**1. Introduction to various sensors and various actuators & its Application. Example are as under**

**a) PIR Motion Sensor.**

**b) Rain Drop Sensor.**

**c) Moisture Sensor.**

**d) Temperature Sensor.**

**e) Touch Sensor.**

**f) Infrared Sensor.**

**g) Servo Motor.**

**h) RFID Sensor.**

**i) Bluetooth Module.**

**j) Wi-Fi Module.**

**a) PIR Motion Sensor:**

A PIR is a **passive infrared sensor**used to detect motion, so a PIR is a passive motion detector that waits for infrared temperature from body heat to trigger an activity. In other words, it can sense motion through changes in temperature.

**PIR Detector Components**

A PIR detector also consists of three main components:

1.PIR Sensor

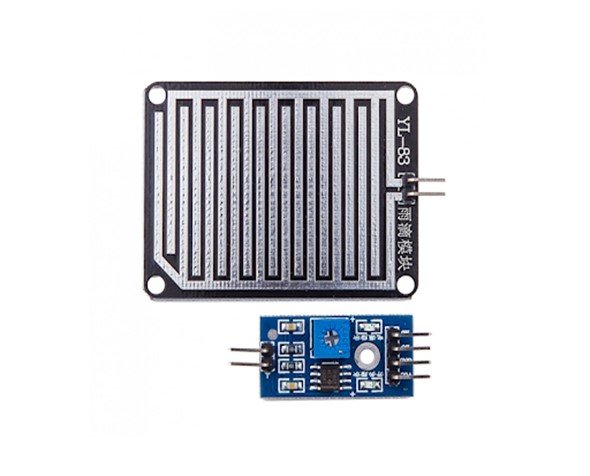
2.Lens

3.Printed Circuit Board



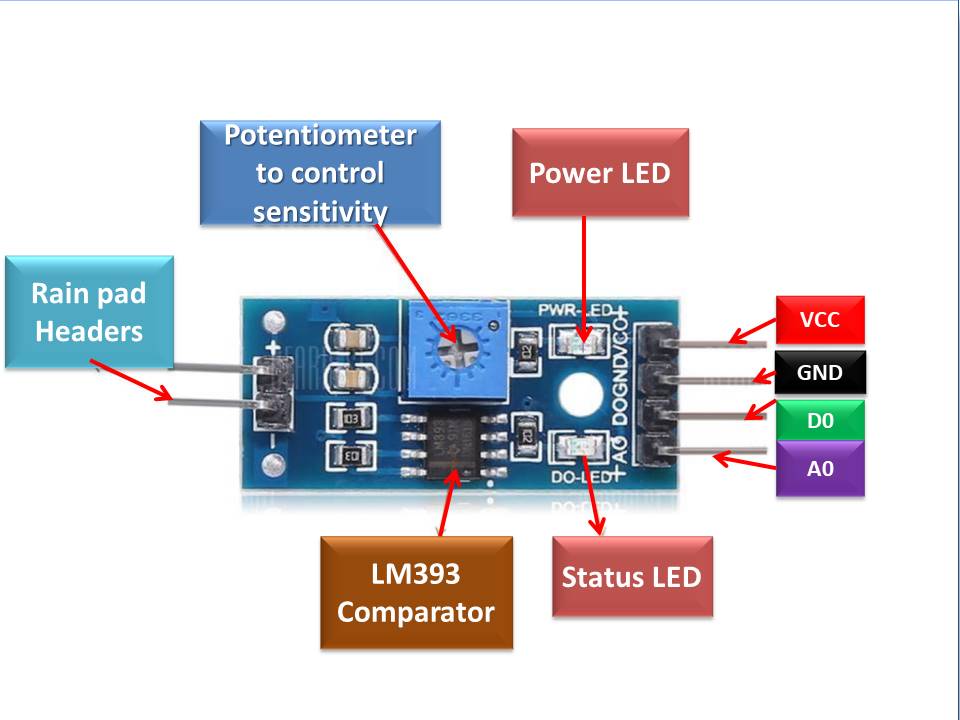
**b) Rain Drop Sensor:**

The rain drop sensor module is a smart and low-cost rain sensing device. It has two parts i.e. a rain sensing pad and a control board. The sensitive sensing pad detects any water present on it while the control board reads these signals and can also binarize them. The rain drop module has a major application in the automobile industry. It can be used to monitor the rain and send closure requests to shutters or windows whenever the rain is detected.



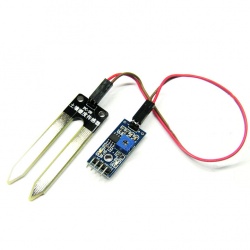
## ****Rain Drop Pinout:****

The rain drop control sensor is embedded with LM393 voltage comparator, current limiting resistors to adjust signal states and divide the voltage and capacitors as biasing elements.



**c) Moisture Sensor:**

The **Moisture sensor** is used to measure the water content(moisture) of soil.when the soil is having water shortage,the module output is at high level, else the output is at low level.This sensor reminds the user to water their plants and also monitors the moisture content of soil.It has been widely used in agriculture,land irrigation and botanical gardening.

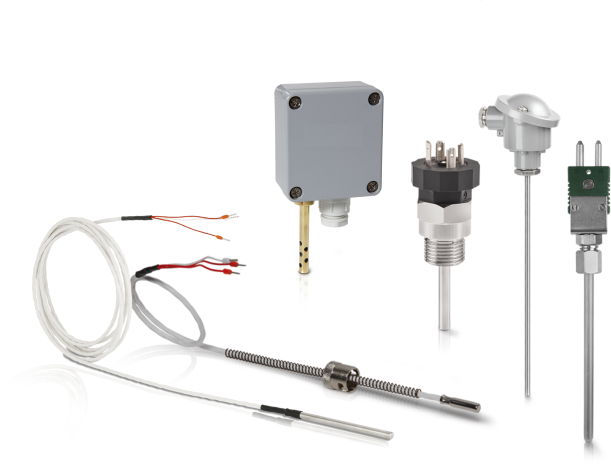


**d) Temperature Sensor.**

A temperature sensor is a device used to measure the temperature (hotness or coolness) of the air, a liquid, or solid matter.

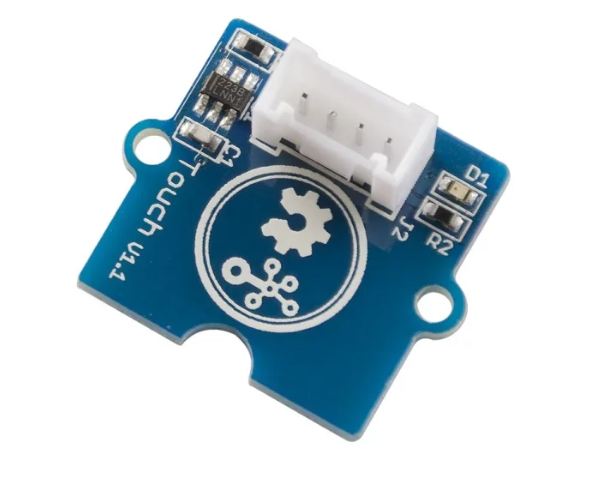
Temperature sensors work by providing a readable temperature measurement via electrical signals inside the probe or device. They contain two metals that generate an electrical voltage or resistance across the diode terminals when a temperature change occurs.

If there is a voltage increase, the temperature also rises, shortly followed by a voltage drop between the terminals and emitter in the diode.



**e) Touch Sensor:**

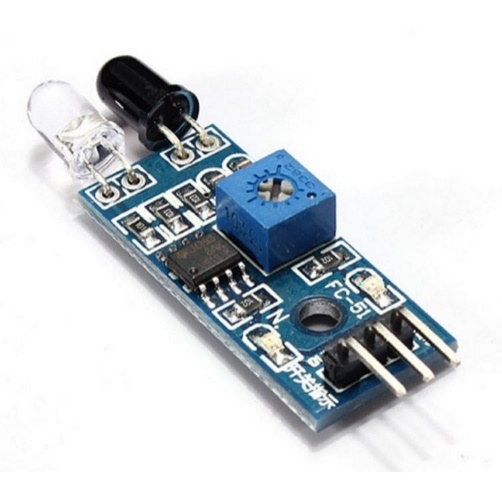
A touch sensor is an electronic sensor used in detecting and recording physical touch. Also known as tactile sensors, it’s a small, simple, low-cost sensor made to replace old mechanical switches we seen in the past.



**f) Infrared Sensor**

IR sensor is a simple electronic device which emits and detects IR radiation in order to find out certain objects/obstacles in its range. Some of its features are heat and motion sensing.

IR sensors use infrared radiation of wavelength between 0.75 to 1000µm which falls between visible and microwave regions of electromagnetic spectrum. IR region is not visible to human eyes. Infrared spectrum is categorized into three regions based on its wavelength i.e. Near Infrared, Mid Infrared, Far Infrared.



**g) Servo Motor**

The servo motor is an electric motor, which enables continuous determination of precise positions, speeds and torque via control electronics (servo controller). A software interface with the control electronics also allows precise parameterization and programming for actuation of the motor, which provides a high degree of dynamism and individuality.

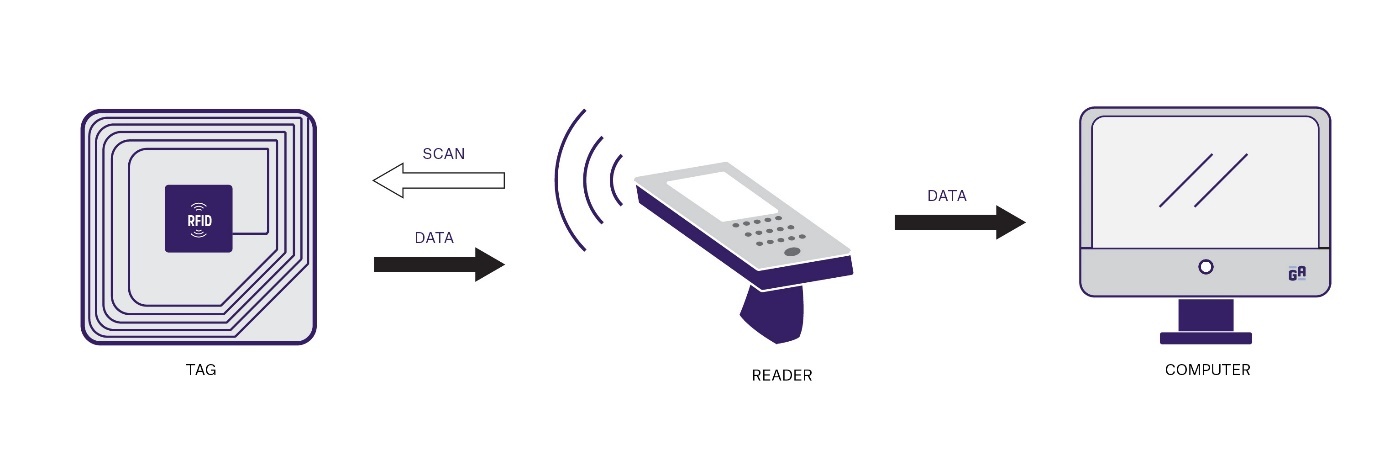


**h) RFID Sensor**

RFID (radio frequency identification) is a form of [wireless](https://www.techtarget.com/searchmobilecomputing/definition/wireless) communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object, animal or person.

There are two main types of RFID tags:

* **Active RFID.** An active RFID tag has its own power source, often a battery.
* **Passive RFID.** A passive RFID tag receives its power from the reading antenna, whose electromagnetic wave induces a current in the RFID tag's antenna.



**i) Bluetooth Module.**

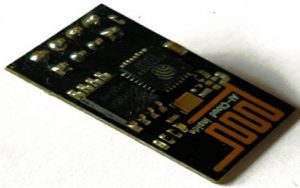
The Bluetooth device is fast and has a wide range of reach while consuming less power. It also has a powerful communication node that is well suited for IoT Sensors. Bluetooth has the ability to work without an active internet connection. It also has competency in providing a wide-scale device network through the Bluetooth mesh. These features of the Bluetooth modules have made it a popular choice in usage of the IoT sensor compared to other forms of connectivity like wifi.



**j) Wi-Fi Module.**

Wifi or wireless is most trending and highly used technology across the world. From television to mobile and from AC’s to home appliances, all are almost equipped with wifi or wireless technology.

An ESP8266 Wi-Fi module is a SOC microchip mainly used for the development of end-point IoT (Internet of things) applications. It is referred to as a standalone wireless transceiver, available at a very low price. It is used to enable the internet connection to various applications of embedded systems.



**2. Experiment using Arduino Uno to measure the distance of any object using Ultrasonic Sensor.**

/\*

https://arduinogetstarted.com/tutorials/arduino-ultrasonic-sensor

\*/

int trigPin = 9; // TRIG pin

int echoPin = 8; // ECHO pin

float duration\_us, distance\_cm;

void setup() {

// begin serial port

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// generate 10-microsecond pulse to TRIG pin

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// measure duration of pulse from ECHO pin

duration\_us = pulseIn(echoPin, HIGH);

// calculate the distance

distance\_cm = 0.017 \* duration\_us;

// print the value to Serial Monitor

Serial.print("distance: ");

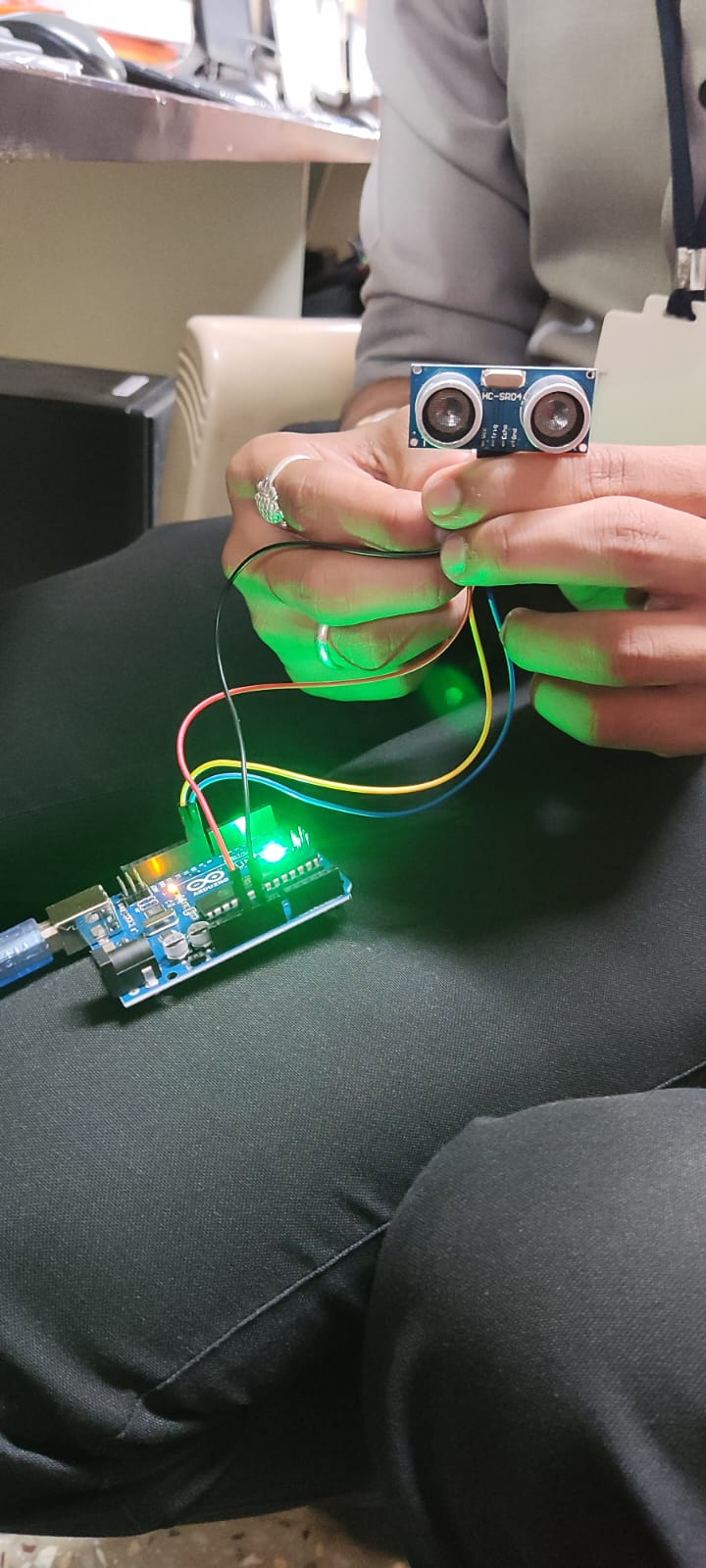
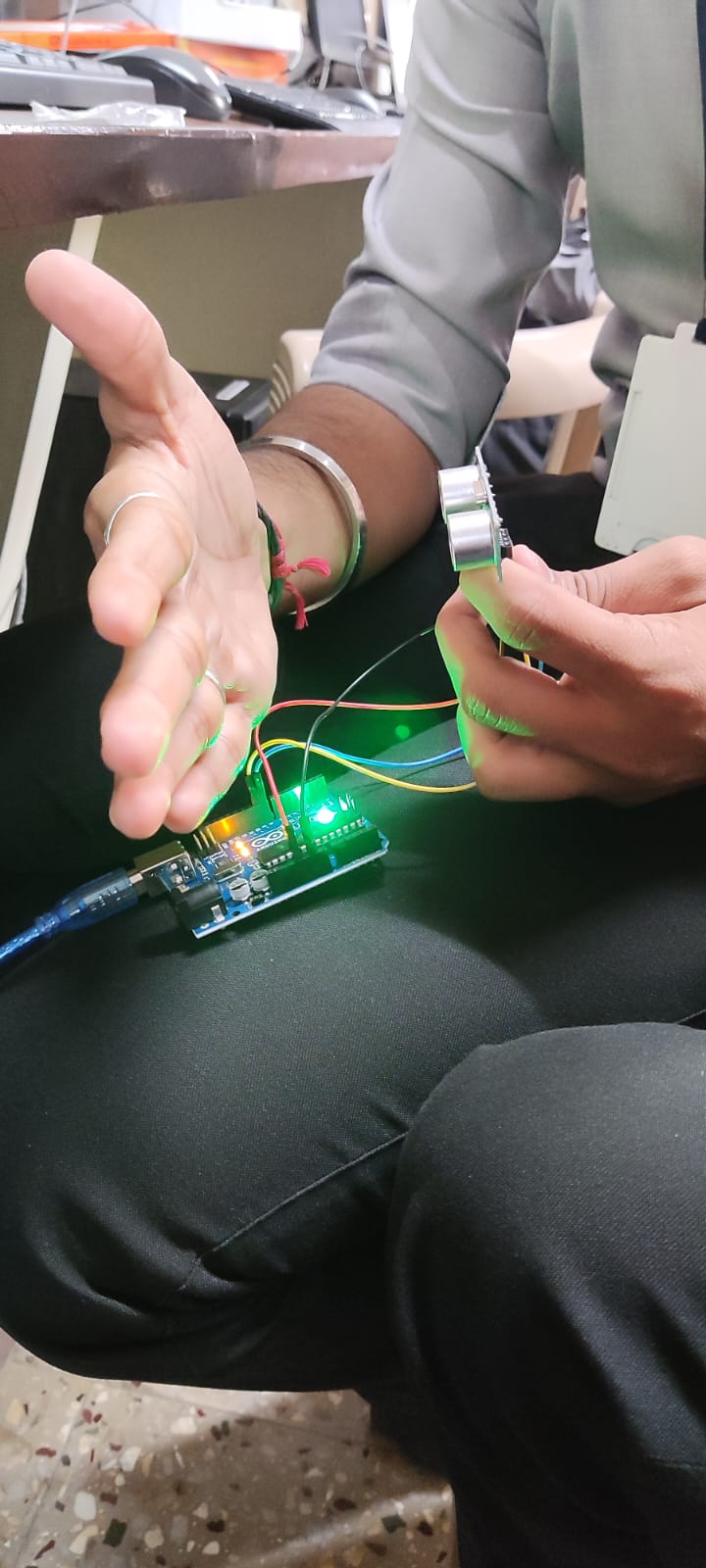
Serial.print(distance\_cm);

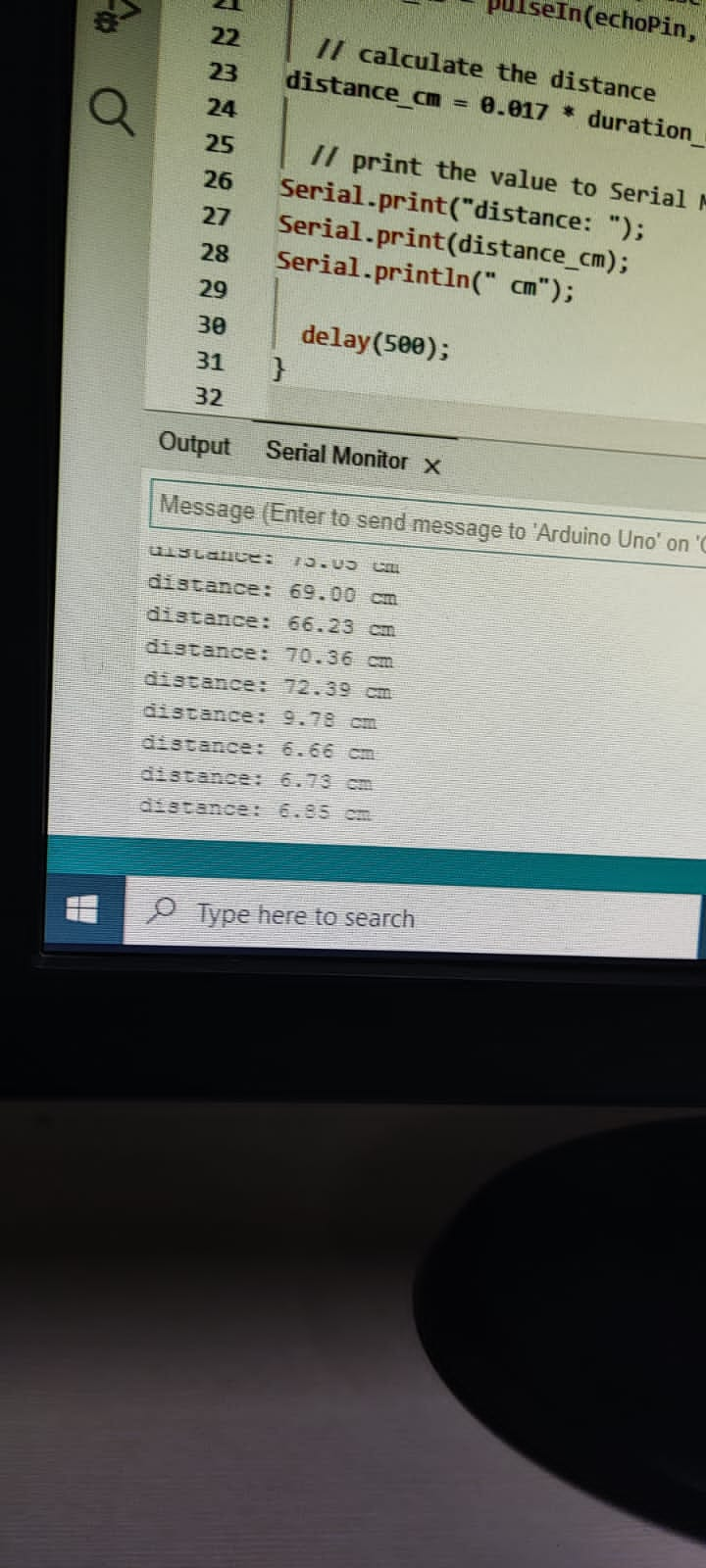
Serial.println(" cm");

delay(500);

}

**Output:-**

** **

****

**3. Create a circuit using Arduino and sensors. Perform experiment using Arduino to Learn Working of Servo Motor.**

/\* Servo motor with Arduino example code. Position and sweep. More info: https://www.makerguides.com/servo-arduino-tutorial/

\*/

// Include the servo library:

#include <Servo.h>

// Create a new servo object:

Servo myservo;

// Define the servo pin:

#define servoPin 9

// Create a variable to store the servo position:

int angle = 0;

void setup() {

// Attach the Servo variable to a pin:

myservo.attach(servoPin);

}

void loop() {

// Tell the servo to go to a particular angle:

myservo.write(90);

delay(1000);

myservo.write(180);

delay(1000);

myservo.write(0);

delay(1000);

// Sweep from 0 to 180 degrees:

for (angle = 0; angle <= 180; angle += 1) {

myservo.write(angle);

delay(15);

}

// And back from 180 to 0 degrees:

for (angle = 180; angle >= 0; angle -= 1) {

myservo.write(angle);

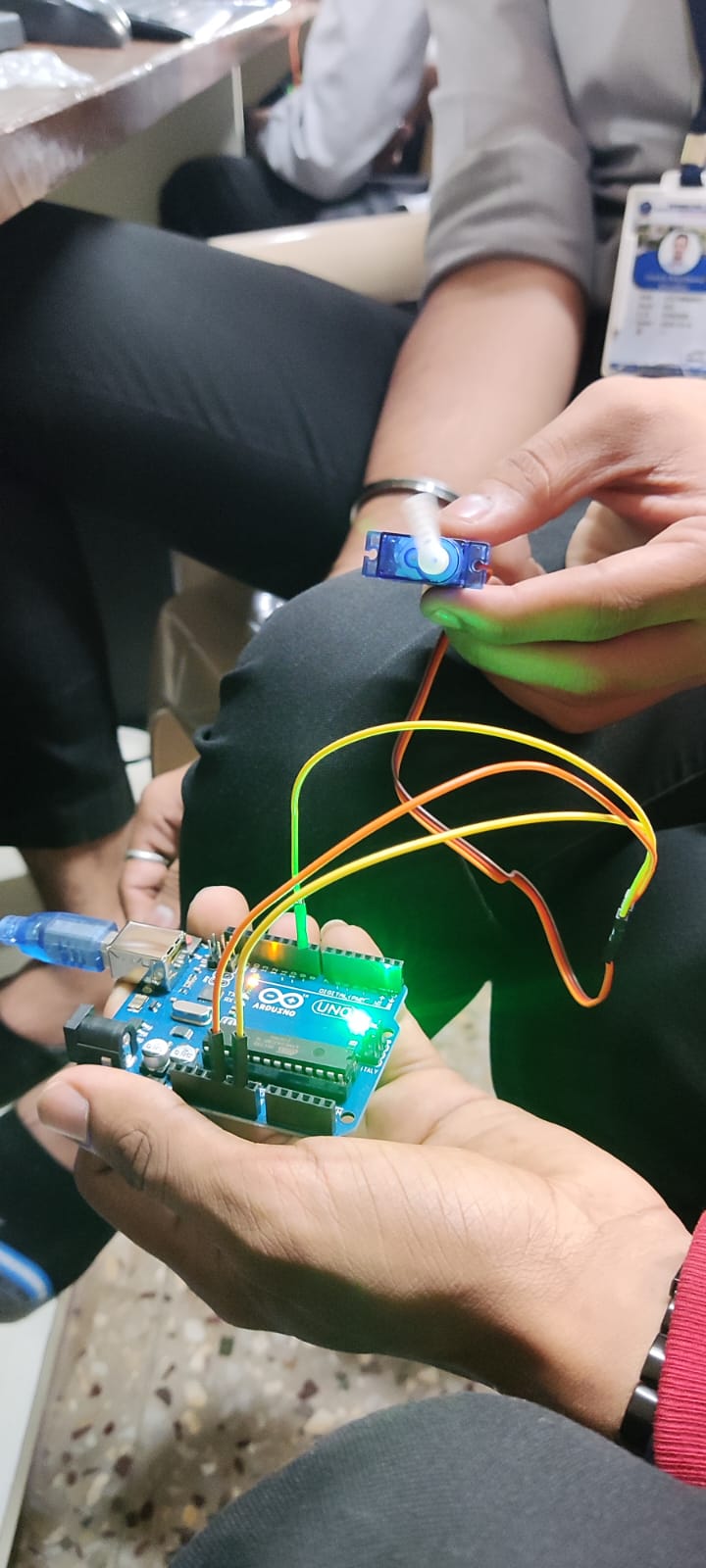
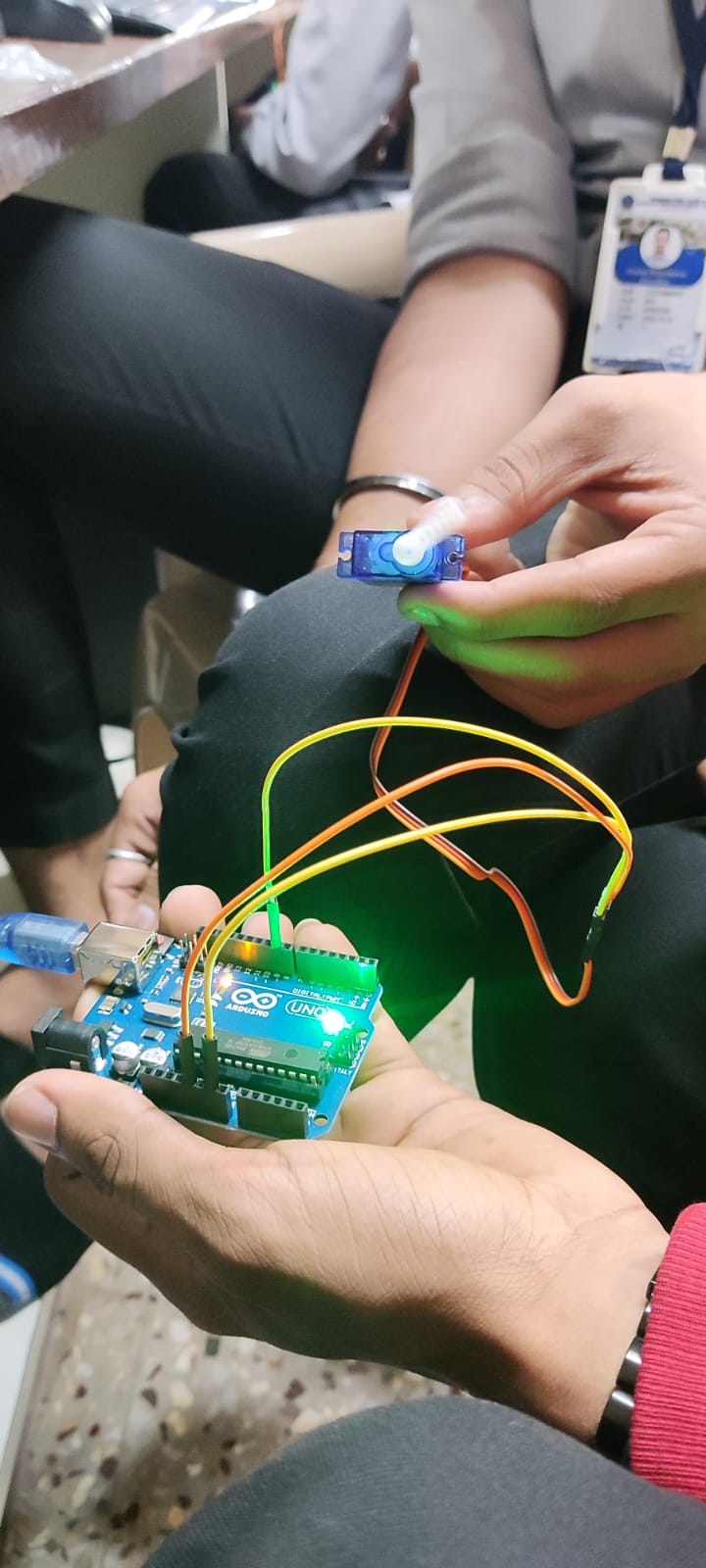
delay(30);

}

delay(1000);

}

**Output-**

** **

**4. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.**

/\*

http://www.arduino.cc/en/Tutorial/Blink

\*/

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(LED\_BUILTIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

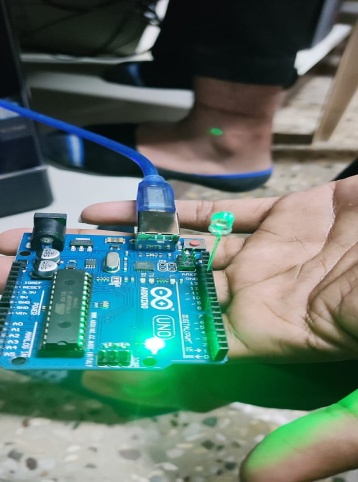
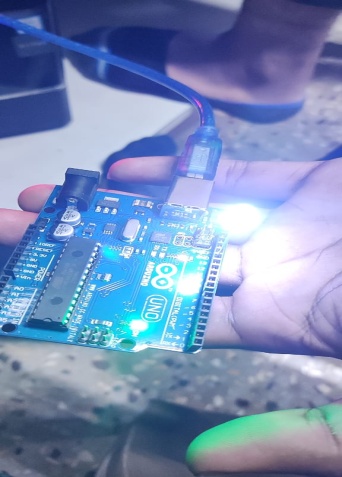
delay(1000);

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltageLOW

delay(1000);

}

**Output:-**

** **

**5. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.**

/\*

Tutorial page: https://arduinogetstarted.com/tutorials/arduino-touch-sensor-led

\*/

const int TOUCH\_SENSOR\_PIN = 7; // Arduino pin connected to the OUTPUT pin of touch sensor

const int LED\_PIN= 3; // Arduino pin connected to LED's pin

void setup() {

Serial.begin(9600); // initialize serial

pinMode(TOUCH\_SENSOR\_PIN, INPUT); // set arduino pin to input mode

pinMode(LED\_PIN, OUTPUT); // set arduino pin to output mode

}

void loop() {

int touchState = digitalRead(TOUCH\_SENSOR\_PIN); // read new state

if (touchState == HIGH) {

Serial.println("The sensor is being touched");;

digitalWrite(LED\_PIN, HIGH); // turn on

}

else

if (touchState == LOW) {

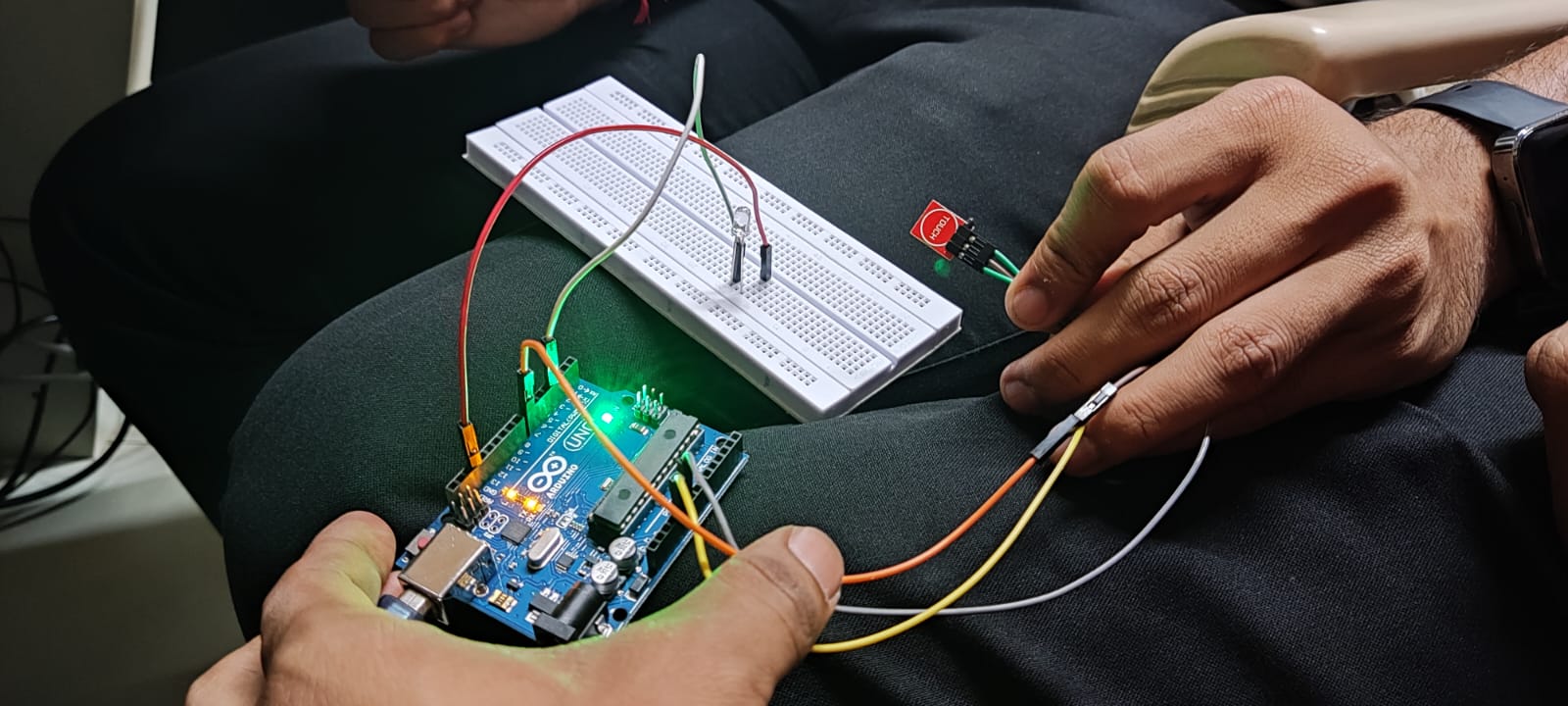
Serial.println("The sensor is untouched");

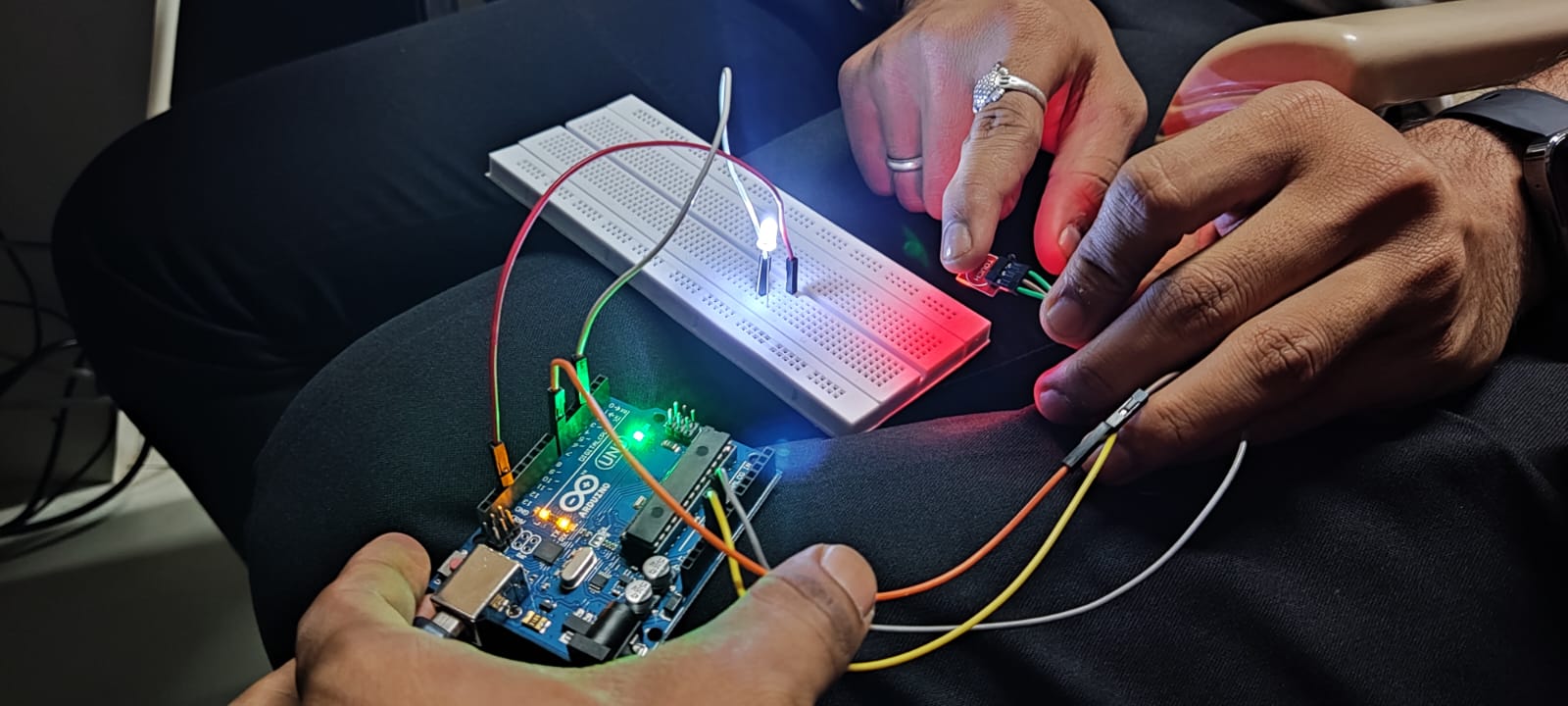
digitalWrite(LED\_PIN, LOW); // turn off

}

}

**Output:-**

****

****

**6. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.**

/\*

Tutorial page: https://arduinogetstarted.com/tutorials/arduino-touch-sensor-servo-motor

\*/

#include <Servo.h>

// constants won't change

const int TOUCH\_SENSOR\_PIN = 7; // Arduino pin connected to touch sensor's pin

const int SERVO\_PIN = 9; // Arduino pin connected to servo motor's pin

Servo servo; // create servo object to control a servo

// variables will change:

int angle = 0;

int lastTouchState;

int currentTouchState;

void setup() {

Serial.begin(9600);

pinMode(TOUCH\_SENSOR\_PIN, INPUT);

servo.attach(SERVO\_PIN);

servo.write(angle);

currentTouchState = digitalRead(TOUCH\_SENSOR\_PIN);

}

void loop() {

lastTouchState = currentTouchState;

currentTouchState = digitalRead(TOUCH\_SENSOR\_PIN);

if(lastTouchState == LOW &&currentTouchState == HIGH) {

Serial.println("The sensor is touched");

// change angle of servo motor

if(angle == 0)

angle = 90;

else

if(angle == 90)

angle = 0;

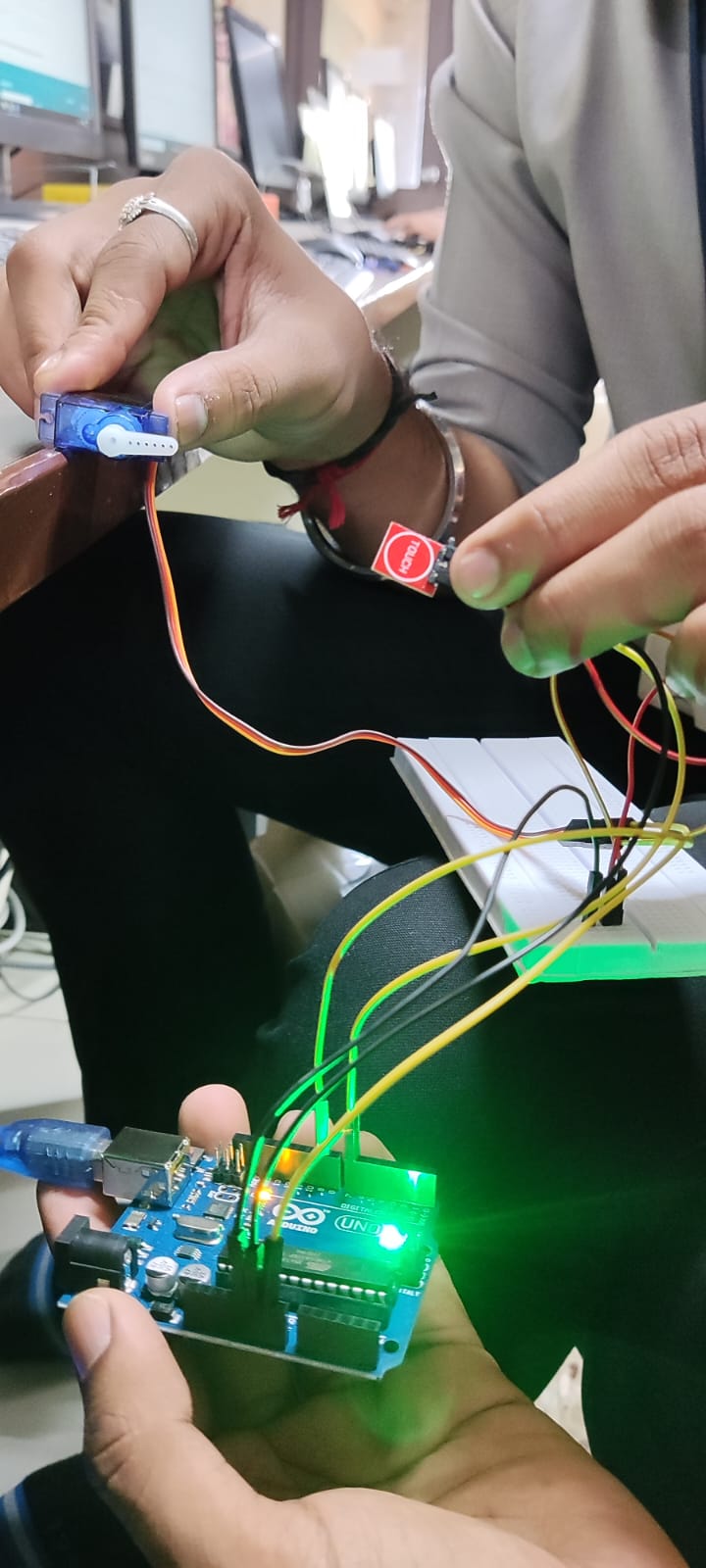
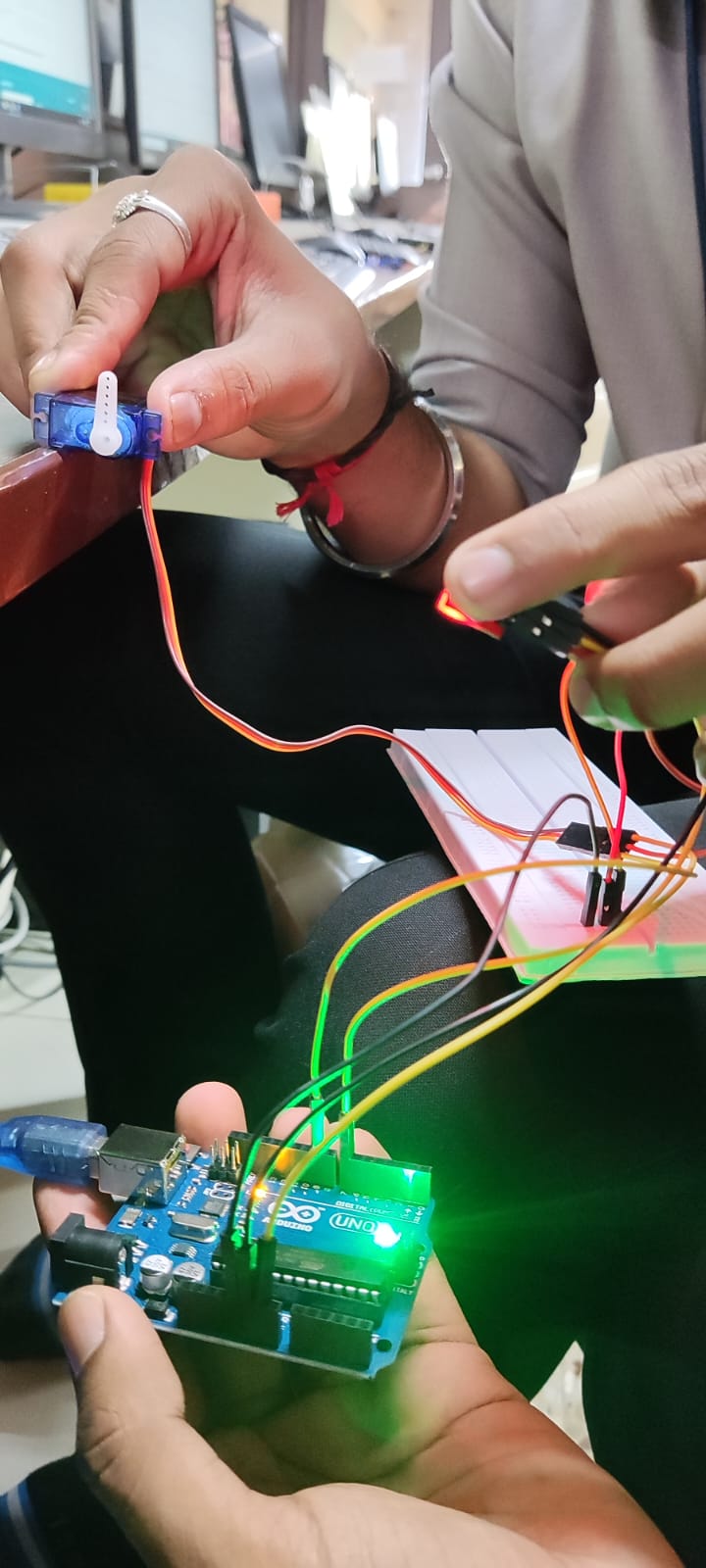
// control servo motor arccoding to the angle

servo.write(angle);

}

}

**Ouput:-**

** **

**7. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.**

/\*

Tutorial page: https://arduinogetstarted.com/tutorials/arduino-temperature-humidity-sensor

\*/

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

dht.begin(); // initialize the sensor

}

void loop() {

// wait a few seconds between measurements.

delay(2000);

// read humidity

float humi = dht.readHumidity();

// read temperature as Celsius

float tempC = dht.readTemperature();

// read temperature as Fahrenheit

float tempF = dht.readTemperature(true);

// check if any reads failed

if (isnan(humi) || isnan(tempC) || isnan(tempF)) {

Serial.println("Failed to read from DHT sensor!");

} else {

Serial.print("Humidity: ");

Serial.print(humi);

Serial.print("%");

Serial.print(" | ");

Serial.print("Temperature: ");

Serial.print(tempC);

Serial.print("°C ~ ");

Serial.print(tempF);

Serial.println("°F");

}

}

**Output:-**

