PREMIER UNIVERSITY CHATTOGRAM

DEPARTMENT OF **COMPUTER SCIENCE AND ENGINEERING**

Lab Report

COURSE NAME	Microcontrollers Laboratory
COURSE CODE	CSE3816
REPORT NO	03
REPORT NAME	Controlling a Servo with a Potentiometer and Arduino.
DATE OF REPORT	7-05-24

SUBMITTED TO

MOHAMMED SAIFUDDIN MUNNA

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

REMARKS	**SUBMITTED BY**	
	NAME	Rimjhim Dey
	ID	0222220005101039
	SEMESTER	4th
	BATCH	42
	SESSION	Spring 2024
	SECTION	A

Experiment Name:

Controlling a Servo with a Potentiometer and Arduino.

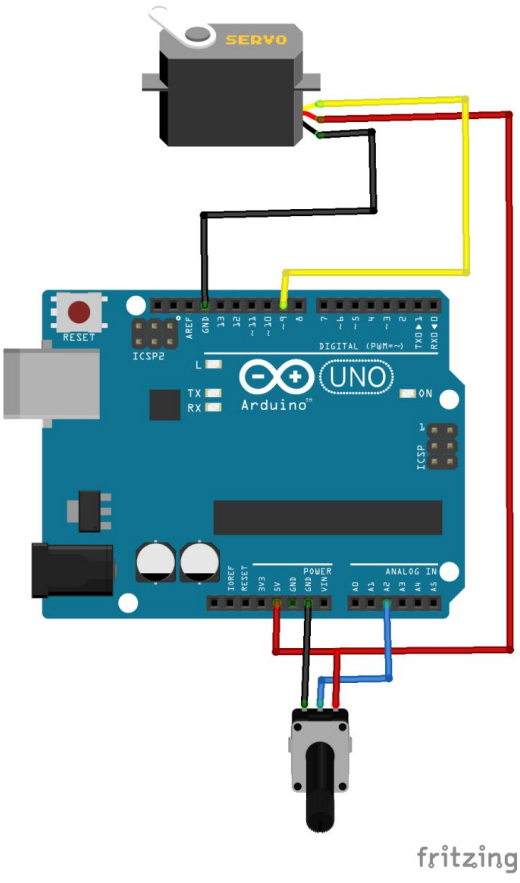
Objective:

The objective of this experiment is to control the position of a servo motor using a potentiometer connected to an Arduino microcontroller. This demonstrates how to read analog input and control a servo motor accordingly.

Instruments Required:

- Arduino Board (e.g., Uno, Nano, Mega)
- Servo Motor
- Potentiometer
- Connecting wires
- Breadboard
- USB cable for programming the Arduino

Circuit Diagram:



Source Code:

```
#include <Servo.h>

Servo myservo;
int apin = A2;
int mot_pin = 9;

void setup() {
  Serial.begin(9600);
  pinMode(apin, INPUT);
  myservo.attach(mot_pin);
}

void loop() {
  int v = analogRead(apin);
  Serial.println(v);
  int d = map(v, 0, 1023, 0, 120);
  myservo.write(d);
  Serial.println(d);
  delay(50);
}
```

Output:

When the circuit is powered on and the code is uploaded to the Arduino, rotating the potentiometer will change the position of the servo motor. The servo motor will move to a position corresponding to the potentiometer's angle, and the values will be displayed on the serial monitor.

Discussion:

In this experiment, we controlled a servo motor using a potentiometer connected to an Arduino. The potentiometer provided an analog input that was read by the Arduino using analogRead(). The value was then mapped to an angle between 0 and 120 degrees using the map() function, and the servo motor's position was set with myservo.write(). Serial communication was used to print the potentiometer and servo position values to the serial monitor, aiding in debugging and visualization. This experiment demonstrates how to interface analog input devices with a microcontroller to control actuators, essential for building responsive and interactive systems.