# **Arduino Motor Workshop**

1. Written by Mark Webster, Ousema Zayati, and Connor Sheeran

Summary

In this workshop learn how to control DC, stepper, and servo motors from an Arduino microcontroller.

## Introduction

Almost every electrically powered project that has moving parts involves motors. Microcontrollers like Arduinos typically do not put out enough current to drive power hungry motors. Instead, the output of a pin on the Arduino is used to trigger larger amounts of current flow using some external device like a power transistor, MOSFET, H-Bridge, or motor controller.

The type of external device depends on the type of motor.

The type of external device depends on the type of motor. There are three main categories of motors: DC, stepper, and servo.

* DC motors don’t turn in increments, they make complete revolutions one way or another, with speed controlled by the voltage.
* Stepper motors are DC motors that move in discrete steps and are used for precise positioning or speed control.
* Servo motors are higher power and precision stepper motors. A stepper motor usually needs a separate rotary encoder to know it’s initial or subsequent position. A servo motor doesn’t need a separate encoder and can be directed to any particular position. Servo motors are often used in 3D printers, CNC machines, or robotic arms.

A quirk of motors is they turn into a little electrical generator when they are turned off and are slowing down. This back-voltage can damage an Arduino or transistorized motor control unit. So a diode is used in parallel with the motor to short out the reverse back-voltage. Often the diode is built into a motor controller module

## DC Motors

If only one way rotation of the motor is required, a simple power transistor or MOSFET can be used. The maximum current draw of the motor determines the power requirement of transistor to use. The gate (base) of the transistor is connected to the output of an Arduino digital pin. A small current from the Arduino can trigger a large current flow through the transistor. When used as a switch, the Arduino must life the base of the transistor to VCC. This means the voltage between Emitter-Collector must be 5V. The motor is placed somewhere in the path between a separate positive voltage supply (such as the + pole of a battery) and a common ground.

The transistor or MOSFET is used in a binary mode, either full on or full off. This is much simpler than a transistor linear amplifier circuit.

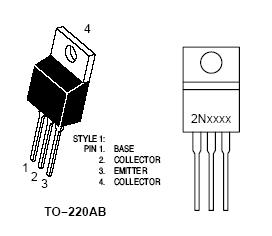
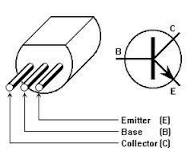
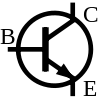
A diode is placed parallel to the motor, but with forward bias opposite the normal current path. No current will flow through the diode when the motor is running, but as the motor is stopping and creating a reverse back-voltage the diode shorts out the motor-generator so the transistor isn’t damaged.

When the DC motor is required to turn either direction, then a slightly more complicated circuit known as an H-bridge is used. This allows two pins on the Arduino to trigger current to flow either direction through the DC motor.

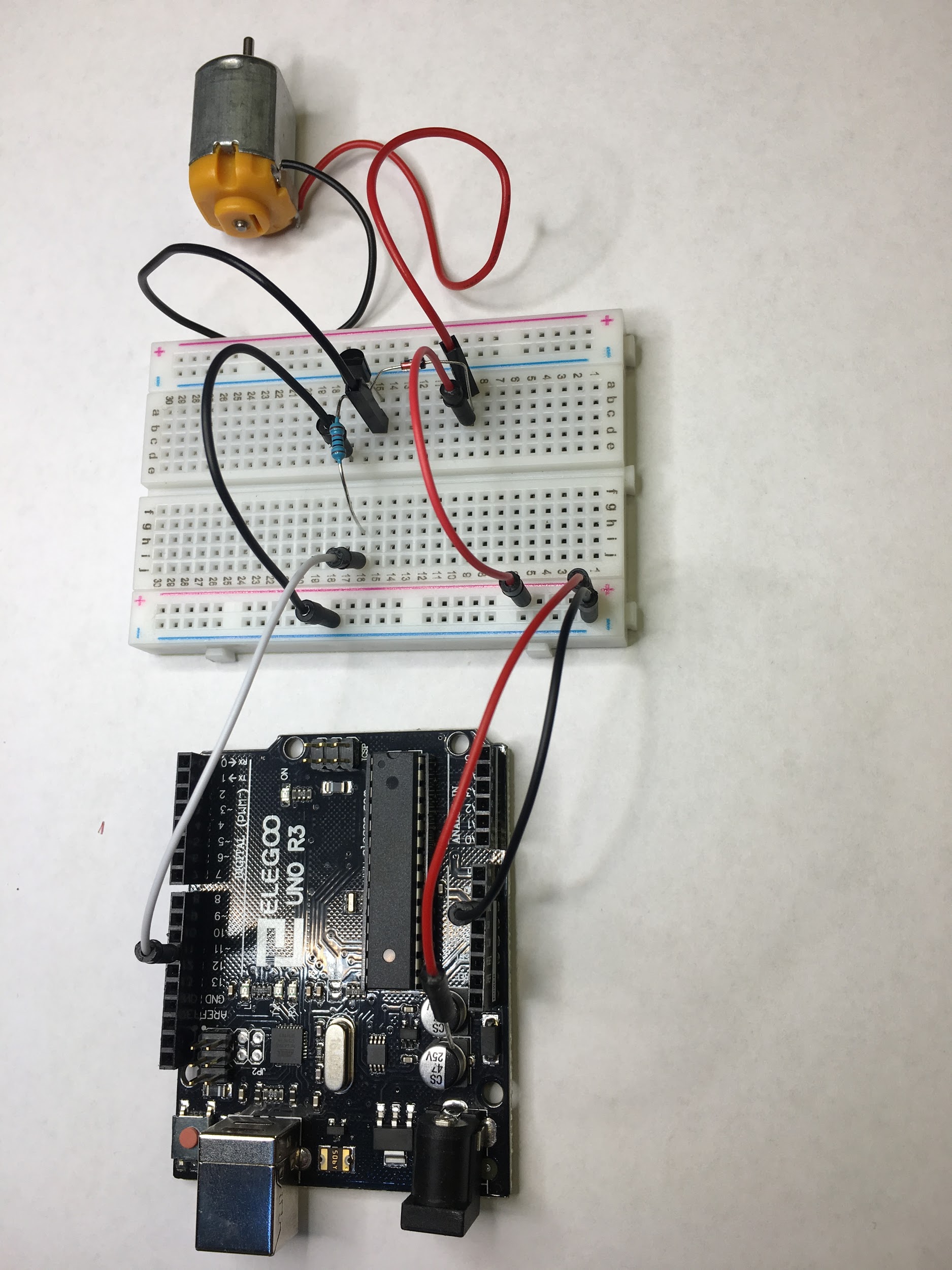
The rotational speed of a DC motor is often controlled using PWM (pulse width modulation). The pulse width will determine how much power flows through the transistor and hence the voltage and thus the rotational speed of the motor.

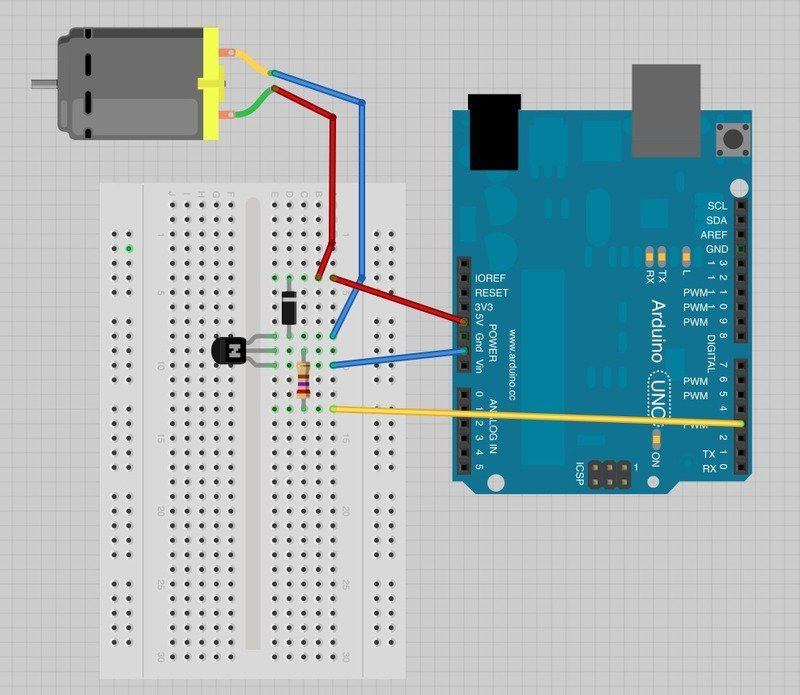
### Example 1: DC motor

Use an Arduino UNO, a digital output pin, and an NPN transistor connected to a low voltage DC motor. The voltage of the power supply must be 5V for the Arduino to turn on the transistor completely. Use the BLINK example, modified to use another digital pin.



NPN transistors, with two different cases.

****



Modify the BLINK program to use “int motorPin = 11’: instead of LED\_BUILTIN.

The motor should alternately spin and stop. Use the “analogWrite(11,value from 0-254)” to control the speed of the motor.

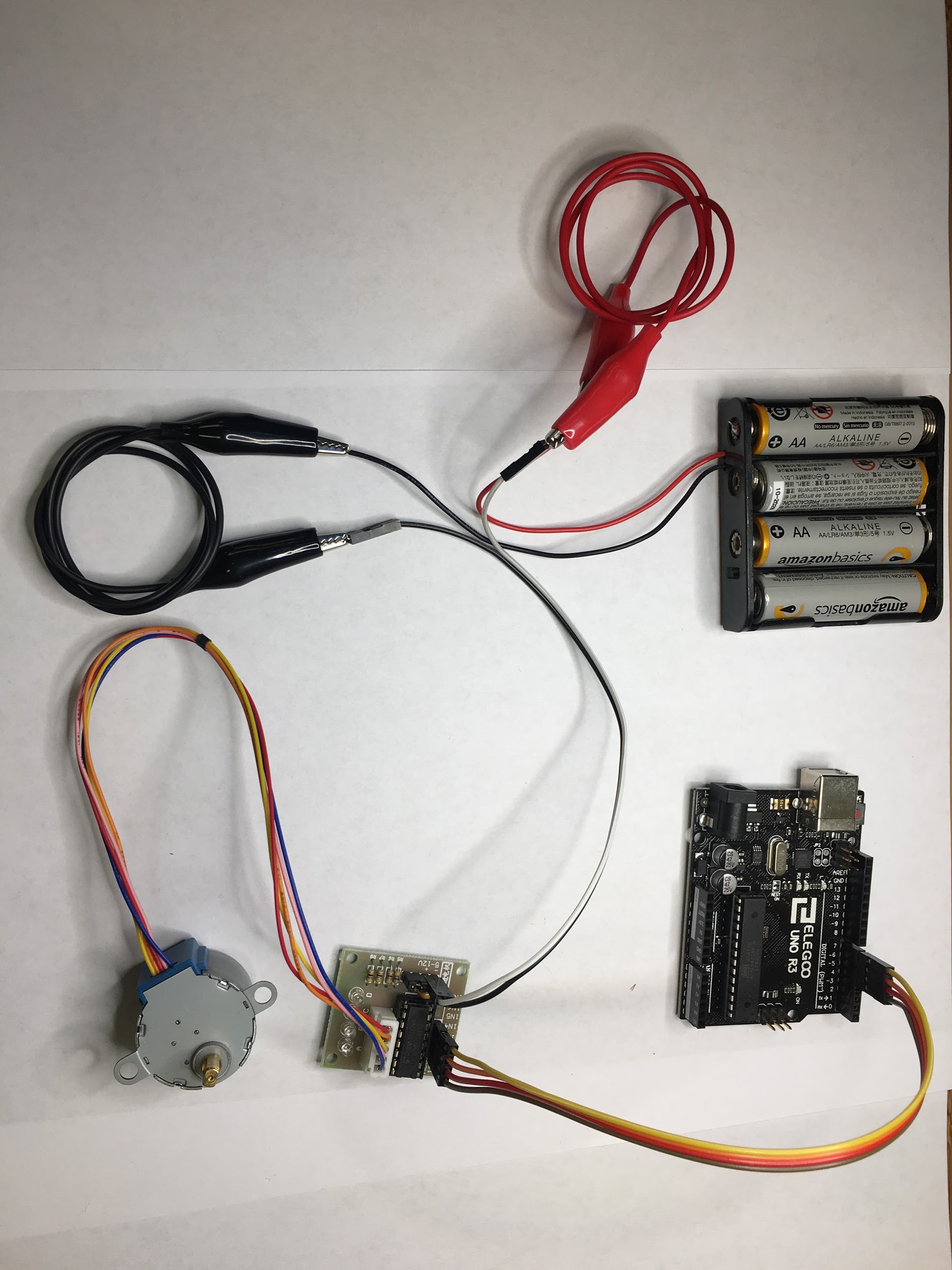
## Stepper Motor

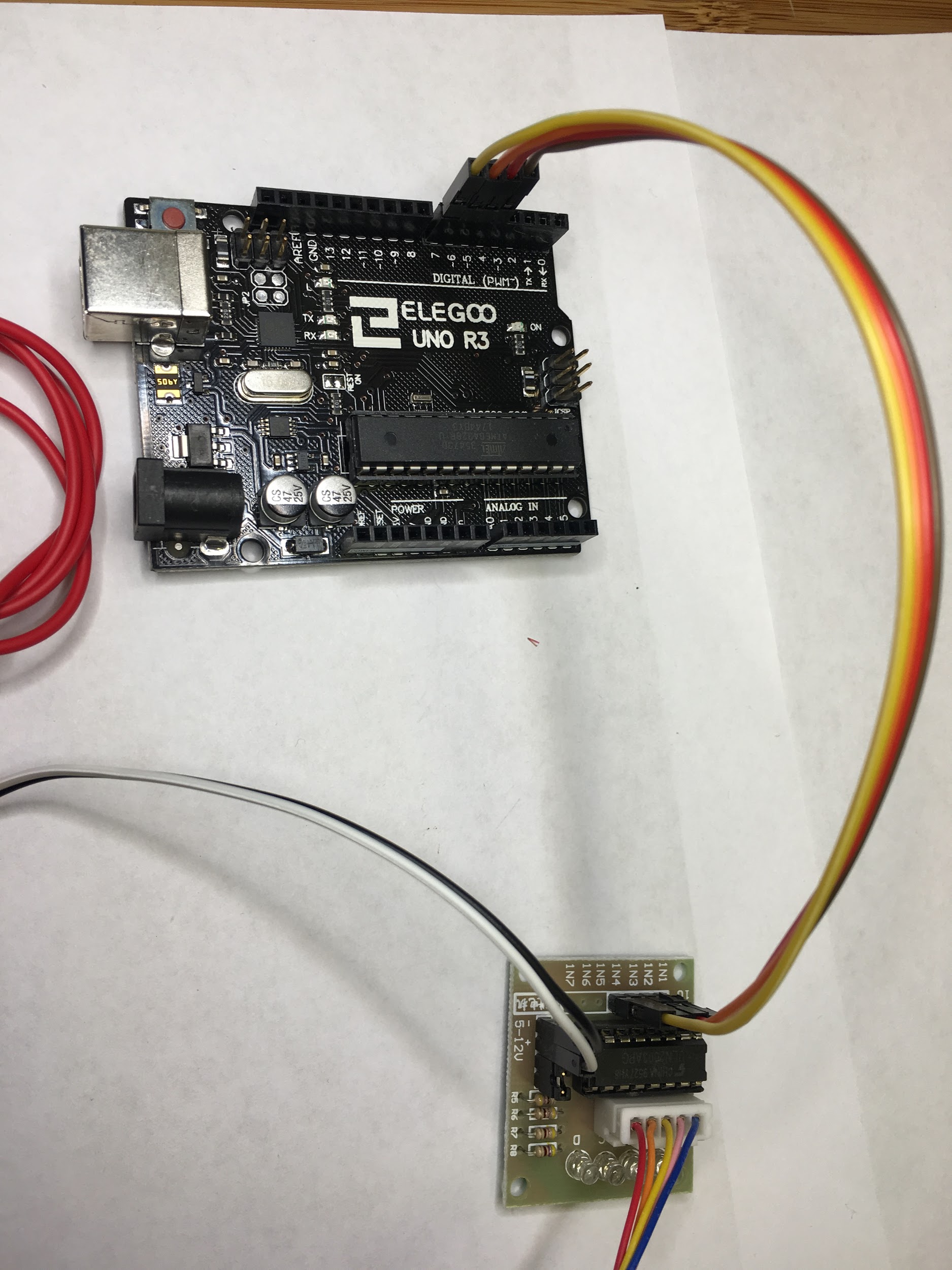
Stepper motors are frequently used to control the speed and position of mechanical systems. The standard Arduino library <Stepper.h> is used which supports unipolar and bipolar stepper motors, two or four pins. A standard Arduino library does not have to be separately downloaded and installed. Below is example code (from the Arduino website). Every stepper motor has a specific number of steps per revolution. As usual with libraries, you create an instance of the stepper object and invoke methods of the object.

### Example 2: Stepper

This example reads a voltage on an analog input pin and moves the motor the number of steps corresponding to the voltage. The Arduino will typically connect to a U2004 Darlington Array if you're using a unipolar stepper or a SN754410NE H-Bridge if you have a bipolar motor.

Here is an example of a two control pin, unipolar UN2003 driver module setup





The code is to control the position with a potentiometer knob.

/\*

\* MotorKnob

\*

\* A stepper motor follows the turns of a potentiometer

\* (or other sensor) on analog input 0.

\*

\* http://www.arduino.cc/en/Reference/Stepper

\* This example code is in the public domain.

\*/

#include <Stepper.h>

// change this to the number of steps on your motor

#define STEPS 100

// create an instance of the stepper class, specifying

// the number of steps of the motor and the pins it's

// attached to

Stepper stepper(STEPS, 8, 9, 10, 11);

// the previous reading from the analog input

int previous = 0;

void setup() {

// set the speed of the motor to 30 RPMs

stepper.setSpeed(30);

}

void loop() {

// get the sensor value

int val = analogRead(0);

// move a number of steps equal to the change in the

// sensor reading

stepper.step(val - previous);

// remember the previous value of the sensor

previous = val;

}

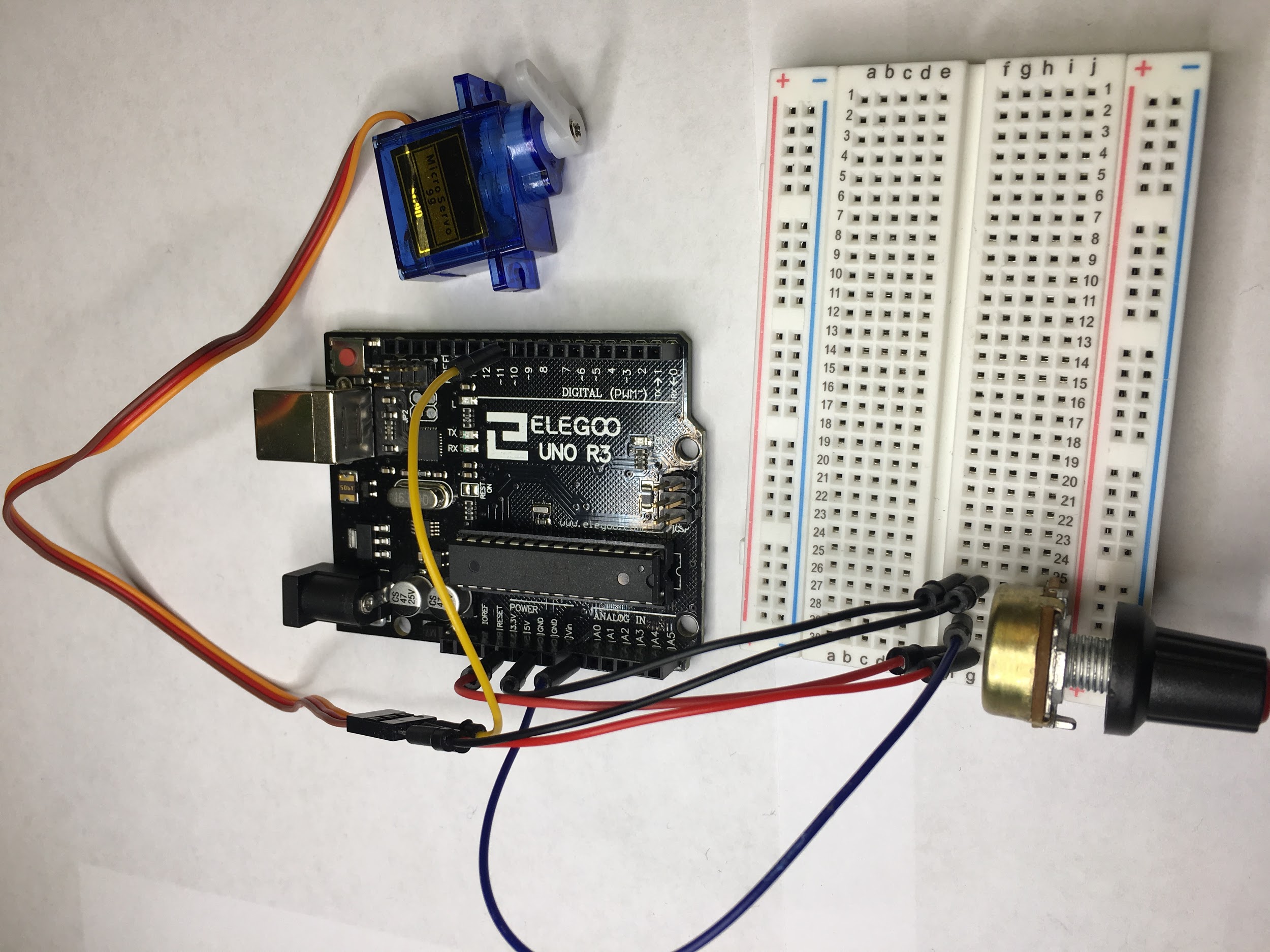
## Servo Motor

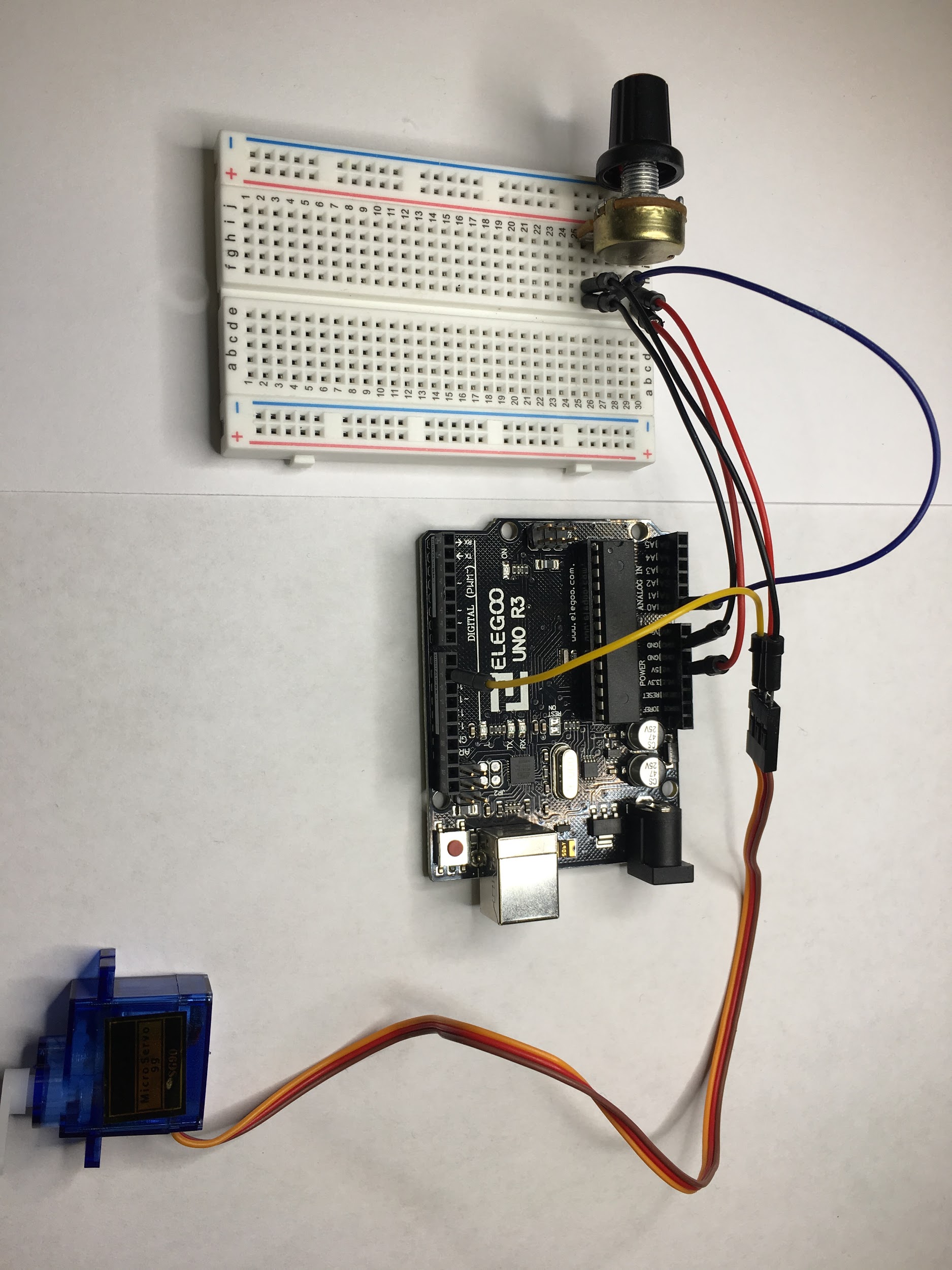
Servo motors on Arduino compatible devices can be controlled with the standard <Servo.h> library (included with the default Arduino IDE). A servo allows the motor to be positioned precisely, typically between 0 and 180 degrees.

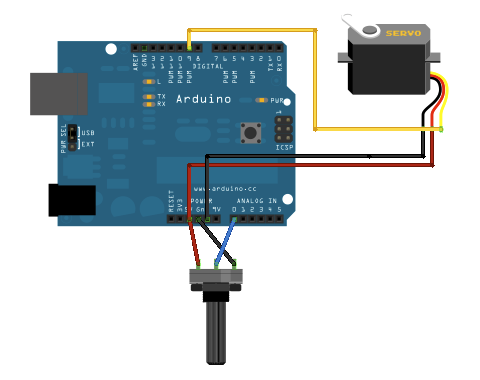
The code example below positions a stepper motor using a potentiometer. The simple fritzing diagram is also from the [www.arduino.cc](http://www.arduino.cc/) website.

### Example 3: Servo Motor

Below are pictures of the servo motor example setup from two different angles.





/\*

Controlling a servo position using a potentiometer (variable resistor)

by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

modified on 8 Nov 2013

by Scott Fitzgerald

http://www.arduino.cc/en/Tutorial/Knob

\*/

#include <Servo.h>

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

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

}

void loop() {

val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)

val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)

myservo.write(val); // sets the servo position according to the scaled value

delay(15); // waits for the servo to get there

}