

Controlling Continues Servo Motor with Pot

<http://arduino.cc/en/Tutorial/Knob>

Control a continues servo motor with your Arduino and a potentiometer. This example makes use of the Arduino servo library. This example it is based in Sweep example made by by BARRAGAN <<http://barraganstudio.com>>

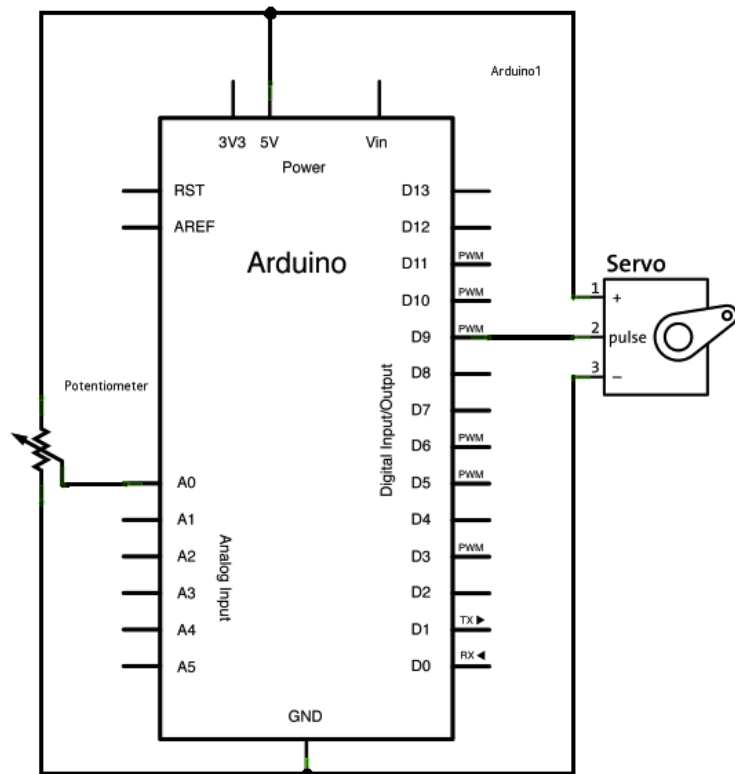
HARDWARE REQUIRED

Arduino Board
 (1) Continues 360° Servo Motor
 (1) Potentiometer

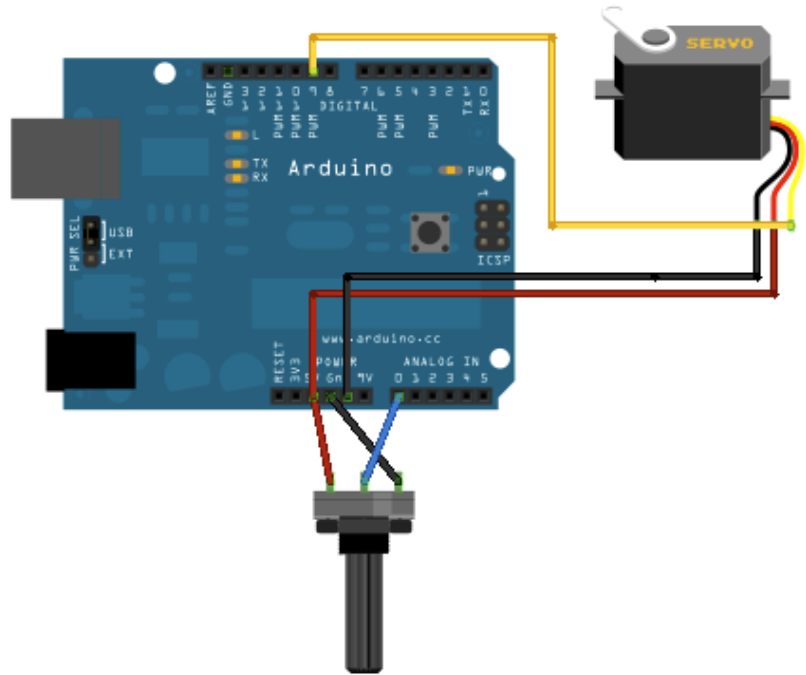
CIRCUIT

Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is typically black or brown and should be connected to a ground pin on the Arduino board. The signal pin is typically yellow or orange and should be connected to pin 9 on the Arduino board. The potentiometer should be wired so that its two outer pins are connected to power (+5V) and ground, and its middle pin is connected to analog input 0 on the Arduino.

SCHEMATIC



IMAGE



CODE

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
               // a maximum of eight servo objects can be created

int pos = 1; // variable to store the servo position
int myDirection=1; // indicates the servo direction
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}

void loop()
{
  val = analogRead(potpin); // reads the value of the potentiometer
                             (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 5.5); // scale it to use it with the servo (value
                                   between 0 and 180)

  if (pos==0 || pos==180)

    // specify servo's direction
    myDirection=-myDirection;
    pos=pos+myDirection;

    // print to Serial the results
    Serial.print(val);
    Serial.print(" ");
    Serial.println(pos);

    // Use 90 to stop the servo
    myservo.write(180-pos); // apply position

    delay(val);
}
```