**Lab Exercise 9- Controlling a Servo Motor with Arduino**

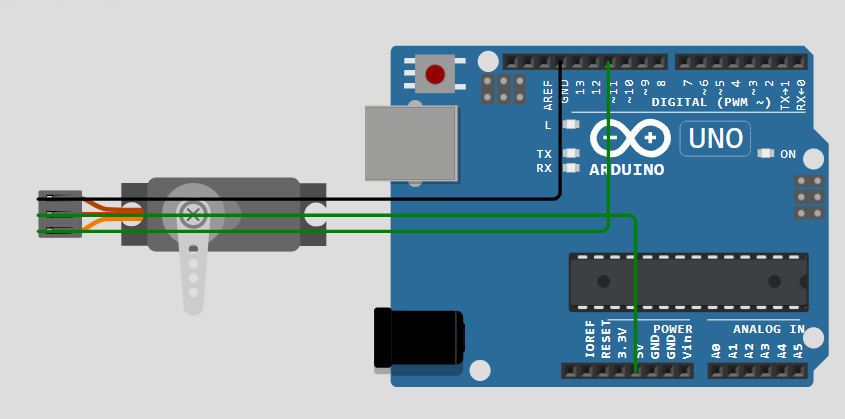
In this lab, you will learn how to control the position of a **servo motor** using an Arduino. The servo motor will be rotated to specific angles based on commands sent from the Arduino.

**Components Required:**

* 1 x Arduino Uno
* 1 x Servo Motor (SG90 or any other)
* 1 x 5V power supply (optional for larger servos)
* Jumper wires
* Breadboard (optional)

**Circuit Diagram:**

* **Servo Motor Connections**:
  + **Red Wire (VCC)**: Connect to the **5V pin** of Arduino.
  + **Black/Brown Wire (GND)**: Connect to the **GND pin** of Arduino.
  + **Yellow/White Wire (Signal)**: Connect to **digital pin 11** (or any PWM pin) of the Arduino.



For larger servos, use an external 5V power supply because the Arduino's onboard power may not be sufficient.

**Arduino Code:**

The Arduino IDE has a built-in **Servo library** that makes it easy to control servo motors. To control the servo motor, use the following code:

**Part-1: Code**

#include <Servo.h>

int servoPin=11;

Servo servo1;

void setup() {

  // put your setup code here, to run once:

  servo1.attach(servoPin);

}

void loop() {

  // put your main code here, to run repeatedly:

  servo1.write(0);

  delay(2000);

  servo1.write(90);

  delay(2000);

  servo1.write(180);

  delay(2000);

}

**Part 2: Code**

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

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

int pos = 0; // Variable to store the servo position

void setup() {

myservo.attach(11); // Attach the servo on pin 9 to the servo object

}

void loop() {

// Sweep the servo from 0 to 180 degrees

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

myservo.write(pos); // Tell the servo to go to position 'pos'

delay(15); // Wait for the servo to reach the position

}

// Sweep the servo from 180 to 0 degrees

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

myservo.write(pos); // Tell the servo to go to position 'pos'

delay(15); // Wait for the servo to reach the position

}

}

**How the Code Works:**

1. **Servo Library**: The code includes the **Servo library** (<Servo.h>) which simplifies controlling the servo motor.
2. **Servo Object**: We create a servo object (myservo) that will control the servo motor.
3. **Setup**:
   * In the setup(), the myservo.attach(9) function links the servo object to pin 9 on the Arduino, which will control the servo's movement.
4. **Loop**:
   * The loop() contains two **for loops**:
     + The first loop moves the servo from **0 to 180 degrees**, incrementing by 1 degree at a time. The delay(15) ensures smooth motion.
     + The second loop moves the servo back from **180 to 0 degrees**.
   * The servo motor continuously sweeps between these two angles.

**Uploading the Code:**

1. Connect your Arduino to your computer using a USB cable.
2. Open the Arduino IDE and paste the code into the editor.
3. Select the correct **Board** and **Port** from the "Tools" menu.
4. Click the **Upload** button to upload the code to your Arduino.

**Expected Result:**

The servo motor should rotate smoothly from **0 degrees** to **180 degrees**, and then back to **0 degrees** continuously. The delay between each step gives the motor time to reach the correct position.

**Modifying the Exercise:**

* You can modify the **angle limits** to move the servo within a smaller range (e.g., from 30° to 150°).
* You can also experiment with **faster or slower movements** by adjusting the delay time between steps.

**Part-2: Controlling Servo via Serial Monitor**

You can also control the servo motor via the **Serial Monitor**, allowing you to input the desired angle manually. Here's an updated version of the code to control the servo angle through the serial input:

#include <Servo.h>

Servo myservo;

int angle; // Variable to store the user-defined angle

void setup() {

myservo.attach(9); // Attach the servo on pin 9

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

Serial.println("Enter an angle between 0 and 180:");

}

void loop() {

// Check if data is available in the serial buffer

if (Serial.available() > 0) {

angle = Serial.parseInt(); // Read the input angle

// Validate the angle input

if (angle >= 0 && angle <= 180) {

myservo.write(angle); // Move the servo to the specified angle

Serial.print("Servo moved to angle: ");

Serial.println(angle);

} else {

Serial.println("Invalid angle. Enter a value between 0 and 180.");

}

}

}

**Explanation of Changes:**

* The **Serial.begin(9600)** function initializes serial communication.
* Inside the loop(), the program waits for a user to input an angle via the **Serial Monitor**.
* The Serial.parseInt() function reads the angle value entered by the user and moves the servo to the corresponding position.

**Testing the Serial Monitor Version:**

1. Open the **Serial Monitor** (Tools > Serial Monitor) after uploading the code.
2. Enter any angle between 0 and 180, and the servo will rotate to that position.

**Conclusion:**

In this lab exercise, you learned how to control a **servo motor** using an Arduino. The servo can be controlled programmatically using loops, or interactively using serial input. This exercise is foundational for building more complex projects like robotic arms, automated systems, or moving parts in IoT projects.