Embedded System Workshops
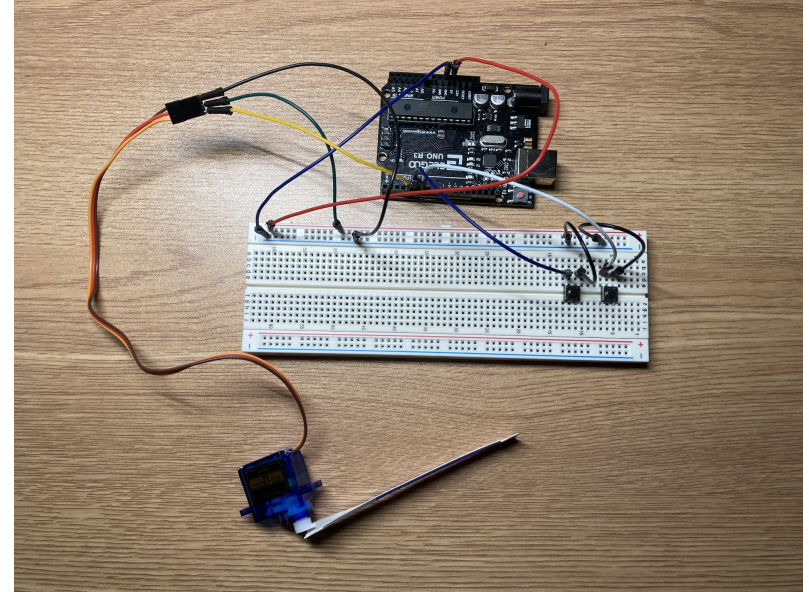
04. Servo Motors
CCA Girls Who Code

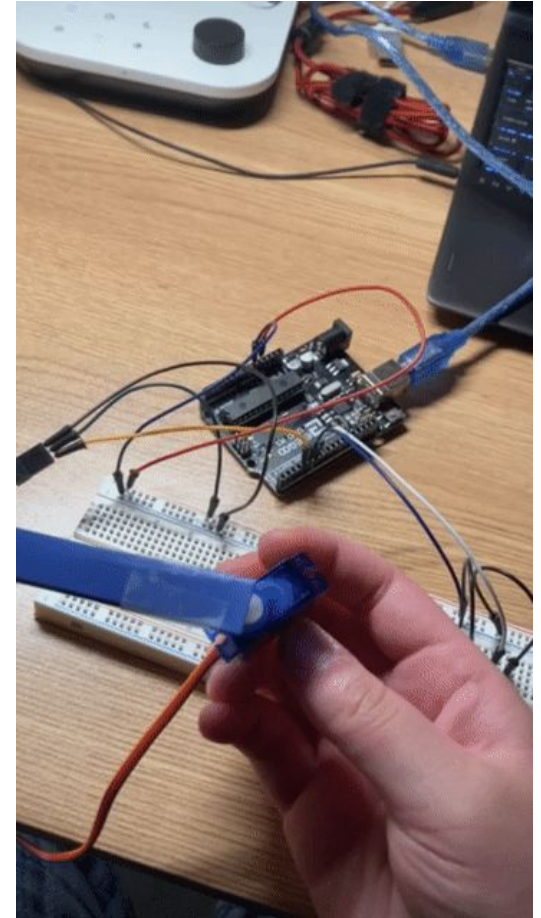# Project Overview

➔ Purpose

   ◆ Introduce the Servo Motor and learn how to control it

➔ Project

   ◆ Use buttons to control a waving servo apparatus

➔ Grab your kit, and let's get started!

What are we making?

➔ Servo Project

  ◆ Use a servo to control rotation with the push of a button

 ◆ Use a set of buttons to make the servo wave an arm in different patterns

Parts List

Below is the list of parts we'll be using during this lesson

- ➔ Arduino UNO R3 Controller Board
- ➔ USB Cable
- ➔ Breadboard
- ➔ Servo Motor (x1) & Servo Horn
- ➔ Buttons (x2)
- ➔ Male-to-male jumper wires
- ➔ (optional) Screwdriver (not included in kit)
- ➔ (optional) Tape and Cardboard to extend the servo horn

Review

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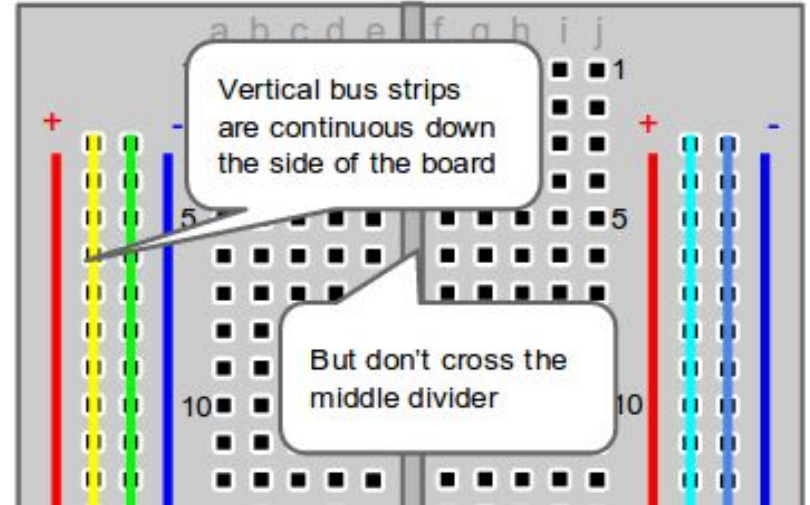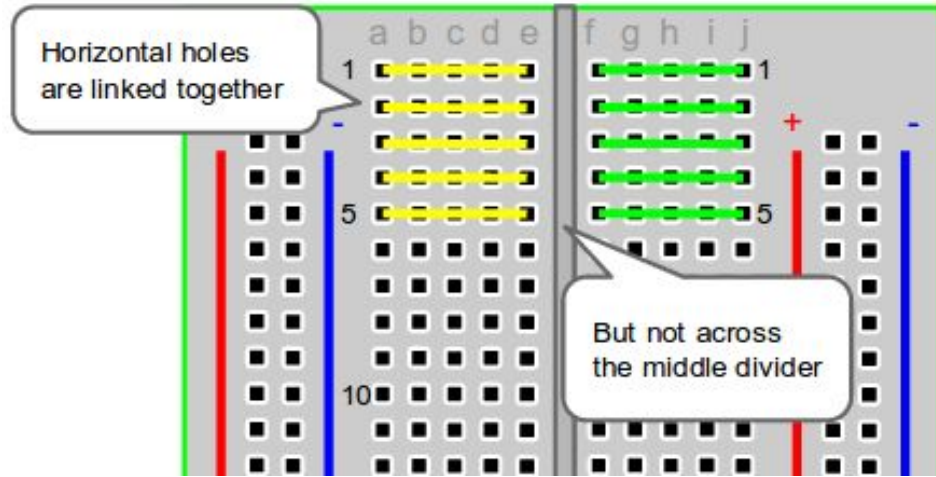
Tip: It is good practice to have your power input connected to the red/positive rail and your ground pin connected to the blue/negative rail.

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
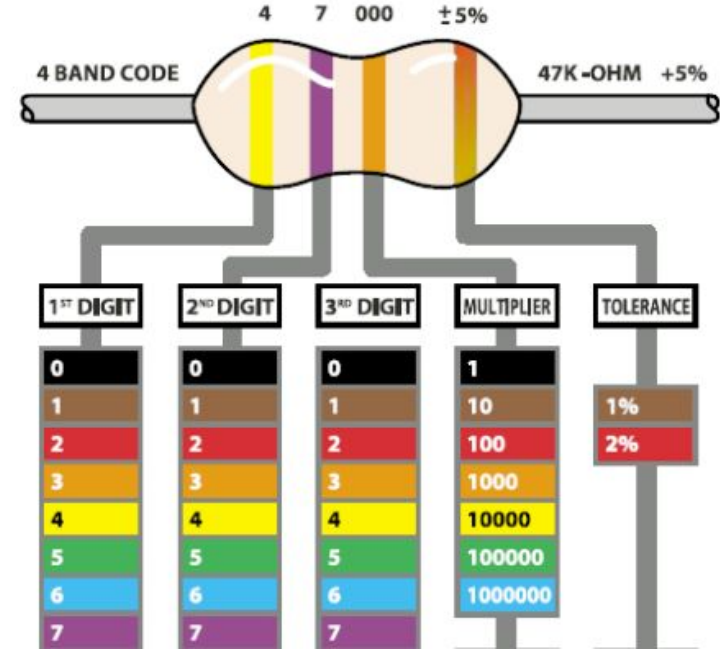
Review: What is a Potentiometer?

➔ A potentiometer is essentially a variable resistor: you can change how much resistance is applied by twisting the knob
➔ This is useful for adjusting brightness, power, etc.
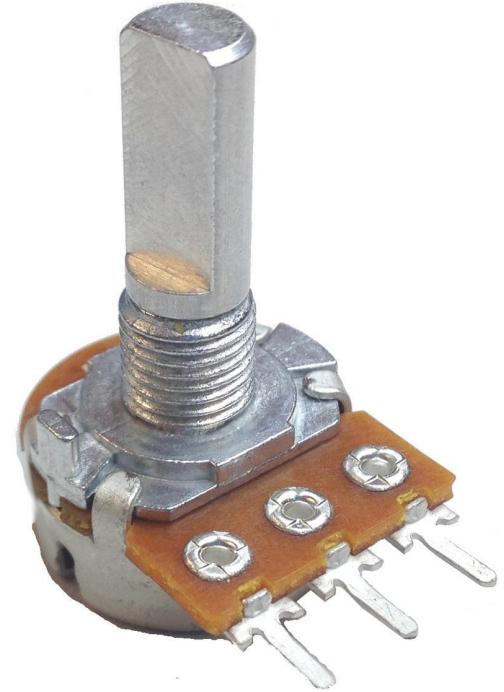
Project

Introduction to Servos

A Servo is a type of motor designed to allow precise rotation within a fixed range of rotational positions. The servos we are using can go from 0 to 180 degrees depending on the input signal.

Servos differ from traditional motors in that they do not provide continuous rotation.

Servo Components

The servo has three wires: red, brown, and orange.

- The Red wire connects to the 5v power supply.
- The Brown wire connects to Ground (GND).
- The Orange wire connects to one of the numbered pins and provides the signal to the servo.
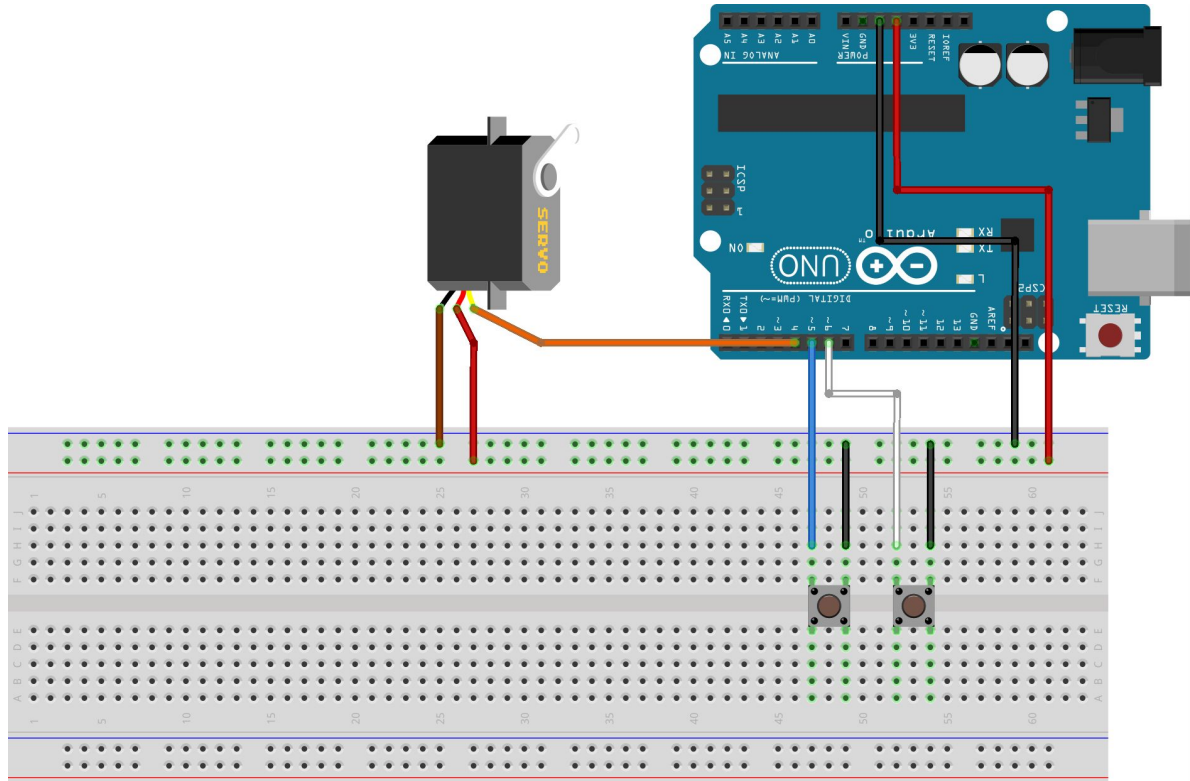
Servo Diagram

NOTE:
On this diagram, the orange wire is replaced by a yellow wire, and the brown wire is replaced by a black one.

Assemble the servo by attaching the servo horn to the servo itself (any of the three types is fine). Use a small screwdriver to securely fasten the two parts together.
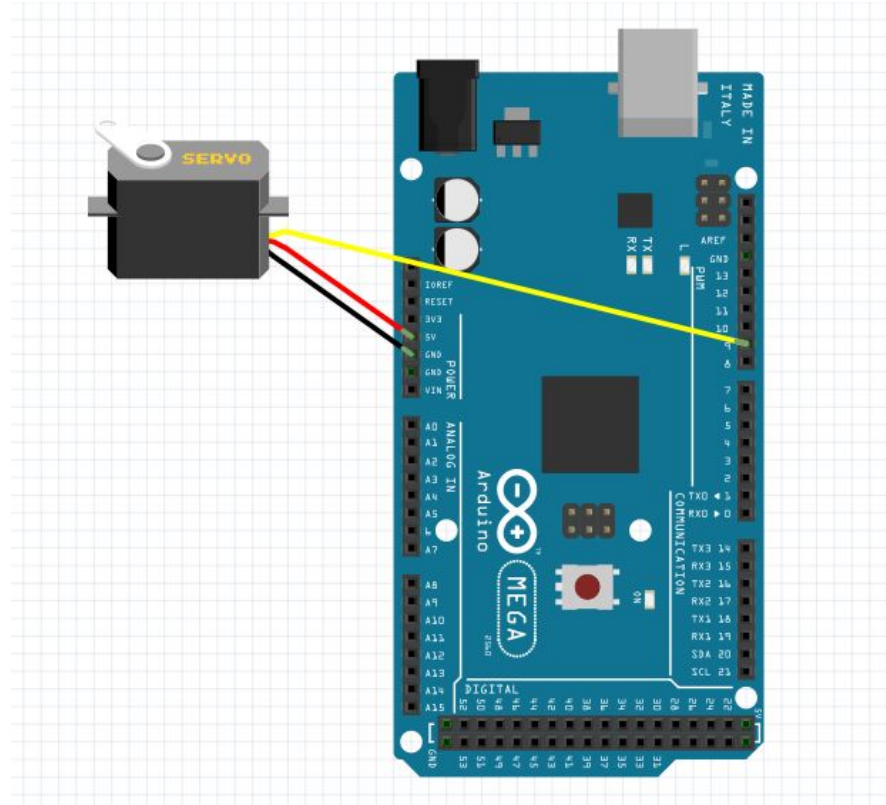
Schematic

Grab the Starter Code from GitHub

➔ 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 lesson4 folder in your repository

This branch is 1 commit behind FTC9837:main.   ⇅ Pull request   ± Compare

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

Create pull request
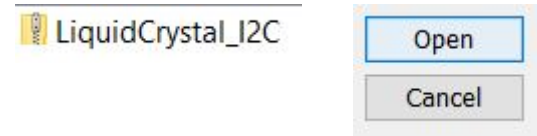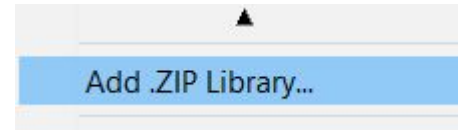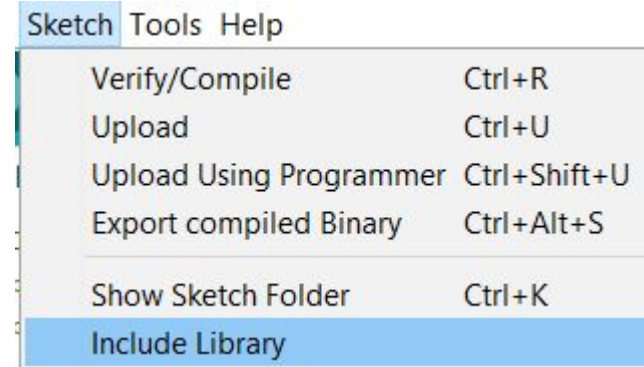
Review: Setting up Arduino

➔ Find and open Arduino on your desktop

➔ Click "File" in the top left corner and click save

➔ Save this tab as "ServoTest"

➔ 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)"
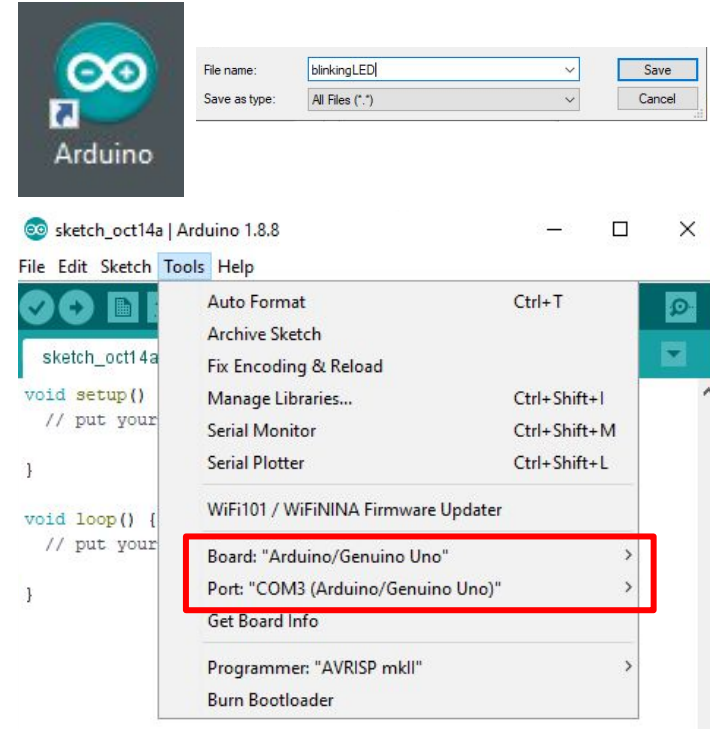
Downloading a Library

➔ We need to download a library in order to use the Servo motors
➔ Download the Servo 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 Servo library.

Sketch	Tools	Help	
Verify/Compile			Ctrl+R
Upload			Ctrl+U
Upload Using Programmer			Ctrl+Shift+U
Export compiled Binary			Ctrl+Alt+S
Show Sketch Folder			Ctrl+K
Include Library			

▲

Add .ZIP Library...

LiquidCrystal_I2C

Open

Cancel

Servo Code

Servo is a type of **Object**.

This segment of code initializes a servo called `myservo`

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

```
servos

#include <Servo.h>

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

void setup(){
  myservo.attach(4);  // attaches the servo on pin 4 to the servo object
}

void loop(){
    myservo.write(0);
    delay(1000);
    myservo.write(90);
    delay(1000);
    myservo.write(180);
    delay(1000);
    myservo.write(90);
    delay(1000);
}
```

`myservo.attach(4);` sets pin 4 to control the servo position

This `loop()` function tells the servo what position to go to, with the delays giving it time to get to that position.
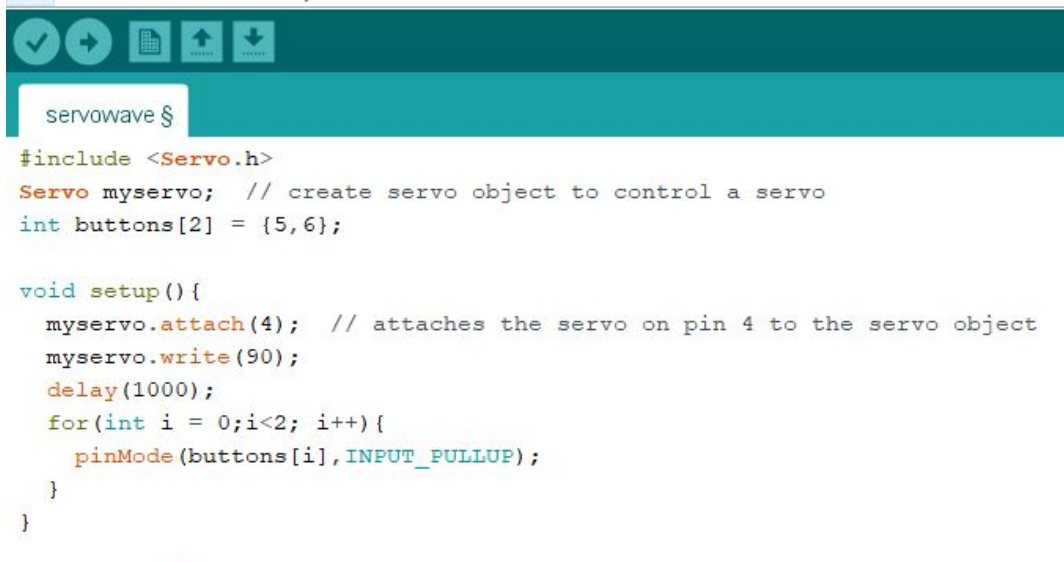
EXERCISE 1

Edit the servo code to change the speed of the wave

EXERCISE 2

Edit the servo code to produce your own movement pattern

Servo Wave Code

```
servowave §
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int buttons[2] = {5,6};

void setup(){
  myservo.attach(4);   // attaches the servo on pin 4 to the servo object
  myservo.write(90);
  delay(1000);
  for(int i = 0;i<2; i++){
    pinMode(buttons[i],INPUT_PULLUP);
  }
}
```

```
void loop(){
    if (digitalRead(buttons[0])==LOW){
      for(int i=0;i<3;i++){
        myservo.write(45);
        delay(750);
        myservo.write(135);
        delay(750);
      }
      myservo.write(90);
      delay(1000);
    }
    if (digitalRead(buttons[1])==LOW){
      for(int i=0;i<3;i++){
        myservo.write(0);
        delay(750);
        myservo.write(90);
        delay(750);
      }
      myservo.write(90);
      delay(1000);
    }
}
```

[View this code on GitHub!](#)

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