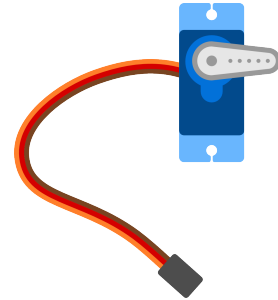


## Servo Motor

Name: \_\_\_\_\_

Servo motors are components that allow you to control the position of a shaft. They can be used in a wide range of applications, from moving robot arms to controlling the rudder on a model boat.



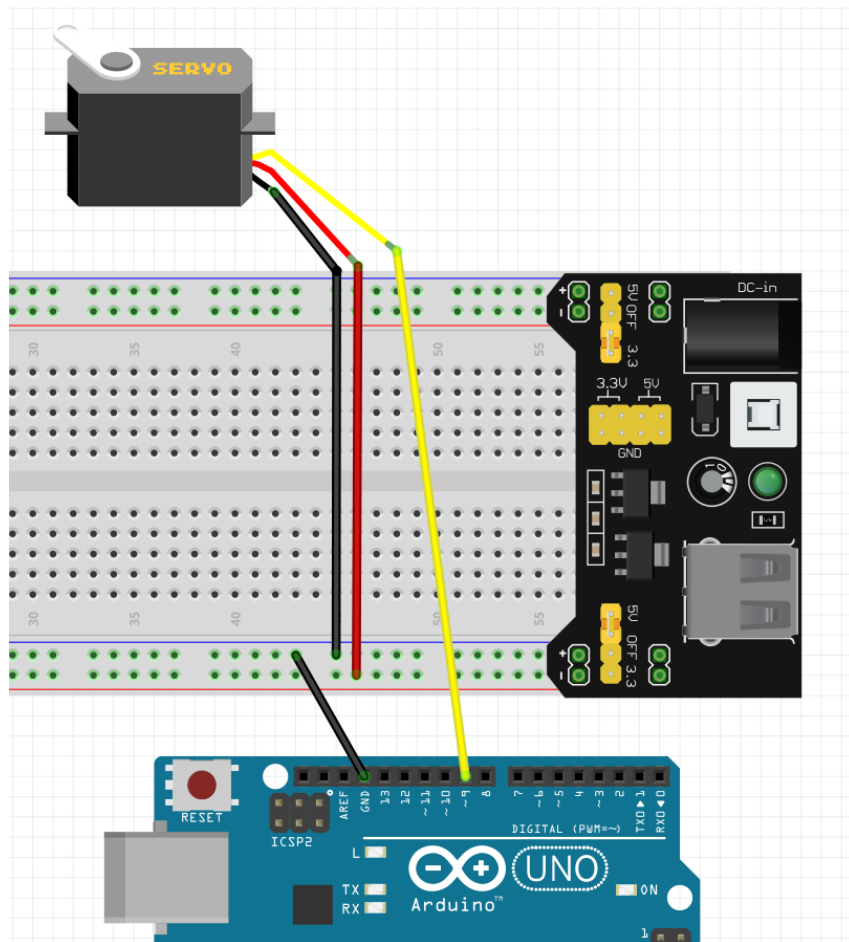
We'll cover how to control a servo motor with Arduino.

### What You'll Need:

- 1 x Arduino Uno board
- 1 x Servo Motor
- 1 x Breadboard
- 1 x Power Supply
- 1 x 9v Battery (optional)
- Jumper wires

## Circuit

Although the Arduino can provide 5V, it can't provide enough current for a motor to work. This is why we use an *external 5V power source*.



**Figure 1:** Servo Motor Circuit

### Servo Wires

1. **Power (Red wire):** Connect the red wire from the servo to the 5V pin on the Arduino.
2. **Ground (Brown/Black wire):** Connect the brown or black wire from the servo to one of the GND pins on the Arduino.
3. **Signal (Orange/Yellow wire):** Connect the orange or yellow wire from the servo to digital pin 9 on the Arduino.

## Example: Motor Sweep

The code below makes the servo go back and forth every 3 seconds

```
1  #include <Servo.h>
2
3  // The servo motor instance
4  Servo motorA;
5
6  void setup() {
7      // Configure the motor on Pin 9
8      motorA.attach(9);
9  }
10
11 void loop() {
12     // Set the motor to 0 degrees
13     motorA.write(0);
14     delay(3000);
15
16     // Set the motor to 180 degrees
17     motorA.write(180);
18     delay(3000);
19 }
```

## Example: Light intensity motor

Control a servo motor based on the photoresistor's analog value.

Use the same photoresistor voltage divider circuit that we used previously.

## Fix the code: Light intensity motor

This code is expected to control the motor according to light sensed on the photoresistor.

The problem is that `light` can range from 0 to 1023 and the motor can not take more than 180 degrees.

What should we add to the code so it behaves as expected ?

```
1  #include <Servo.h>
2
3  // The analog light value
4  int light = 0;
5
6  // The servo motor instance
7  Servo lightMotor;
8
9  void setup() {
10     // Configure the motor on Pin 9
11     lightMotor.attach(9);
12 }
13
14 void loop() {
15     // Read the analog value on Pin A0
16     light = analogRead(A0);
17
18     // IMPORTANT:
19     // Convert light to a 0 to 180 value
20
21     // Send the analog value to the servo
22     lightMotor.write(light);
23
24     // Wait, too much `write` calls can overload a motor
25     delay(150);
26 }
```