Title of Experiment:

Interfacing Potentiometer for Position Control Simulation

Aim:

To interface a potentiometer with Arduino and simulate position control by controlling a servo motor using the analog input.

Apparatus:

- Arduino Uno board
- USB cable
- Breadboard
- Potentiometer (10kΩ)
- Servo motor (e.g., SG90)
- Jumper wires
- Arduino IDE installed on PC

Precautions:

1. Ensure the servo motor is powered properly; do not overload it.
2. Connect the potentiometer correctly to avoid incorrect readings.
3. Avoid rapidly turning the potentiometer to prevent sudden servo movement.
4. Do not apply external force to rotate the servo horn manually.
5. Disconnect USB power while changing the wiring.

Theory:

A **potentiometer** is an analog input device that provides a varying voltage based on the rotation of its knob. The Arduino reads this voltage using the **analogRead()** function.

A **servo motor** is a position-controlled device that rotates within a specific range (typically 0–180°). The position is controlled using **PWM (Pulse Width Modulation)** signals from a digital PWM pin on Arduino.

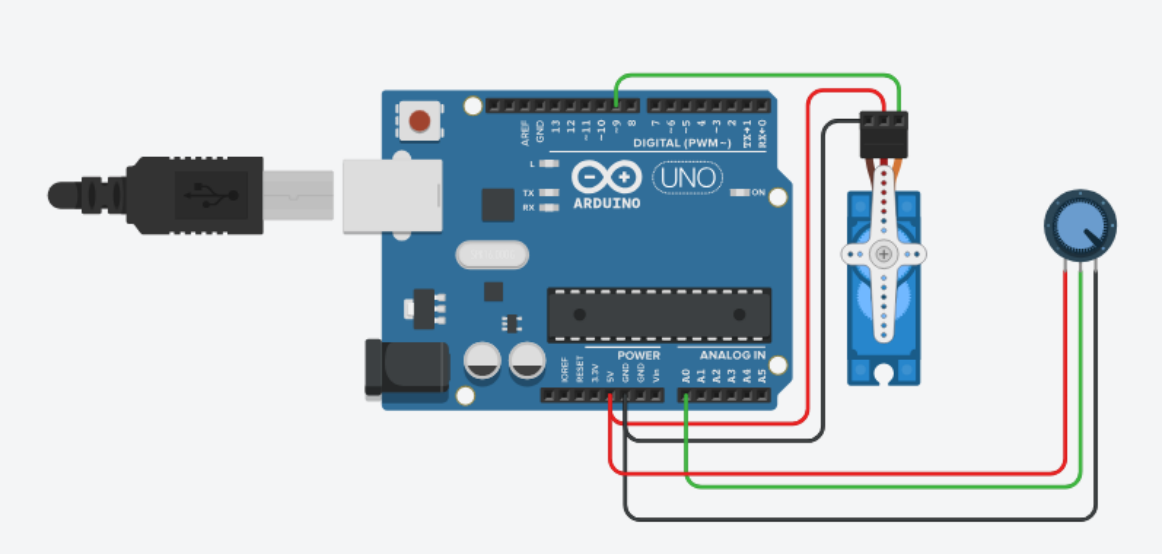
By reading the analog value (0–1023) from the potentiometer and mapping it to a servo angle (0–180), we can simulate **position control**—as the potentiometer is rotated, the servo position changes accordingly.

Circuit Diagram:

Connections:

- **Potentiometer**
  - One end to 5V
  - Other end to GND
  - Center pin (wiper) to A0
- **Servo Motor**
  - Control wire to digital pin 9
  - VCC to 5V
  - GND to GND

Circuit Diagram:



Procedure:

1. Connect the potentiometer to Arduino's analog pin A0.
2. Connect the servo motor's control wire to digital PWM pin D9.
3. Supply power (5V and GND) to both potentiometer and servo.
4. Write the Arduino code to read the potentiometer value and map it to the servo angle.
5. Upload the code using Arduino IDE.
6. Rotate the potentiometer and observe the servo's position change.
7. Observe angle response in Serial Monitor if needed.

Arduino Programming

```
#include <Servo.h>          // Include Servo library

Servo myServo;              // Create servo object

const int potPin = A0;      // Potentiometer connected to analog pin A0

int potValue = 0;           // Variable to store analog value

int angle = 0;              // Variable to store mapped servo angle


void setup() {

  myServo.attach(9);        // Attach servo to digital pin 9 (PWM)

  Serial.begin(9600);       // Start serial communication

}


void loop() {

  potValue = analogRead(potPin);          // Read analog value (0–1023)

  angle = map(potValue, 0, 1023, 0, 180);   // Map analog value to 0–180°

  myServo.write(angle);                   // Rotate servo to the mapped angle
```

```
  // Print values to Serial Monitor

  Serial.print("Potentiometer: ");

  Serial.print(potValue);

  Serial.print(" -> Servo Angle: ");

  Serial.println(angle);



  delay(15);   // Small delay for smooth movement

}
```

Conclusion:

The experiment demonstrated how a potentiometer can be used as an input device to control the position of a servo motor. The analog signal from the potentiometer was successfully converted into angular movement, simulating a basic position control system using Arduino.

---

4 Short Questions and Answers:

Q1: What is the role of the potentiometer in this experiment?
Answer: The potentiometer acts as an analog input device that varies voltage based on its rotation, which is used to control the servo's angle.

Q2: Why is the `map()` function used in the code?
Answer: The `map()` function converts the potentiometer's analog value (0–1023) to the servo angle range (0–180°).

Q3: What is the purpose of the `Servo` library in Arduino?
Answer: The `Servo` library simplifies the process of generating PWM signals required to control a servo motor.

Q4: Which type of signal is used to control the servo motor?
Answer: The servo motor is controlled using a **PWM (Pulse Width Modulated)** signal from a digital pin (e.g., D9).