



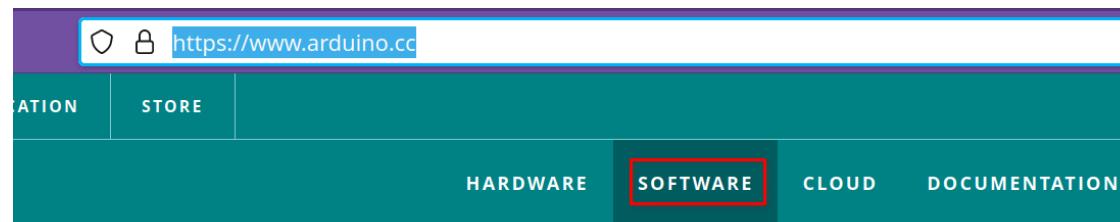
Arduino Workshop

Slides and Code

- <https://github.com/MorganDenes/Arduino>

Install Arduino IDE

- Go to <https://www.arduino.cc/>
- Select SOFTWARE from menu bar



Install Arduino IDE

- Select installation process for your device

The screenshot shows the Arduino website's navigation bar at the top, with "SOFTWARE" highlighted. Below the header, there's a large callout for "Arduino IDE 2.0.1" featuring its logo and a brief description of its features. A link to the "Arduino IDE 2.0 documentation" is provided. To the right, a "DOWNLOAD OPTIONS" section lists links for Windows (MSI installer and ZIP file), Linux (AppImage and ZIP file), and macOS (ZIP file). The "Windows MSI installer" and "macOS ZIP file" links are highlighted with red boxes.

HARDWARE SOFTWARE CLOUD DOCUMENTATION ▾ COMMUNITY ▾ BLOG ABOUT

 Arduino IDE 2.0.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits

Install IDE

- Follow the installer...

Open the IDE



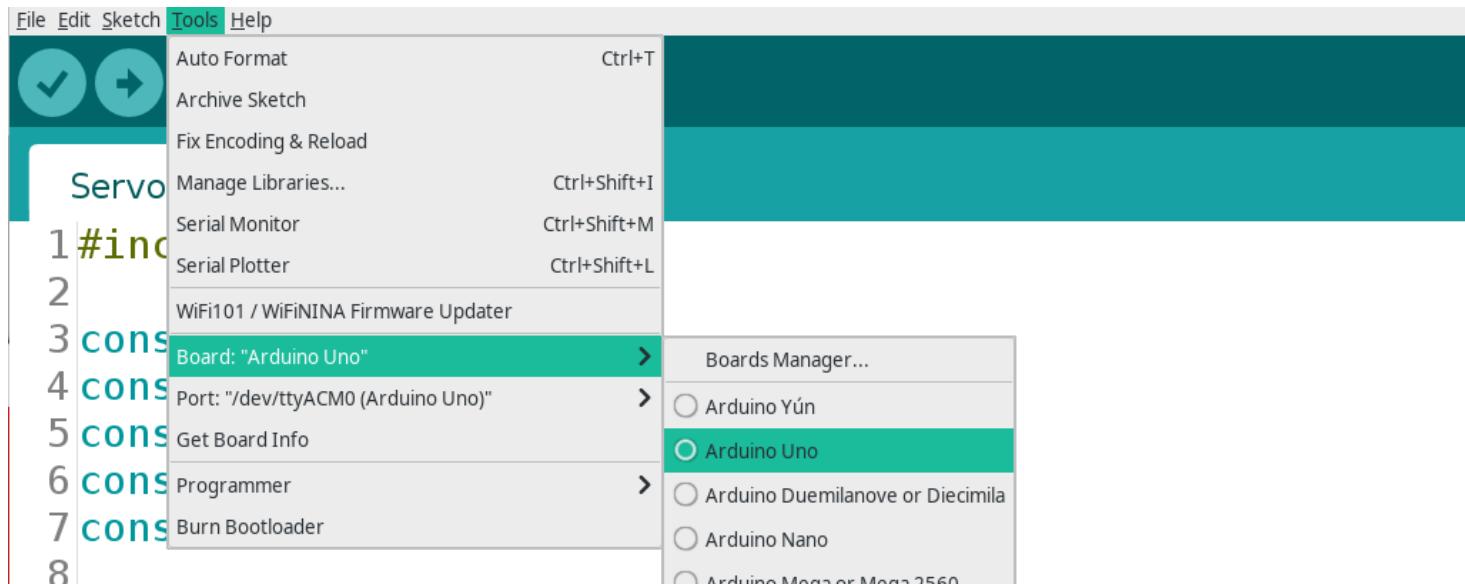
Connect the Arduino

- You know.



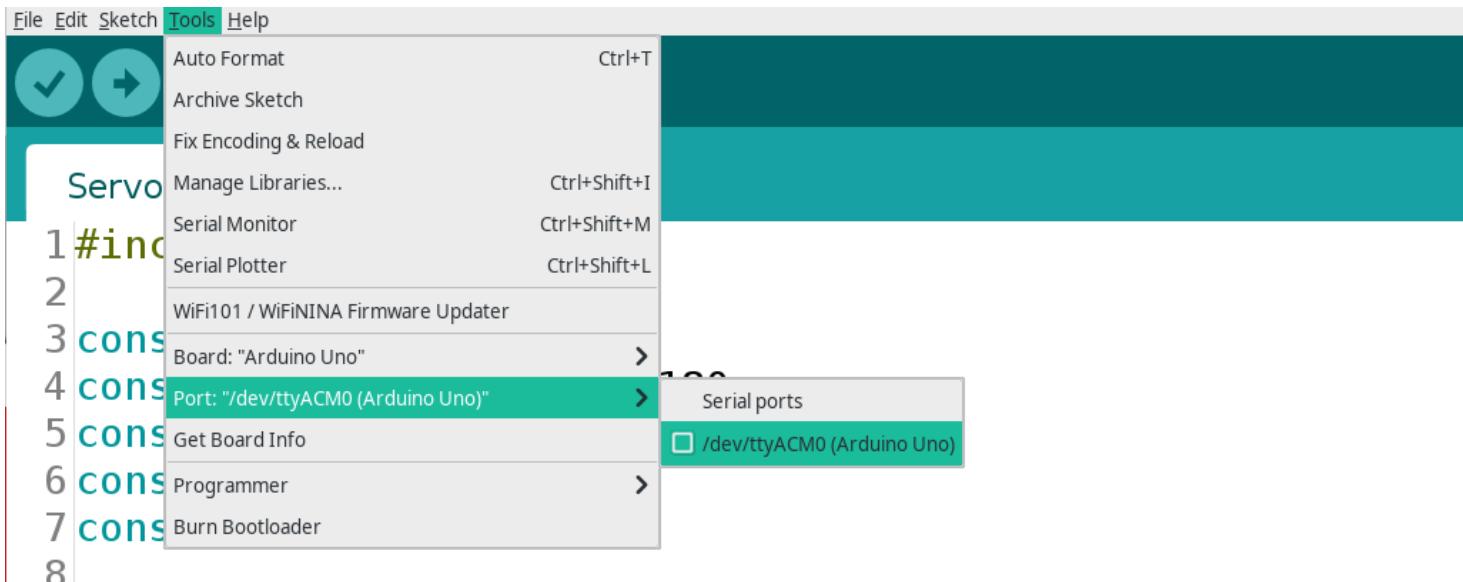
Select Correct Arduino Type

- Tools > Board > Arduino Uno



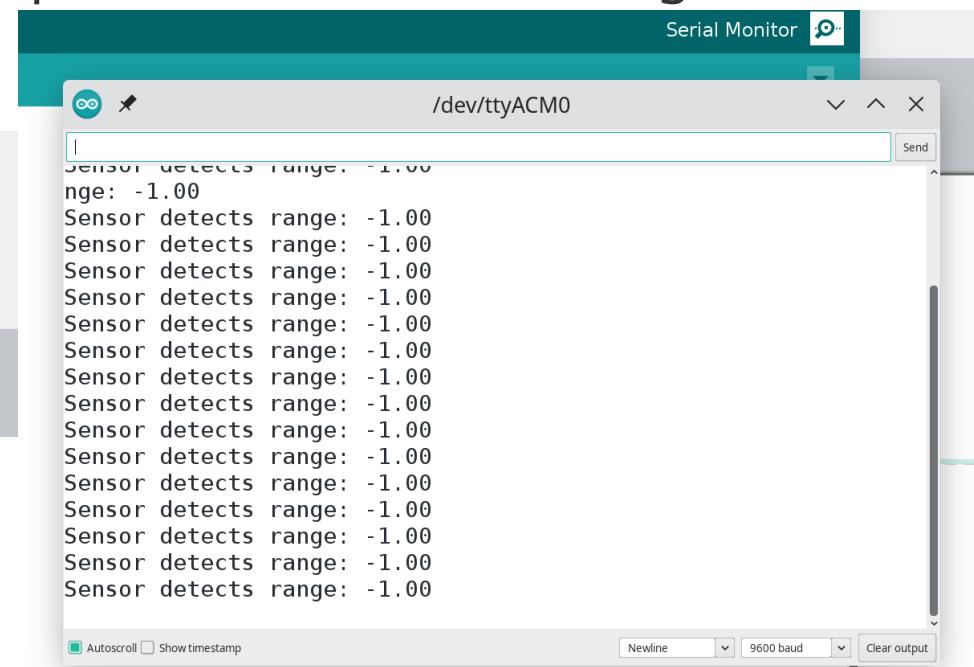
Select Correct Port

- Tools > Port > [Whatever Port]



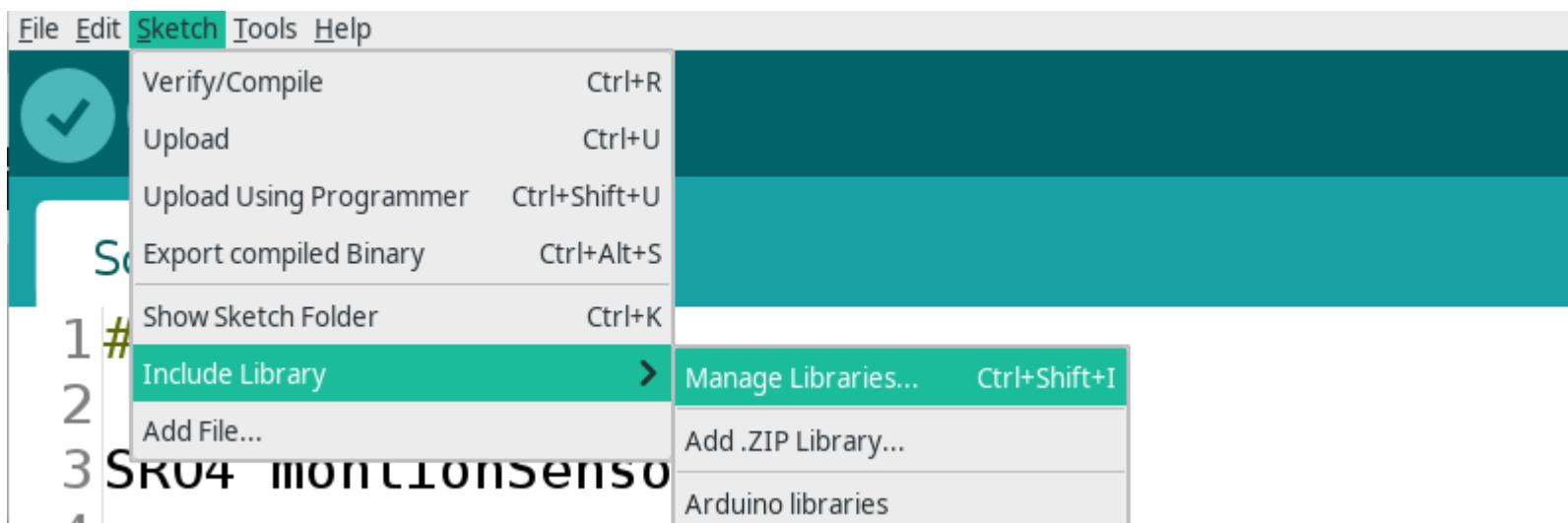
Verify Connection

- Top right of the IDE click the 'magnifying glass'
- The Serial Monitor will open if the connection is good



Install a Library

- Sketch > Include Libraries > Manage Libraries...



Install a Library

- On the top bar type in the name of the library
- Click install on the desired library

The screenshot shows the Arduino Library Manager interface. At the top, there are two dropdown menus: 'Type' set to 'All' and 'Topic' set to 'All'. A search bar contains the text 'SR04'. Below the search bar, a list of libraries is displayed. The first item in the list is 'utilize one trigger with multiple echoes simultaneously.' by 'Martin Sosic'. The second item is 'HCSR04' by 'Martin Sosic'. Both items have a 'More info' link below them. In the bottom right corner of the main list area, there is a dropdown menu set to 'Version 2....' and a green 'Install' button.

Type All Topic All SR04

utilize one trigger with multiple echoes simultaneously.
[More info](#)

HCSR04
by **Martin Sosic**
Library for HC-SR04 ultrasonic distance sensor. You can measure distance in centimeters.
[More info](#)

Version 2....

Libraries to Get

- **Servo** by *Arduino*
- **Stepper** by *Arduino*
- **LiquidCrystal** by *Arduino*
- **HCSR04** by *Martin Sosic*

Breadboard

- Designed for rapid prototyping
- Plug & Play

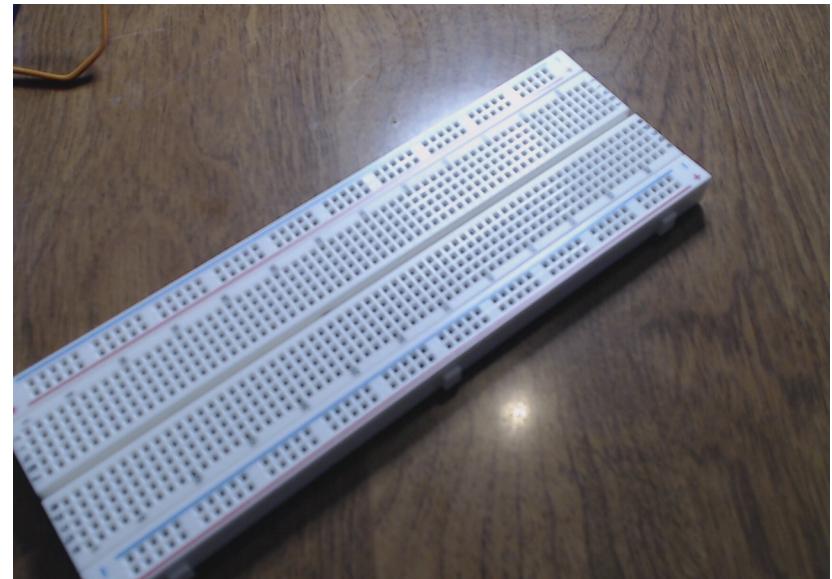
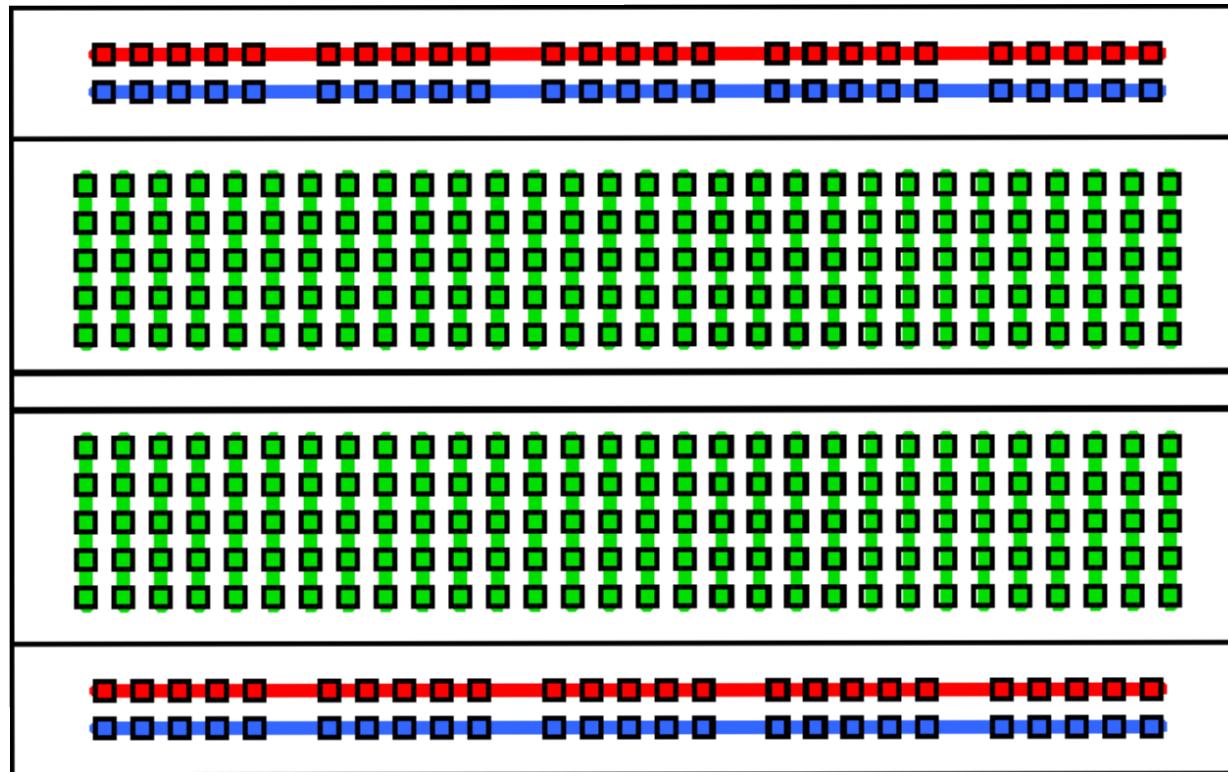
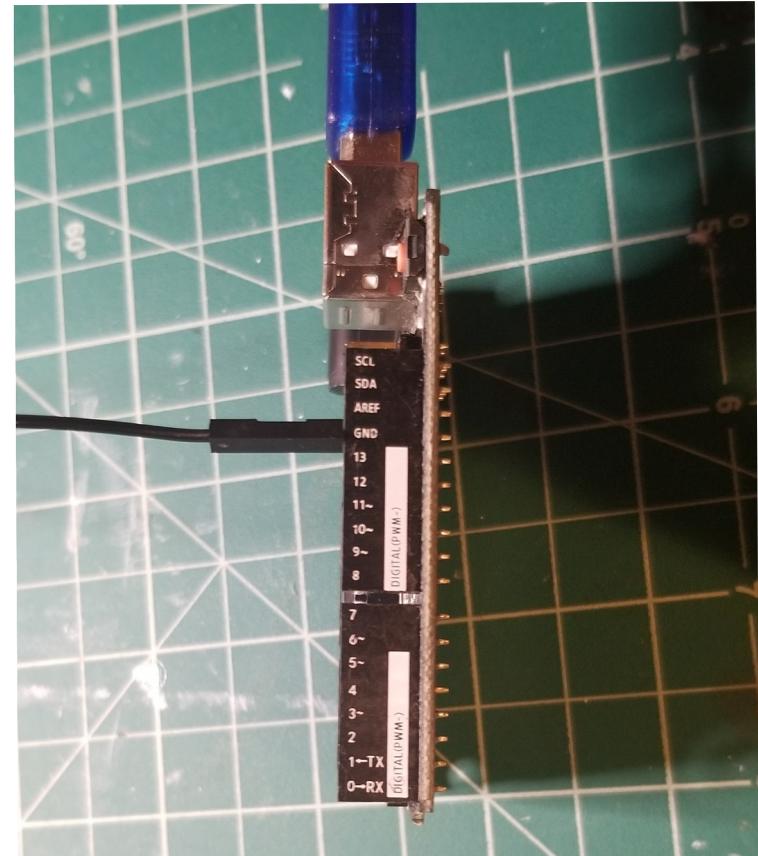


Diagram of Breadboard



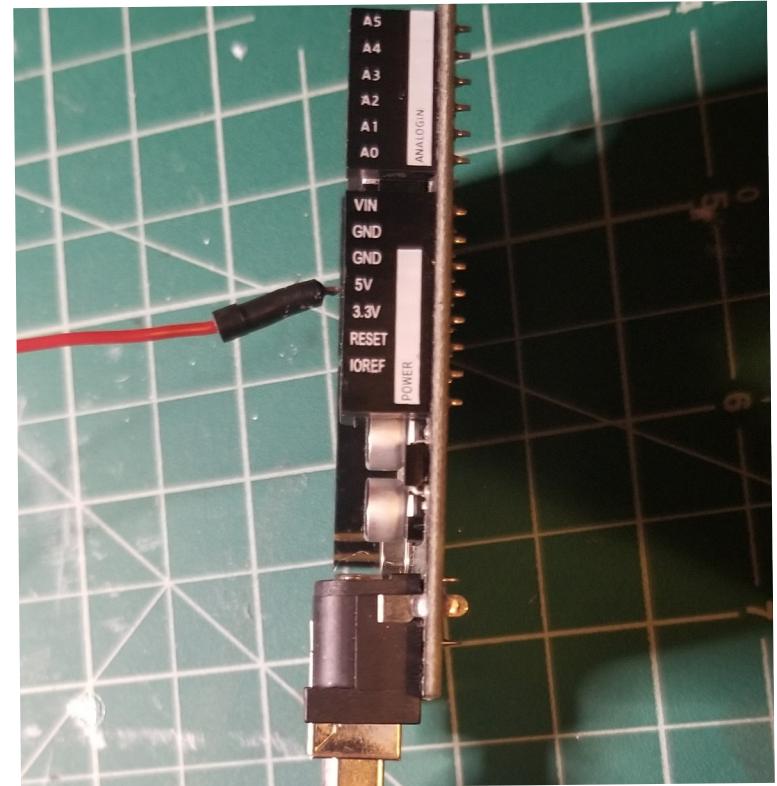
Connect to the Breadboard

- Black wire in GND



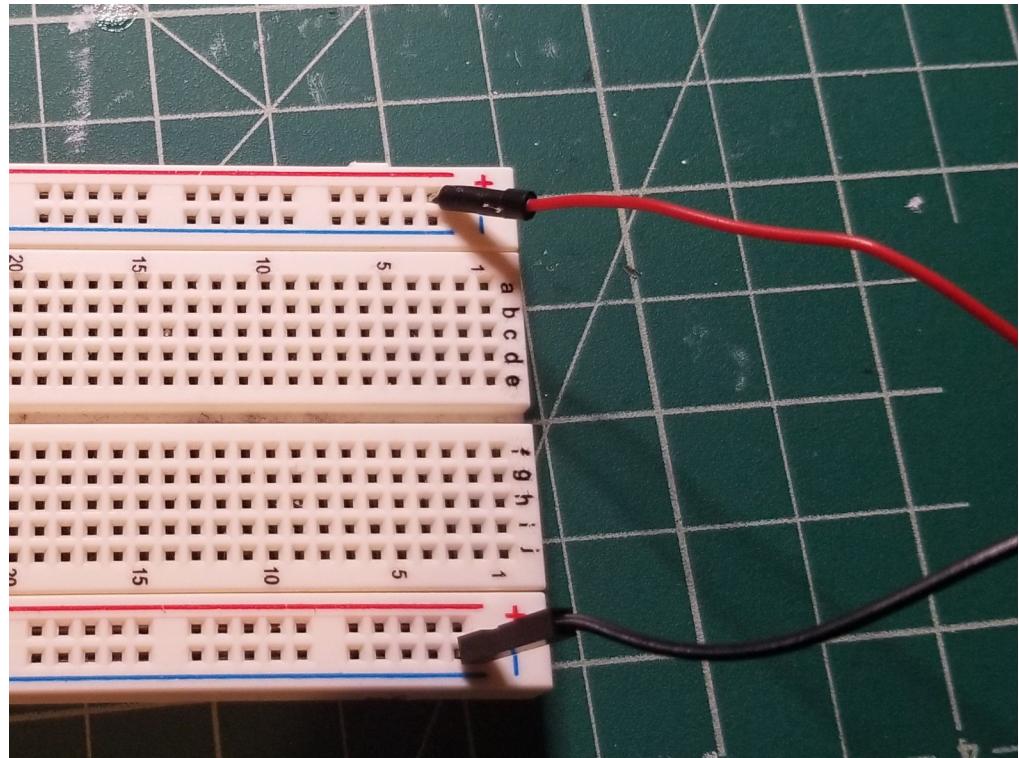
Connect to the Breadboard

- Red wire in 5V

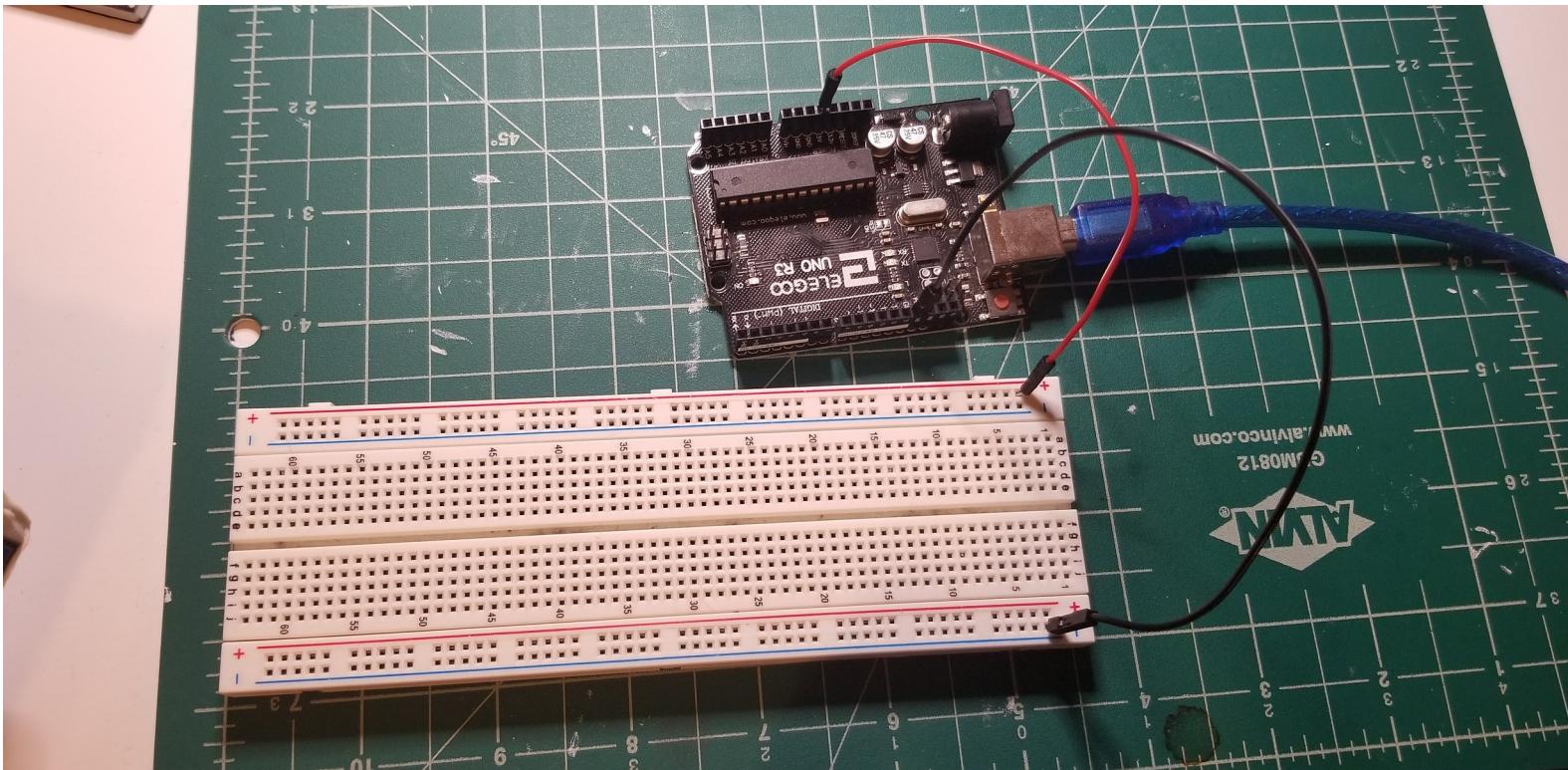


Connect to the Breadboard

- Red wire to red rail
- Black wire to blue rail



Connect to the Breadboard



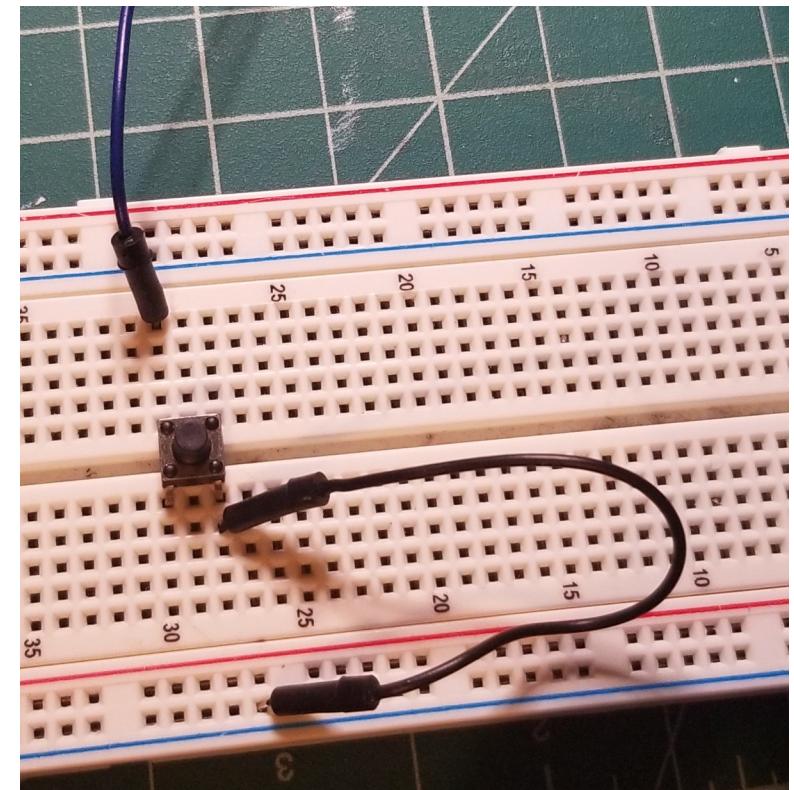
Button

- Opposing legs are already connected
- Clicking in the button connects all legs together
- Lets us control code flow



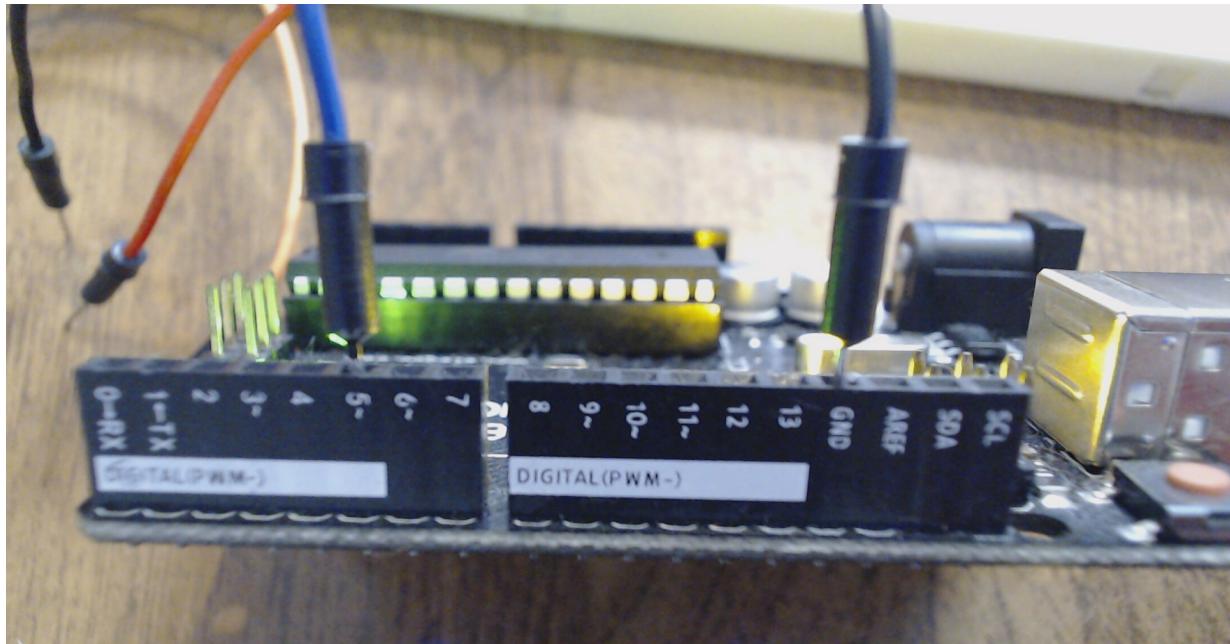
Wire the Button

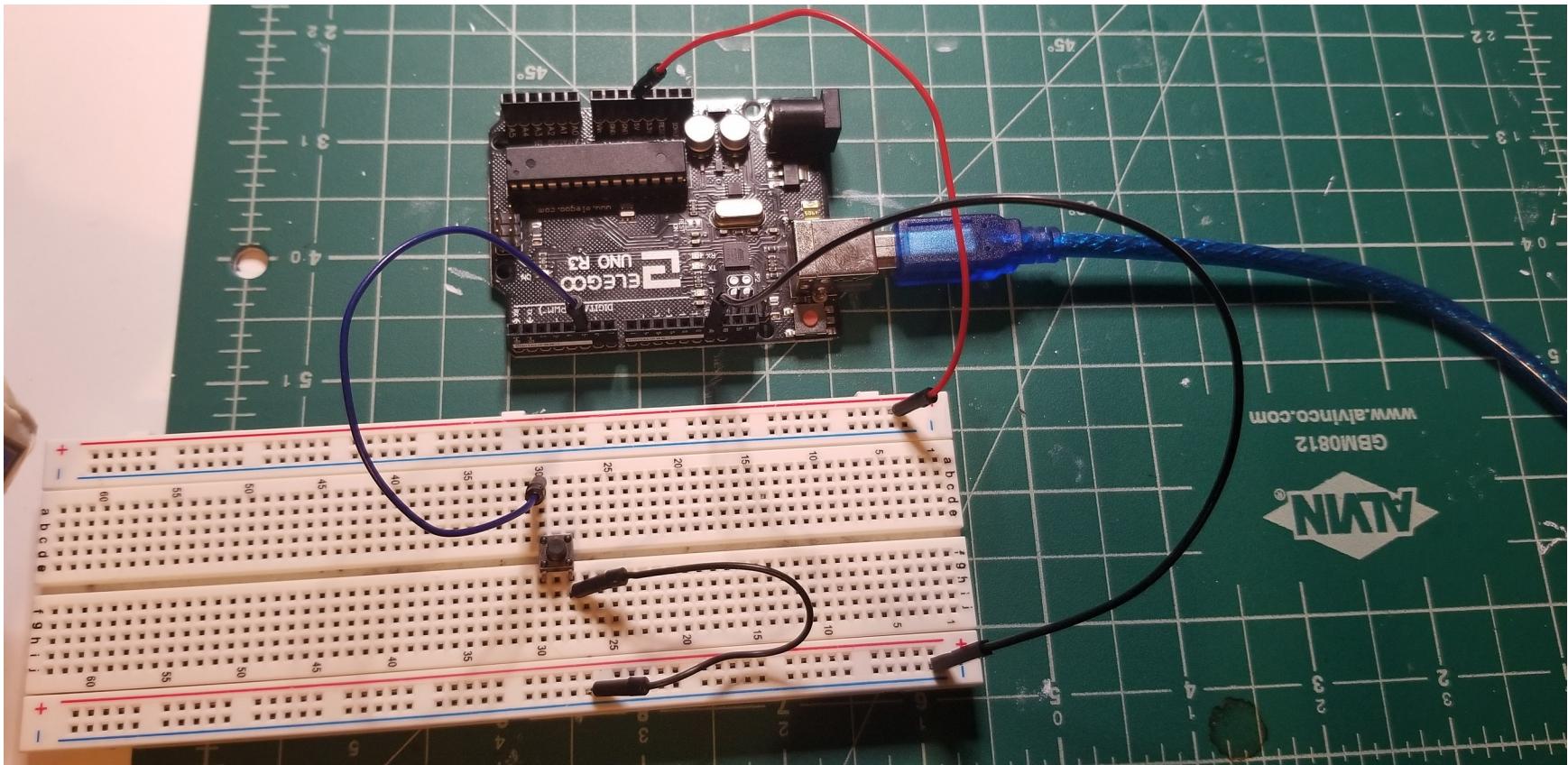
- Put the button over the middle of the breadboard
- Connect one side to black wire
- Connect other side of black to blue rail
- Connect the other side to blue wire



Wire Arduino

- Blue wire to digital 5
- Black wire to GND



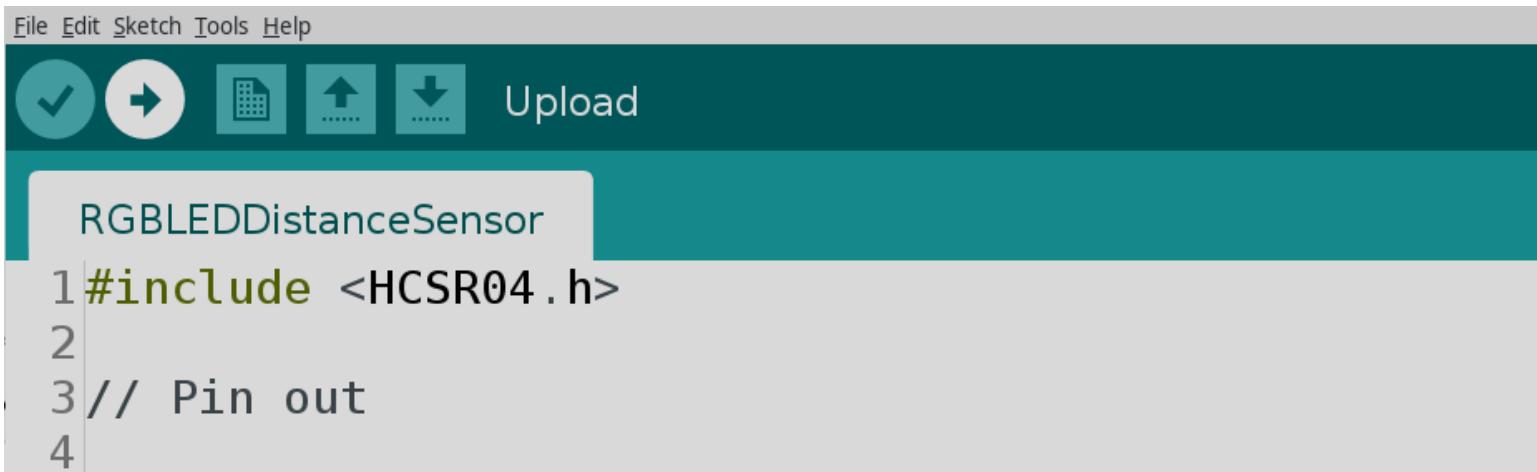


Button Code

- Button
 - Shows that Arduino is able to read the button state

Run a Program

- Click the arrow in the top left of the IDE

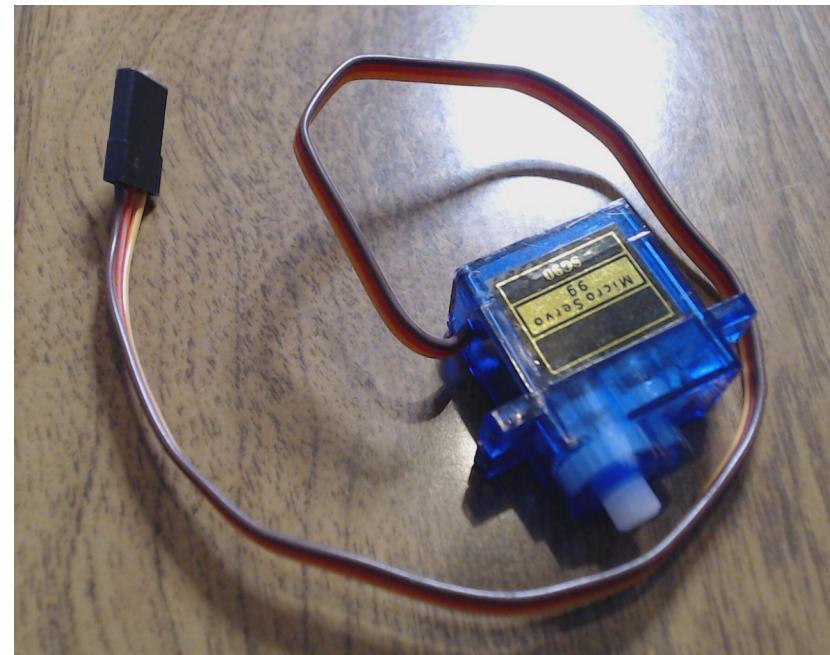


The screenshot shows the Arduino IDE interface. At the top, there's a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu is a toolbar with several icons: a checkmark, a right-pointing arrow, a file folder, an upload symbol, and a download symbol. To the right of these icons is the word 'Upload'. The main area of the IDE displays a sketch titled 'RGBLEDDistanceSensor'. The code in the sketch is as follows:

```
1 #include <HCSR04.h>
2
3 // Pin out
4
```

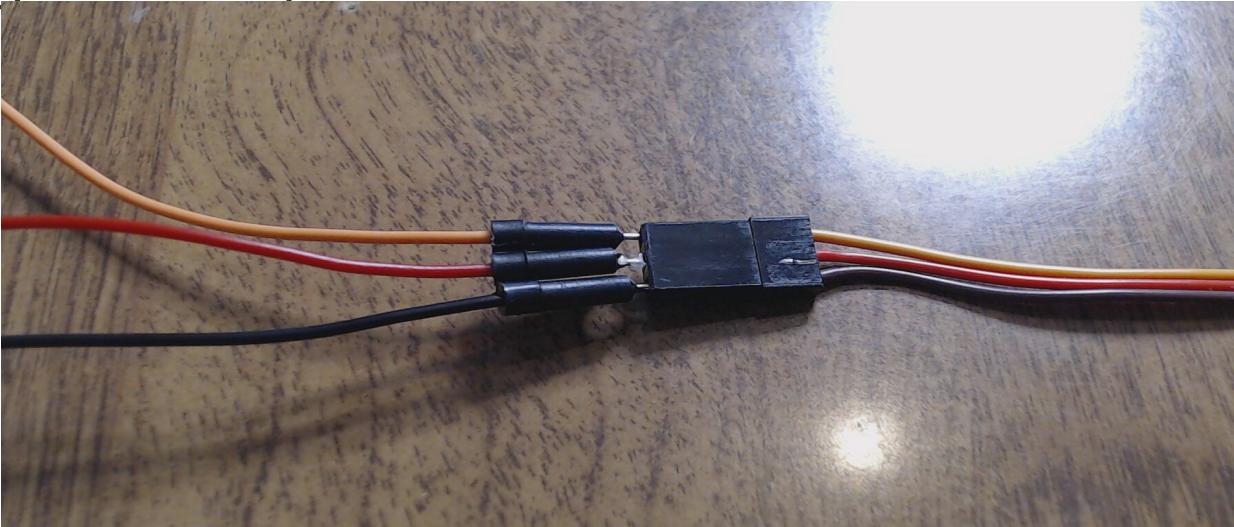
Servo

- Only 180° of rotation
- Control it by setting angle
- Needs to be wired to ~PWM
- Good for hobby work



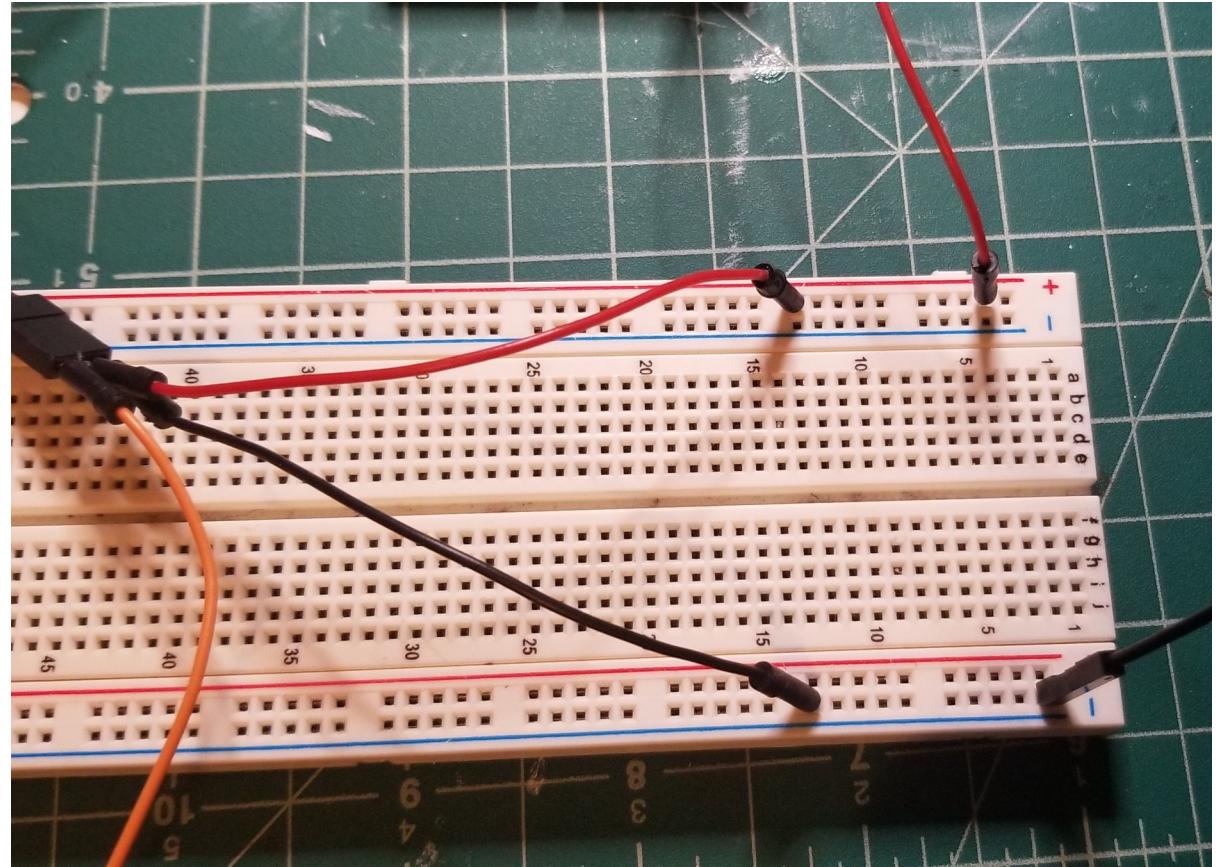
Wire the Servo

- Red to red
- Brown to black
- Orange to orange



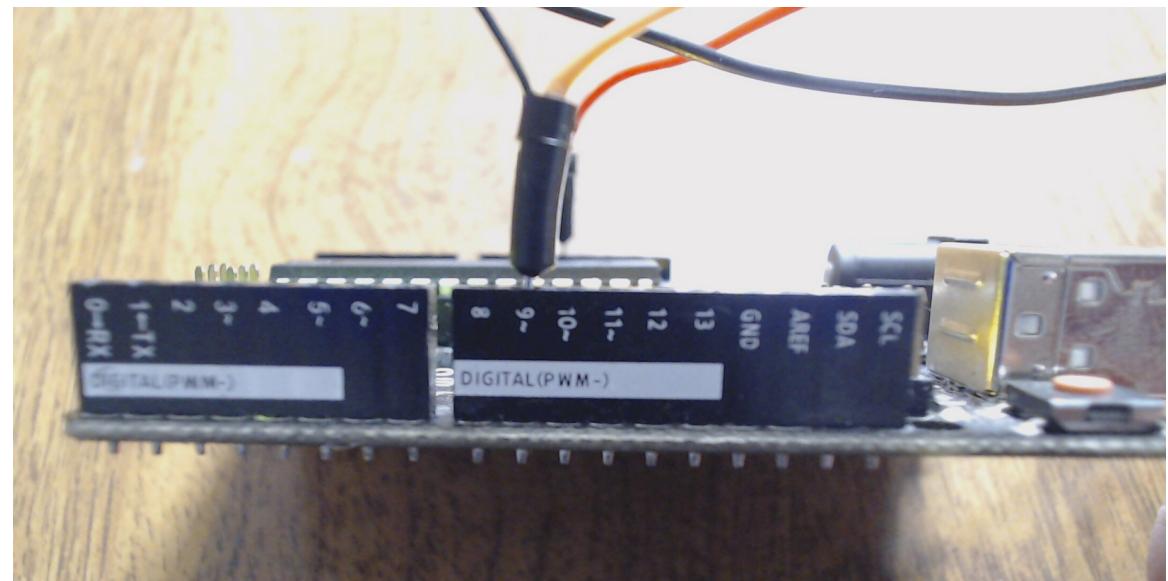
Wire Arduino

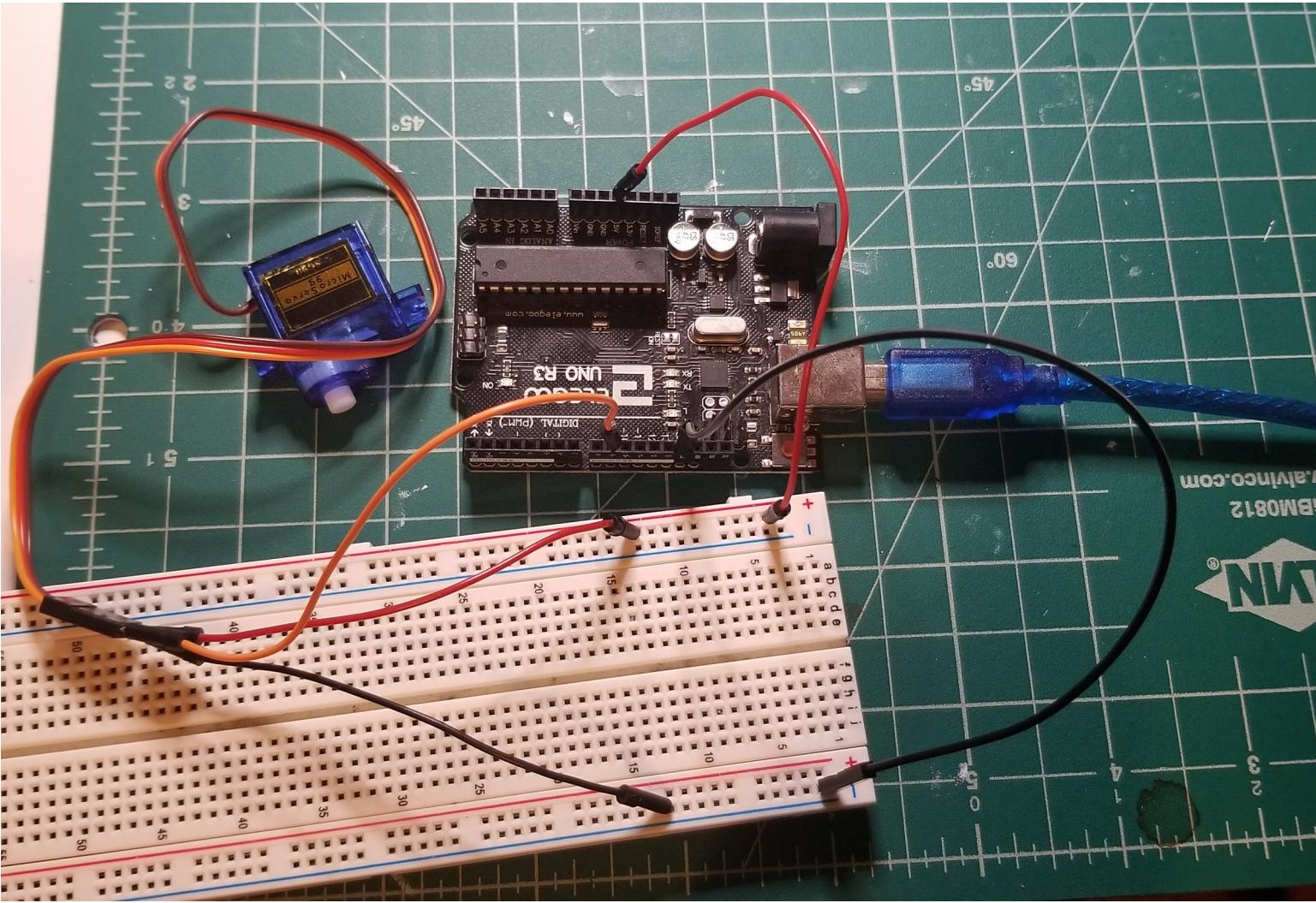
- Red wire to red rail
- Black wire to blue rail



Wire Arduino

- Orange to digital 9



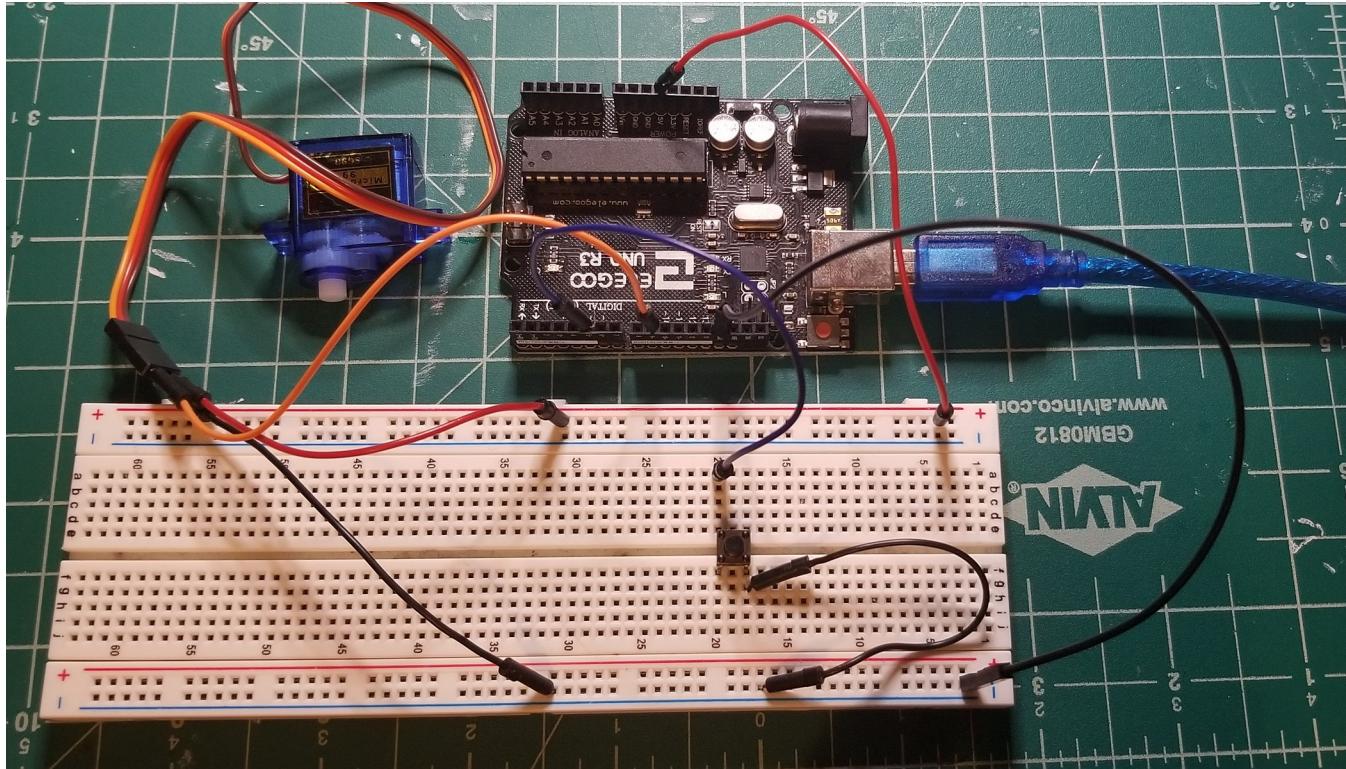


Servo Programs

- ServoSimple
 - A minimum program to demonstrate a servo
- ServoWithVairables
 - Has variables at the top that you can control

Button and Servo

- Use the button to control the Servo

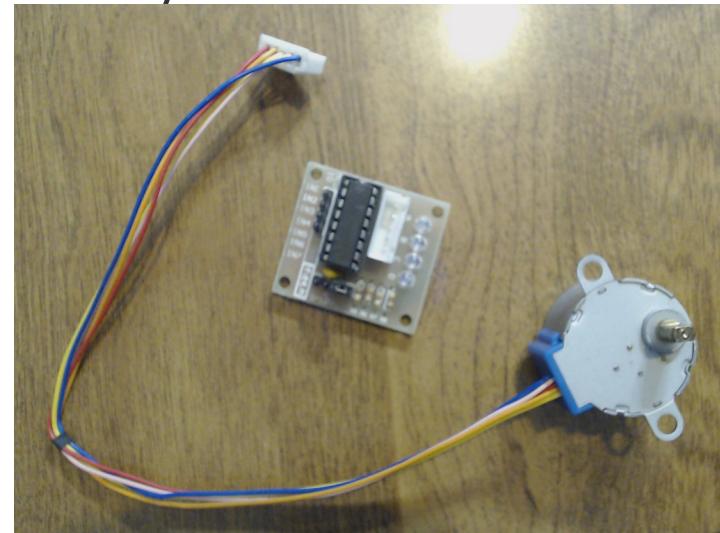


Servo and Button Code

- ServoActiveButton
 - Button must be held to turn servo
- ServoPassiveButton
 - Button is clicked to turn servo
- ServoPassiveButtonWithVariables
 - Same as ServoPassiveButton but with extra variables

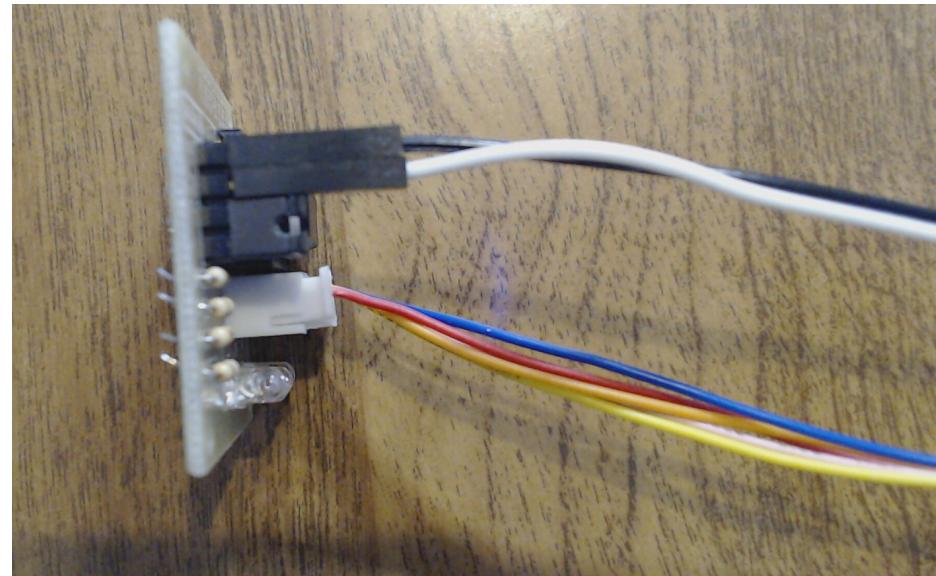
Stepper

- Most complex to wire up
- Control how far it rotates
- 3D printers use them for their accuracy



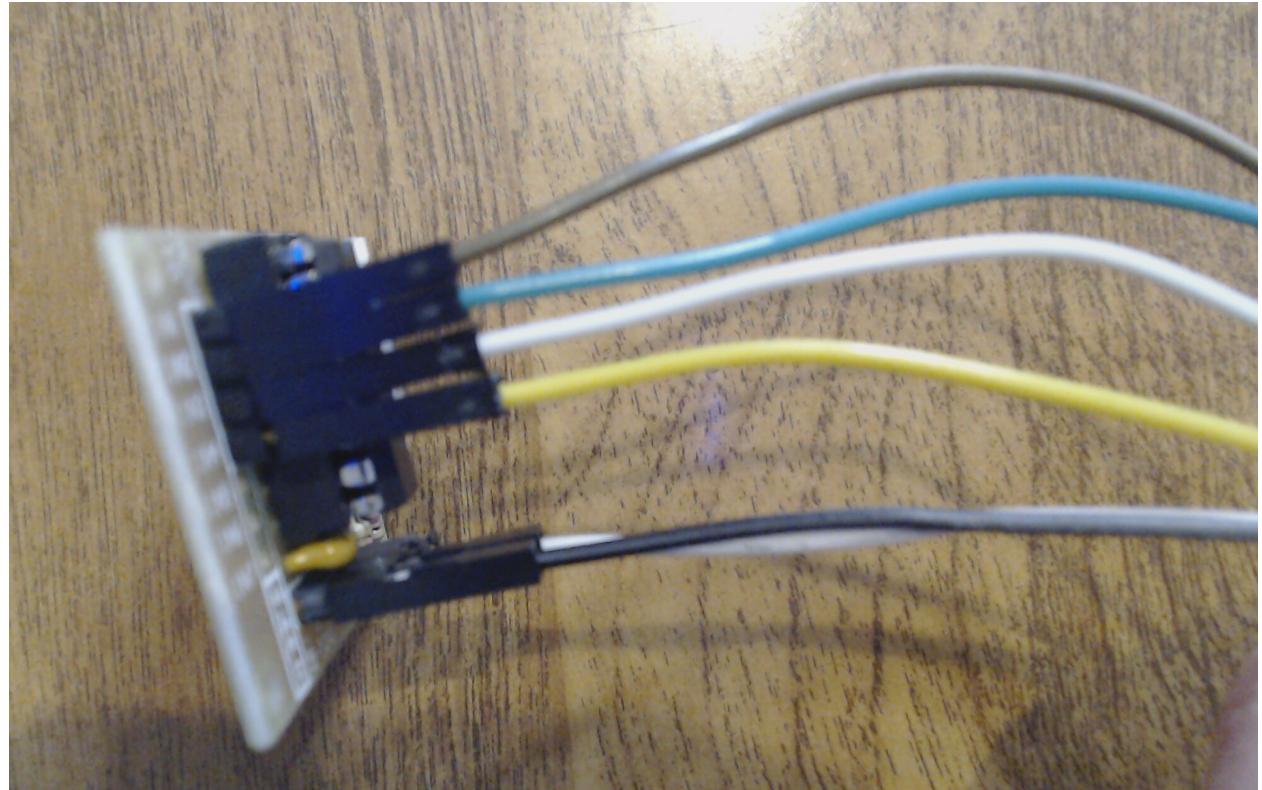
Wire Stepper Motor

- Plug the motor into it's extra board
- Connect 5V '-' to black
- Connect 5V '+' to "red"



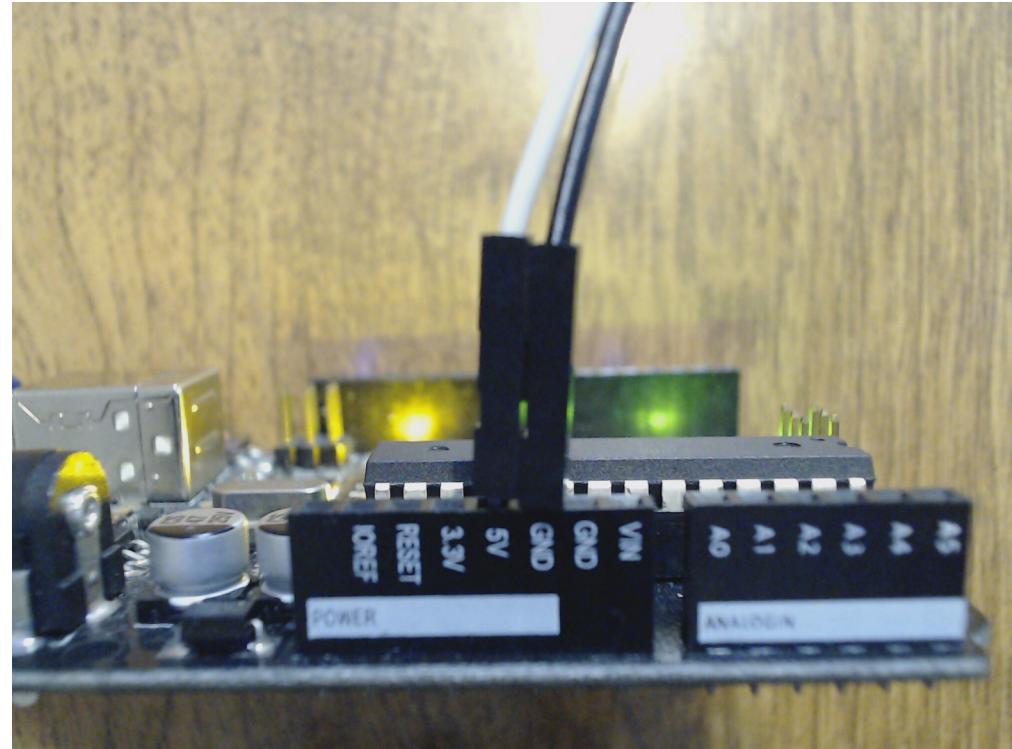
Wire Stepper Motor

- 1N1 to brown
- 1N2 to green
- 1N3 to white
- 1N4 to yellow



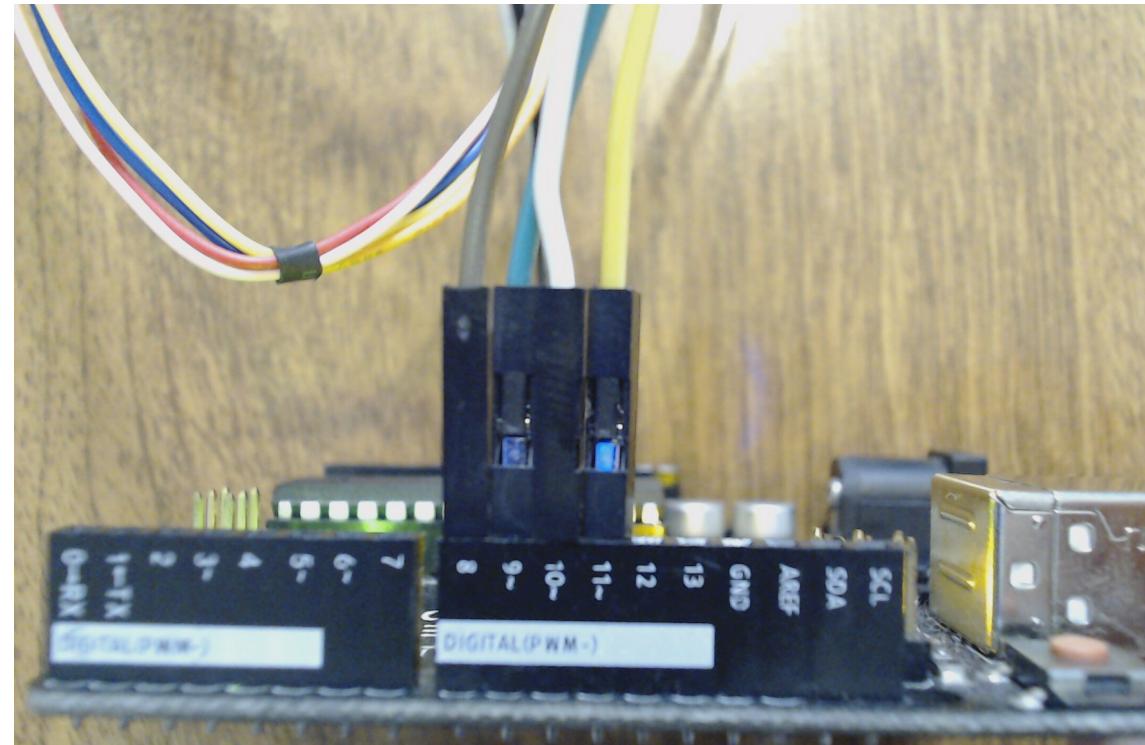
Wire Arduino

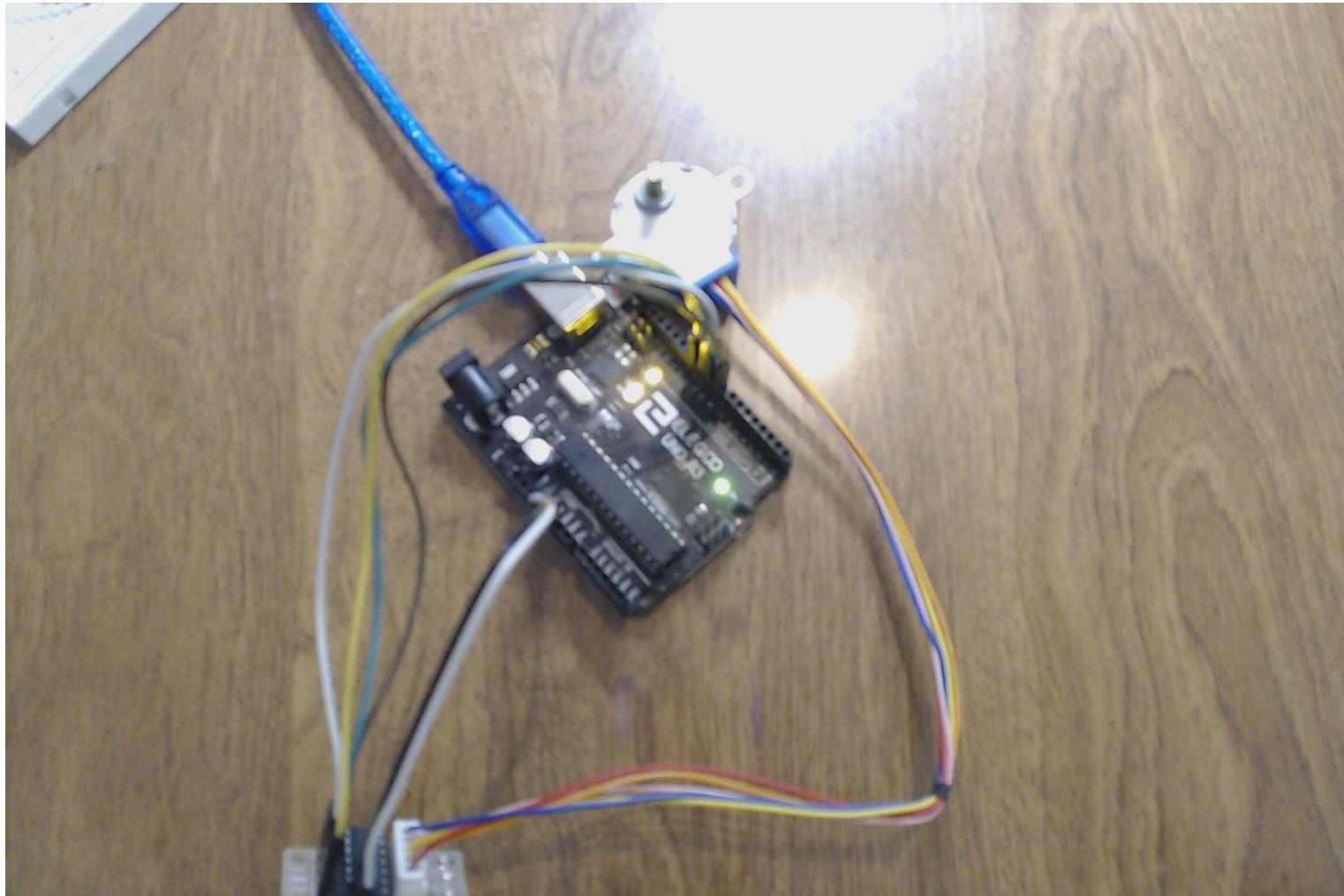
- Black wire to GND
- “Red” wire to 5V



Wire Arduino

- Brown to digital 8
- Green to digital 9
- White to digital 10
- Yellow to digital 11

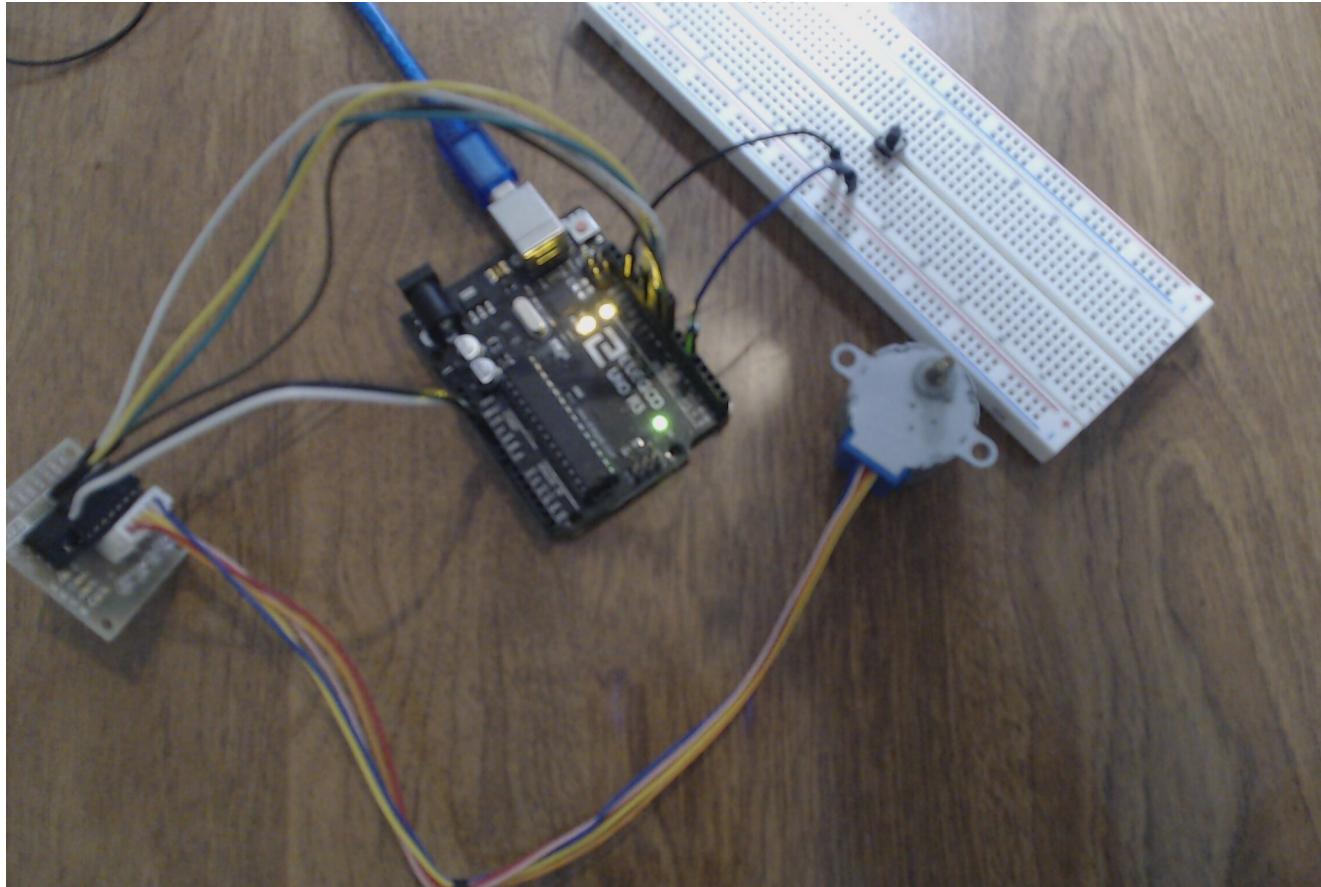




Stepper Code

- StepperSimple
 - Simple example of the Stepper working

Stepper and Button



Stepper and Button Code

- StepperActiveButton
 - Set the rotation direction with the button
- StepperActiveButtonWithVariables
 - Same as StepperActiveButton but with more variables

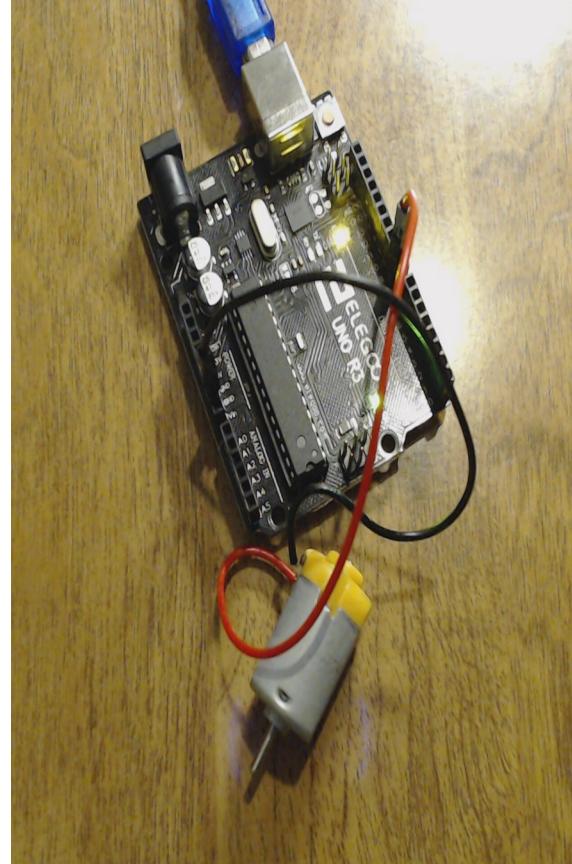
DC

- DC means Direct Current
- Easiest to wire
- Used for Tesla cars, fans, and RC cars



Wire DC to Arduino

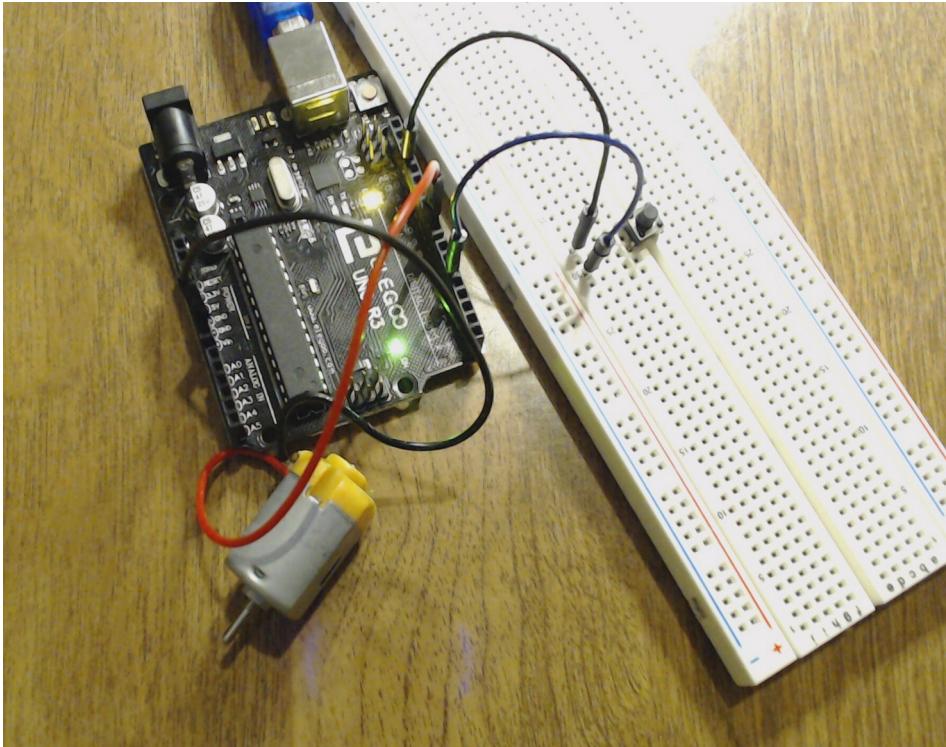
- Red wire to digital 9
- Black wire to GND



DC Code

- DC
 - Simple code for working DC

DC and Button



DC and Button Code

- DCButton
- DCPassiveButton
 - Like ServoPassiveButton but with a DC motor

Footnote on DC

- Can wire DC to spin both ways
 - Not doing it in this workshop

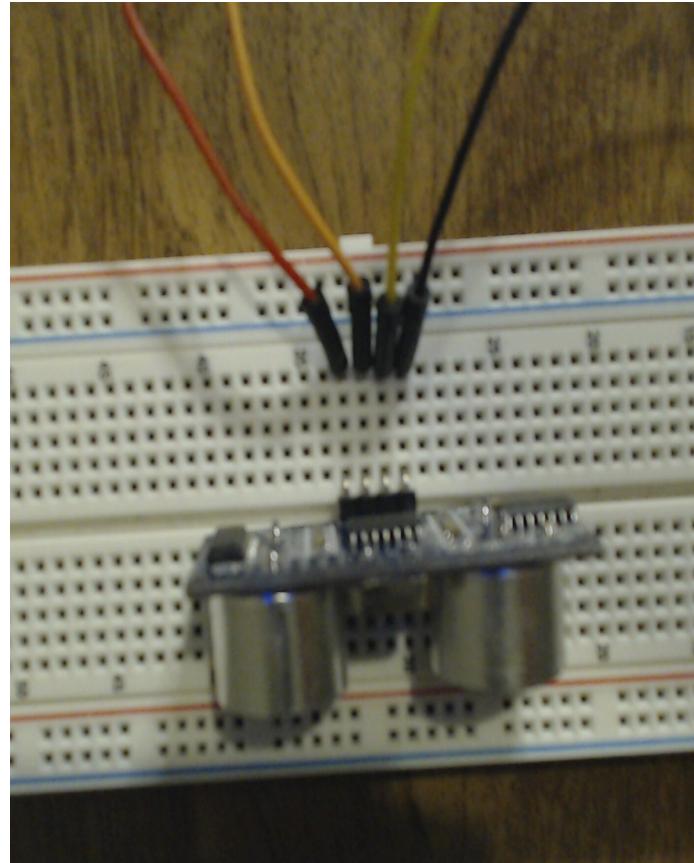
Distance Sensor

- Can estimate the distance from it
- We'll be treating it like a button



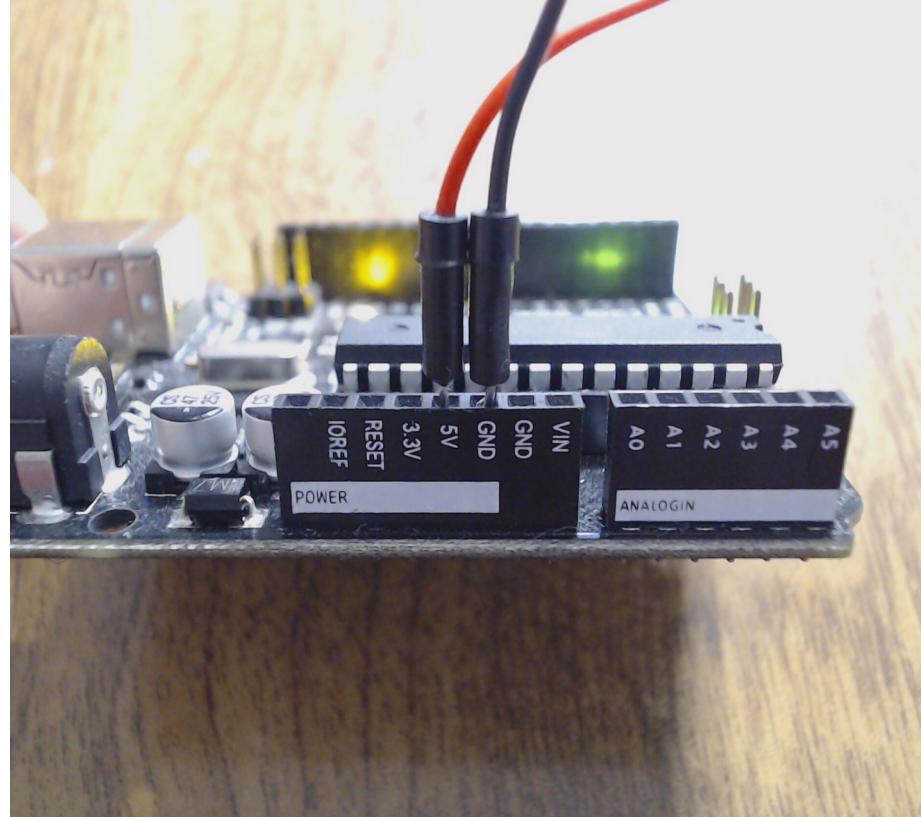
Wire Distance Sensor

- Vcc to Red
- Trig to Orange
- Echo to Yellow
- GND to black



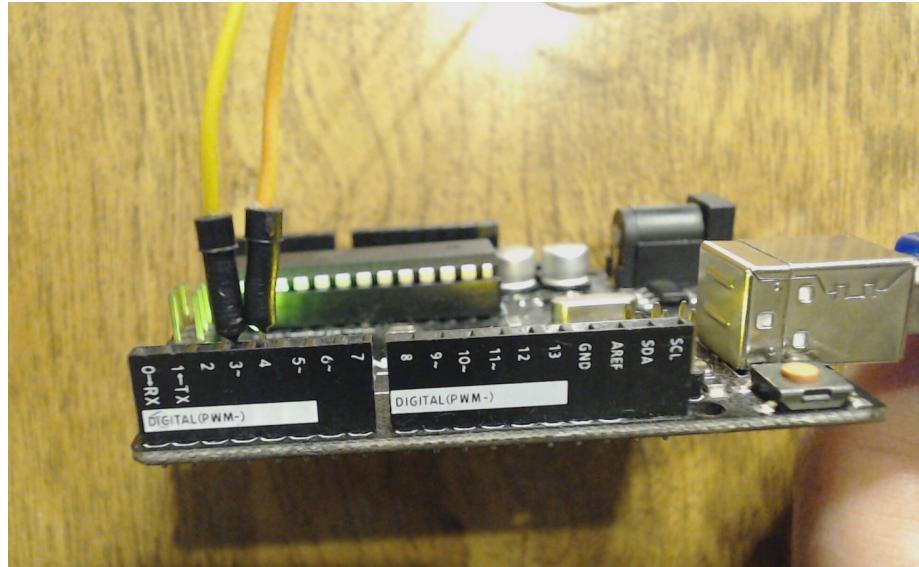
Wire Arduino

- Red to 5V
- Black to GND



Wire Arduino

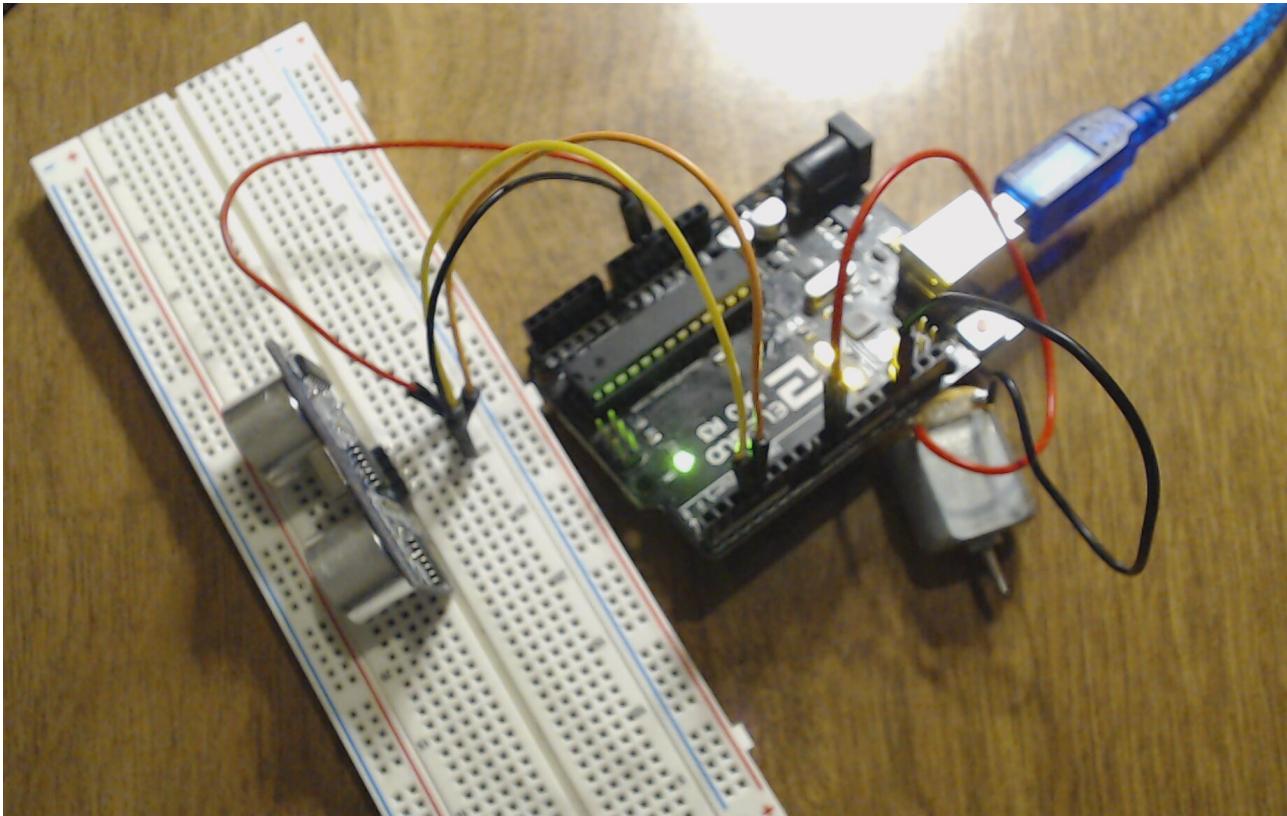
- Orange to digital 4
- Yellow to digital 3



Distance Sensor Code

- DistanceSensor
 - Simple code to use the distance sensor

Distance Sensor and DC



Distance Sensor and DC Code

- DC Distance Sensor
 - Control DC motor with Distance Sensor

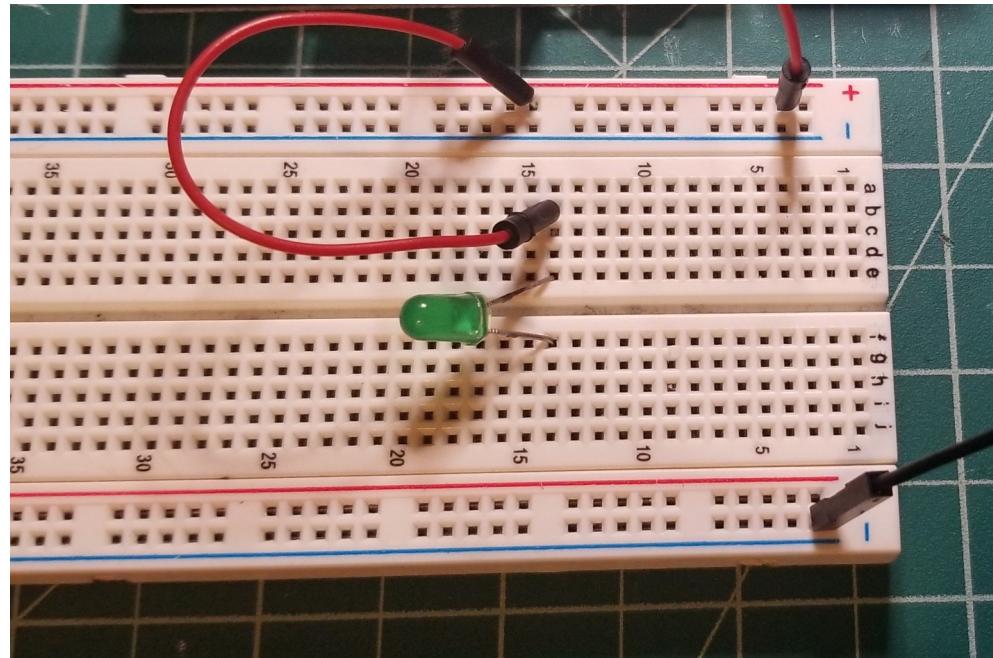
LED

- Light Emitting Diode
- Only let's electricity flow one way
- Long leg is positive



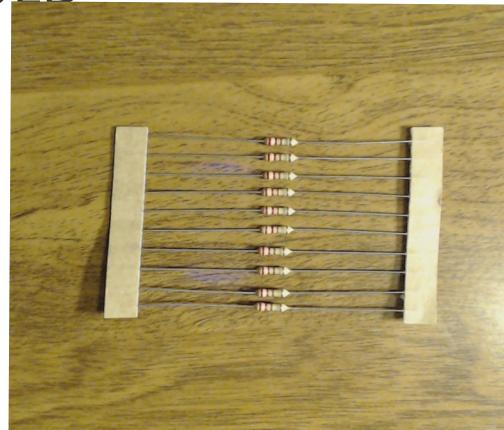
Connect LED

- Run red wire from red power rail to far side of breadboard
- Place LED across center with long leg connected to red wire

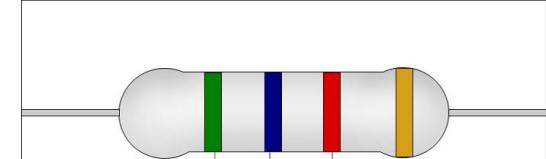


Resistor

- Sets resistance
- Necessary to not break the LEDs
- Pick out ones with the following bands
 - RED RED BROWN GOLD



4 Band Resistor

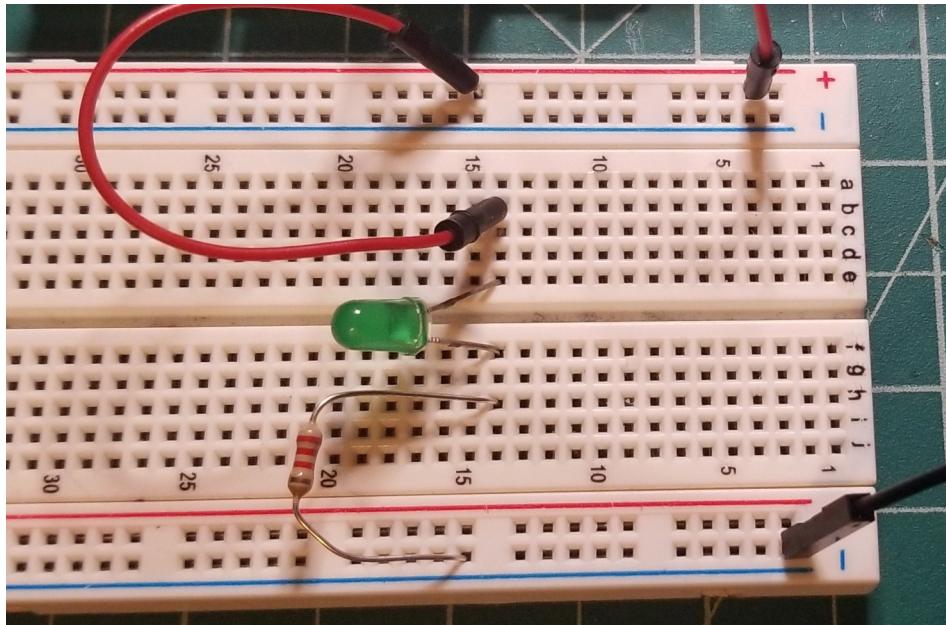


4 Band Resistor			
	First Digit	Second Digit	Multiplier
Black	Nil	0	1
Brown	1	1	10
Red	2	2	100
Orange	3	3	1000
Yellow	4	4	10000
Green	5	5	100000
Blue	6	6	1M
Violet	7	7	10M
Grey	8	8	100M
White	9	9	1G
Gold	Nil	Nil	±10%
Silver	Nil	Nil	±10%

www.CircuitsToday.com

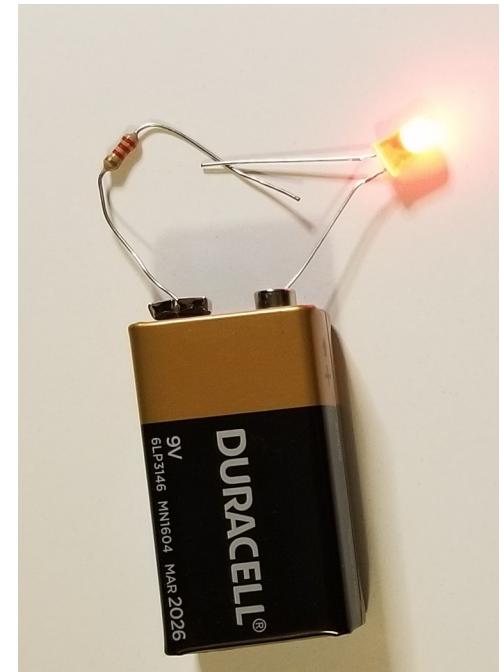
Connect Resistor

- Connect output of LED to the blue rail
- And there was light



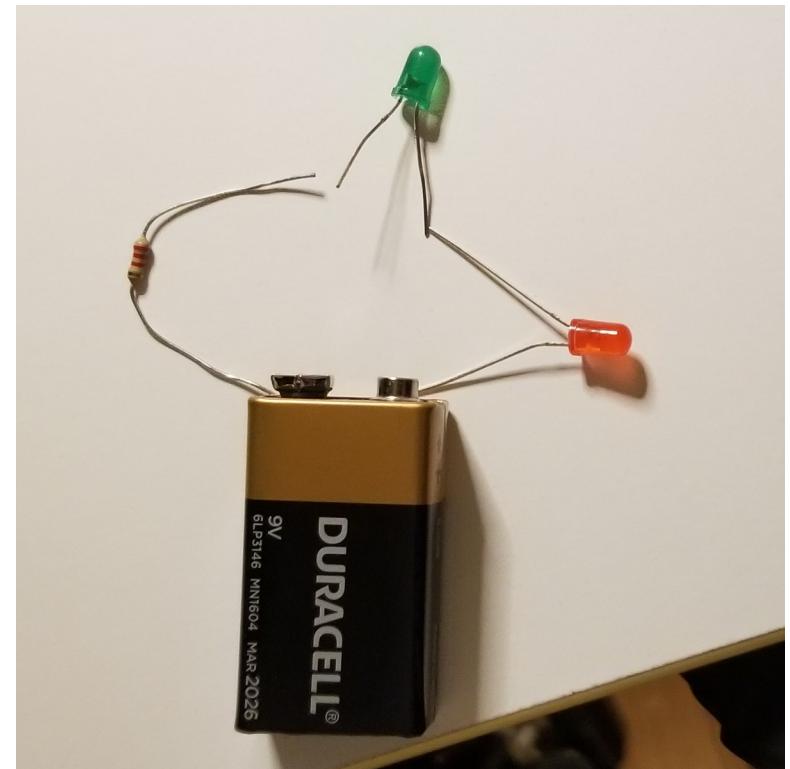
Circuits an Intermission

- Electronics are nothing but fancy circles
- Every component requires POWER and GND minimum



Circuits an Intermission

- Able to connect LEDs in series
- Ensure each LED connects with longest leg connected to the POWER



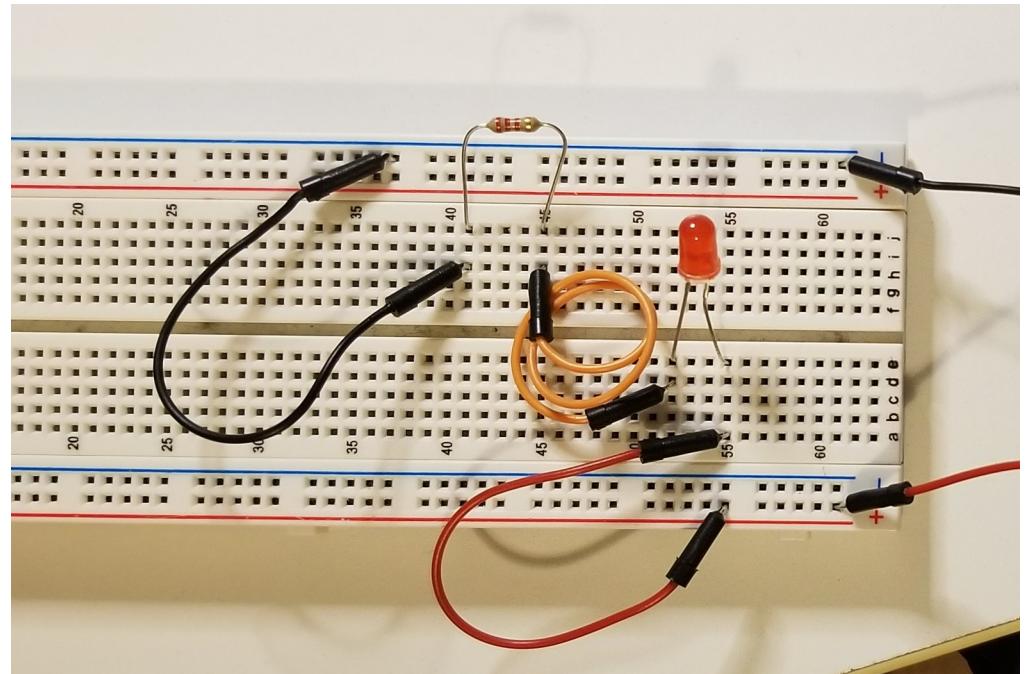
Circuits an Intermission

- Able to connect in parallel
- Again, longest leg should point toward
POWER



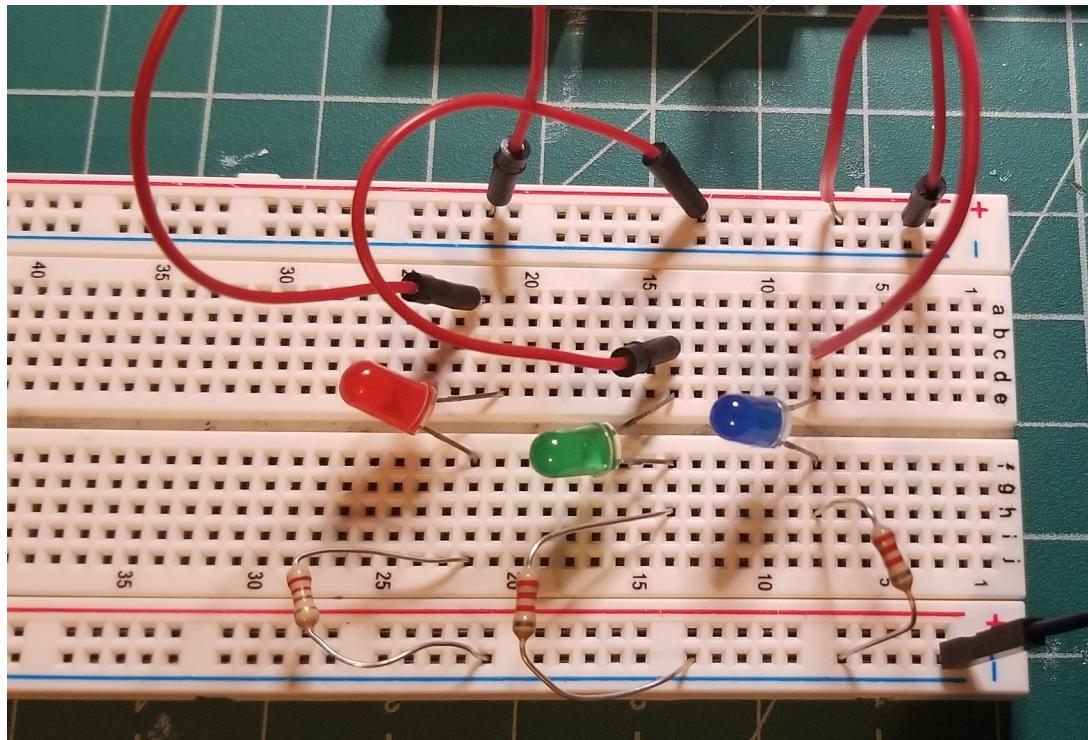
Circuits an Intermission

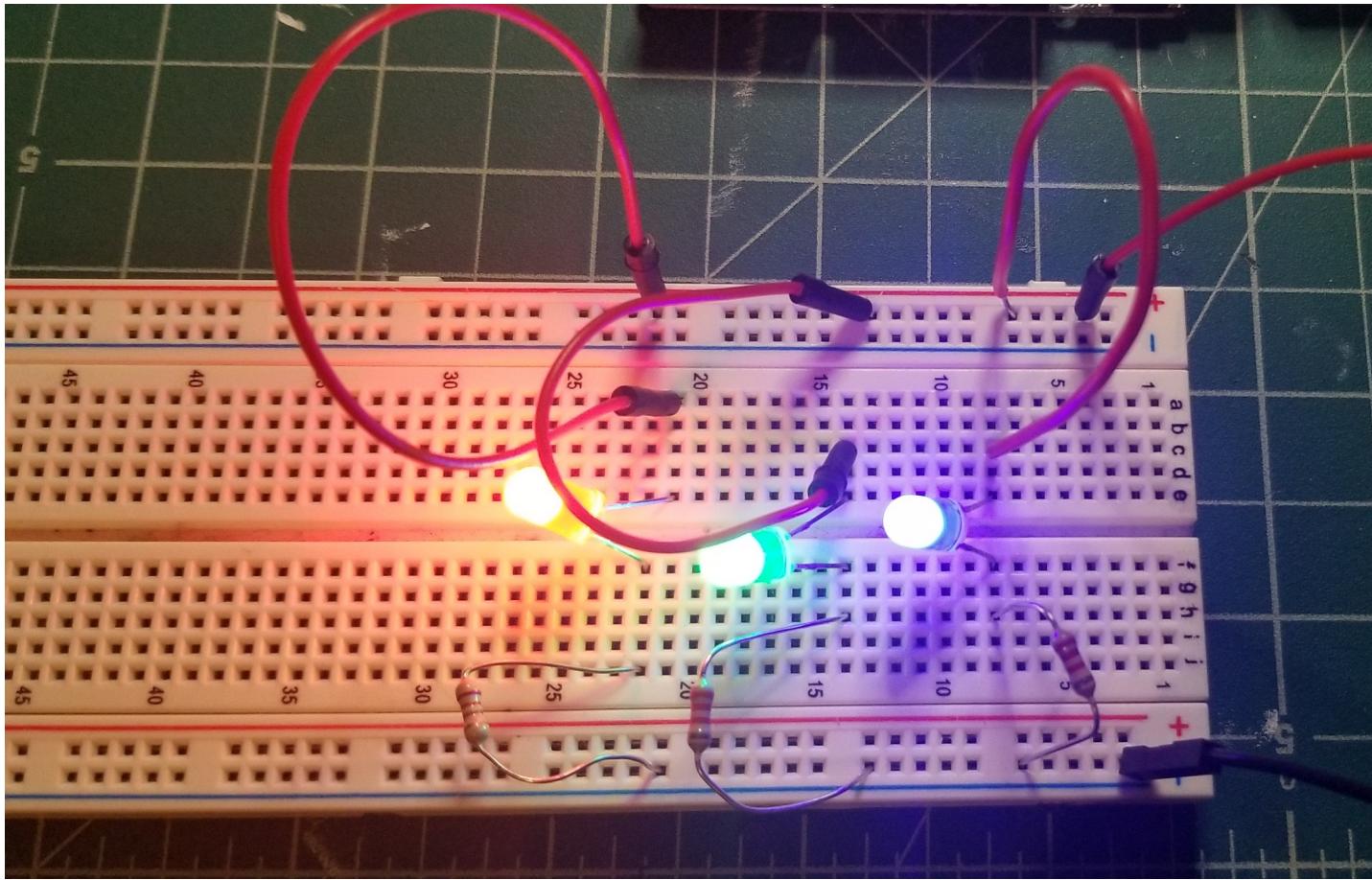
- The layout of the components doesn't matter as long as the circuit is completed.



Repeat Twice more

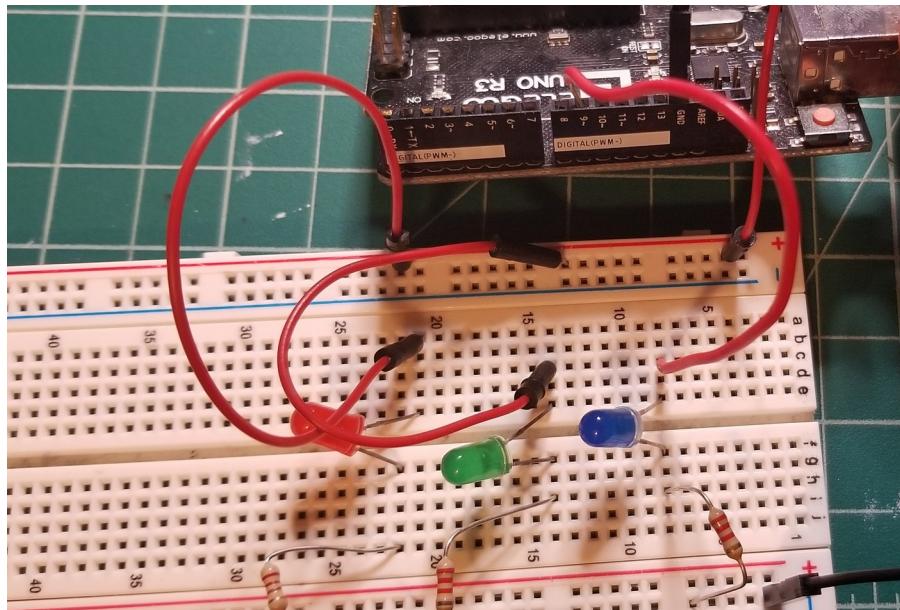
- Use the colors red then green then blue





Move Wires

- Take the wire connecting the red rail and the blue LED
 - Move the power rail side over to digital 11

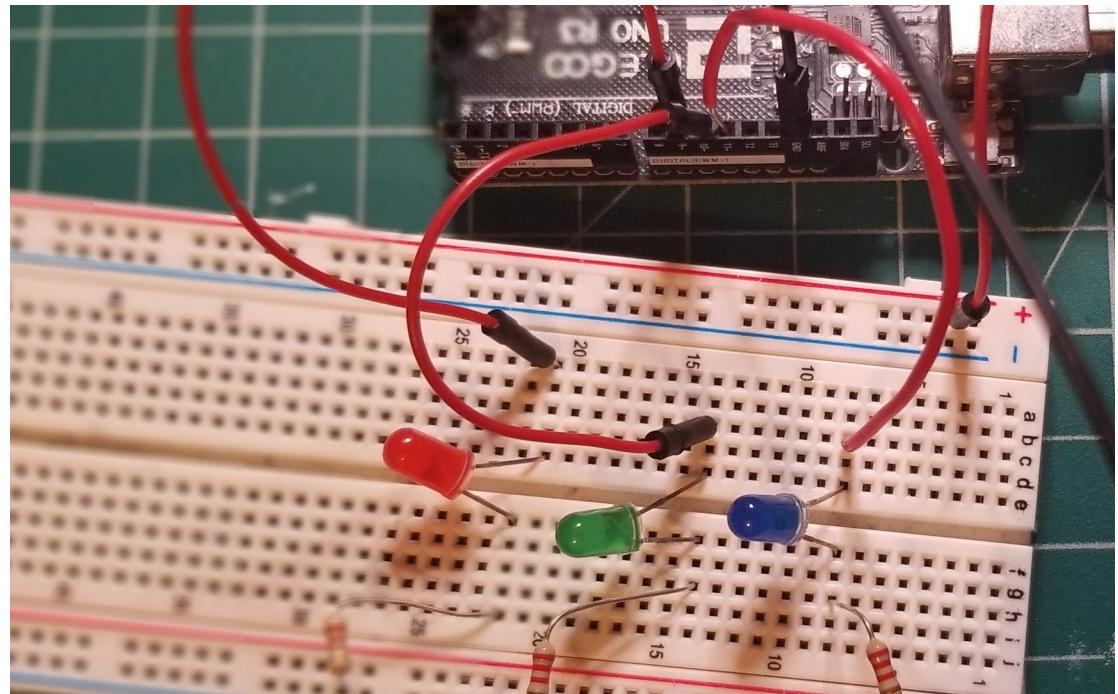


Blink Blue Light Code

- BlinkBlue
 - Simple example to control an LED
- BlinkBlueWithVariables
 - Same as BlinkBlue but with variables

Move Wires

- Then do the same for green
 - Use digital 10
- Then do the same for blue
 - Use digital 9



Blink LEDs Code

- RGBLEDs

RGB LED

- Tricolor LED that gives you any color
- Longest leg is GND
- Single leg on right of ground is red
- Shortest leg is blue
- Leg left of ground is green
- BLUE GREEN GND RED



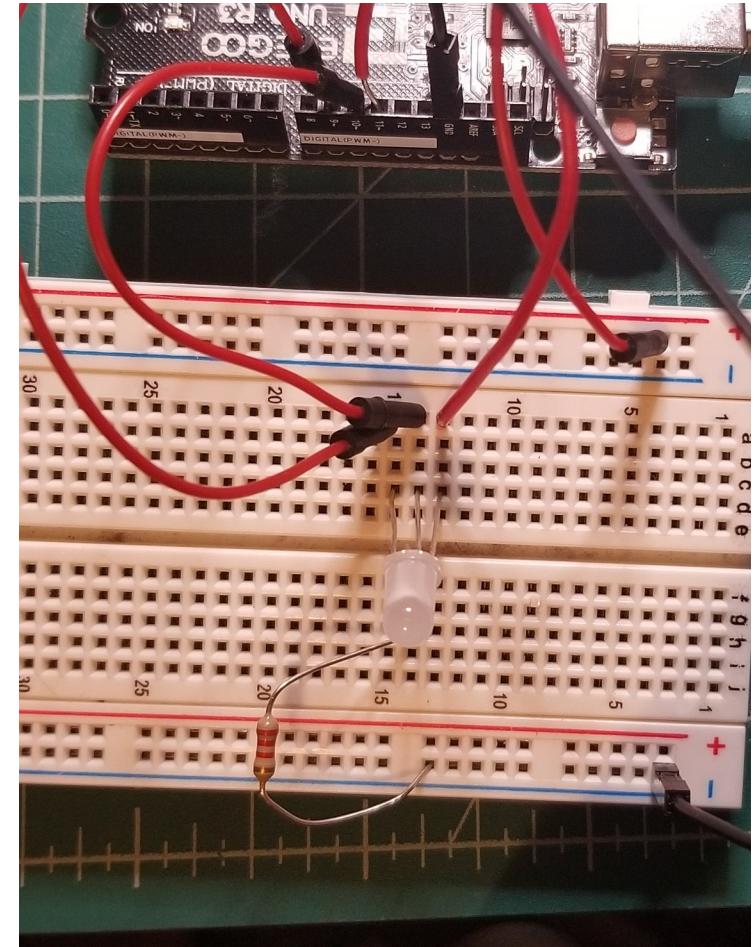
Bend GND Away

- Using the orientation
- BLUE GREEN GND RED
- Bend GND away from you



Replace all the LEDs

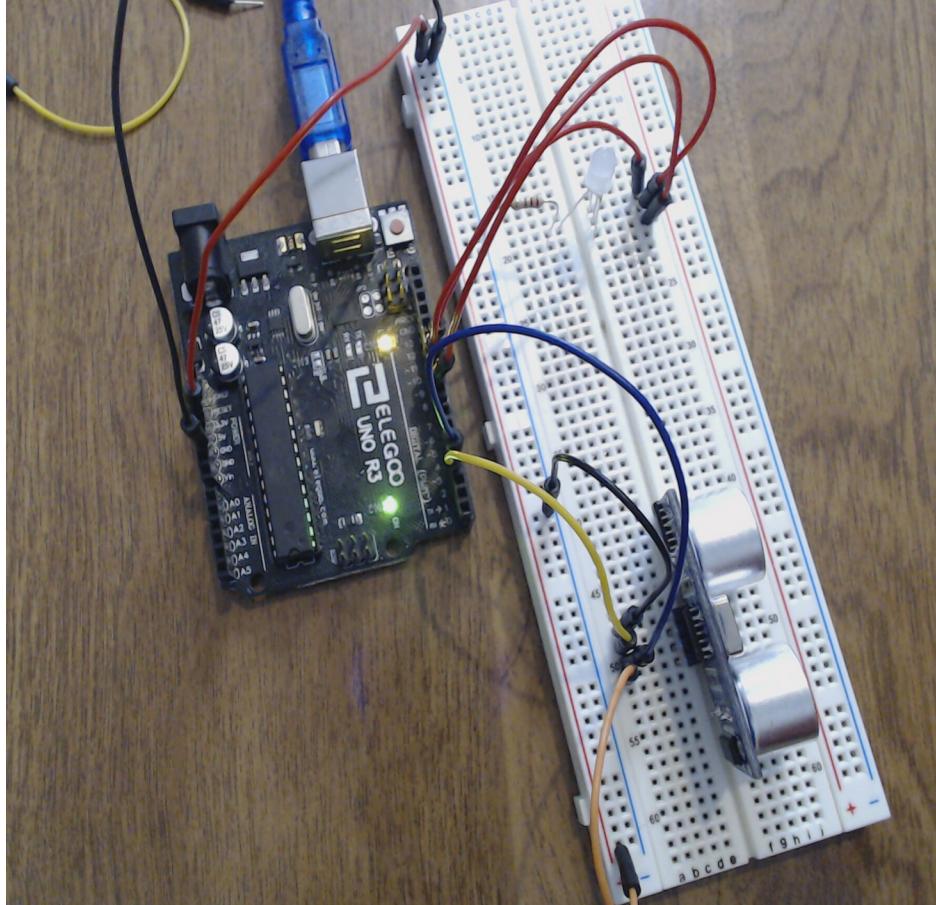
- Replace the green LED with the RGB LED
- GND should plug into the resistor
- Move the digital wires to the RGB LED
 - Red to red
 - Green to green
 - Blue to blue



RGB LED Code

- RGBLEDS
 - This same program will work with the RGB LED

RGB LED and Distance Sensor



RGB LED and Distance Sensor Code

- RGBLEDDistanceSensor
 - Control color based on set distance

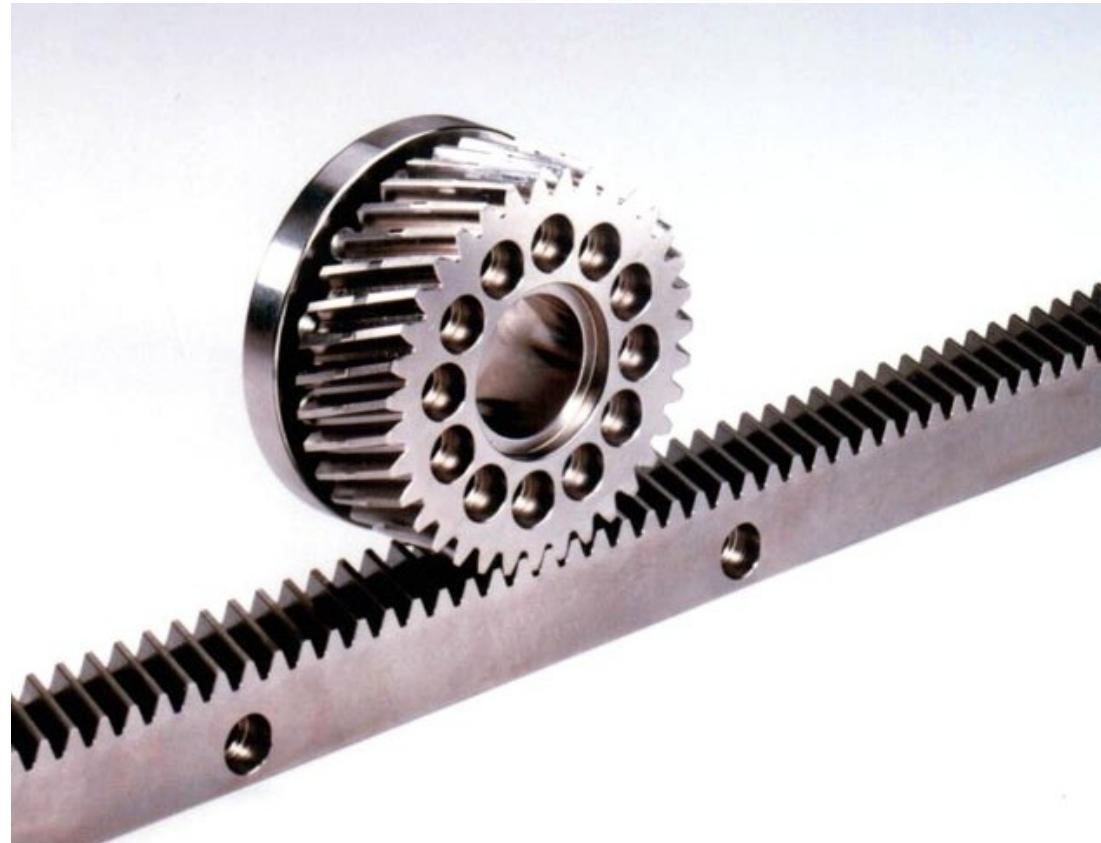
Disconnecting the Computer

- Unplug the computer from the Arduino
- Get a 9V battery and the connector
- Plug it into the Arduino

Rotation to Linear







Arduino Reference

- From arduino.cc go to Documentation > Reference
- See all the builtin funtions and constants
- Find section Standard Libraries
- Download Servo and Stepper