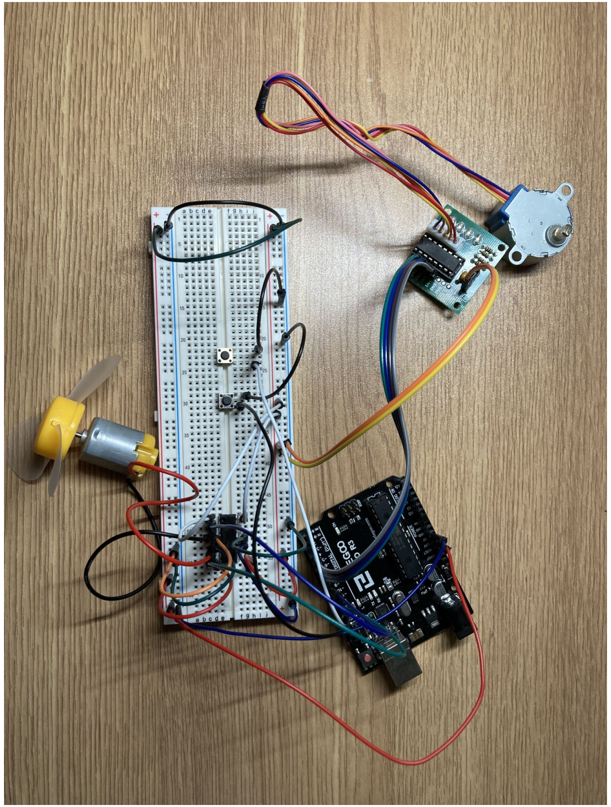
# Embedded System Workshops

## 06. DC Motors and Stepper Motors

*CCA Girls Who Code*

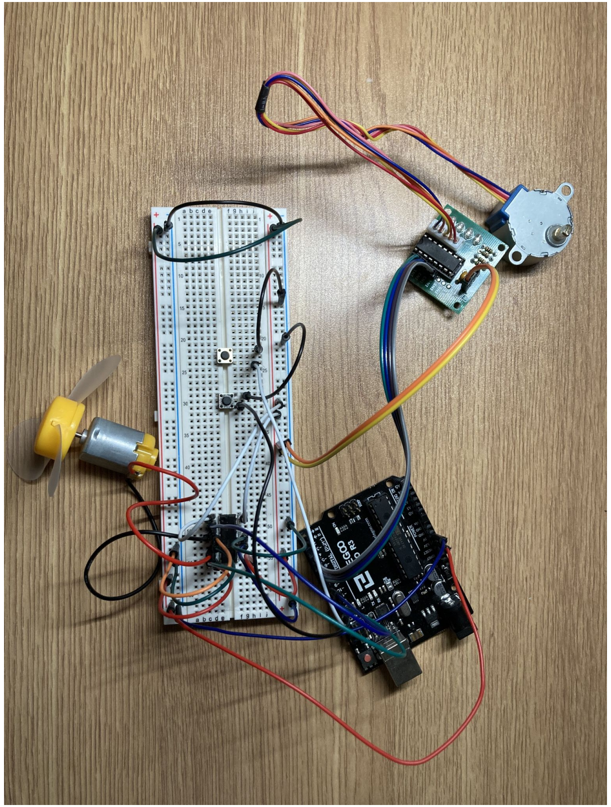
# Project Overview

- Purpose
  - Introduce the DC Motor and Stepper Motor and how to control them.
- Project
  - Create a motor-controlled fan
  - Create a stepper motor circuit controlled by buttons (optional)
- Grab your kit, and let's get started!



# What are we making?

- Motors Project
  - Make a fan powered by a DC motor controlled by a PWM module.
    - Turn this fan on using a button
  - (If time allows) Make a stepper motor controlled by a pair of buttons



# Parts List

DC Motor Project

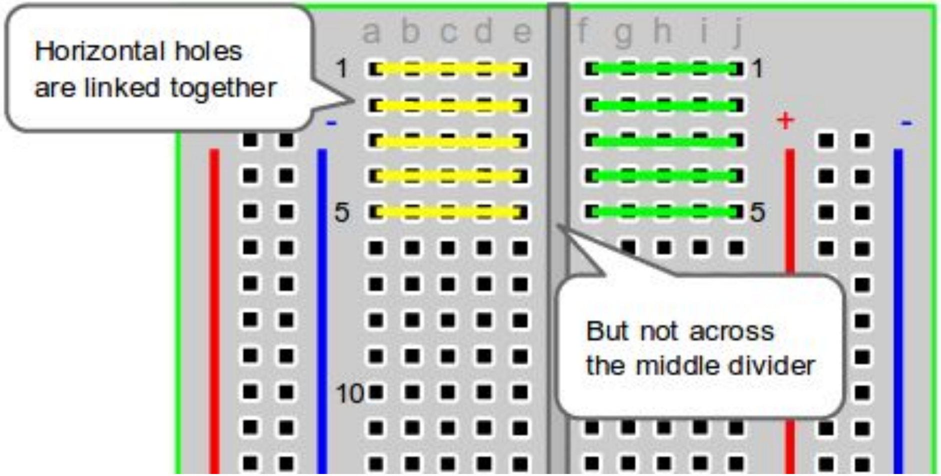
- Arduino UNO R3 Controller Board
- USB Cable
- Breadboard
- DC Motor
- L293D Motor Control Chip
- Male-to-male jumper wires

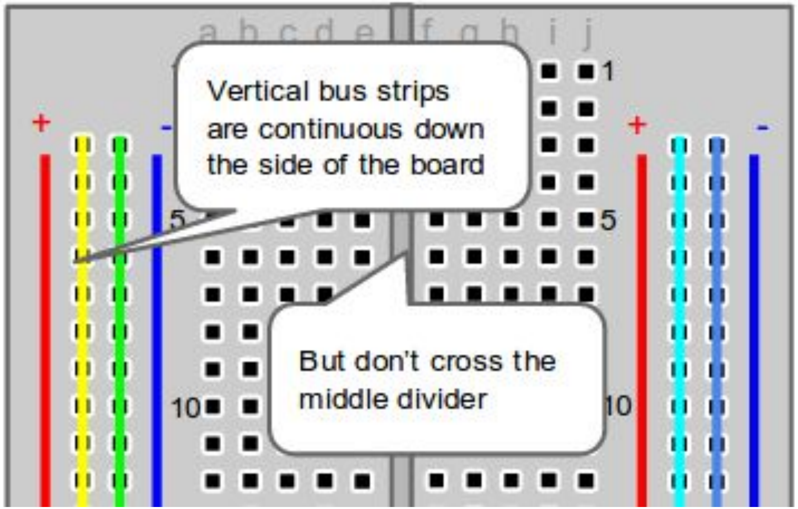
Stepper Motor Project (if time allows)

- Stepper Motor
- ULN2003 Stepper Motor Controller
- Buttons (x2)
- Male-to-female jumper wires

# Review

# Review: Breadboards Explained

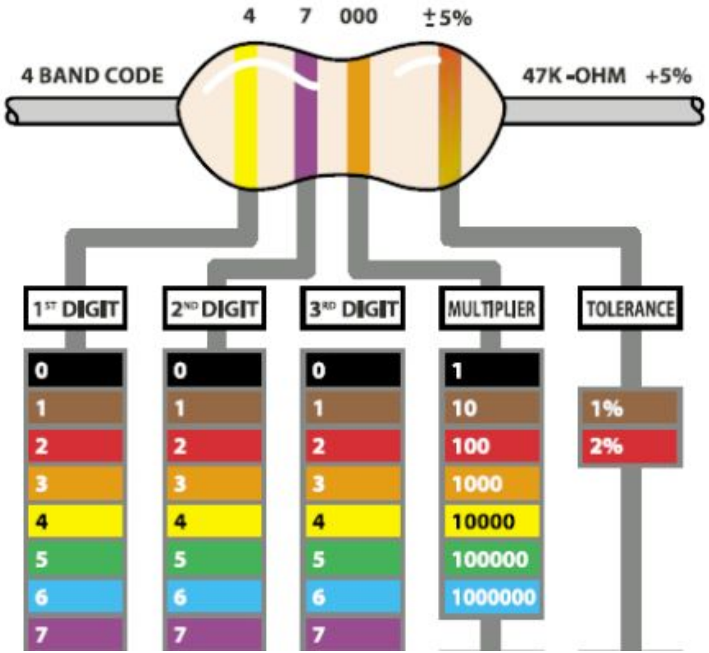

Horizontal holes are linked together
But not across the middle divider
a b c d e
f g h i j
1
5
10
+
-


Vertical bus strips are continuous down the side of the board
But don't cross the middle divider
a b c d e
f g h i j
1
5
10
+
-

Tip: It is good practice to have your power input connected to the red/positive rail and your ground pin connected to the blue/negative rail.

# Review: Resistors

- Resistors slow the electric current, and control where and how fast the current flows
- Resistance value is measured in ohms Ω, which is represented by colored stripes on the body of the resistor
- Each stripe has a different value depending on the color and location as shown in the reference chart
- A potentiometer is a variable resistor



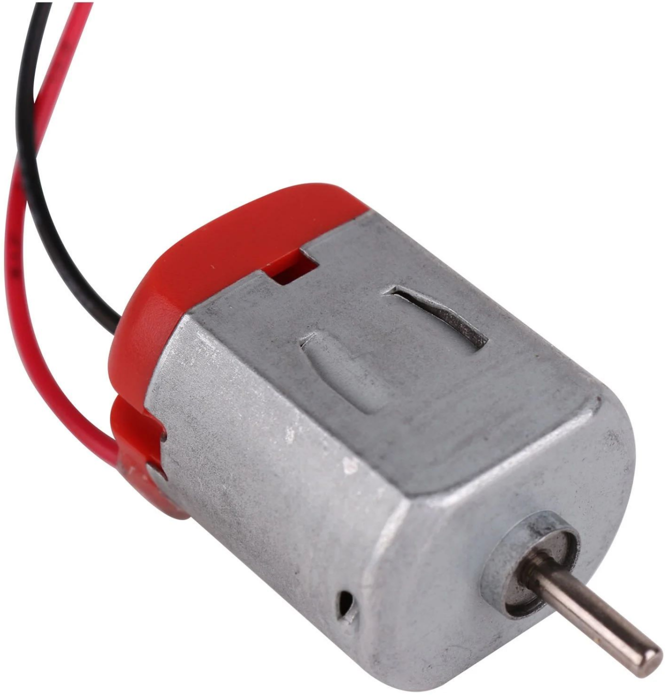
| $1^{ST}$ DIGIT | $2^{ND}$ DIGIT | $3^{RD}$ DIGIT | MULTIPLIER | TOLERANCE |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | |
| 1 | 1 | 1 | 10 | 1% |
| 2 | 2 | 2 | 100 | 2% |
| 3 | 3 | 3 | 1000 | |
| 4 | 4 | 4 | 10000 | |
| 5 | 5 | 5 | 100000 | |
| 6 | 6 | 6 | 1000000 | |
| 7 | 7 | 7 | | |

# Project Part 1: DC Motors

# Introduction to DC Motors

A **motor** is a device that converts electrical energy into rotational motion. The motors we are using run on DC (Direct Current) power.

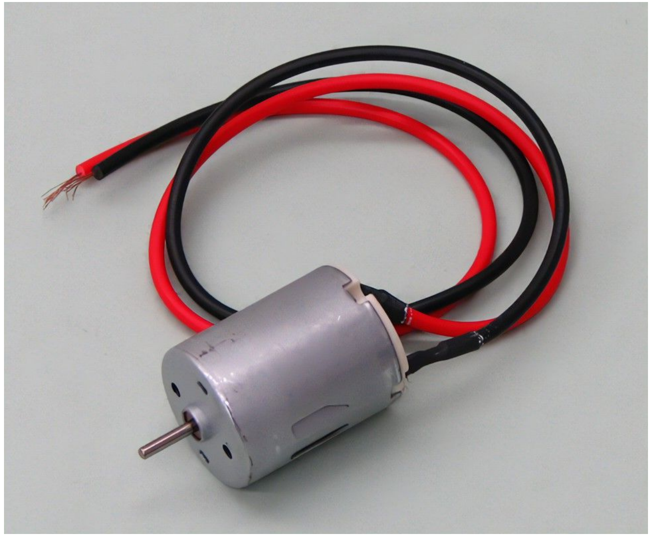
Electric motors are used in many areas of transportation like starter motors of internal combustion engines, electric and hybrid vehicles, and rail locomotives, although motors for traction applications primarily use AC (Alternating Current) power.



# Motor Components

The motor has two wires, a red and black wire.

However, unlike most other devices with color-coded wires, switching which wire is connected to power/ground will not harm the motor. It will instead reverse the direction of the motor's rotation.

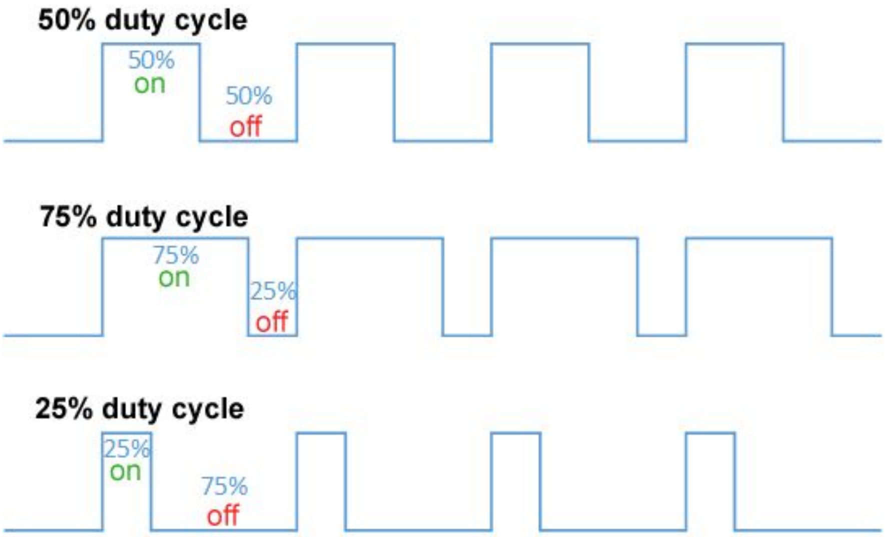


# Introduction to PWM

Because our motor has no internal control circuits, we need something else to control its power and direction. This is where **Pulse-Width Modulation** (PWM) comes in.

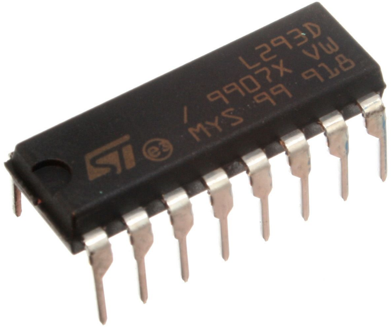
PWM does not change the actual current to the motor. Rather, it changes the duration at which that current is applied.

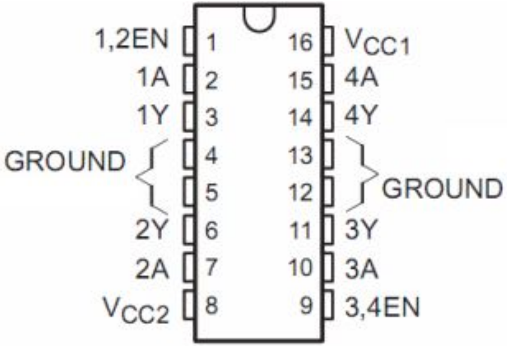
For instance, a 50% duty cycle will apply full power half the time.



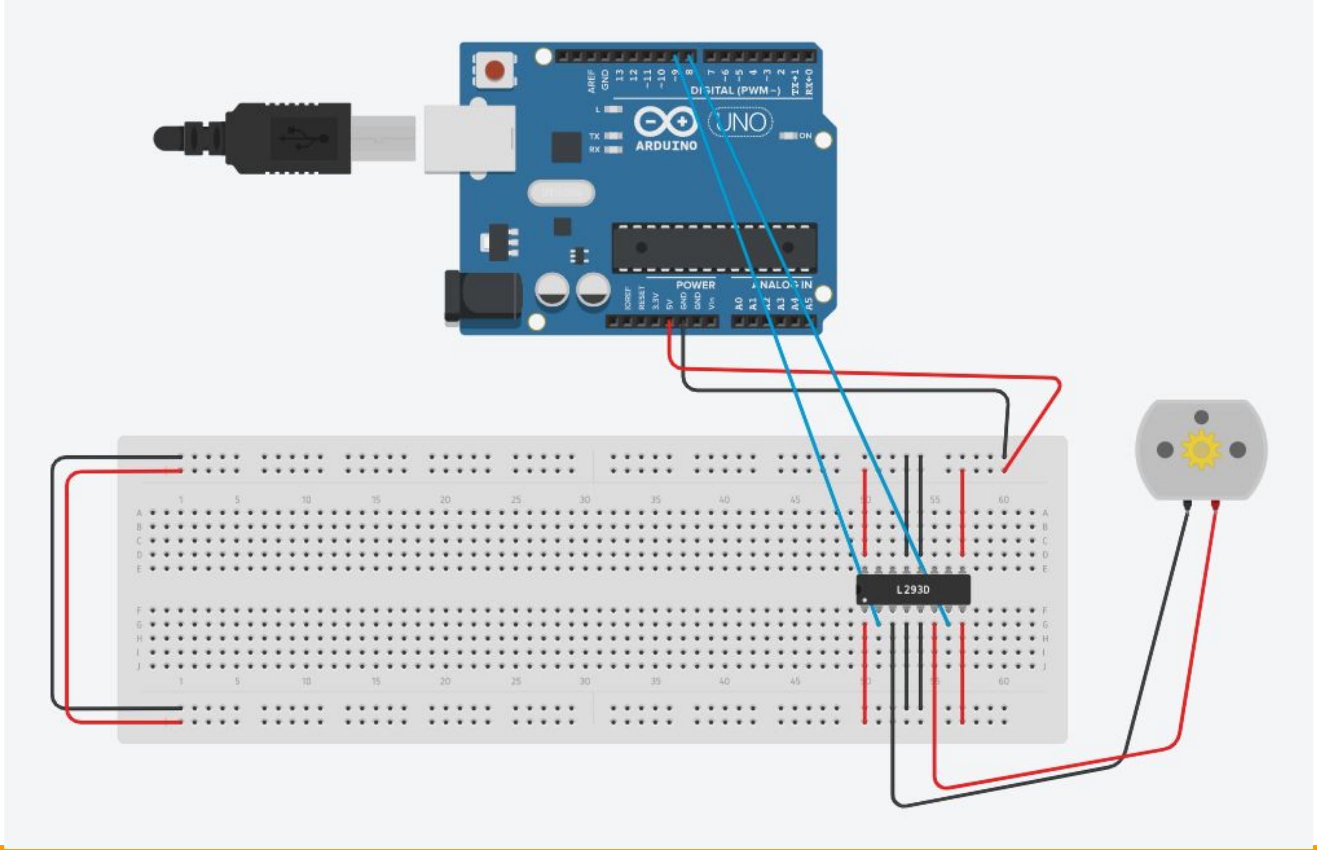
# L293D Motor Driver

The **L293D** is a PWM chip that can simultaneously control the direction and PWM of up to two motors (assuming it has the current to do so). We will be using one of these in today’s project.





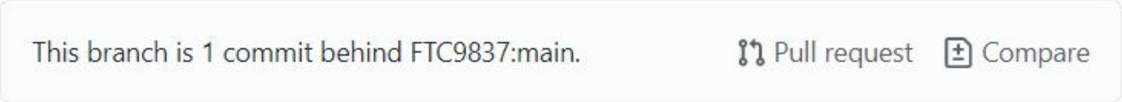
# Schematic


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
UNO
ARDUINO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
L293D

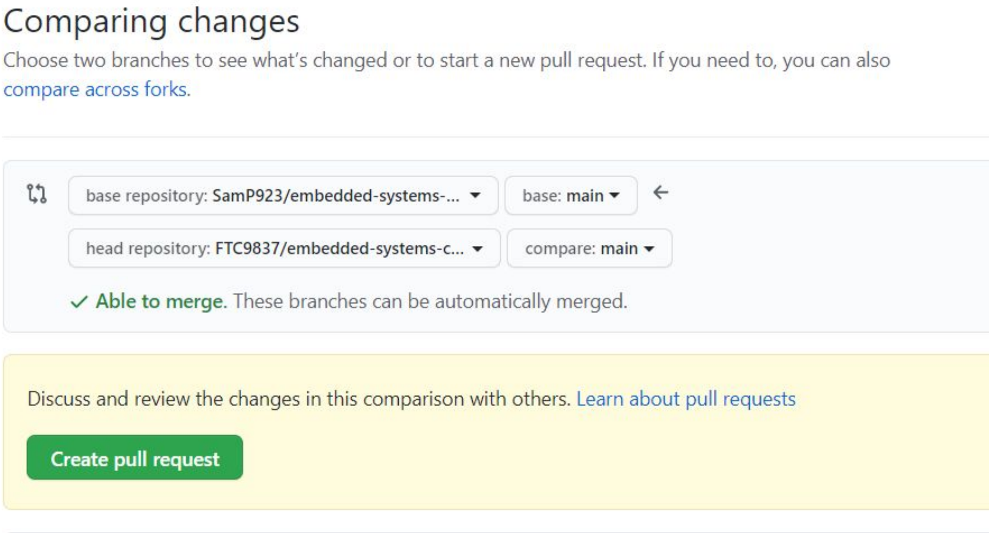
# Grab the Starter Code from GitHub

If you haven’t made the repository yet, check these slides

- Go to your repository (username/embedded-systems-course)
- Click “Compare”
- Switch the repos so yours is the base repository and FTC9837 is the head
- Create pull request (x2)
- Merge pull request (and confirm)
- You should have the lesson6 folder in your repository



This branch is 1 commit behind FTC9837:main.



Comparing changes

Choose two branches to see what’s changed or to start a new pull request. If you need to, you can also compare across forks.

base repository: SamP923/embedded-systems-...

base: main

head repository: FTC9837/embedded-systems-c...

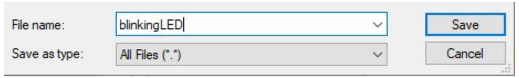
compare: main

✓ Able to merge. These branches can be automatically merged.

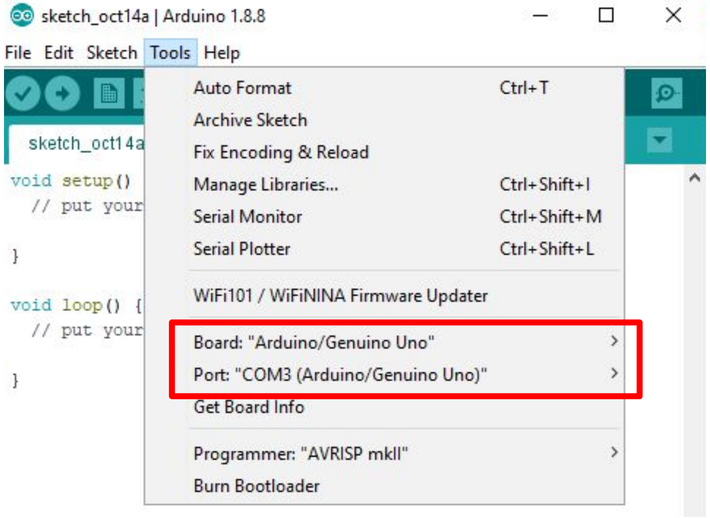
Discuss and review the changes in this comparison with others. Learn about pull requests

# Review: Setting up Arduino

- → Find and open Arduino on your desktop
- → Click “File” in the top left corner and click save
- → Save this tab as “MotorTest”
- → Connect the USB cord in your kit to the Arduino and the computer (USB port is on the left side of the monitor)
- → Open the “Tools” Window and make sure the board has been recognized and the port is “COM#(Arduino/Genuino Uno)”


File name: blinkingLED

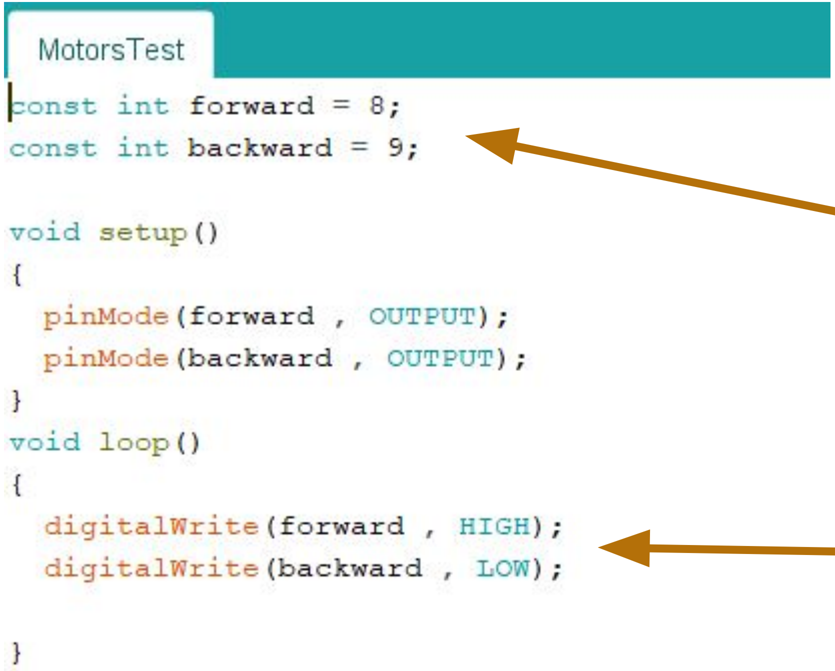
Save as type: All Files (*.*)


sketch_oct14a | Arduino 1.8.8

```
void setup() {
  // put your

}

void loop() {
  // put your

}
```

File
Edit
Sketch
Tools
Help
sketch_oct14a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM3 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

# Basic Motor Code

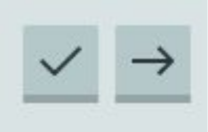


```
const int forward = 8;
const int backward = 9;

void setup()
{
  pinMode(forward , OUTPUT);
  pinMode(backward , OUTPUT);
}
void loop()
{
  digitalWrite(forward , HIGH);
  digitalWrite(backward , LOW);

}
```

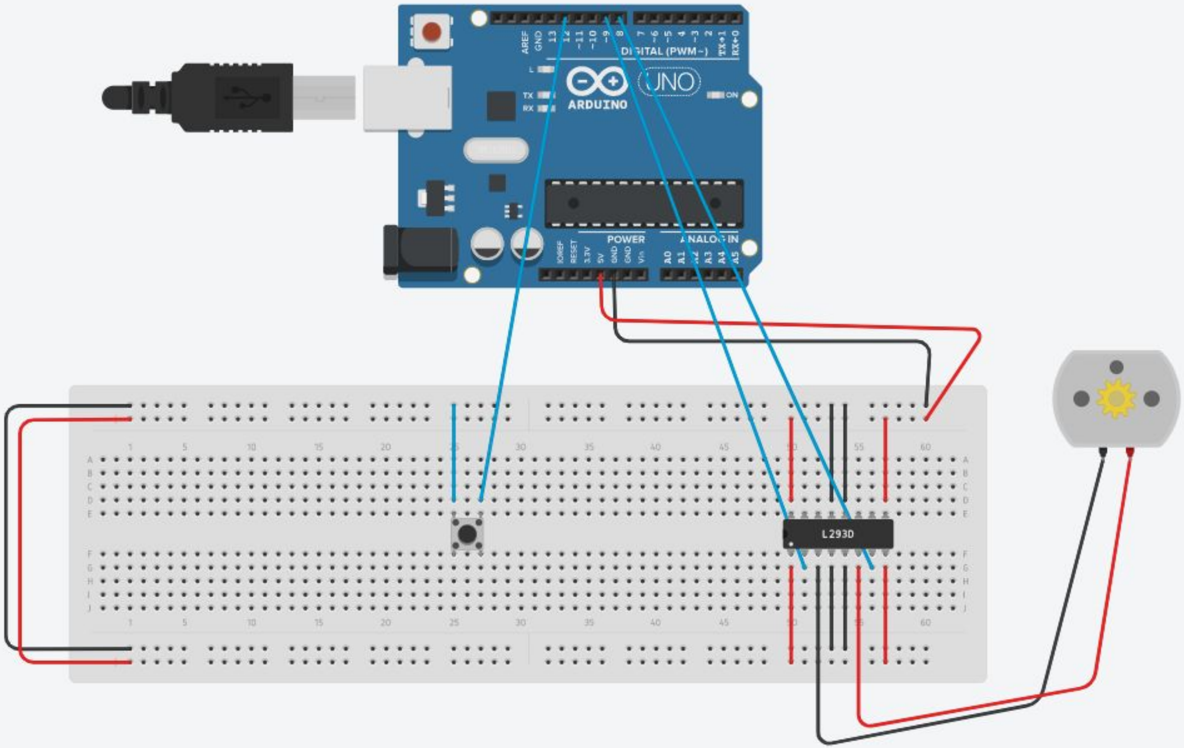
Initializes the forward and backward pins

Makes the motor start turning in the forward direction

To test your code, click the checkmark then the arrow!



# Schematic With Button

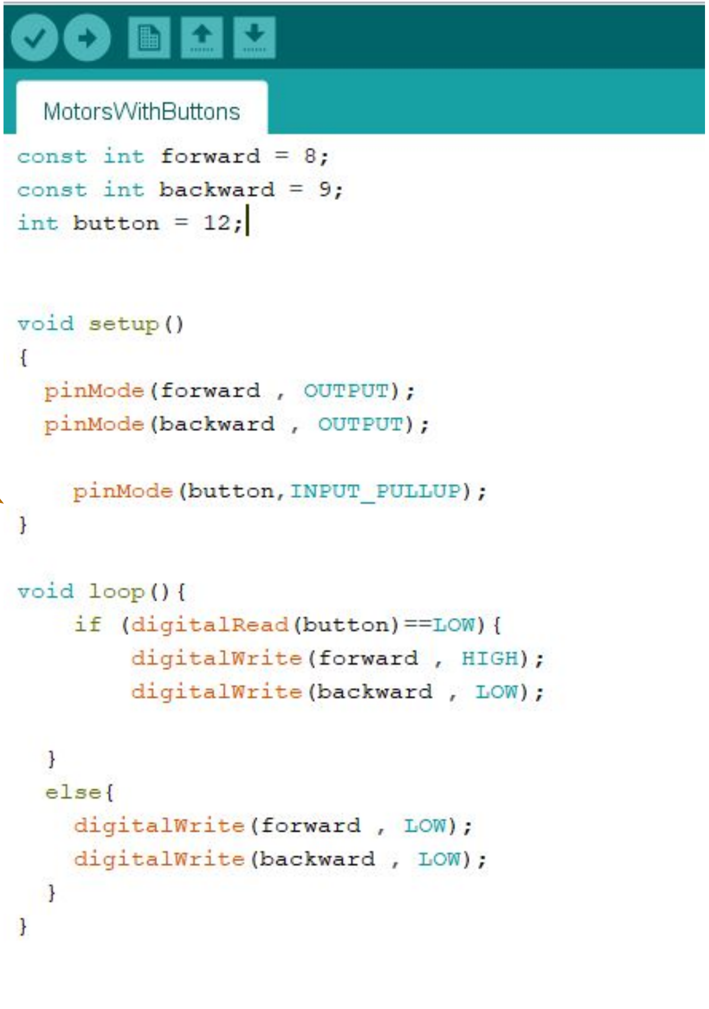


# Motor with Button Code

Initialize another pin for the button and set the pinMode

**If** the button is pressed, the motor should turn. **Else**, the motor should be off.

View this code on GitHub!



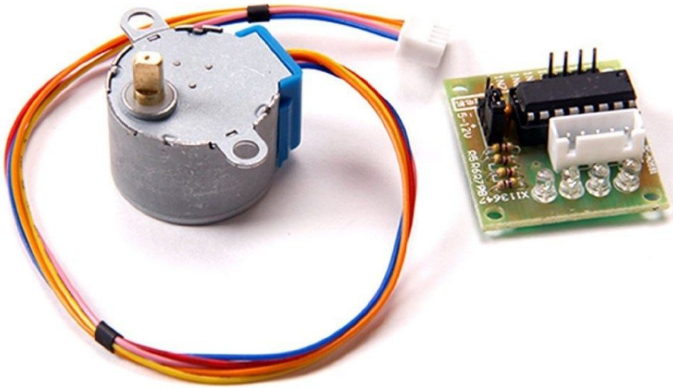
```
const int forward = 8;
const int backward = 9;
int button = 12;


void setup()
{
  pinMode(forward , OUTPUT);
  pinMode(backward , OUTPUT);

    pinMode(button,INPUT_PULLUP);
}

void loop(){
    if (digitalRead(button)==LOW){
        digitalWrite(forward , HIGH);
        digitalWrite(backward , LOW);

  }
  else{
    digitalWrite(forward , LOW);
    digitalWrite(backward , LOW);
  }
}
```

# Project Part 2: Stepper Motors

# Introduction to Stepper Motors

Like normal motors, stepper motors can provide continuous rotation. Unlike normal motors, however, a stepper motor will hold its current position (or “step”) until given a command to move clockwise or counterclockwise a certain number of steps.

Stepper motors are often used in CNC machines, laser cutters, and 3D printers where precise motion is required.

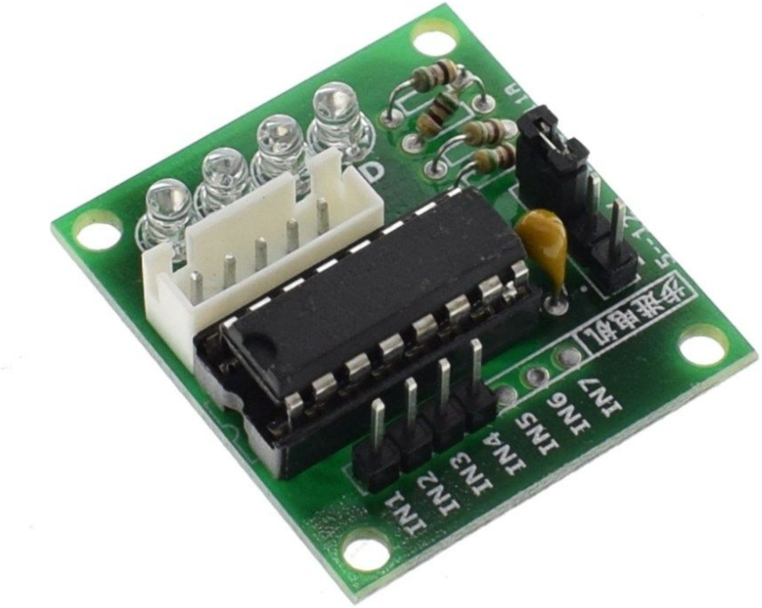


# Stepper Motor Controller

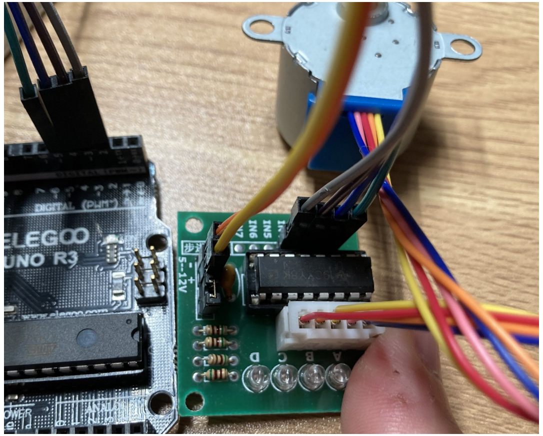
This controller is designed to control the motion of a stepper motor. We will be using it to control our stepper motor.

Connect IN1 to Pin 5, IN2 to Pin 4, IN3 to Pin 3, and IN4 to Pin 2. Connect the + to 5V, the - to Ground.

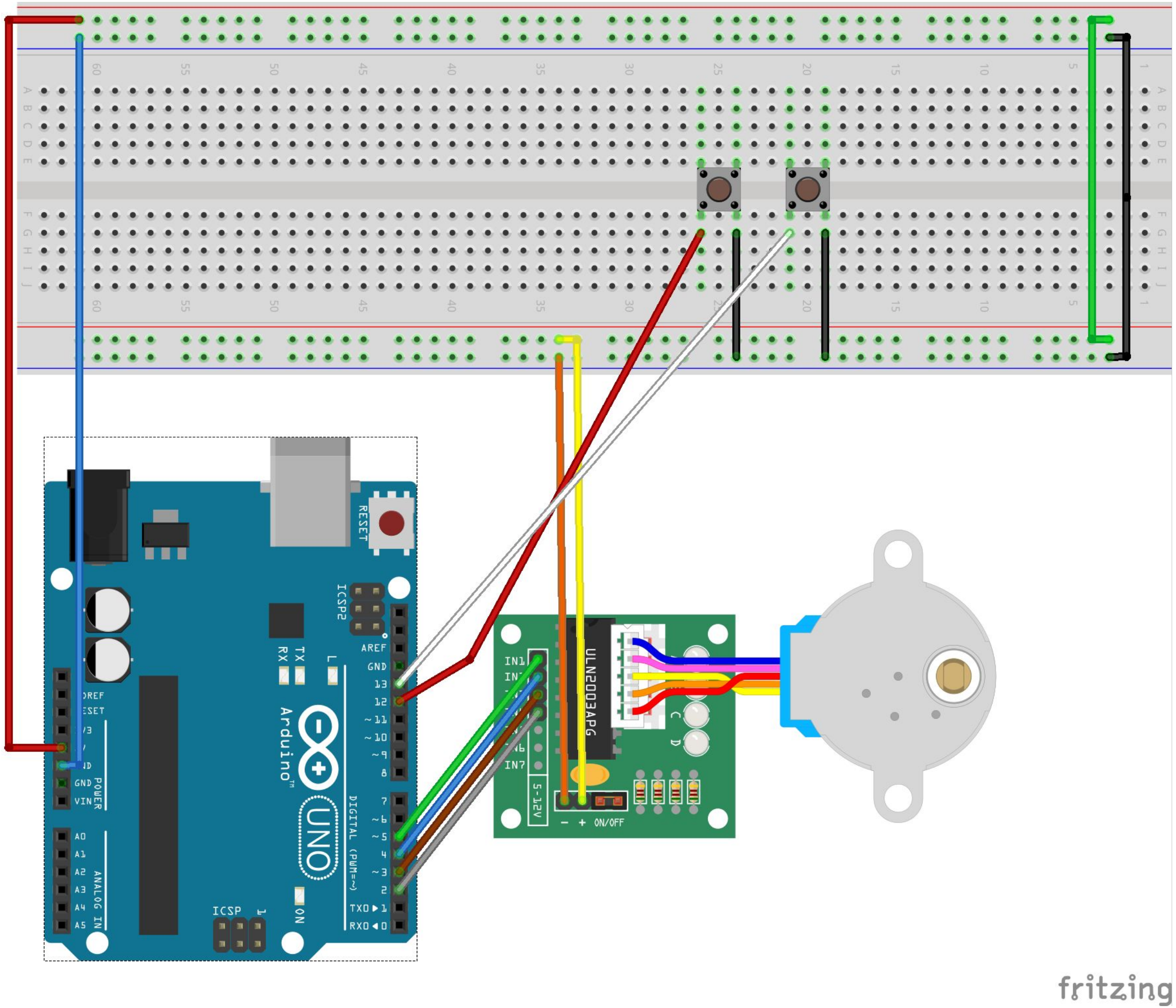
Plug the stepper motor into the white socket



# Schematic



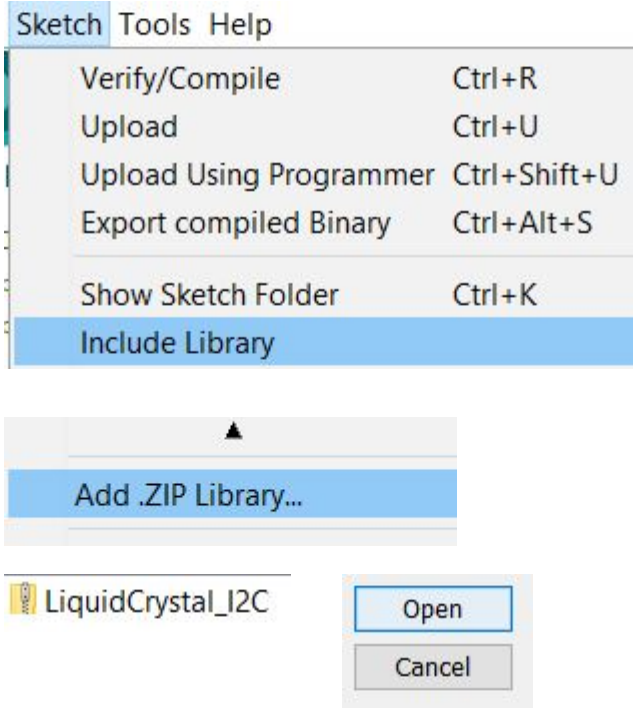
ELEGOO
UNO R3
DIGITAL (PWM~)
IN5
IN6
IN7
5-12V
D
C
B
A



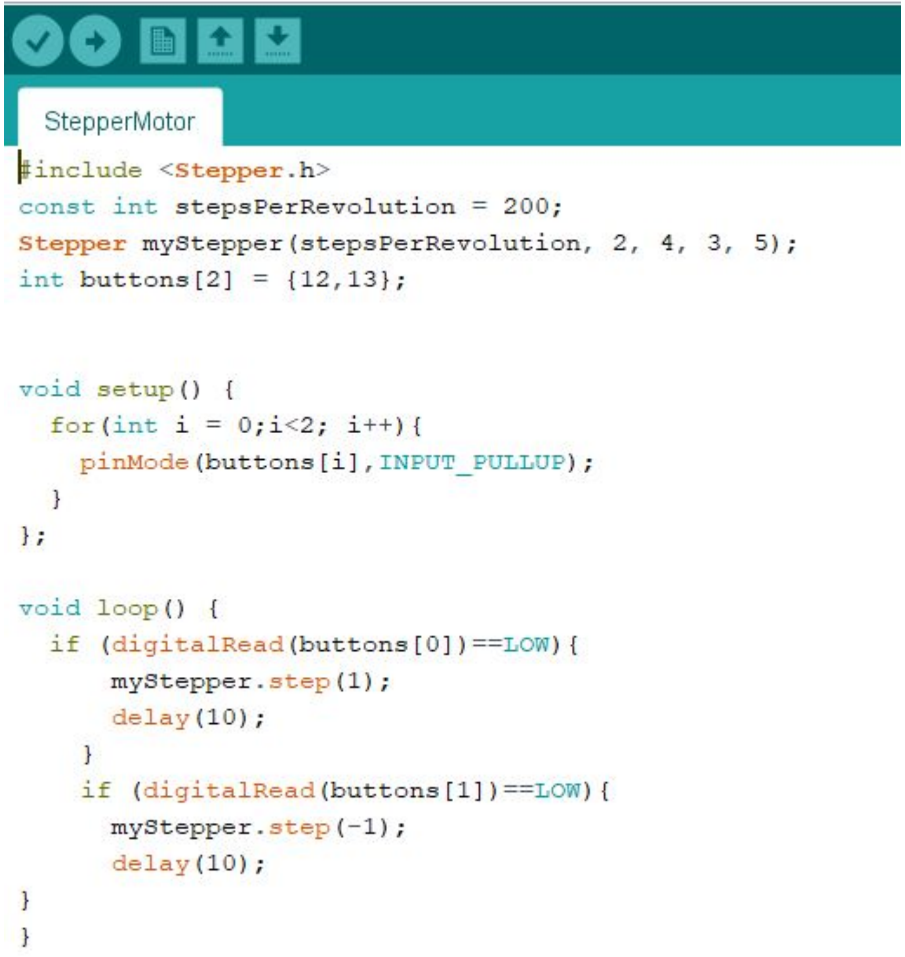
Arduino
UNO
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM=~)
TX
RX
L
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
ULN2003APG
IN1
IN2
IN7
5-12V
- + ON/OFF
C
D
fritzing

# Downloading a Library

- We need to download a library in order to use the Stepper motors
- Download the Stepper Motor library here
  - Save it to your downloads folder
- Go to the Sketch Tab ⇒ Include Library ⇒ Add .Zip Library.
- Navigate to your downloads folder and select and open the Stepper Motor library.



# Stepper Motor Code



```
#include <Stepper.h>
const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 2, 4, 3, 5);
int buttons[2] = {12,13};


void setup() {
  for(int i = 0;i<2; i++){
    pinMode(buttons[i],INPUT_PULLUP);
  }
};

void loop() {
  if (digitalRead(buttons[0])==LOW){
      myStepper.step(1);
      delay(10);
    }
    if (digitalRead(buttons[1])==LOW){
      myStepper.step(-1);
      delay(10);
}
}
```

# Thank you!

CCA Girls Who Code
ccagirlswhocode@gmail.com
www.ccagirlswhocode.weebly.com

# Production Team

Curriculum Director: Stefan Prestrelski

Teaching Assistants: Sarah Luo, Samantha Prestrelski