Arduino Motor Workshop

Written by Mark Webster, Ousema Zayati, and Connor Sheeran, Ed Shattuck

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.

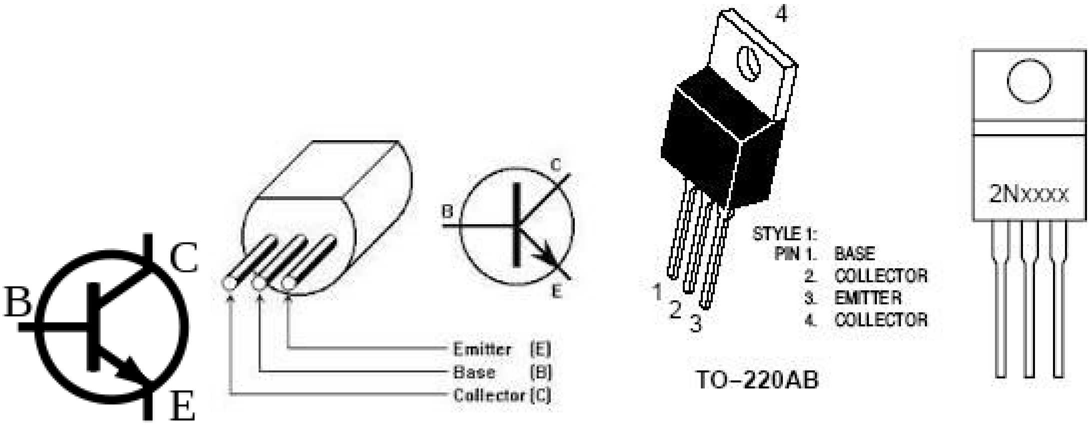
A diode is placed parallel to the motor, but with forward bias opposite the normal current path. No current will flow through the diode when the motor is running, but as the motor is stopping and creating a reverse back-voltage the diode shorts out the motor-generator so the transistor isn’t damaged.

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.

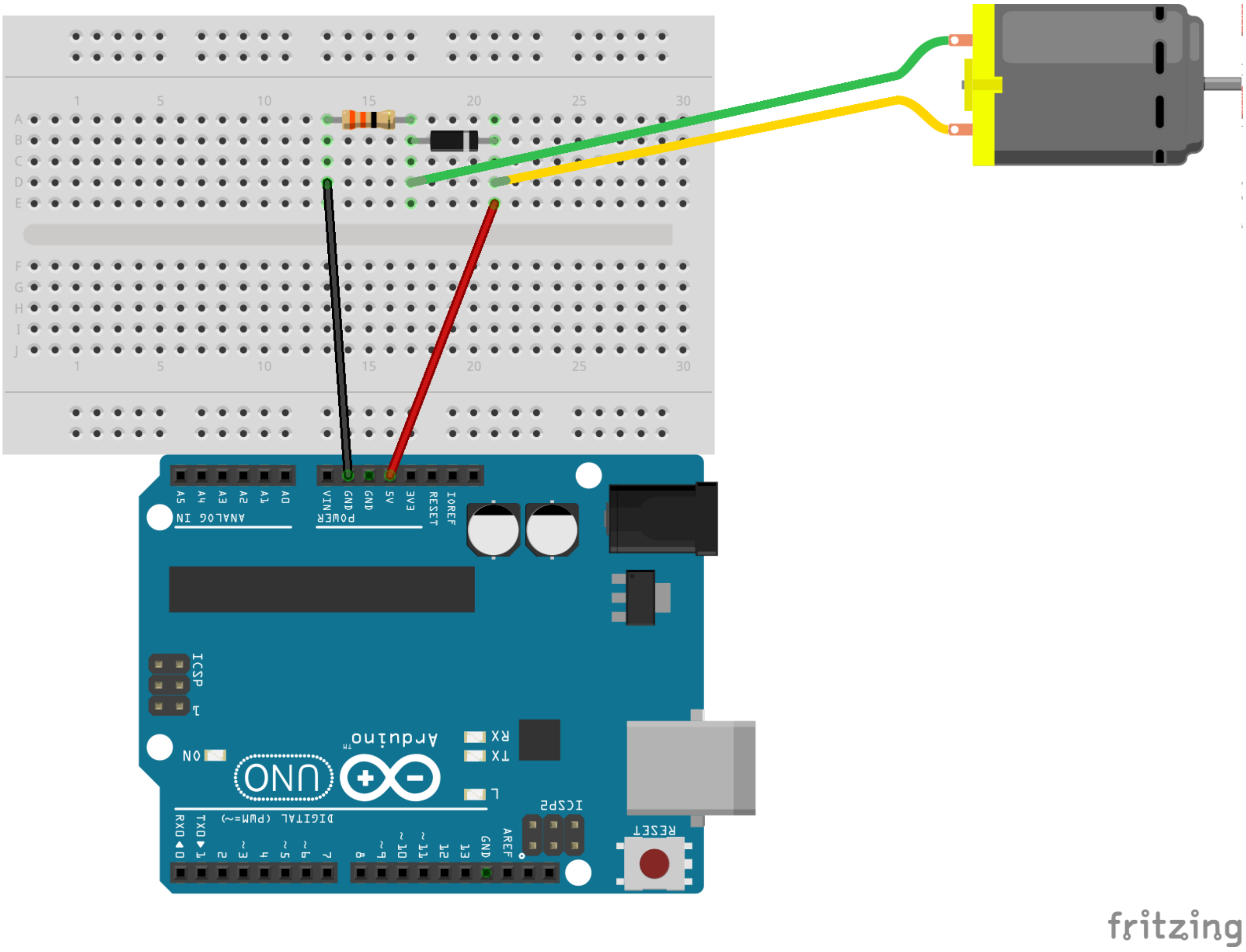


NPN transistors, with two different cases.

Parts List for DC Motor example.
Arduino Uno
NPN like 2N5551
Breadboard
290 ohm resistor (approximate)
33 ohm resistor (approximate)
1N4001 diode
Various male-male jumper cables
Low power DC motor

Example 1.1 Verify Motor

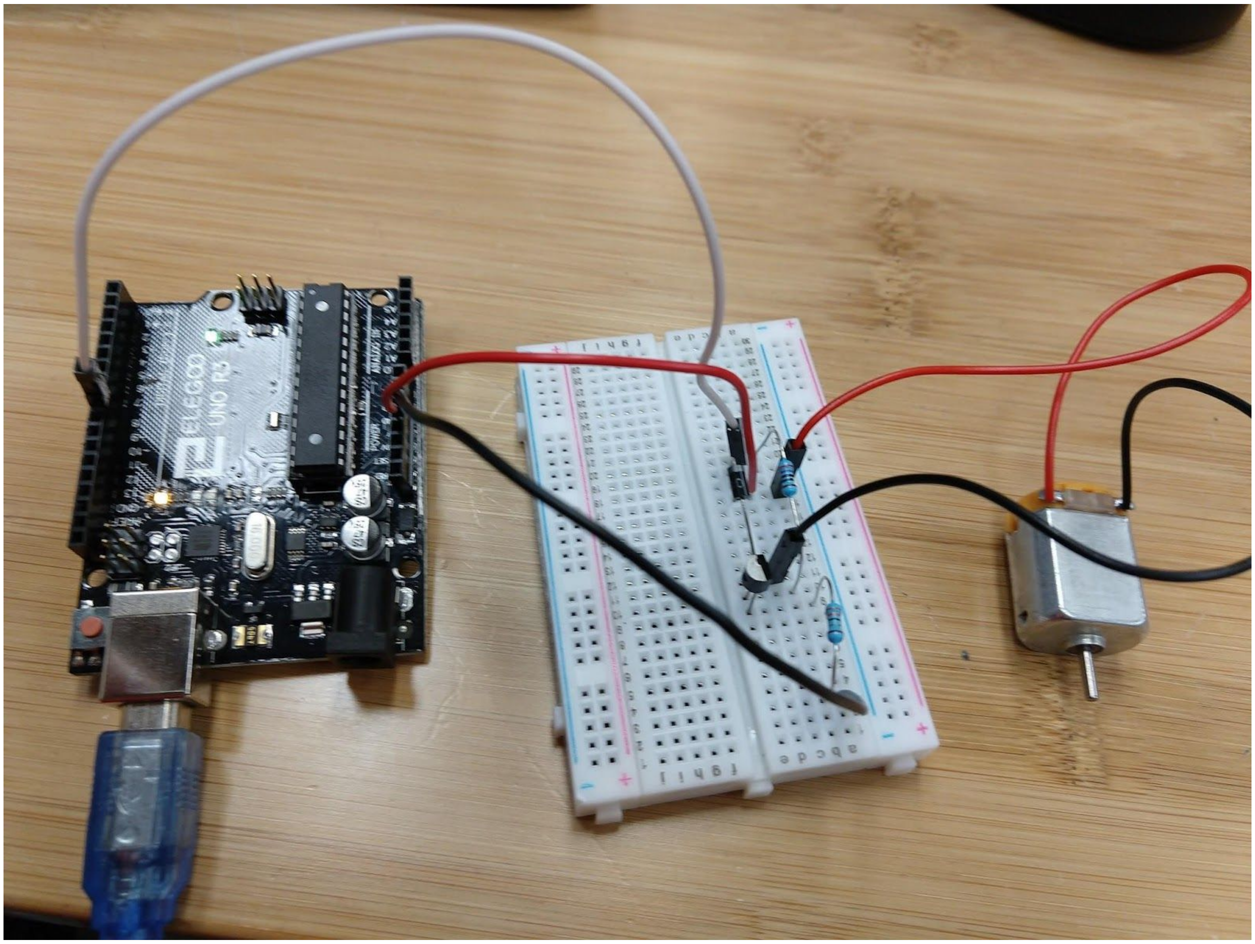
First connect the diode, motor, 33 ohm resistor and Arduino together to verify the motor work.

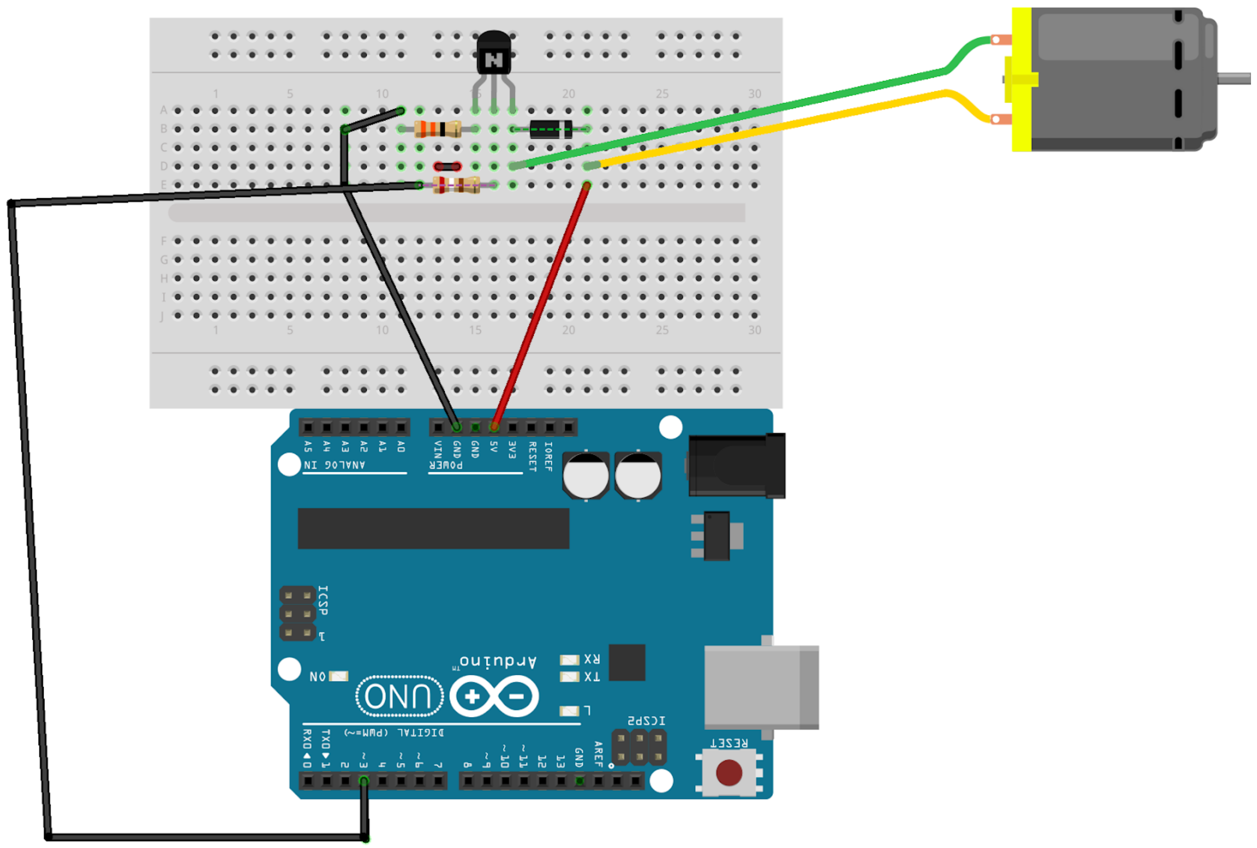


BEHAVIOR:
When the Arduino is connected to power (such as USB or a 9V battery), the motor should start turning full speed.

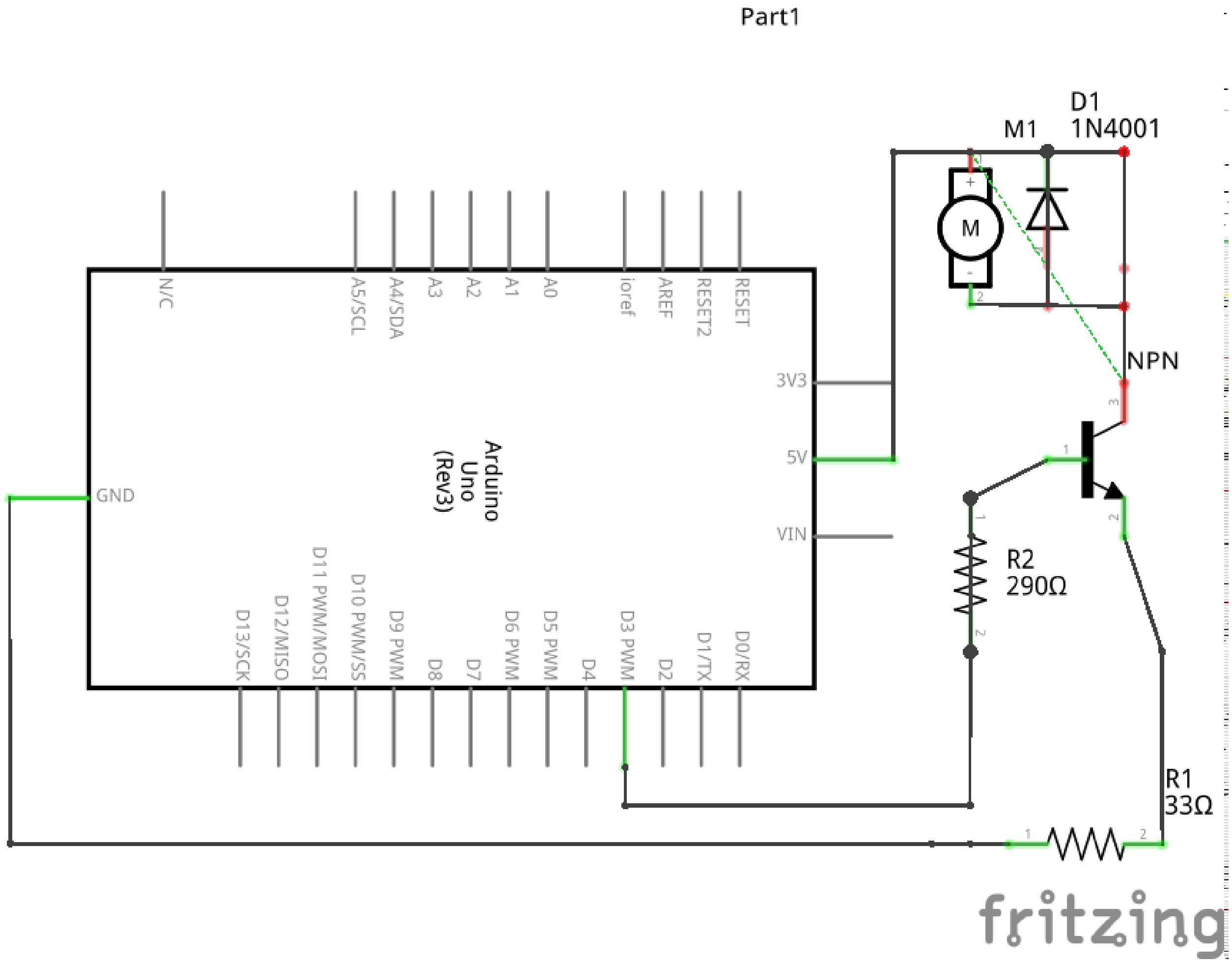
Exercise 1.2 With a transistor

Once that works, include the transistor and second diode. The pin 3 connects through the 290 ohm resistor to the base of the transistor.


ELEGOO
UNO R3


A
B
C
D
E
F
G
H
I
J
1
5
10
15
20
25
30
N
A5 A4 A3 A2 A1 A0
ANALOG IN
VIN GND GND 5V 3V3 RESET IOREF
POWER
ICSP
1
ON
UNO
Arduino™
RX
TX
L
ICSP2
RESET
DIGITAL (PWM=~)
RX0 0
TX0 1
2
~3
4
~5
~6
7
8
~9
~10
~11
12
13
GND
AREF

fritzing



.

Arduino Code Example

```
/*

Turn on DC Motor with NPN transistor


motorPin is connected through a 270 ohm resistor to transistor base

Transistor emitter is connected through 33 ohm resistor to ground

Transistor collector is connected to the black lead of the motor

transistor collector is also connected to a reverse biased diode, like a 1n4001
```

```
Red lead of motor is connected to +5 V of Arduino Uno. And is connected to the other end of diode

*/


int motorPin = 3;

void setup() {

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

pinMode(motorPin,OUTPUT);

}

// Turn motor on, then off

void loop() {

  // put your main code here, to run repeatedly

  digitalWrite(motorPin,HIGH);

  delay(1000);

  digitalWrite(motorPin,LOW);

  delay(1000);

}
```

Exercise 1.3 PWM Variable speed

```
// Advanced version. Must be on a PWM pin

In the loop() {

   Int i;

   for(i=100,i<255;i++) {

       analogWrite(motorPin,i);

      delay(20);

   }

digitalWrite(motorPin,LOW);
```

```
delay(500);
```

Use analogWrite(motorPin, 150-255); instead of the digitalWrite. This allows variable speed using PWM

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

Parts list:

Stepper motor (5 wire unipolar)

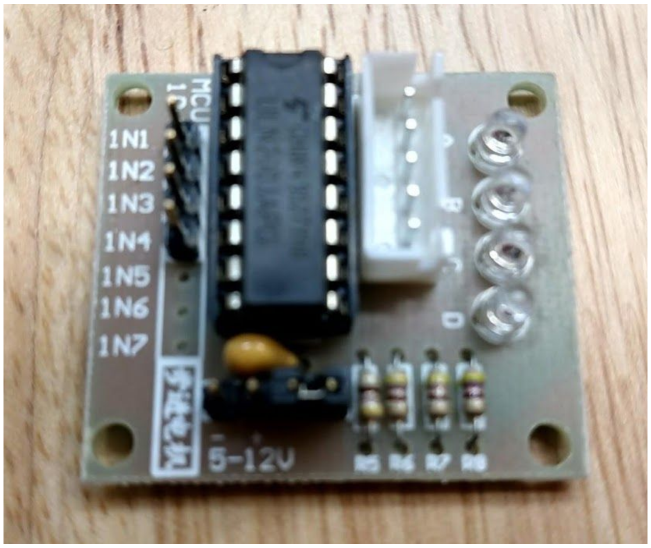
H bridge controller, ULN2003, on a board

Arduino Uno

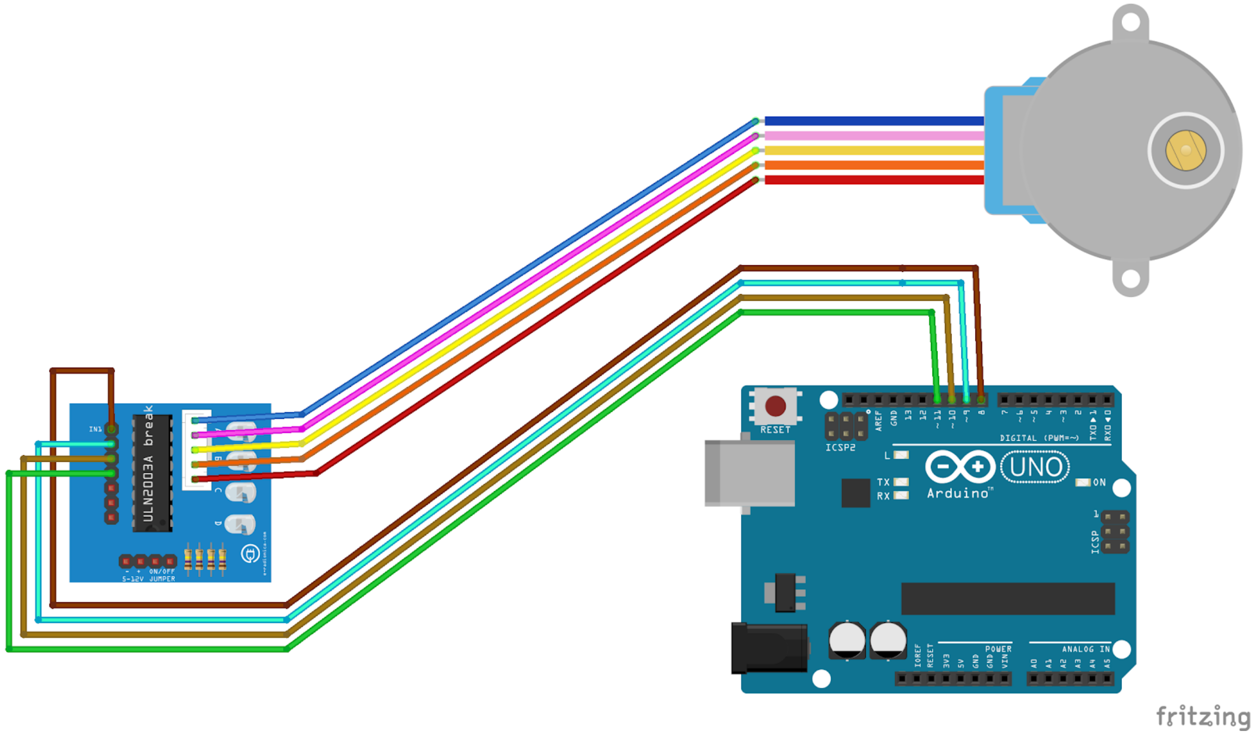
Battery (4 aa) or connect directly to +5 v on Arduino if the stepper is low power

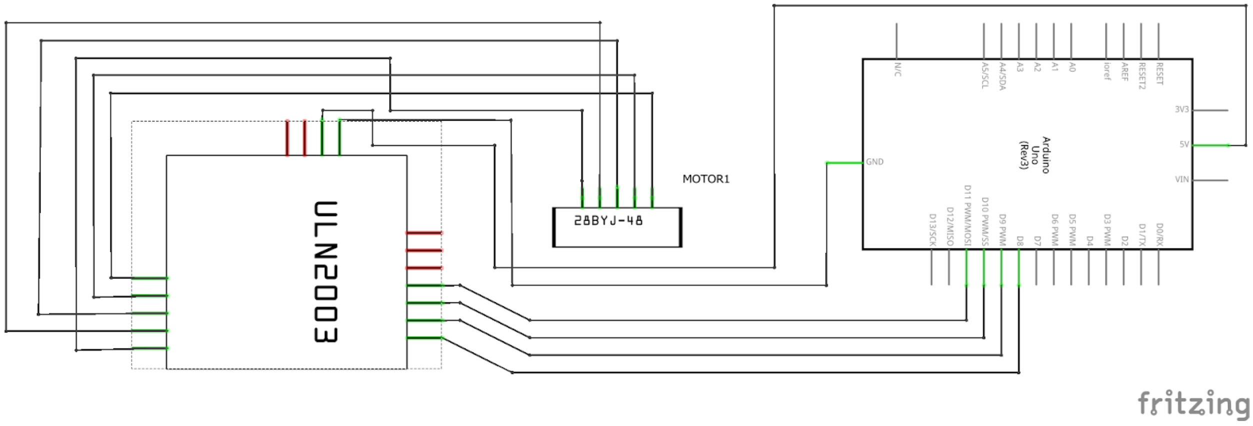
wires

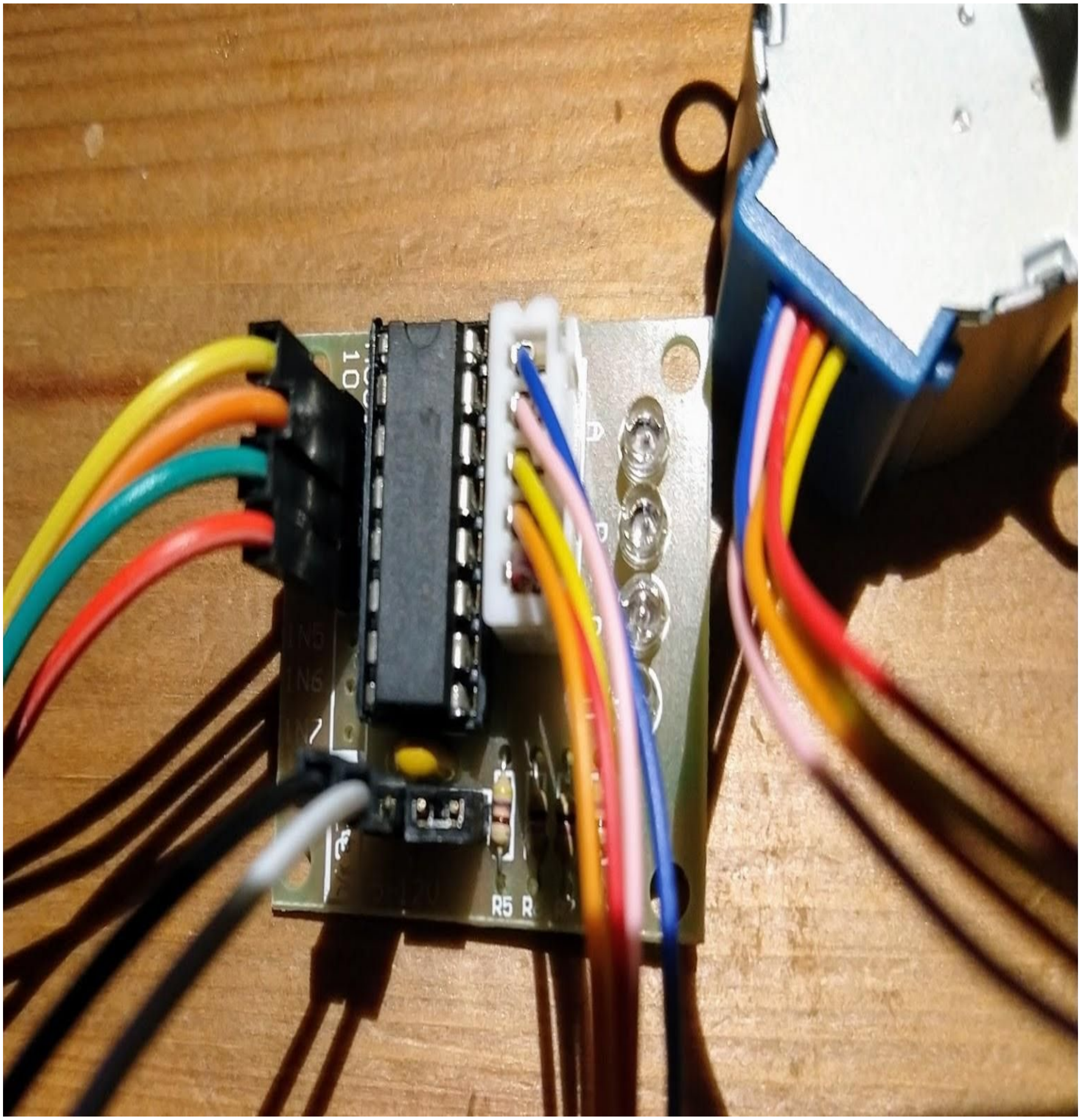
The Arduino will typically connect to a ULN2003 H-bridge chip for a servo motor controller.

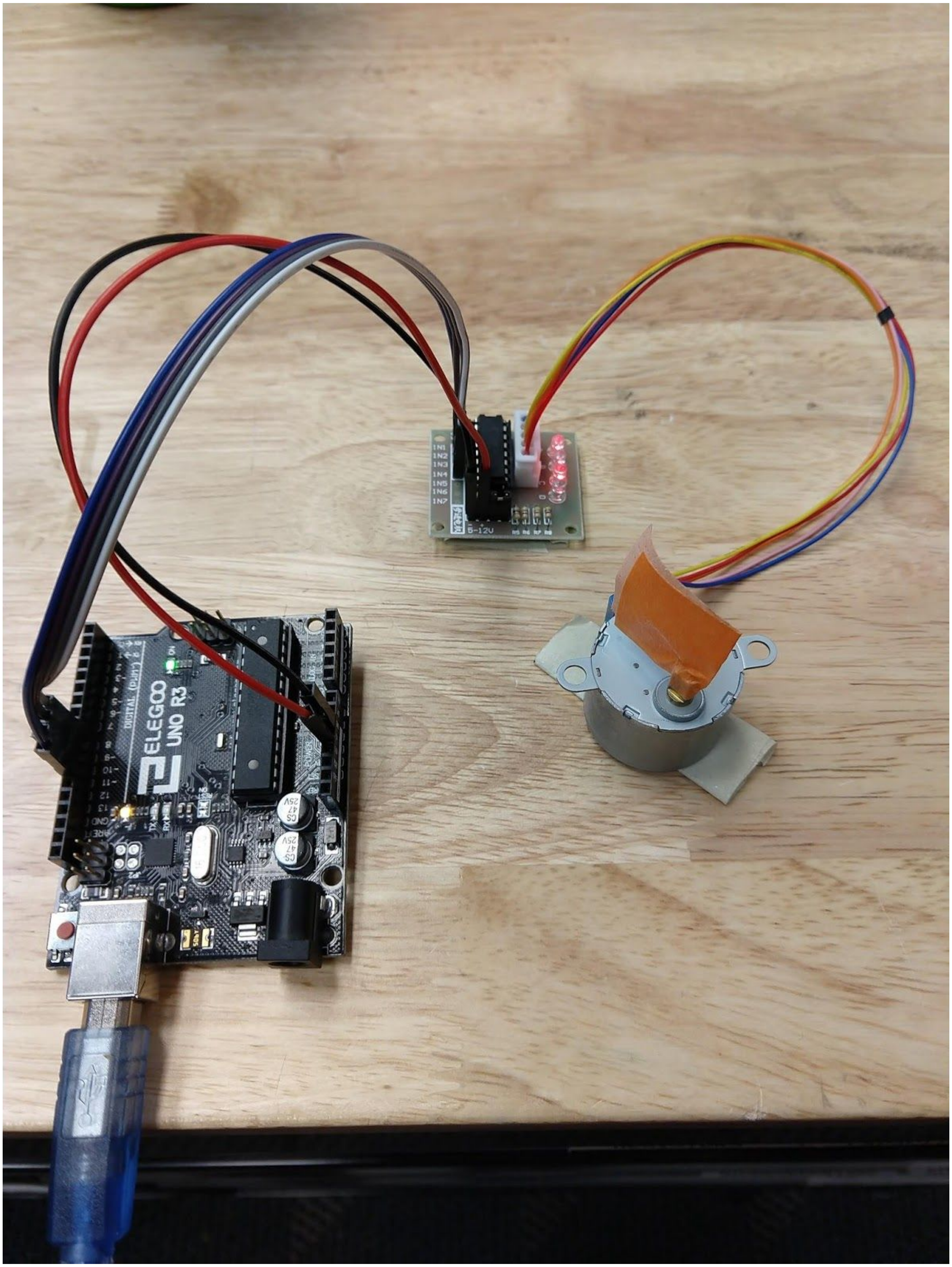


Here is an example of a two control pin, unipolar ULN2003 driver module setup




ULN2003
MOTOR1
28BYJ-48
Arduino
Uno
(Rev3)
N/C
A5/SCL
A4/SDA
A3
A2
A1
A0
ioref
AREF
RESET2
RESET
3V3
5V
VIN
GND
D13/SCK
D12/MISO
D11 PWM/MOSI
D10 PWM/SS
D9 PWM
D8
D7
D6 PWM
D5 PWM
D4
D3 PWM
D2
D1/TX
D0/RX
fritzing


IN5
IN6
R5


ELEGOO
UNO R3
IN1
IN2
IN3
IN4
IN5
IN6
IN7
5-12V

https://lastminuteengineers.com/stepper-motor-l298n-arduino-tutorial/

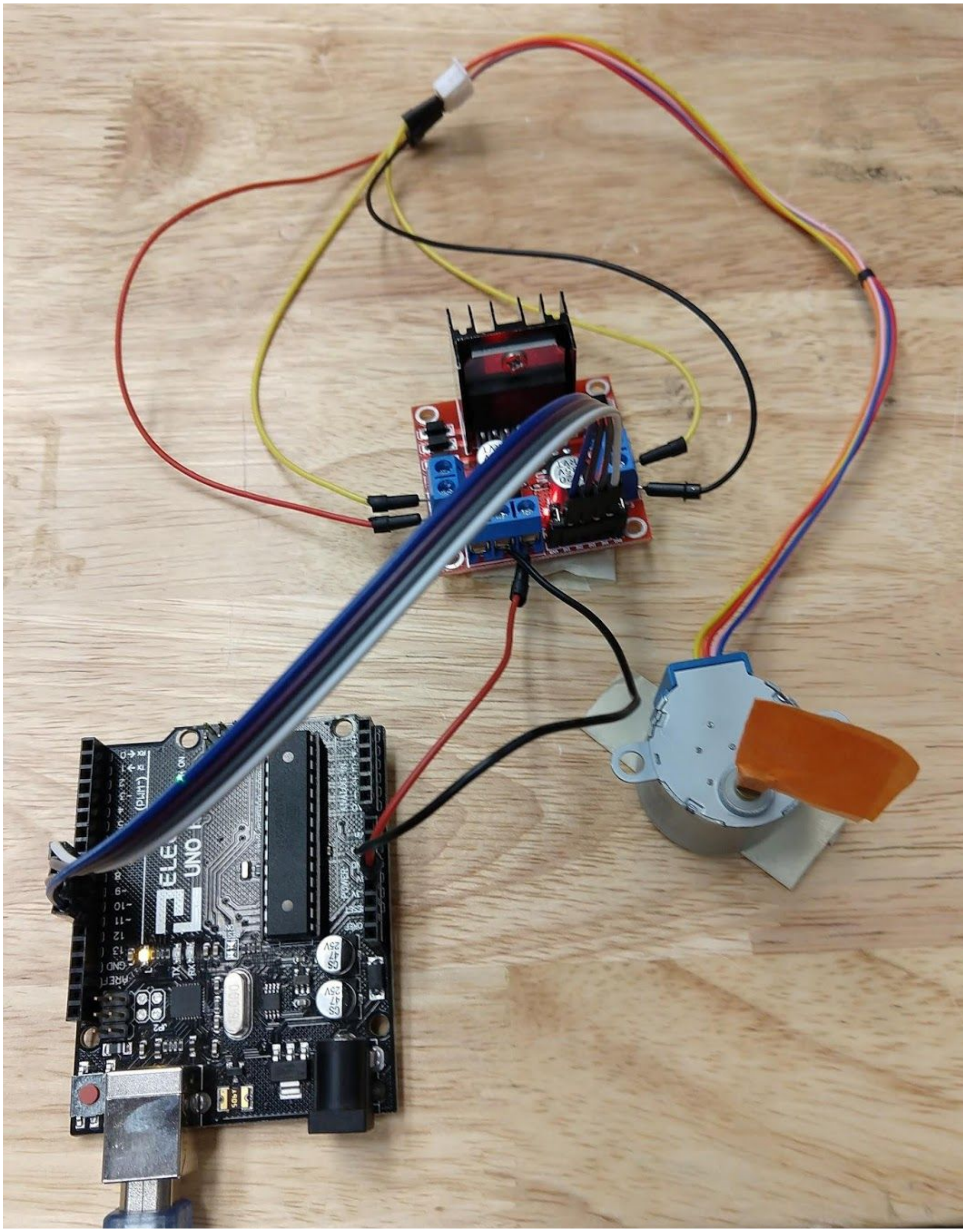
```
// Wires are connected to Arduino pins 8,9,10,11. Pin 8 connects to IN4, etc


#include <Stepper.h>

// change this to the number of steps on your motor
#define stepsPerRevolution 2048

// create an instance of the stepper class, specifying
// the number of steps of the motor and the pins it's
// attached to
Stepper stepper(stepsPerRevolution, 8, 9, 10, 11);



void setup() {

}

void loop() {

        // Turn fast
        stepper.setSpeed(10);
        stepper.step(stepsPerRevolution);
        delay(1000);

        // Turn slow
        stepper.setSpeet(1);
        stepper.step(stepsPerRevolution);
        delay(1000);


}
```

BEHAVIOR:

Stepper motor should turn fairly quickly for a while, pause, the turn slowly for a while.

Exercise 2.2 Dual H-bridge L298N

The L298N is designed to run two DC motors, or one 4 wire bipolar servo motor. In theory a 5 wire unipolar servo can be hooked up. The same code as for the ULN2003 would be used.



Servo Motor

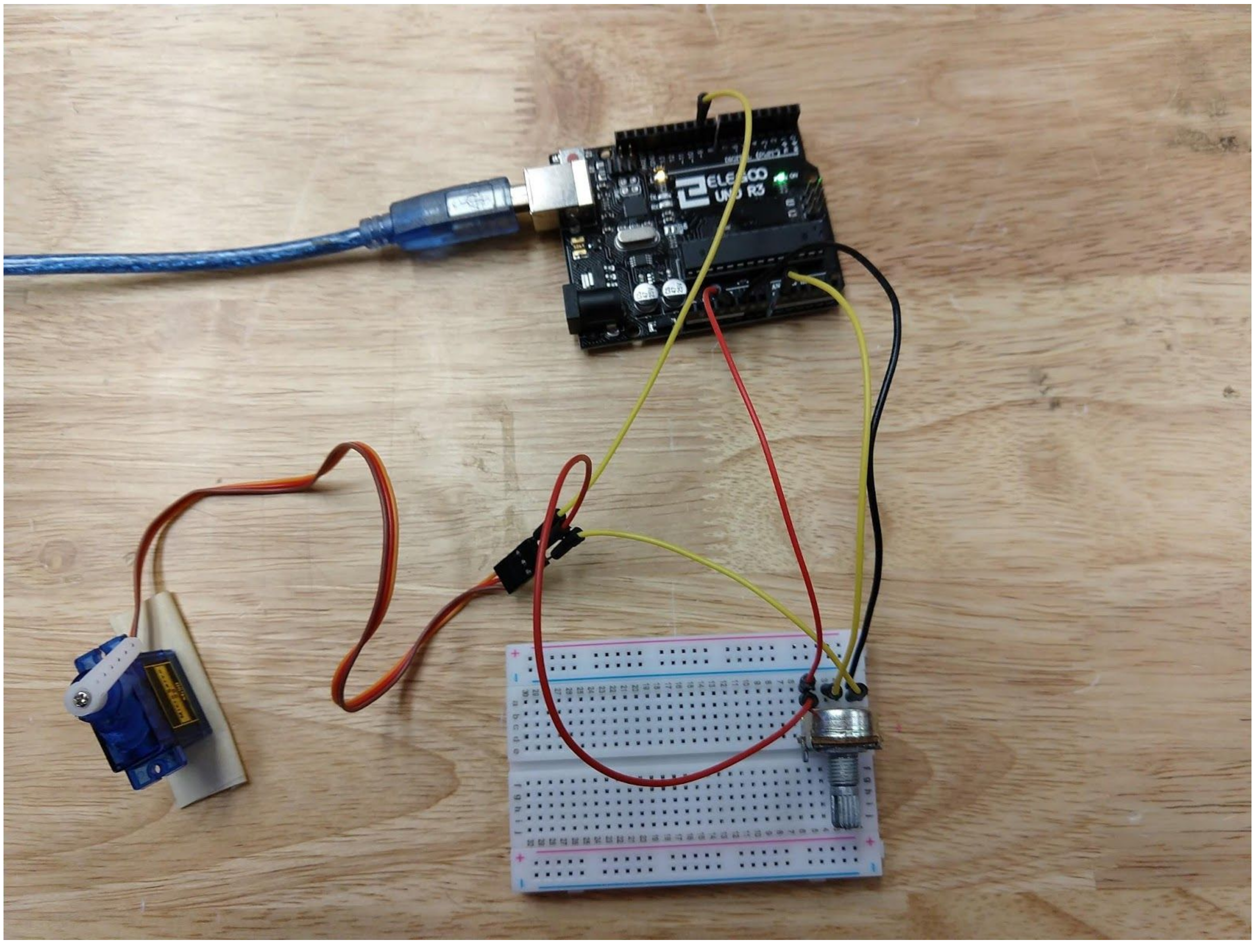
A servo motor turns to a specific angle. This would be useful for controlling a clock or a robot arm.
Using the <Servo.h> library built into the Arduino IDE, just send a value/angle to the servo motor which is attached to a certain pin.
Servo motors on Arduino compatible devices can be controlled with 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.

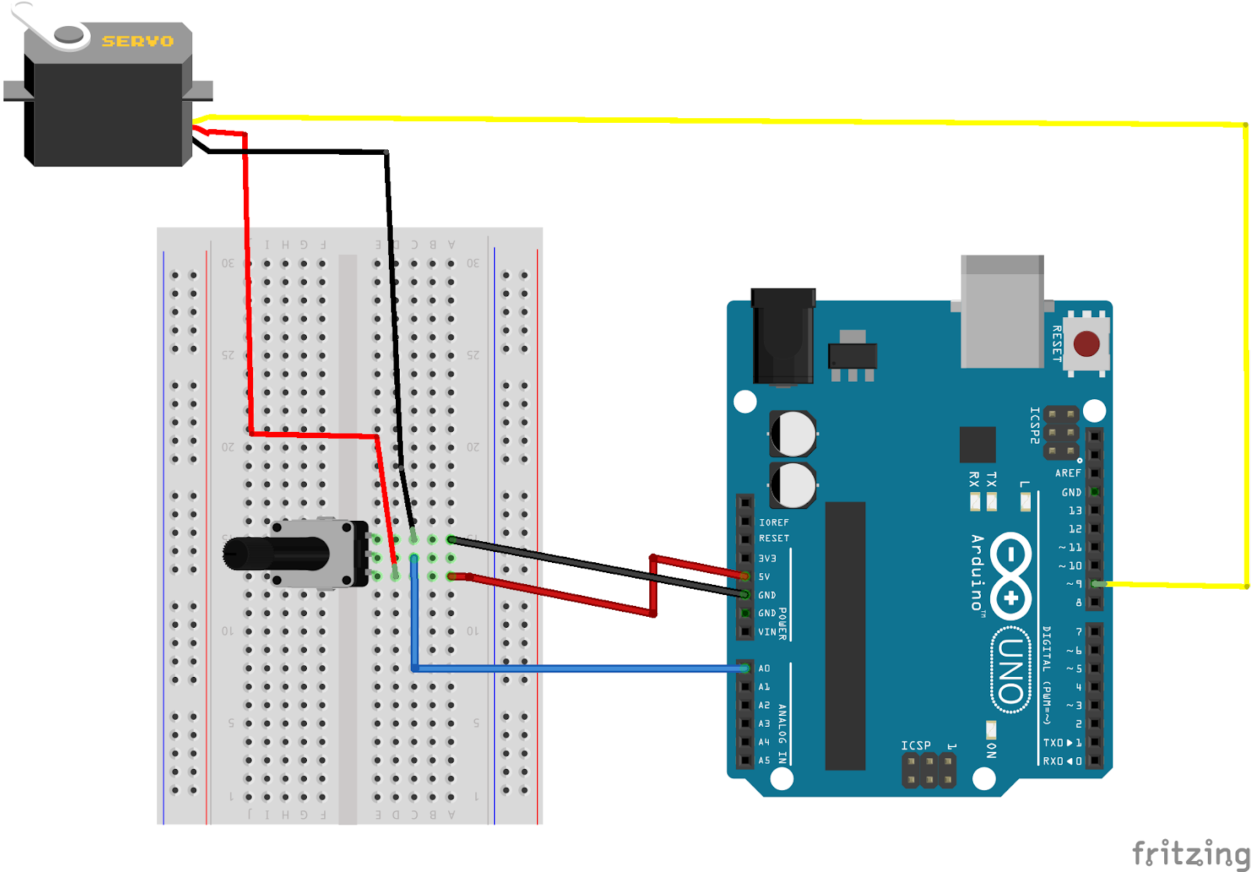
The code example below positions a stepper motor using a potentiometer. The simple fritzing diagram is also from the www.arduino.cc website.

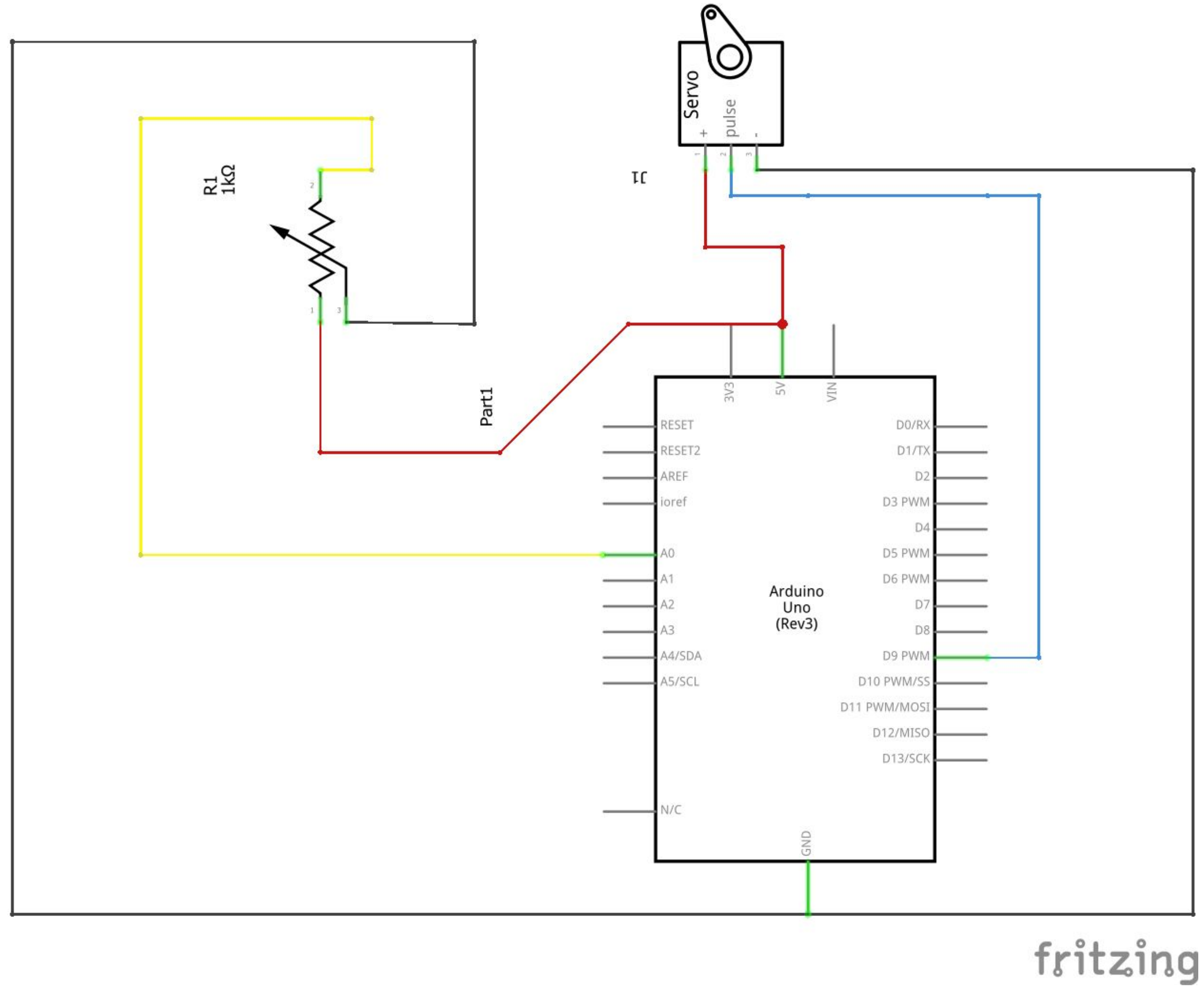
Example 3: Servo Motor

Parts List:
Servo motor (SG90)
Arduino Uno
Wires
Potentiometer 1K or 10K
Breadboard

Below is a picture of the servo motor setup.


ELEGOO
UNO R3


SERVO
RESET
ICSP2
TX
RX
L
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX0 ▶ 1
RX0 ◀ 0
DIGITAL (PWM=~)
Arduino™
UNO
ON
ICSP
1
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
fritzing



```
#include <Servo.h>

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

int potpin = 0;  // analog A0 pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin
int servoPin =9; // pin to connect to data line of servo

void setup() {
  myservo.attach(servoPin);  // attaches the servo motor  to the servo object
}

void loop() {
   val = analogRead(potpin);   // read a number 0-1023 from potentiometer on
   val = map(val, 0, 1023, 0, 180);  // Convert 0-1023 to the proportionate value 0-180
   myservo.write(val);  // send the position value to the servo motor
   delay(15);  // Give the servo time to move into position
}
```

Behavior:

As the potentiometer is rotated the servo will also rotate.