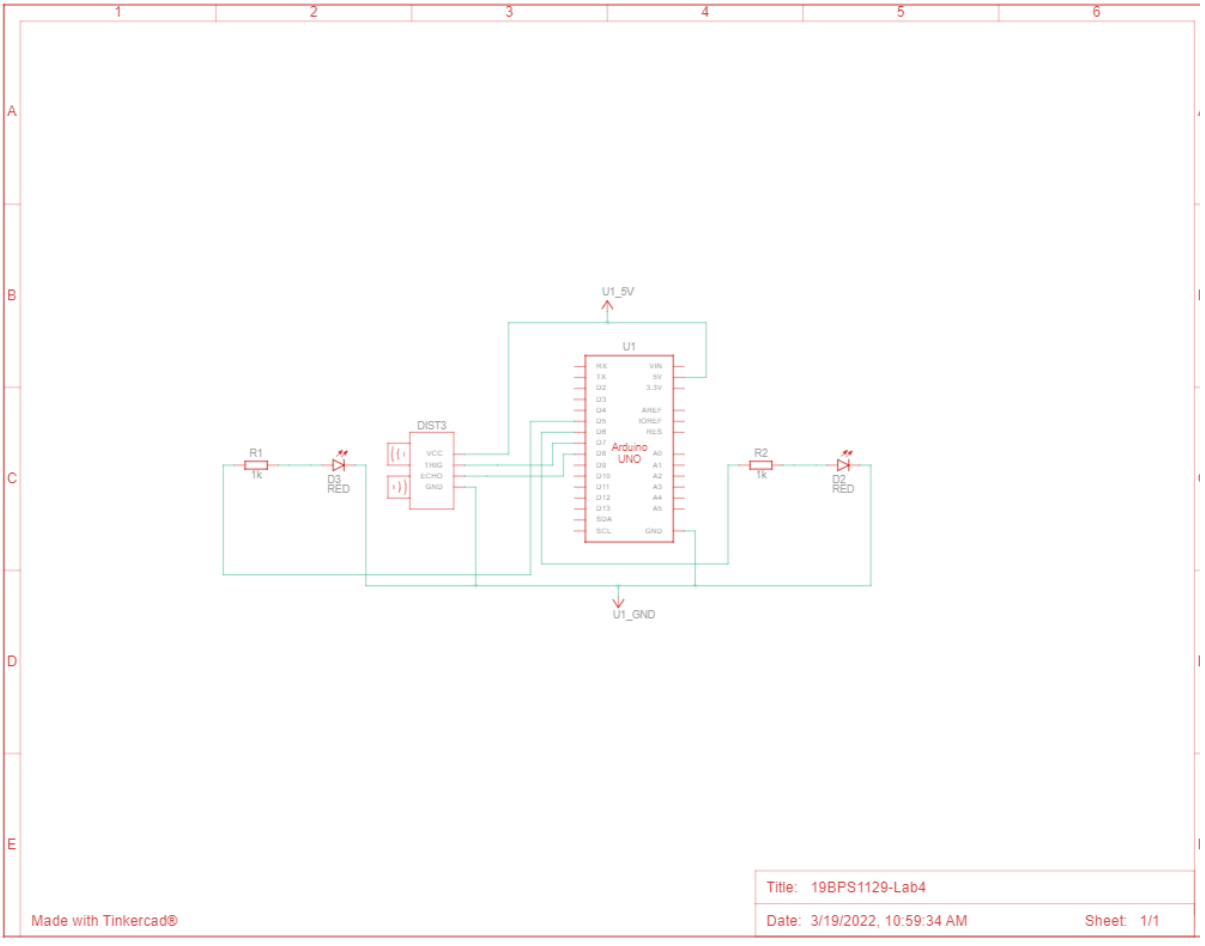
Exp.4. Interfacing optical actuators

Aim: To interface optical actuator

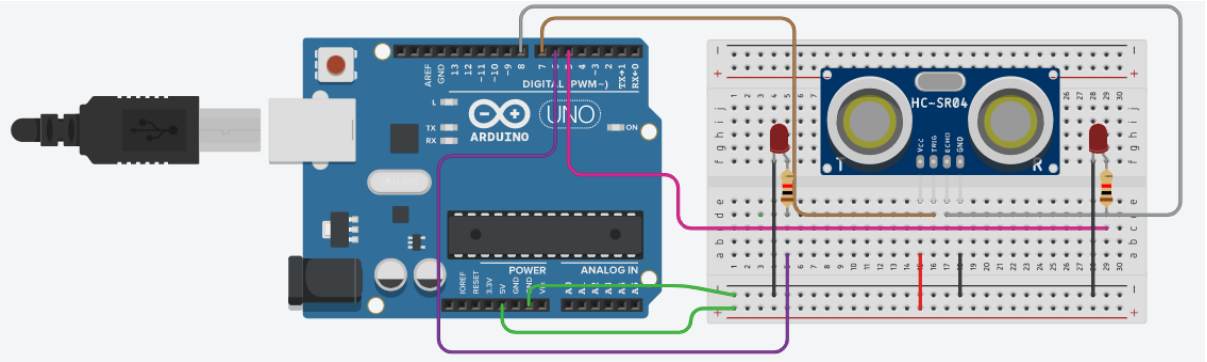
Component Used:

1. Arduino UNO
2. Ultrasonic distance sensor
3. Resistors
4. LED
5. Breadboard
6. Jumper wires

Schematic diagram:



Electric Wiring Diagram

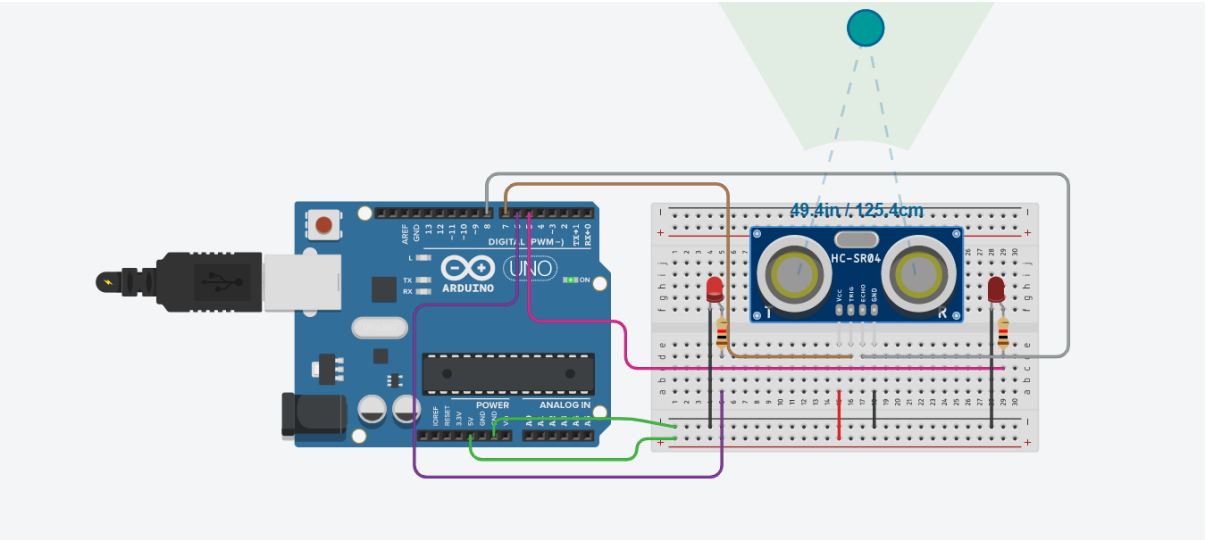


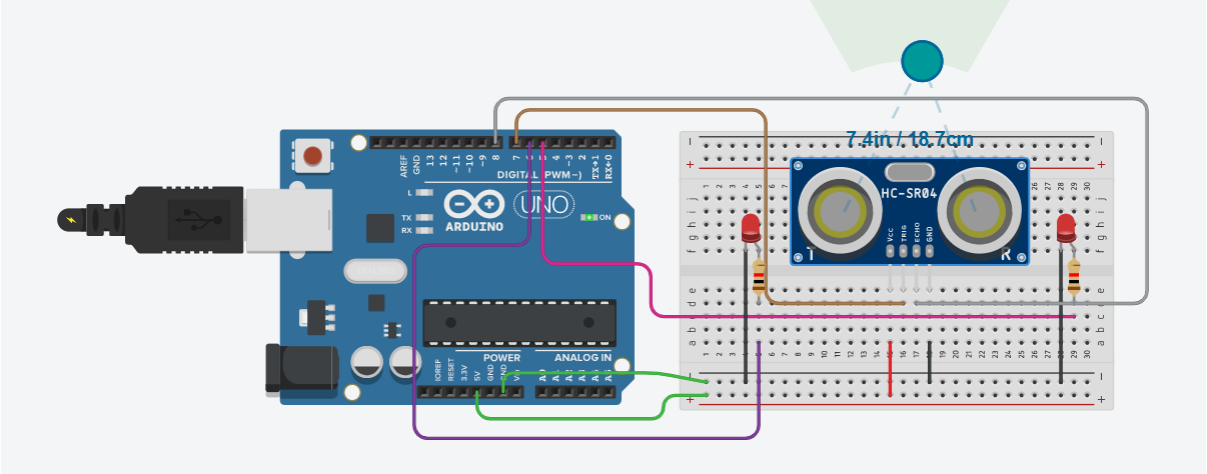
Program:

```
// C++ code
//
int sensor = 0;


long readUltrasonicDistance(int triggerPin, int echoPin)
{
pinMode(triggerPin, OUTPUT); // Clear the trigger
digitalWrite(triggerPin, LOW);
delayMicroseconds(2);
// Sets the trigger pin to HIGH state for 10 microseconds
digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);
pinMode(echoPin, INPUT);
// Reads the echo pin, and returns the sound wave travel time in microseconds
return pulseIn(echoPin, HIGH);
}
```

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

 pinMode(6, OUTPUT);
pinMode(5, OUTPUT);
}

void loop()
{
sensor = 0.01723 * readUltrasonicDistance(7, 8);
Serial.println(sensor);
if (30 < sensor) {
digitalWrite(6, HIGH);
digitalWrite(5, LOW);
} else {
digitalWrite(5, HIGH);
}
delay(10); // Delay a little bit to improve simulation performance
}
```

Snapshot of the Output:





Inference

Therefore on changing the distance we see that one of the LEDs glows and for a certain distance both the LEDs glow.

Result:

Optical actuator is interfaced using LED and ultrasonic distance sensor

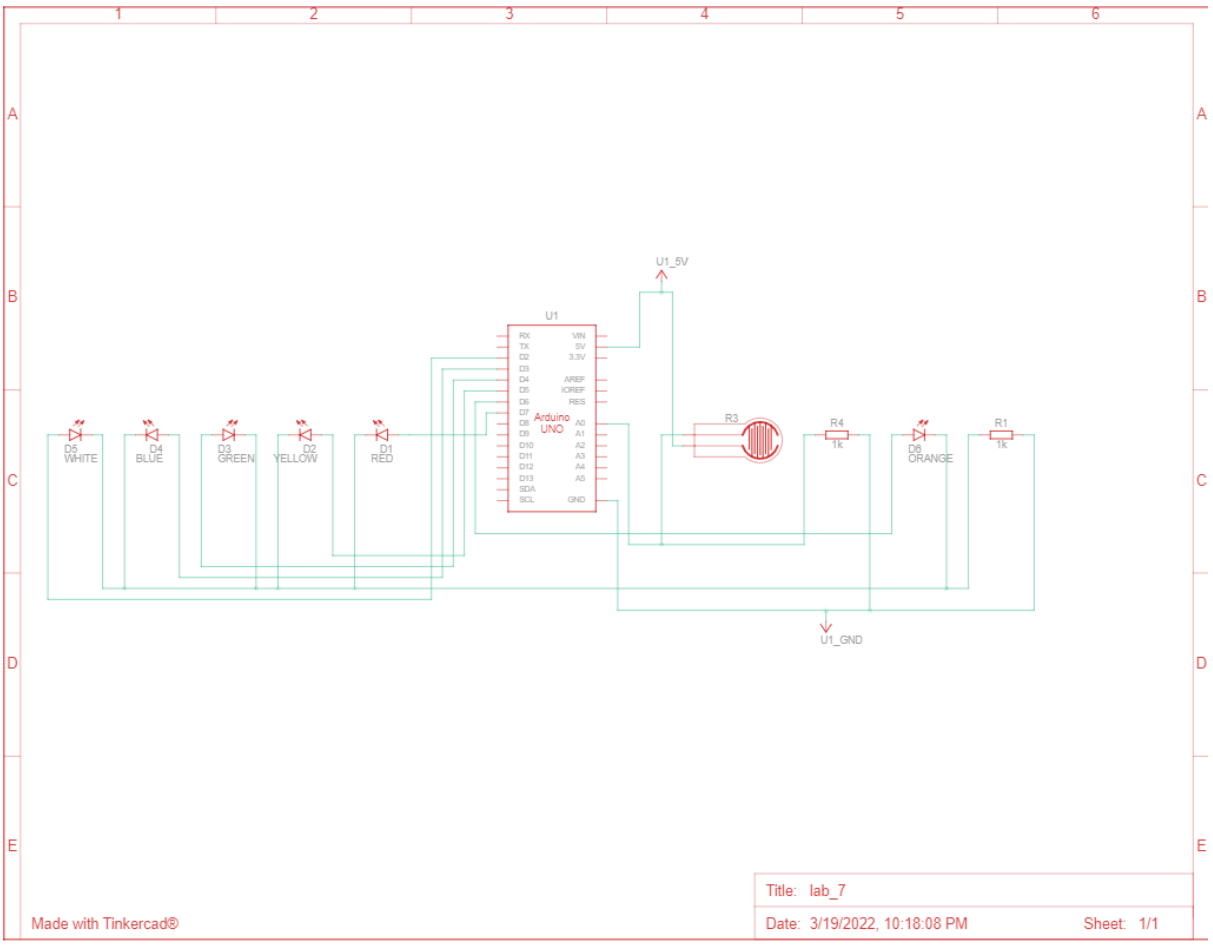
Exp.5. Interfacing force sensor and conversion of sensor output

Aim: To interface force sensor and conversion of sensor output

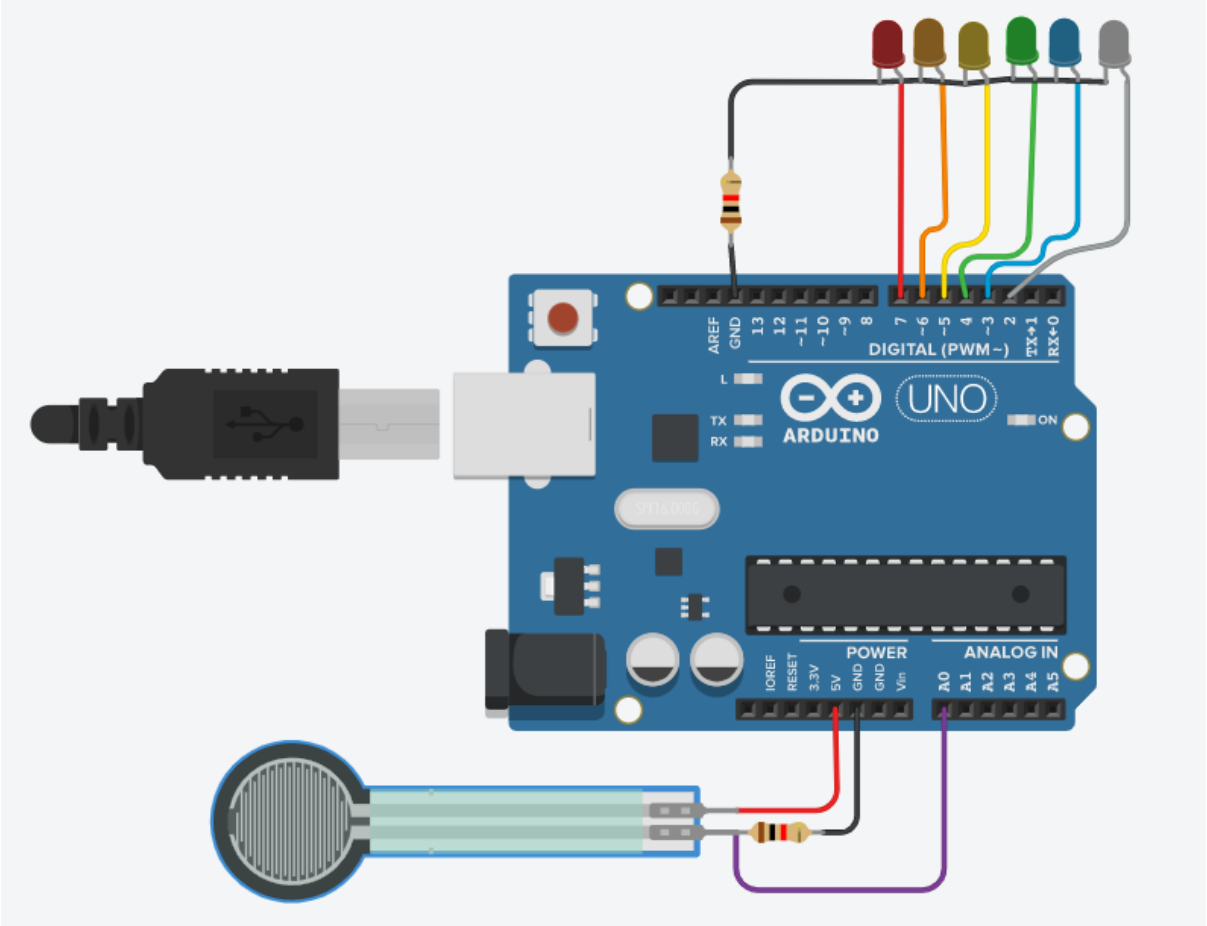
Component Used:

1. Arduino UNO
2. Force sensor
3. Resistors
4. LEDs
5. Breadboard
6. Jumper wires

Schematic diagram:



Electric Wiring Diagram



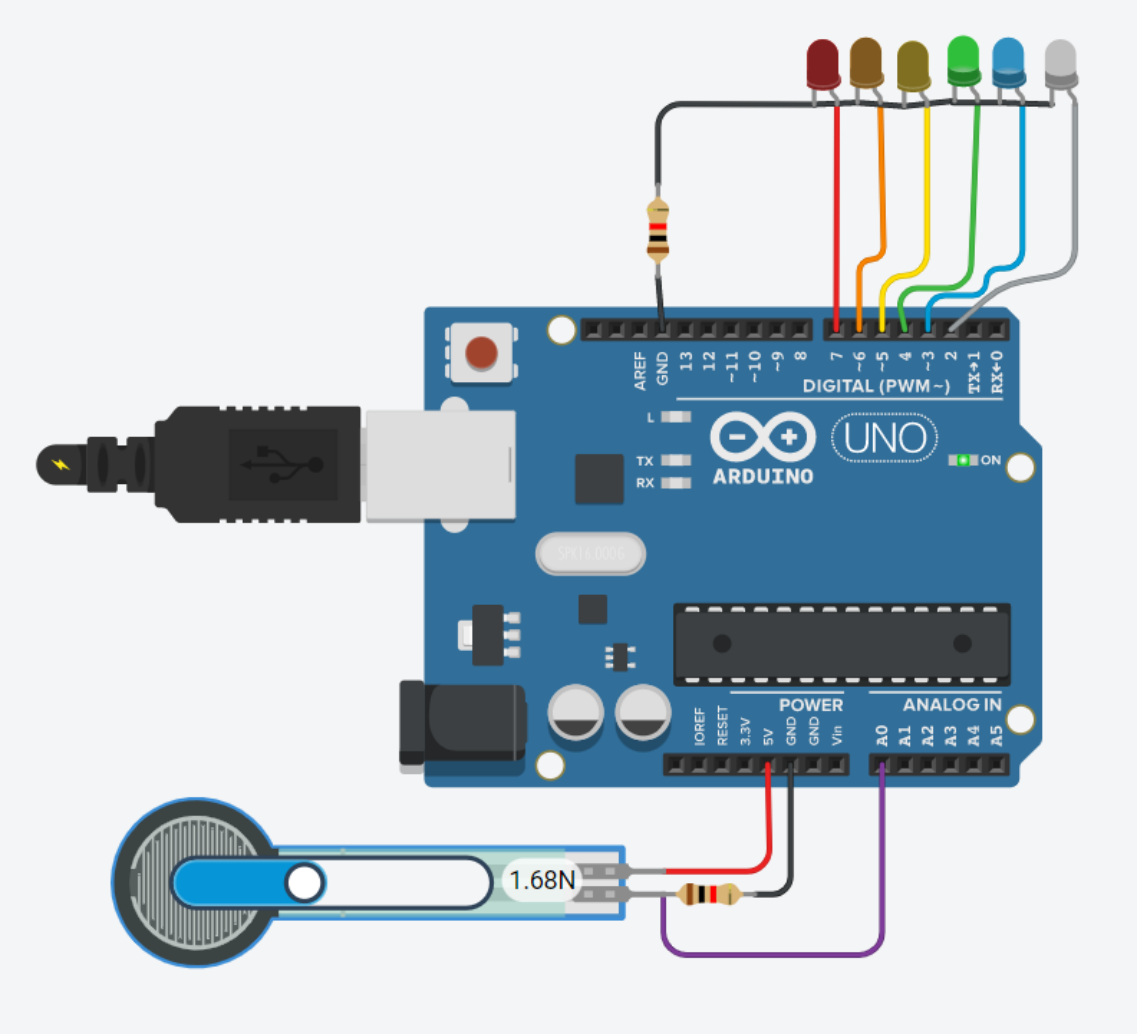
Program:

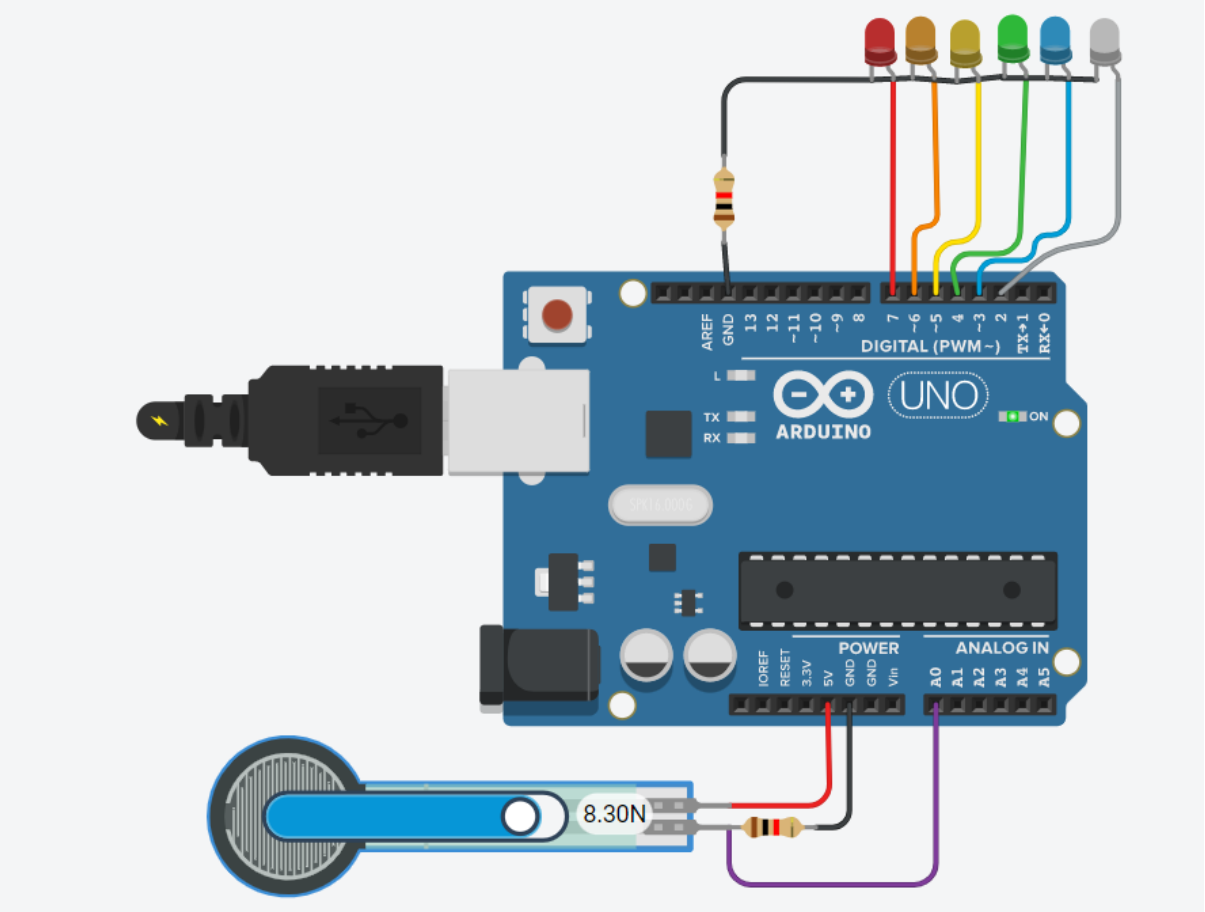
```
#define fsrpin A0

#define led1 2

#define led2 3

#define led3 4

#define led4 5

#define led5 6

#define led6 7


int fsrreading;

void setup() {

Serial.begin(9600);
```

```
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
pinMode(led5, OUTPUT);
pinMode(led6, OUTPUT);
}
void loop() {
fsrreading = analogRead(fsrpin);
Serial.println(fsrreading);
if (fsrreading > 50) {
digitalWrite(led1, HIGH);
}
else digitalWrite(led1, LOW);
if (fsrreading > 100) {
digitalWrite(led2, HIGH);
}
else digitalWrite(led2, LOW);
if (fsrreading > 150) {
digitalWrite(led3, HIGH);
}
else digitalWrite(led3, LOW);
if (fsrreading > 200) {
digitalWrite(led4, HIGH);
}
else digitalWrite(led4, LOW);
```

```
if (fsrreading > 250) {
digitalWrite(led5, HIGH);
}
else digitalWrite(led5, LOW);
if (fsrreading > 300) {
digitalWrite(led6, HIGH);
}
else digitalWrite(led6, LOW);
}
```

Snapshot of the Output:





Inference

We can infer that in changing the values of the force sensor the consecutive LED glows and thereby all the LED glows one after the another

Result:

Force sensor is interfaced and sensor output is converted

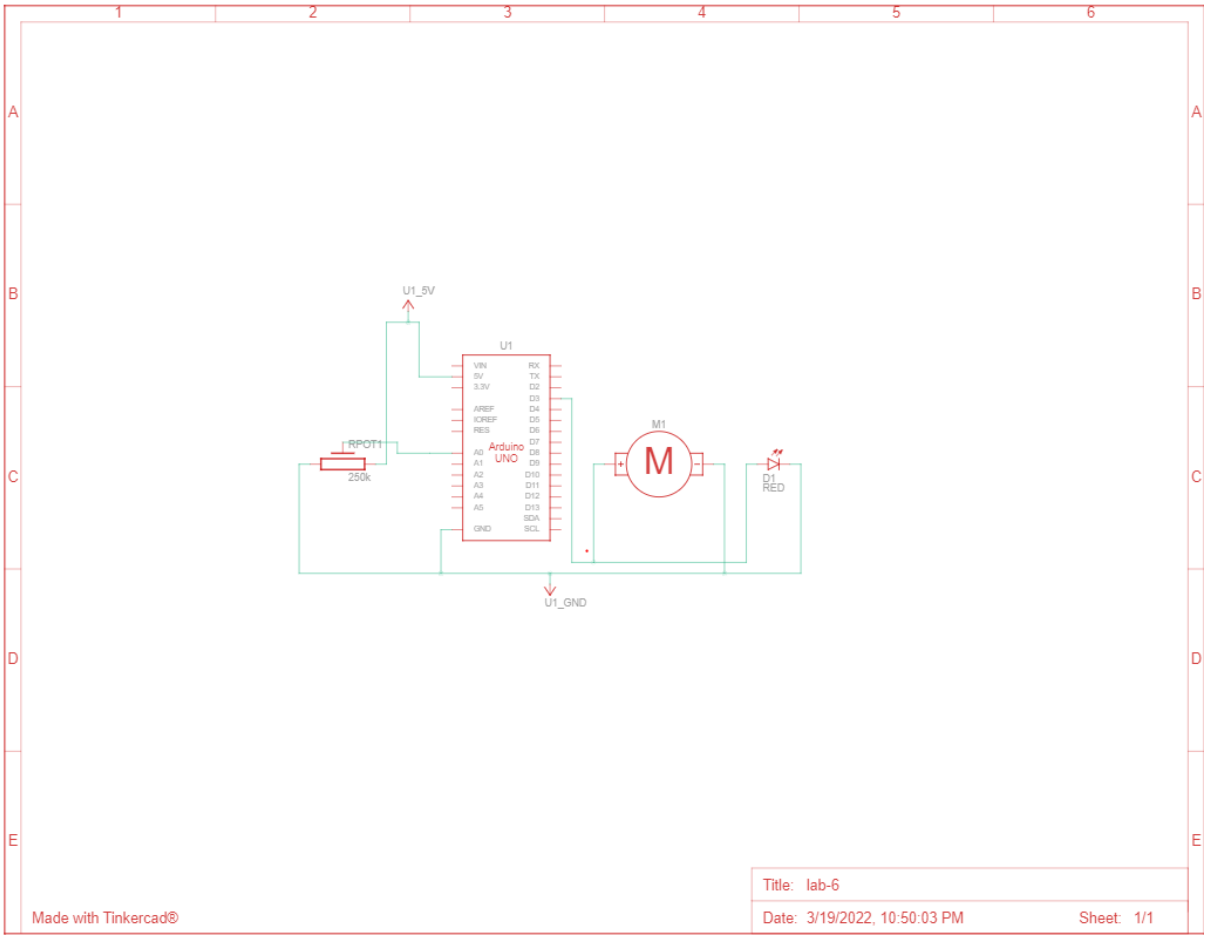
Exp.6. Interfacing velocity sensor and conversion of sensor output

Aim: To interface velocity sensor and conversion of sensor output

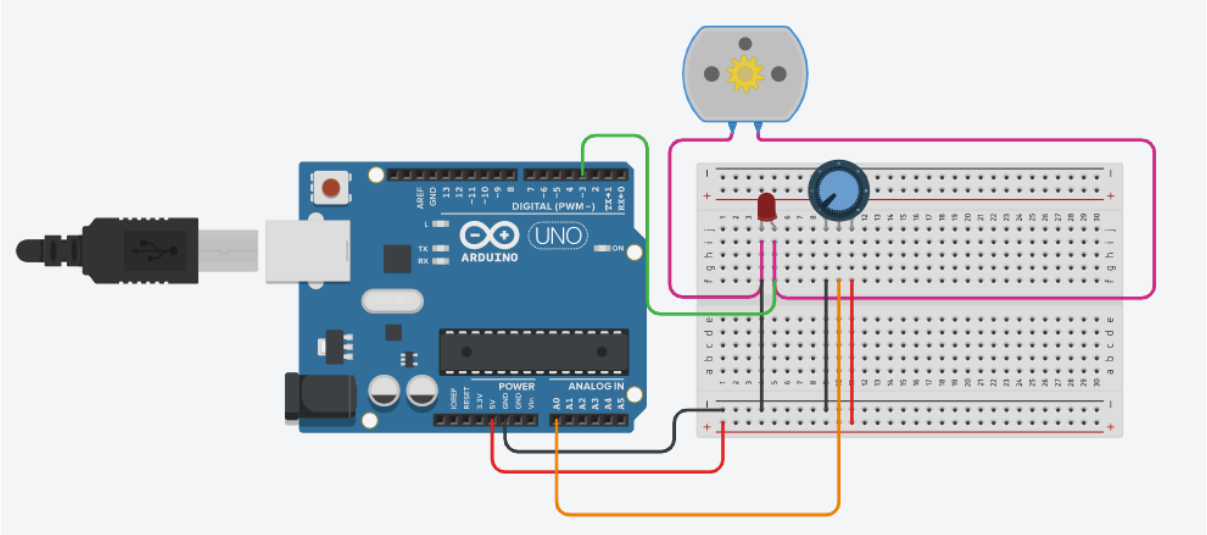
Component Used:

1. Arduino UNO
2. Potentiometer
4. DC Motor
5. LED
6. Breadboard
7. Jumper wires

Schematic diagram:



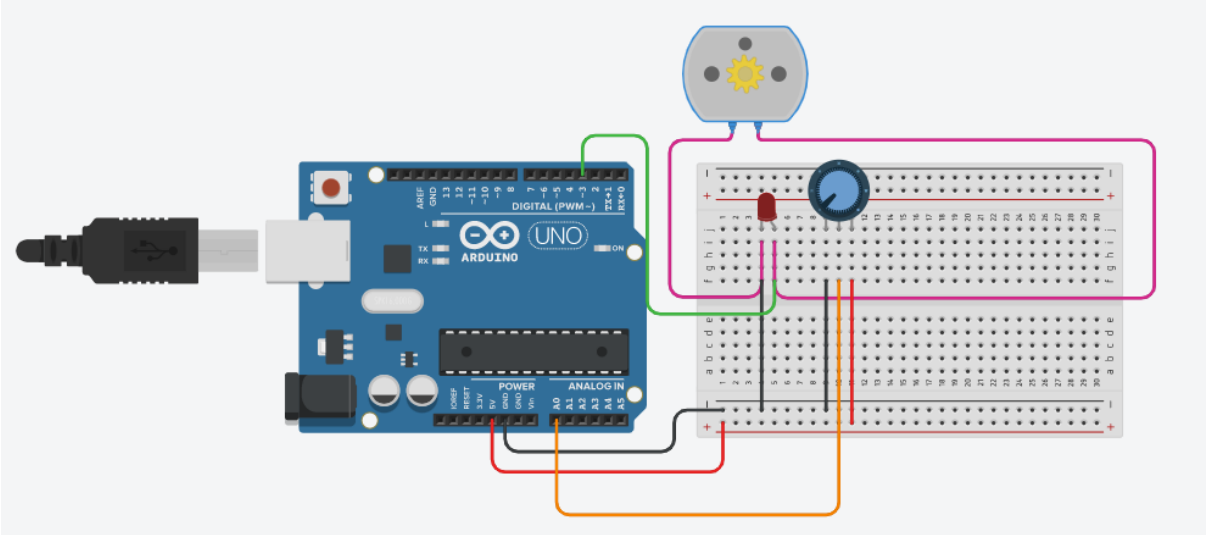
Electric Wiring Diagram

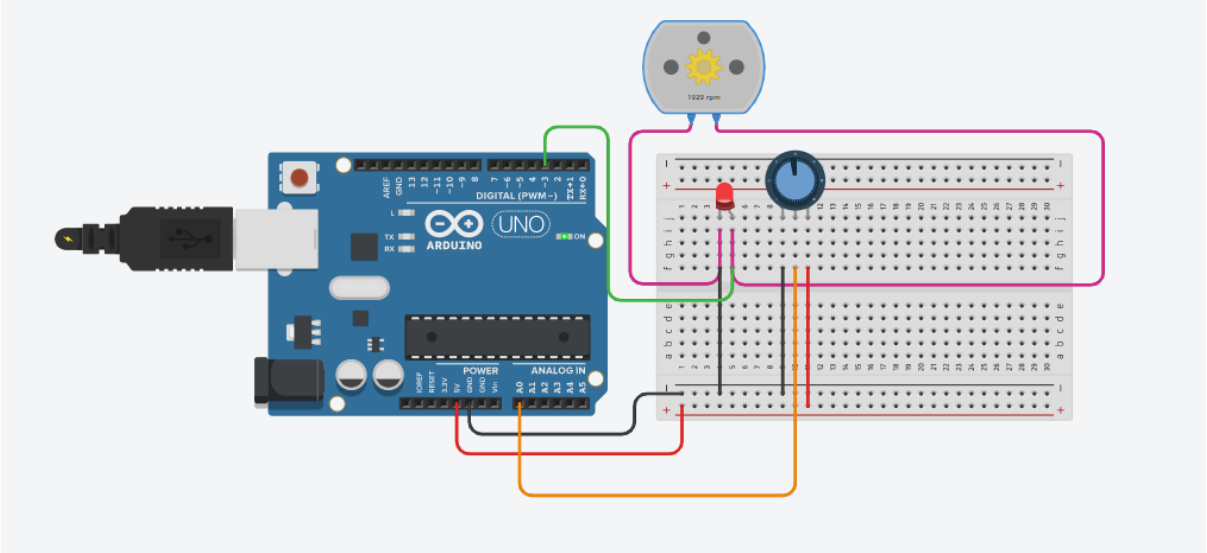


Program:

```
#define pinLed 3

#define pinPot A0


void setup() {

pinMode(pinLed, OUTPUT);

pinMode(pinPot, INPUT );

}


void loop() {

byte value = analogRead(pinPot)/4;

analogWrite(pinLed,value);

delay(10);

}
```

Snapshot of the Output:





Inference

Changing the values of the potentiometer the value of the DC motor changes and thereby the LED glows or dims accordingly

Result:

Velocity sensor and sensor output conversion is interfaced.

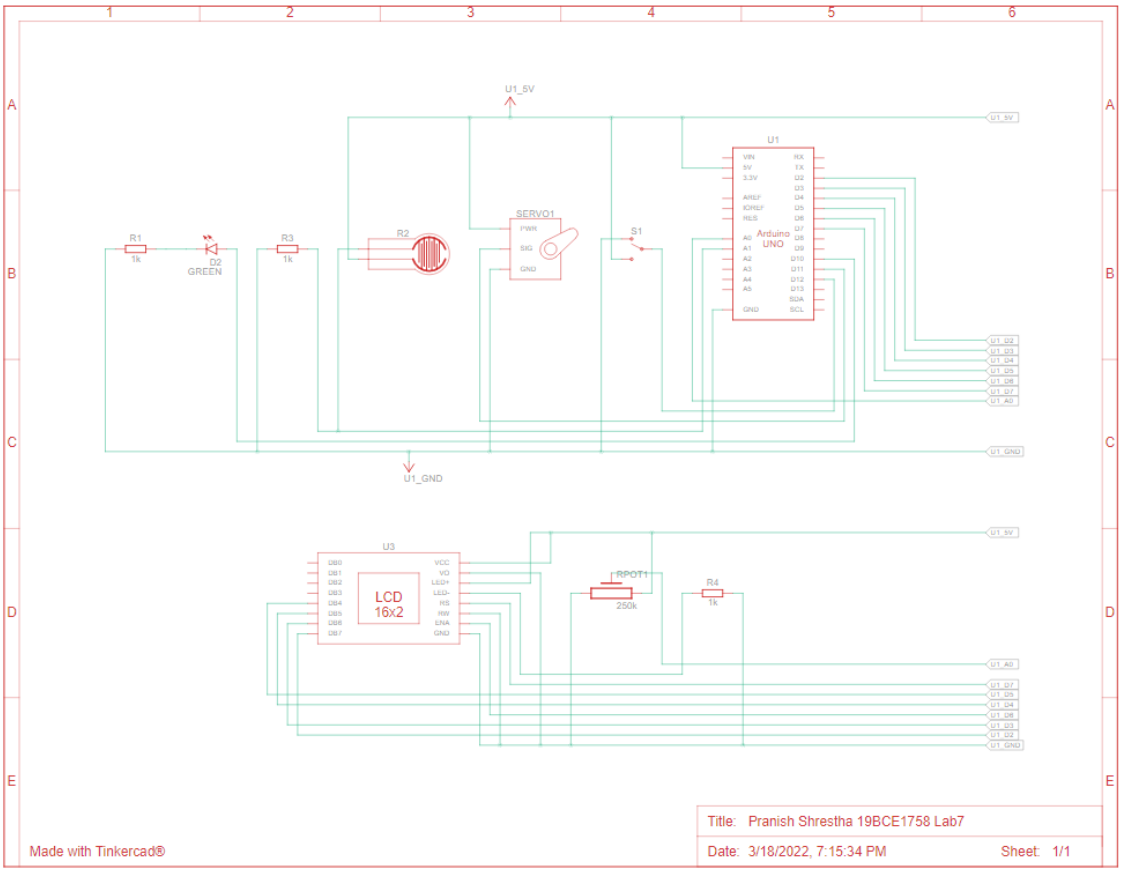
Exp.7. Interfacing of mechanical actuators

Aim: Show the working of force sensor in arduino

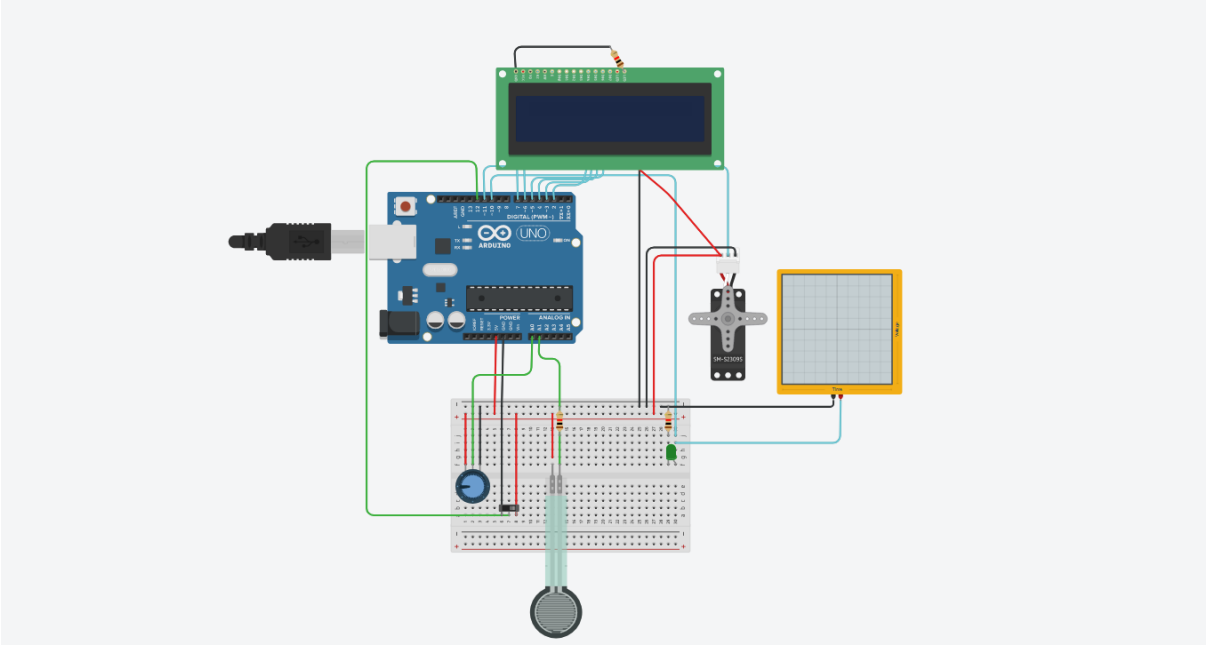
Component Used:

Arduino Uno R3, LCD, Breadboard, Oscilloscope, Micro Servo, Potentiometer, Slideswitch, Resistors, Wires and LED

Schematic diagram:



Electric Wiring Diagram

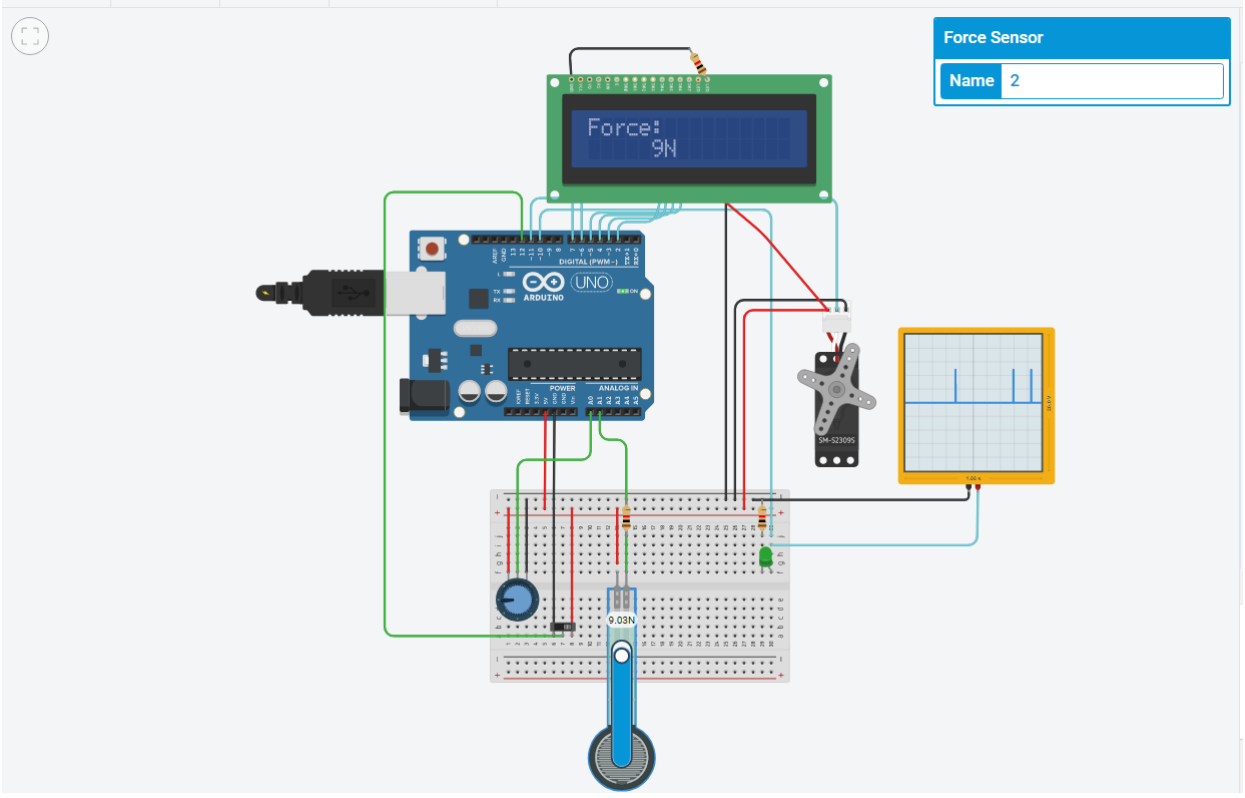


Program:

```
# include <Servo.h>
#include<LiquidCrystal.h>

int reading = 0;
int duty;
int angle;

const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

Servo servo_11;

void setup()
{
        pinMode(12, INPUT);
        pinMode(A0, INPUT);
        pinMode(10, OUTPUT);
        servo_11.attach(11);
        pinMode(A1, INPUT);
        lcd.begin(16,2);
   lcd.setCursor(0,0);
        lcd.print("Force: ");
}
```

```
void loop()
{
        if (digitalRead(12) == 0)
        {
        reading = analogRead(A0);
                duty= map(reading,0,1023,0,255);
                analogWrite(10, duty);
                angle= map(reading,0,1023,0,180);
        if(angle>180)
        angle=360-angle;
                servo_11.write(angle);
        }
        else
        {
        reading = analogRead(A1);
        int force = map(reading, 0, 466, 0, 10);
        lcd.setCursor(5,1);
                lcd.print(force);
        lcd.print('N');
                duty= map(reading,20,466,0,255);
                analogWrite(10, duty);
                angle= map(reading,20,466,0,255);
        if(angle>180)
        angle=360-angle;
                servo_11.write(angle);
        }
        delay(100);
}
```

Snapshot of the Output:



Inference

The resistance of the force sensor decreases when more force is applied to it.

Result:

We were able to successfully implement the force sensor in Arduino using TinkerCad