**LAB TASKS**

**Q1:**

**CODE  
RESULTS**

#include <Servo.h> //library is already in the Arduino

#define servoPin 6 //define pin 6 by use of servoPin as a variable  
Servo myservo; // creates servo object to control a servo

int pos; // pos as a variable to store the servo position

void setup() {

Serial.begin(9600); //baud rate at 9600

myservo.attach(servoPin); // use attach() method to specify the pin #

}

void loop() {

// Sweep from 0 to 180 degrees

for (pos = 0; pos <= 180; pos++) {

myservo.write(pos); //get run acc. to for loop (pos value)

Serial.println(pos); //new line for pos

delay(100); delay for 1 milliseconds

}

// Sweep back from 180 to 0 degrees

for (pos = 180; pos >= 0; pos--) {

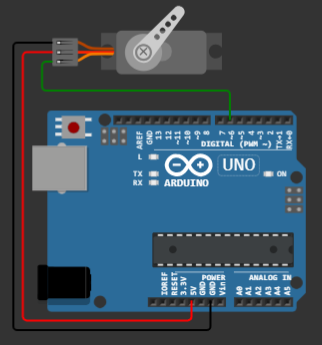
myservo.write(pos); //get run acc. to for loop (pos value)

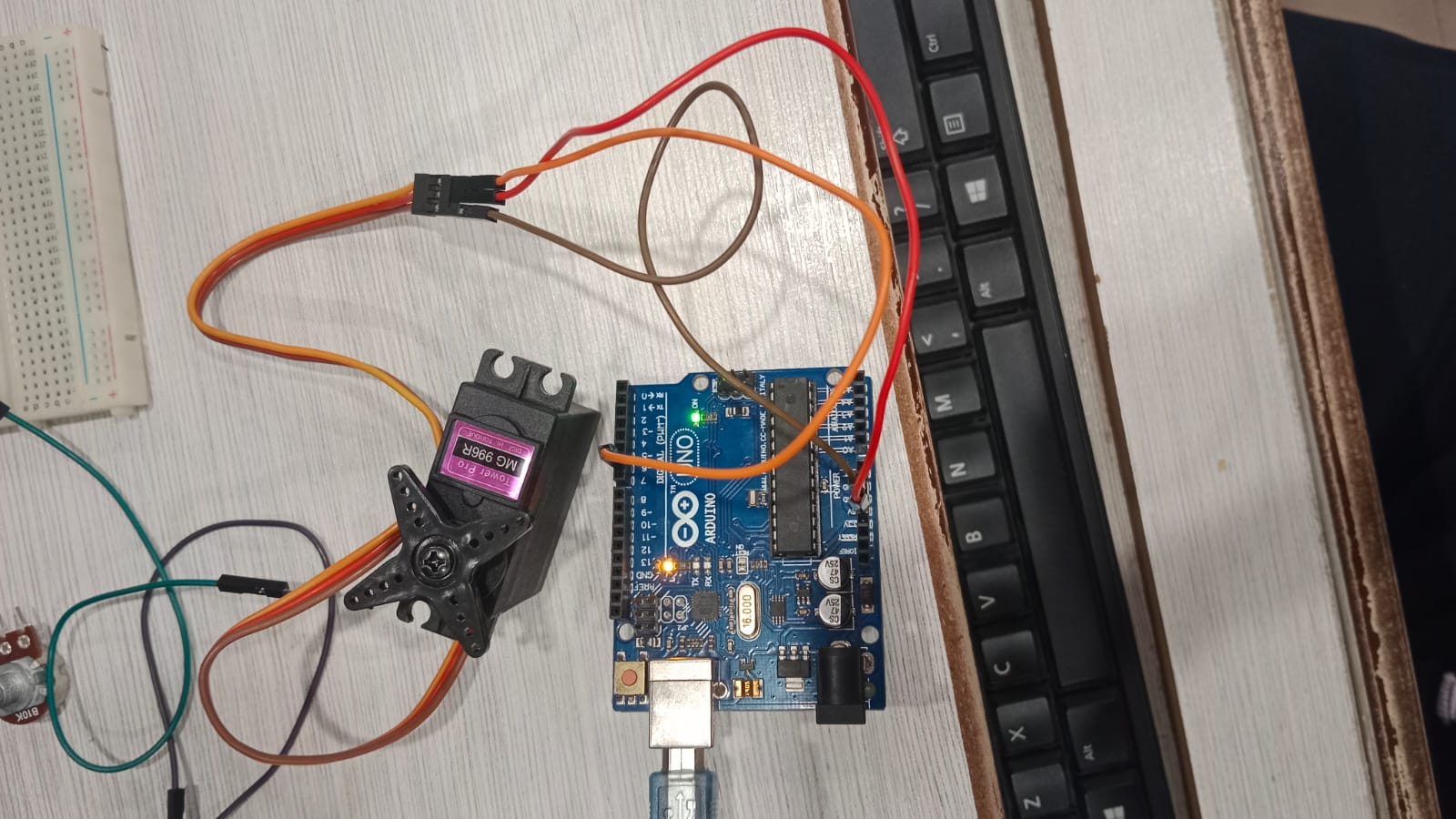
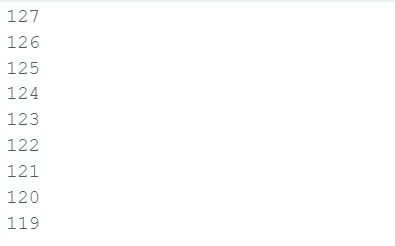
Serial.println(pos); //new line for pos

delay(100); delay for 1 milliseconds

}

}

**Working during lab session on Question 1**



## Angle on serial monitor

**Q2:**

**CODE**

#include <Servo.h> //The library is already included in Arduino.

#define servoPin 6 //define pin 6, servoPin as a variable.

#define potpin A1 //define analog pin A1, potpin as a variable.

Servo myservo; // creates a servo object to control a servo

int val; //integer, use val as a variable

int Mval; //integer, use Mval as a variable

void setup() {

Serial.begin(9600); //baud rate at 9600

myservo.attach(servoPin); // use attach() method to specify the pin #

}

void loop() {

val = analogRead(potpin); // read the value from the potentiometer

Mval = map(val, 0, 1023, 0, 180); // map the potentiometer value to the servo motor range (0 to 180 degrees)

myservo.write(Mval); // set the servo position based on the mapped value

Serial.print("Potentiometer: "); //print potentiometer in serial monitor

Serial.print(val); //print val (value) in serial monitor

Serial.print(" / Motor: "); //print / Motor: in serial monitor

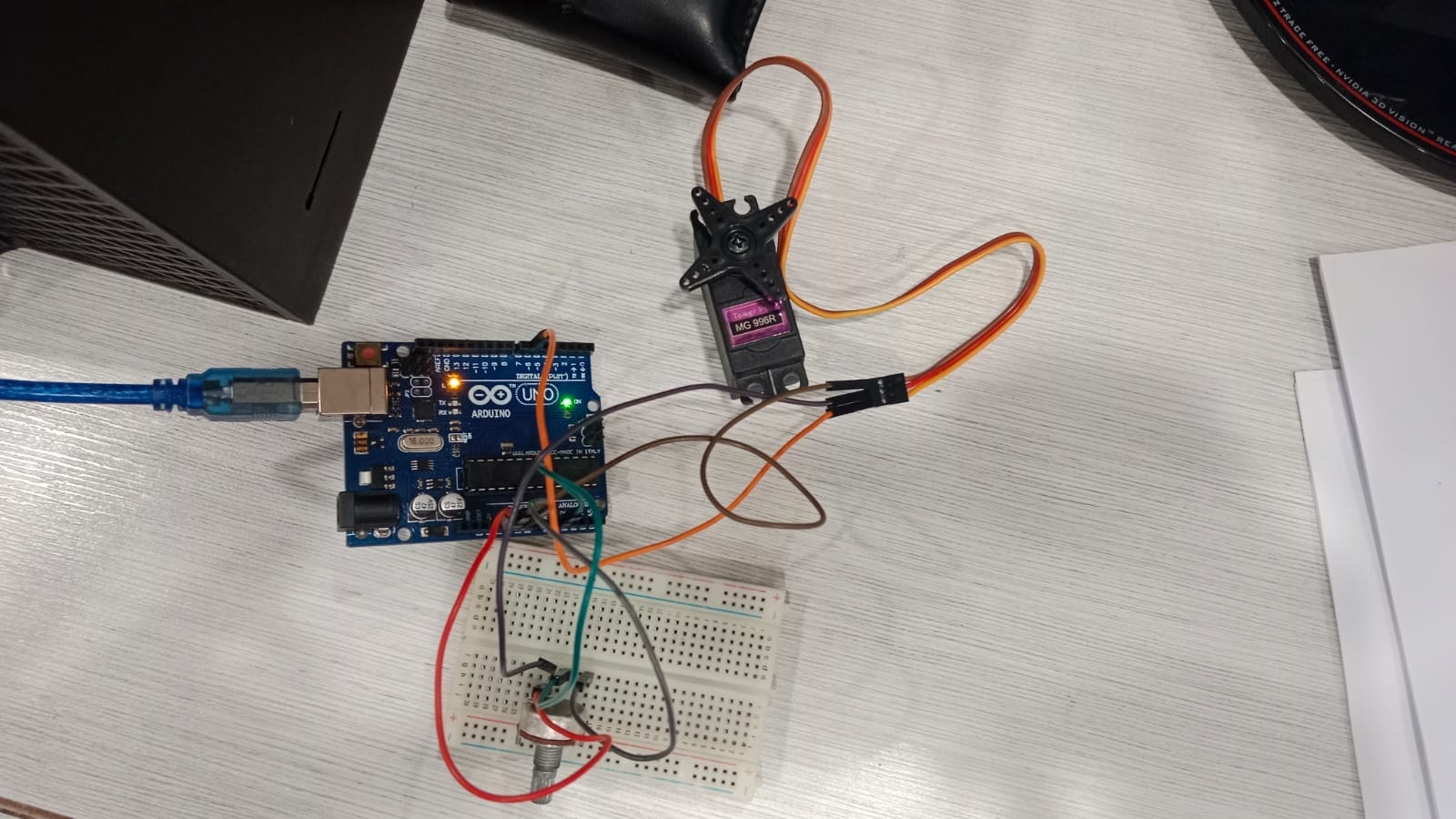
Serial.println(Mval); //Mval value will be in new line

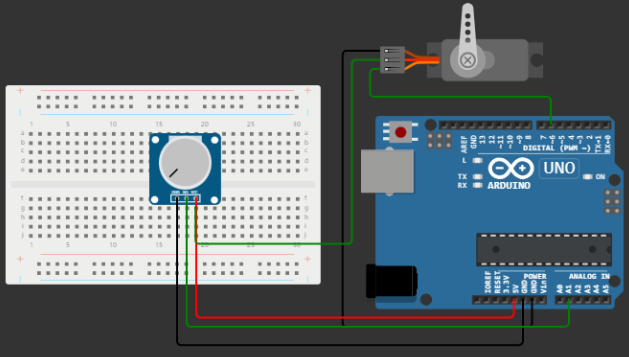
delay(100); // a 1 milliseconds delay to avoid rapid servo movement

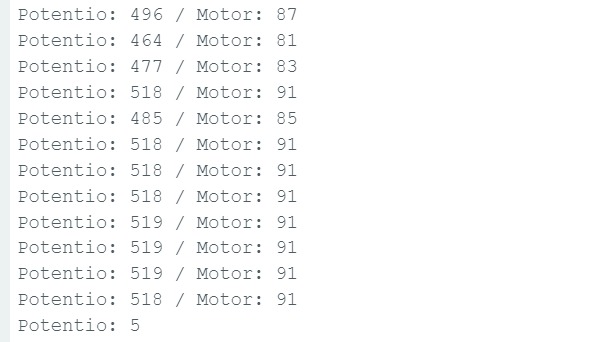
}

**RESULTS**

**Working during lab session on Question 2**







**POST LAB TASKS**

**Q1: Give any 2 examples of practical applications of a 180° servo motor.**

**ANSWER:** a 180° servo motor is commonly used where precise angular control is required.   
here are two practical applications:

* Robotics – Joint Movement
* Camera Pan and Tilt Systems

In both examples, 180**°** servo motors provide an efficient solution for achieving controlled angular movements.

**Q2: By expanding the application of question # 2, write a code that includes the blinking of an LED if the motor has reached its maximum rotation (in other words when the potentiometer output is 1023). Explain your code by commenting on each line and draw the circuit diagram**

**CODE**

#include <Servo.h>

// Define constants for servo and LED pins

#define servoPin 6

#define potPin A1

#define ledPin 13

Servo myservo; // Create a servo object

int potValue; // Variable to store potentiometer value

void setup() {

Serial.begin(9600); //baud rate 9600

myservo.attach(servoPin); // Attach the servo to its pin

pinMode(ledPin, OUTPUT); // Set LED pin as an output

}

void loop() {

potValue = analogRead(potPin); // Read potentiometer value

// Map the potentiometer value to the servo's range (0 to 180)

int servoPosition = map(potValue, 0, 1023, 0, 180);

myservo.write(servoPosition); // Set servo position

// Check if the servo is at its maximum rotation (potentiometer output is 1023)

if (potValue == 1023) {

// Blink the LED if the servo is at its maximum rotation

digitalWrite(ledPin, HIGH);

delay(500);

digitalWrite(ledPin, LOW);

delay(500); //delay

} else {

// Turn off the LED if the servo is not at its maximum rotation

digitalWrite(ledPin, LOW);

}

// Print potentiometer and servo values to the Serial Monitor

Serial.print("Potentiometer: ");

Serial.print(potValue);

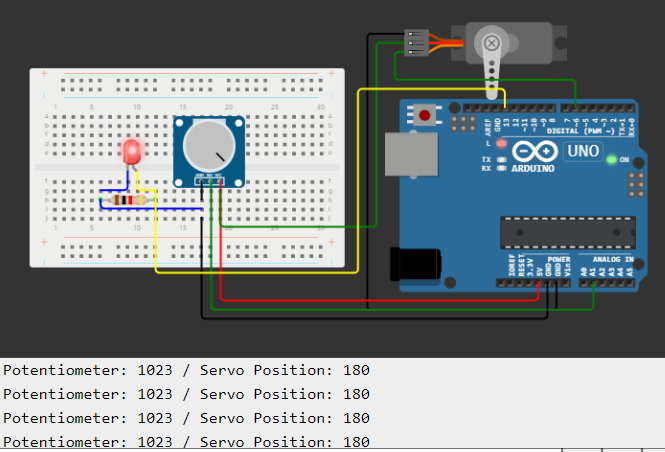
Serial.print(" / Servo Position: ");

Serial.println(servoPosition);

delay(100); // Add a short delay for stability

}

**CIRCUIT**



**Comment to the report: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of Lecturer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**