Workshop: Introduction to Servo Motor with Arduino

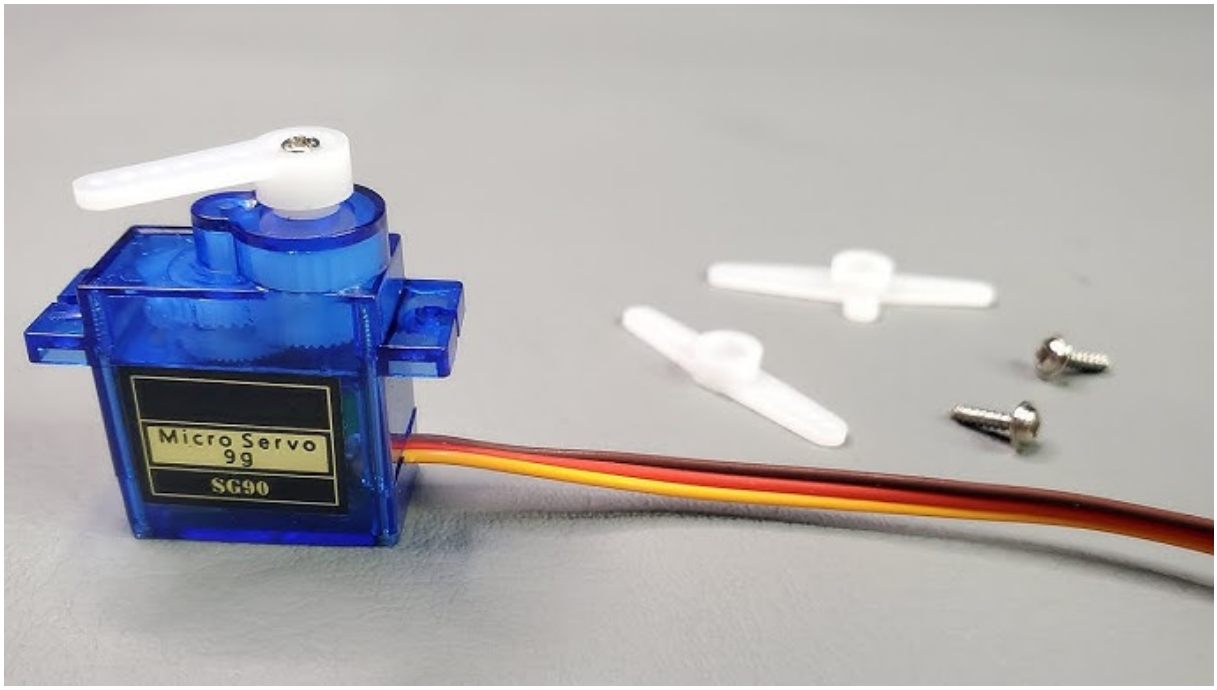

Figure 1: Servo Motor

In this workshop, we will connect and program a servo motor to function like a metronome.

We will explore how to control the motor's swinging speed with a variable and how to adapt the system for different needs.

Required Materials

- 1x Arduino Uno board
- 1x SG90 servo motor
- Connecting wires
- USB cable for Arduino

Part 1: Set Up the Circuit

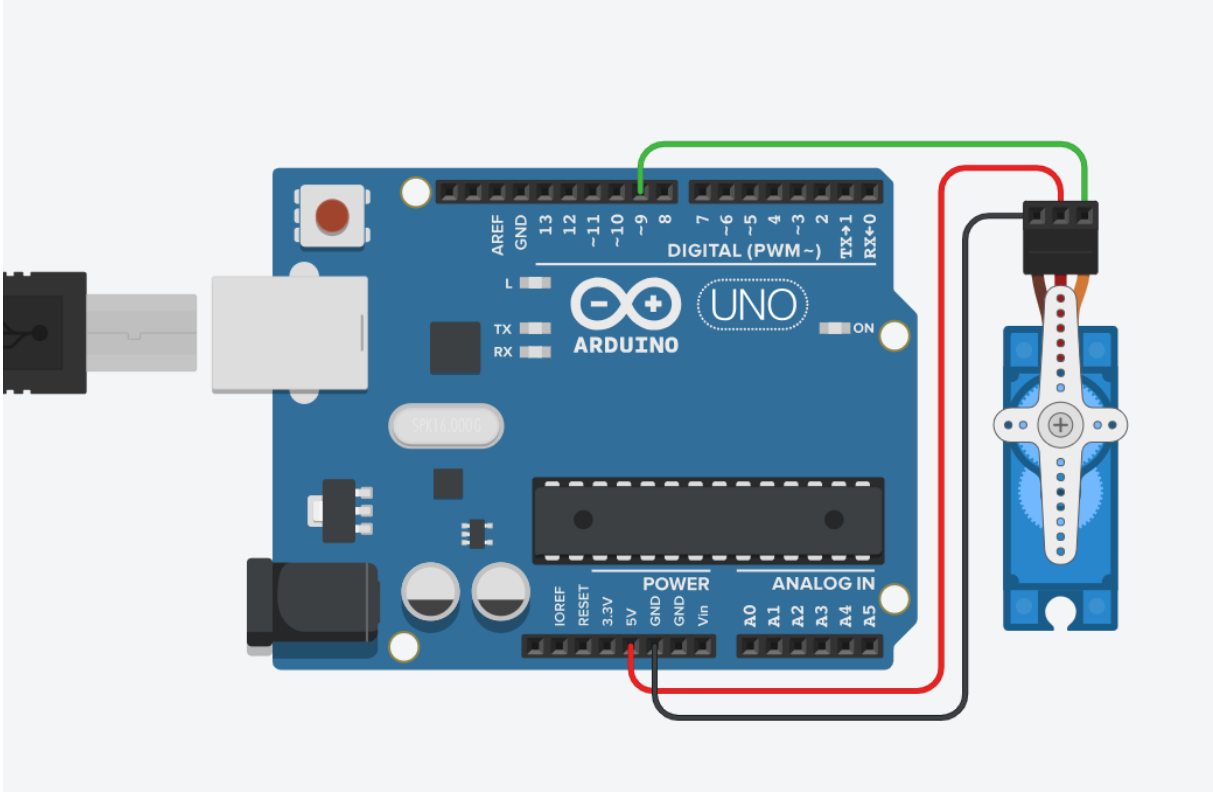

Figure 2: Circuit Setup

- **Red Wire (Power):** Connect it to the 5V pin on the Arduino.
- **Black or Brown Wire (Ground):** Connect it to a GND pin on the Arduino.
- **Yellow or Orange Wire (Signal):** Connect it to digital pin 9 on the Arduino.

Part 2: Program the Arduino

Open the Arduino IDE. We'll use a simple sketch to control the servo's movement.

Step 1: Include the Servo Library

Add this line at the beginning of your code to use the Servo library:

```
#include <Servo.h>
```

Step 2: Create a Servo Object

Create an object `myServo` to control the servo motor:

```
Servo myServo;
```

Step 3: Set Up the Servo

In the `setup()` function, attach the servo to a specific Arduino pin:

```
void setup() {
   myServo.attach(9); // Attach the servo to pin 9
}
```

Step 4: Servo Movement Loop

Use the `loop()` function to move the servo between two positions:

```
void loop() {
   myServo.write(0);   // Move servo to 0 degrees
   delay(1000);        // Wait 1 second
   myServo.write(180); // Move servo to 180 degrees
   delay(1000);        // Wait 1 second
}
```

Since the `loop()` executes continuously, the servo will move from `0` to `180` degrees and back to `0` degrees every `1000` milliseconds.

Step 5: Create a Variable to Change the Delay

Create a variable `delayTime` to control the metronome's speed:

```
int delayTime = 1000;
```

Modify the `loop()` function to use this variable:

```
void loop() {
   myServo.write(0);
   delay(delayTime); // Use the delayTime variable
   myServo.write(180);
   delay(delayTime); // Use the delayTime variable
}
```

Changing the value of `delayTime` allows you to control the metronome speed without modifying the rest of the code.

Part 3: Exploration Questions

1. Modify the value of `delayTime`. How does it affect the metronome?
2. How can you modify the code to read the value from a potentiometer and use it to control `delayTime`?
3. Instead of behaving like a metronome, how could you control the motor angle ranging from `0` to `180` with a potentiometer?