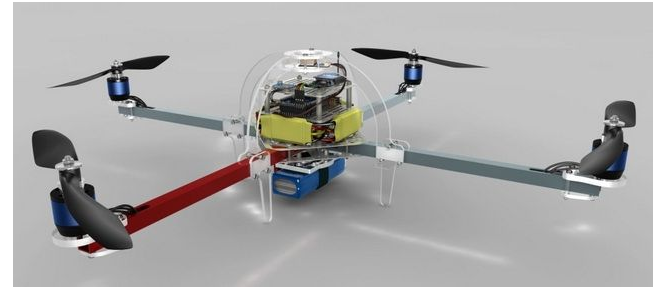
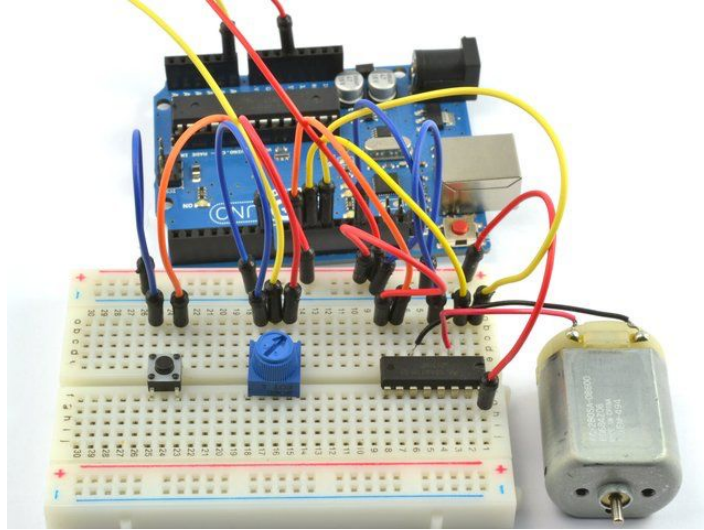




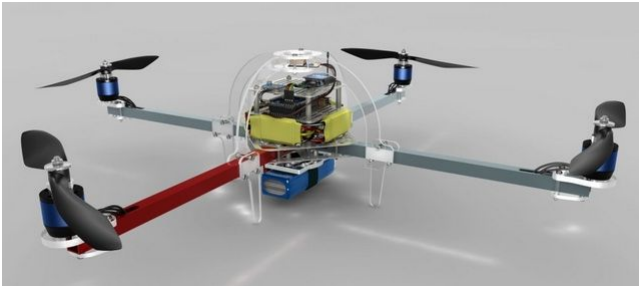
Do Now

What is the role of a magnet in a motor?





Intro to Digital Motors



Goals

- Discuss how basic commercial motors work
- Identify 3 common types of motors
- Identify and justify which type of motor to use in different scenarios



Motors

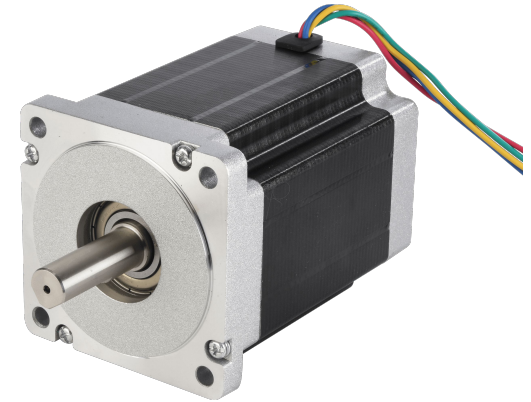
- Devices that convert electrical energy into rotational kinetic energy



**Basic DC
Motors**



**Servo
Motors**



**Stepper
Motors**

Standard DC Hobby Motors

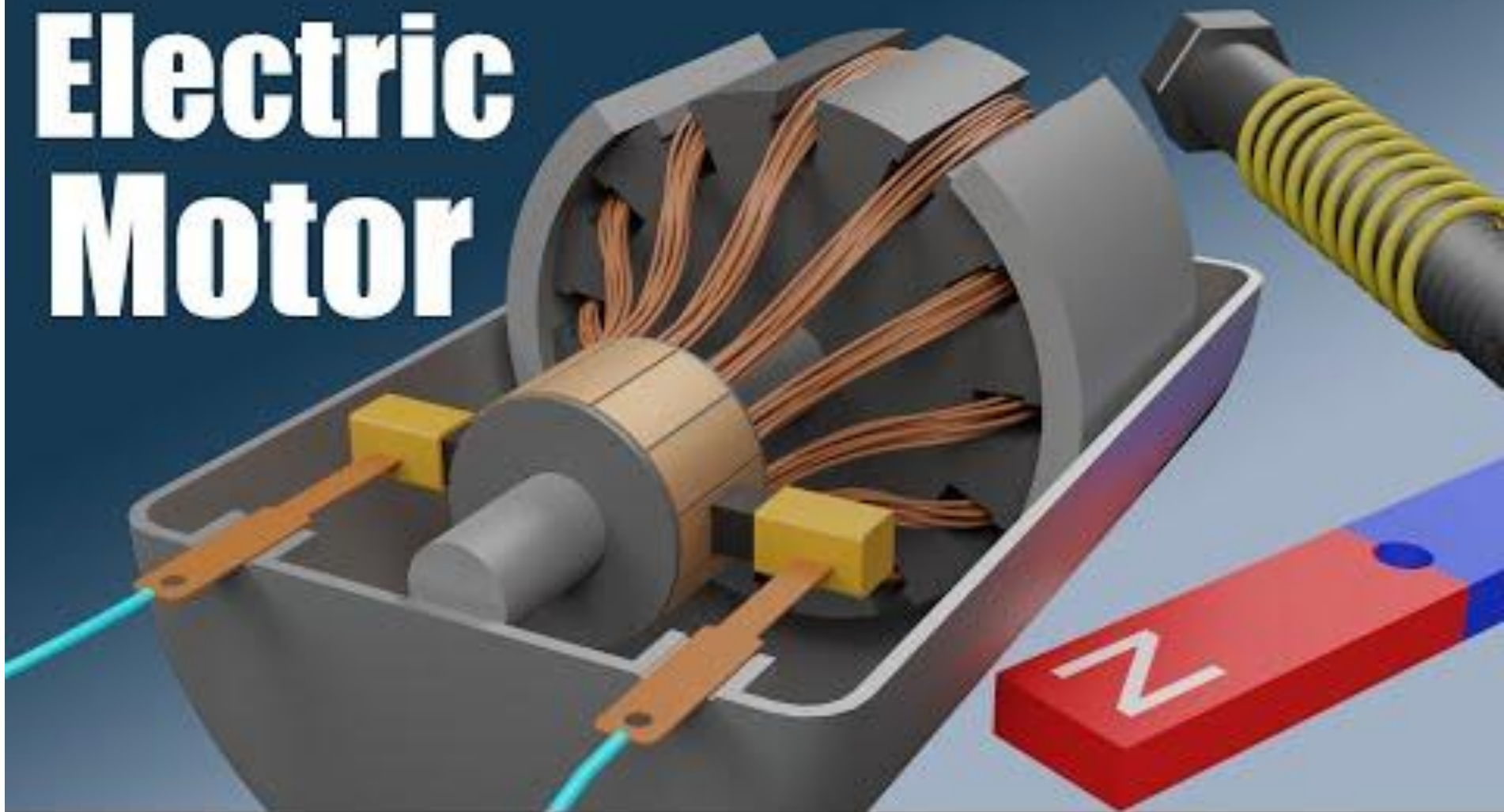
- Can spin CW or CCW
- Crude: either on or off
 - Speed determined by voltage
 - Direction determined by polarity of current
- Unlimited rotation in one direction
- What are some devices that could use hobby motors?



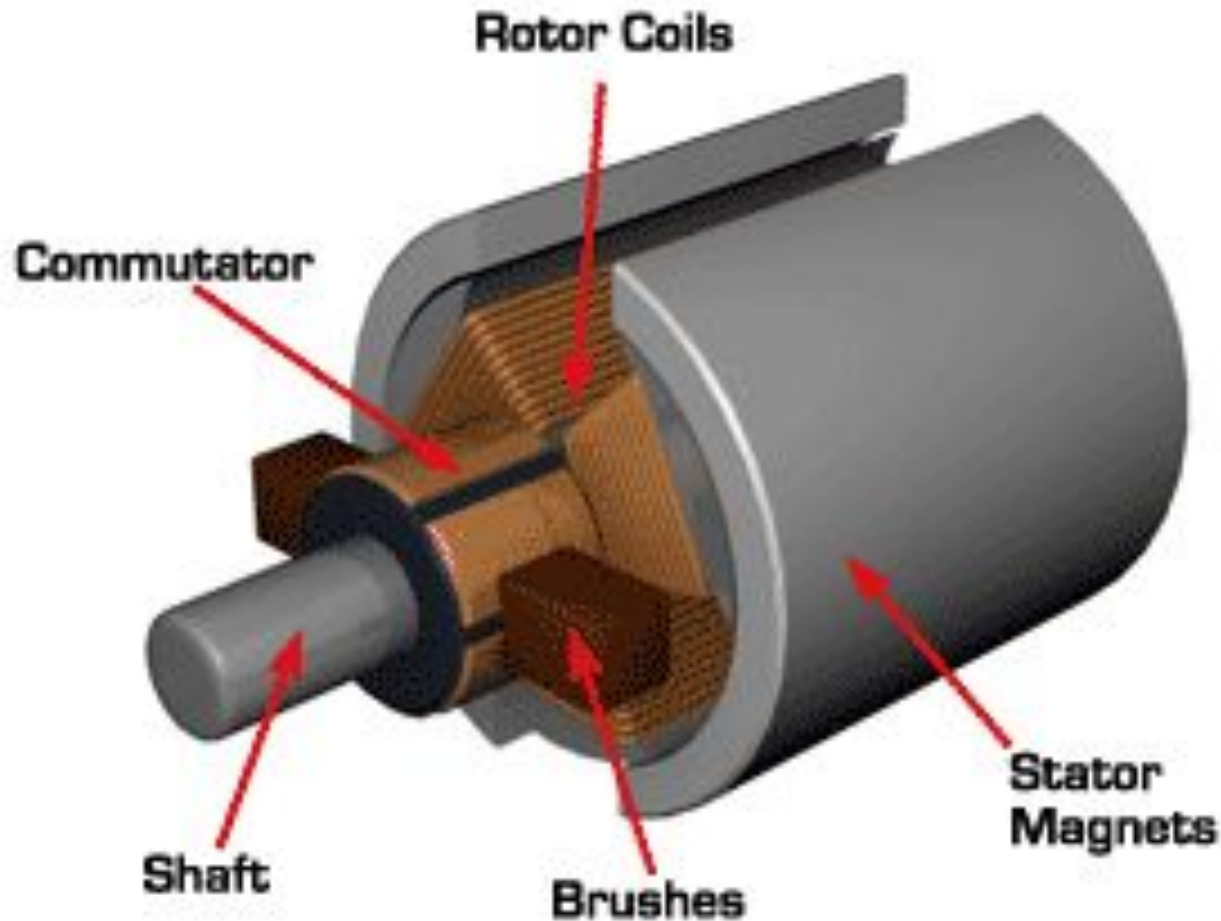
Photo by ElectroPeak

gearbox motor

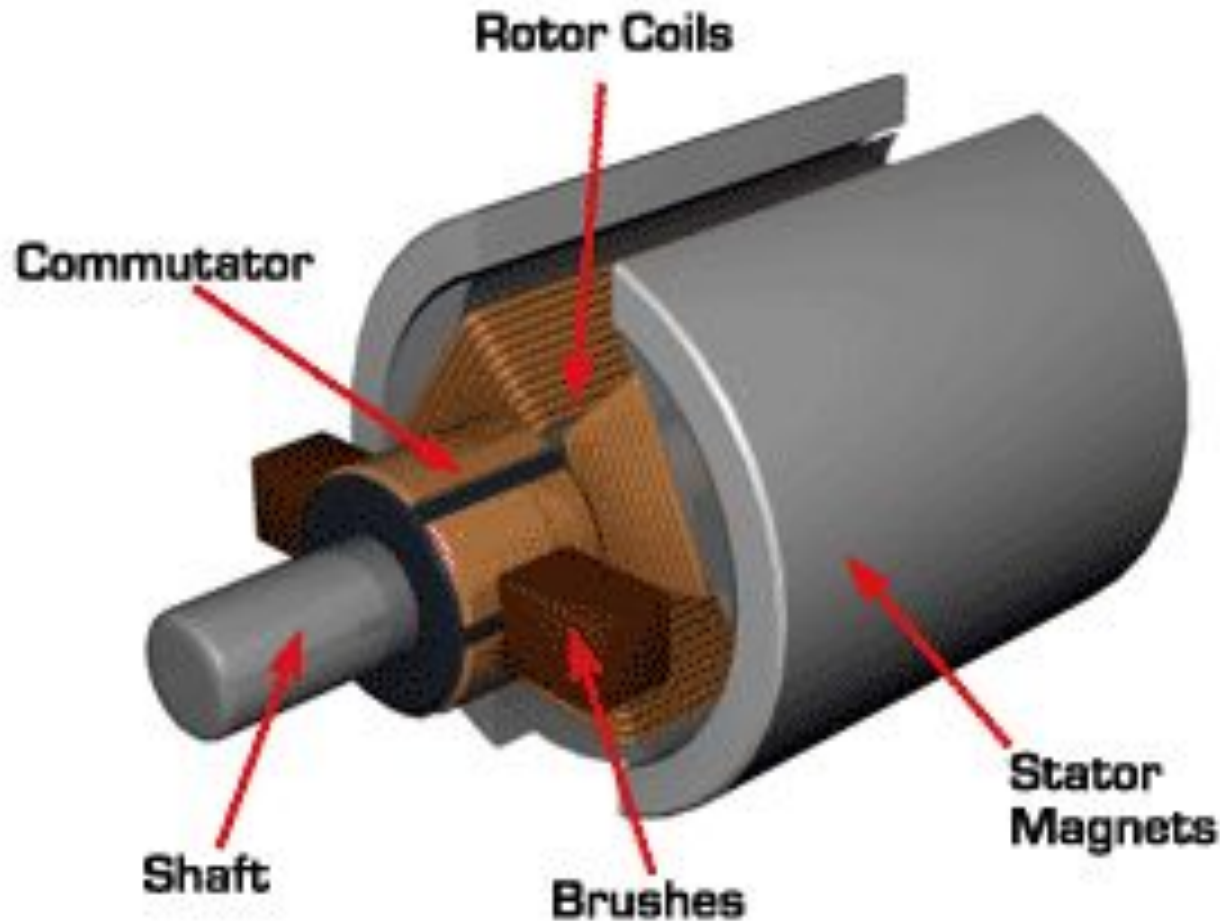
Electric Motor



What's Inside a DC Hobby Motor?



What's Inside a DC Hobby Motor?



How is this design an improvement on the basic motors we built yesterday?

Motors

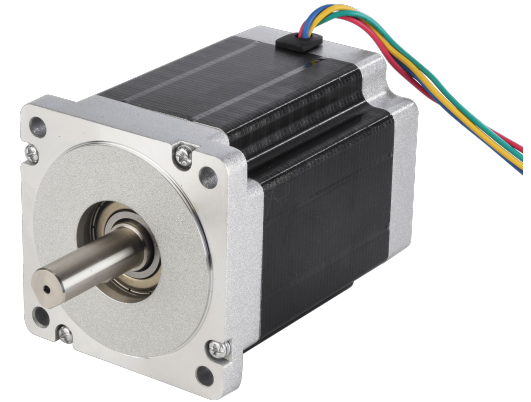
- Devices that convert electrical energy into rotational kinetic energy



**Basic DC
Motors**



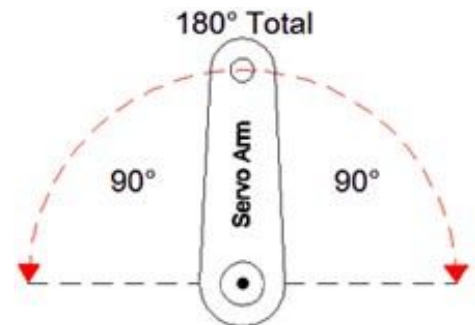
**Servo
Motors**



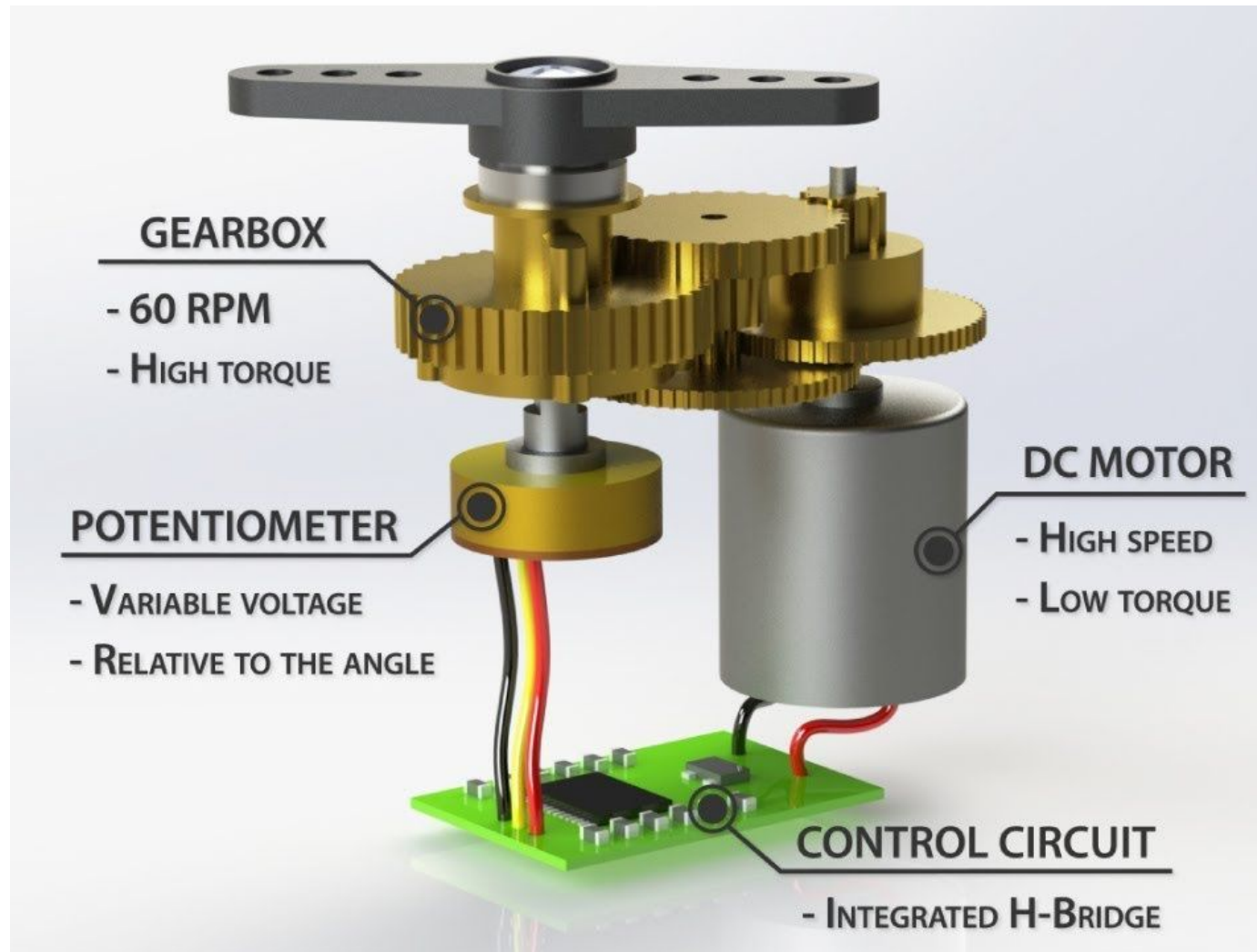
**Stepper
Motors**

Standard Servos

- Precise: Can rotate to a specific angle
- Limited rotation
 - often 180°
- High torque at high speeds
- Good for back and forth, open and closed, dials
- What are some devices that could use servo motors?



What's Inside a Servo?



Motors

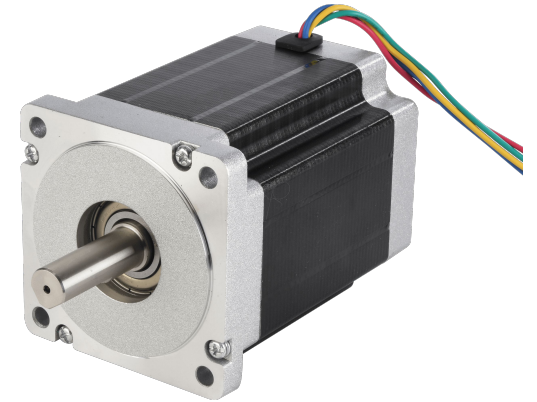
- Devices that convert electrical energy into rotational kinetic energy



**Basic DC
Motors**



**Servo
Motors**



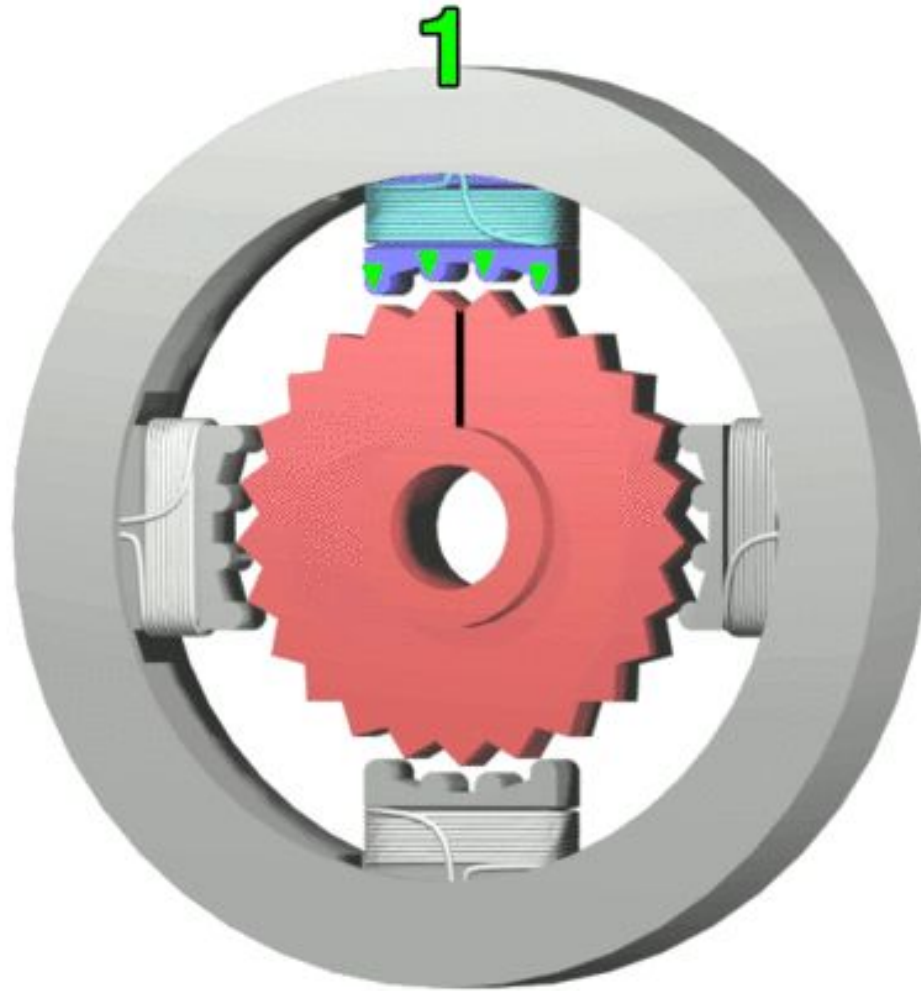
**Stepper
Motors**

Standard Stepper Motors

- Precise: Can program to rotate to a specific angle or step
- Unlimited rotation in either direction
- High torque at low speeds
- What are some devices that could use stepper motors?



What's Inside a Stepper Motor?



Motor Applications

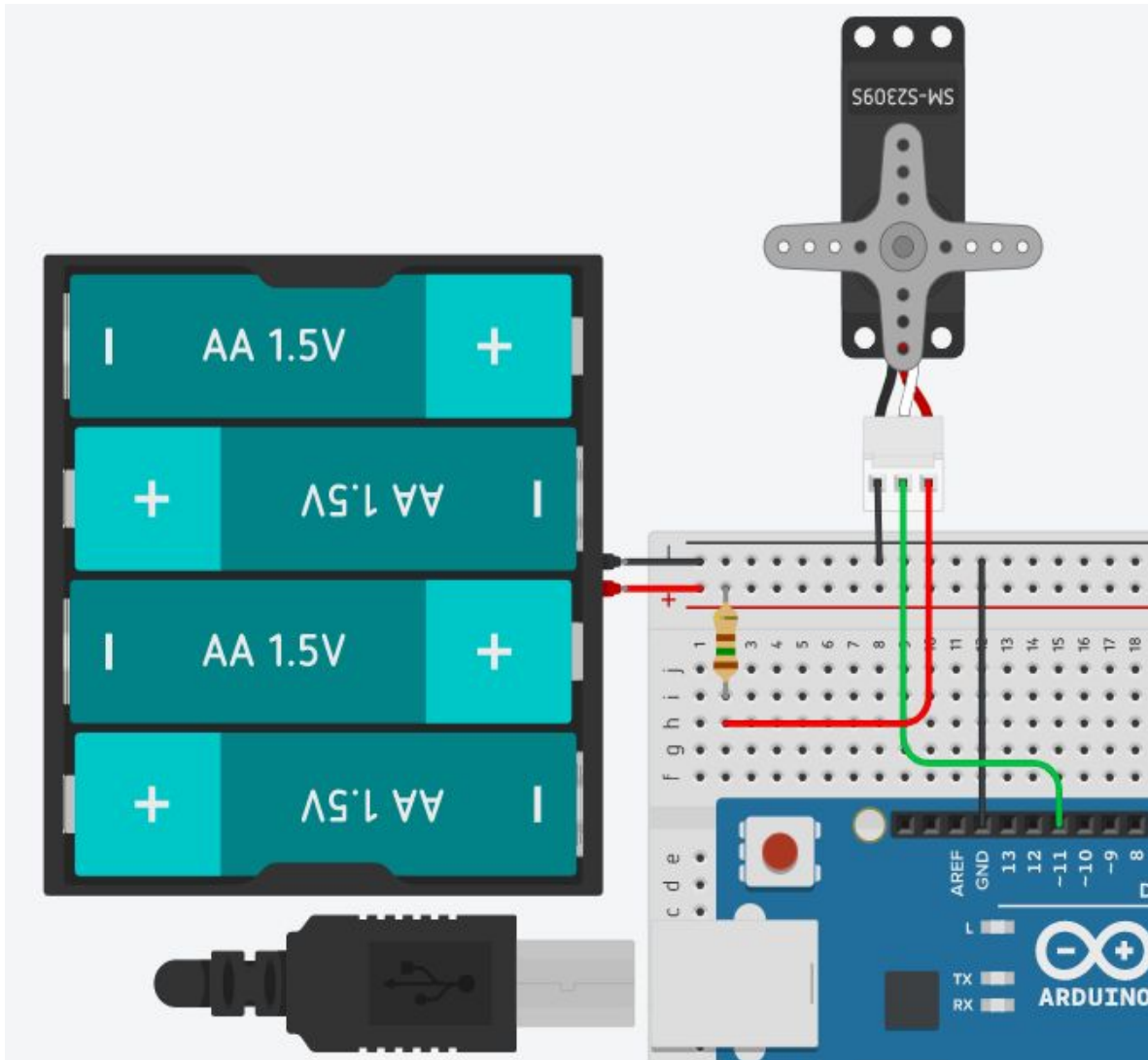
With your partner, determine which type of motor you would use for each device and explain your reasoning

1. Automatic doors
2. Laptop fan
3. Elevator
4. Escalator
5. Electric toothbrush
6. Grocery store conveyor belt

Powering a Motor

- Even though the Arduino can output 5V, the motor can draw more current than what the Arduino may safely be able to output
- Therefore we usually connect the motor to an external power source (like a battery pack), and use the Arduino just to send the signal

Powering a Motor



NOTE:

Check the **voltage rating** of your motor and measure the actual voltage output of your battery pack.

If the voltage supplied by your battery pack is above the maximum threshold for your motor, **you will need a resistor** as shown here.

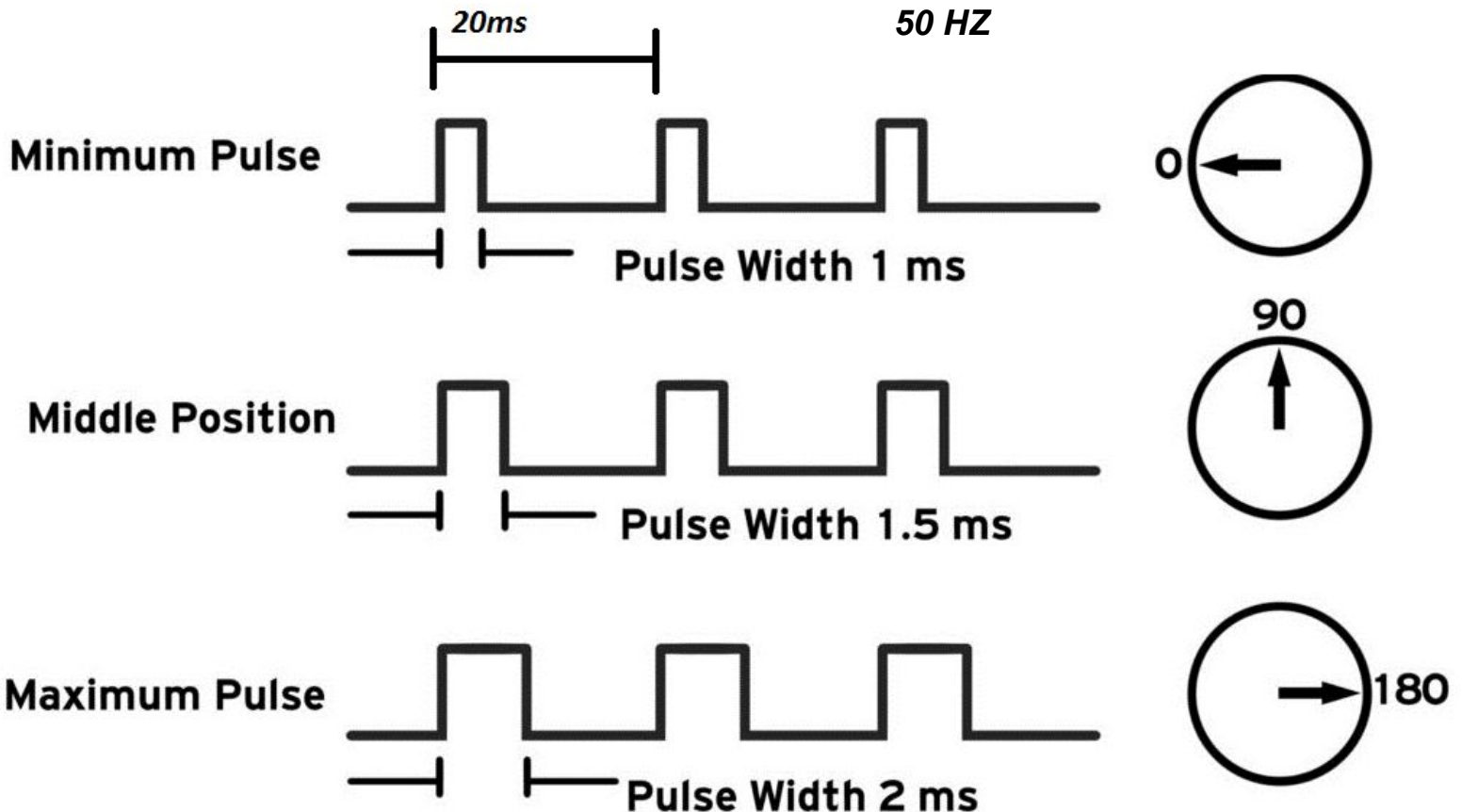
Otherwise, you will burn out your Arduino.

Session 1 Servo Motors and Libraries



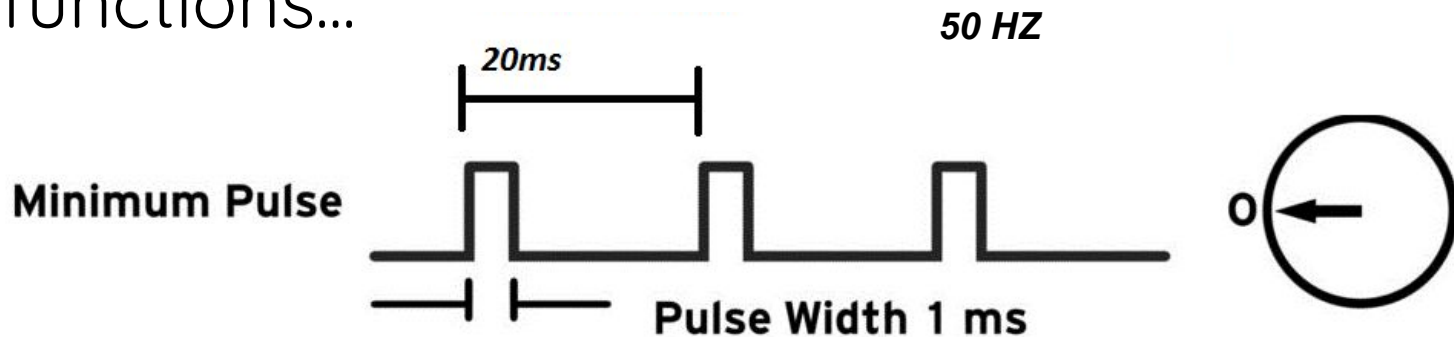
Controlling a Servo

- Using PWM, the size of the pulse width determines the angle/position of the servo shaft



Controlling a Servo

- **analogWrite()** isn't set to this frequency
- You could do this without any special functions...



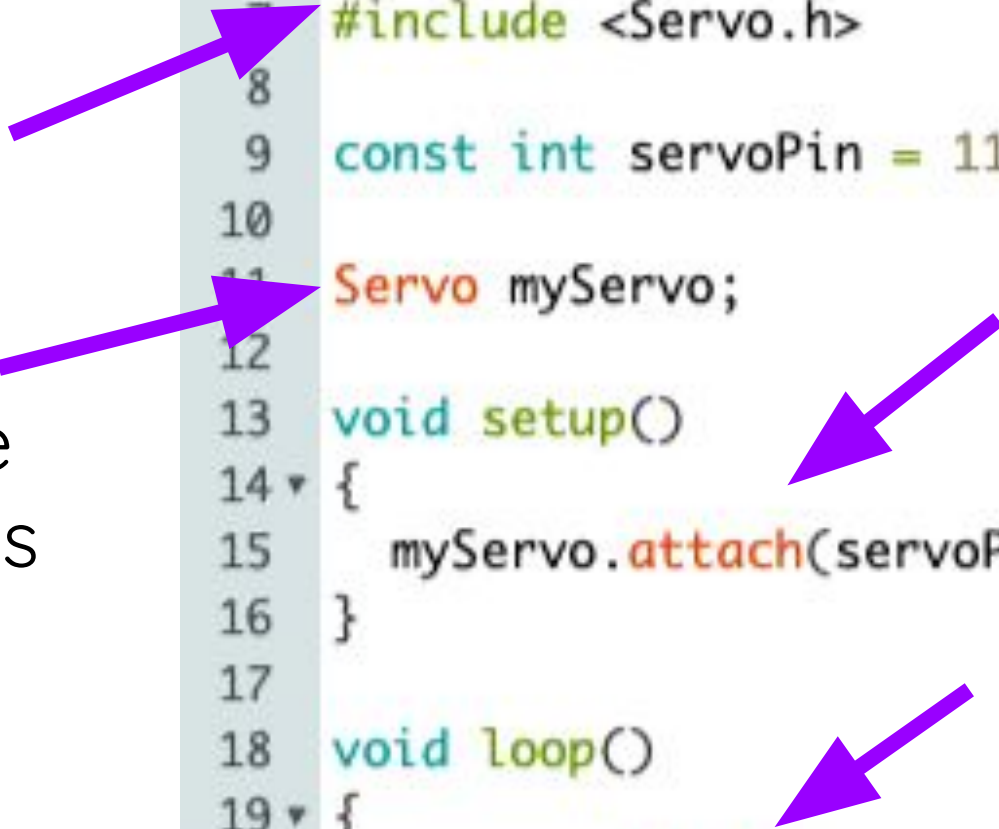
```
digitalWrite(11, HIGH);  
delayMicroseconds(1000);  
digitalWrite(11, LOW);  
delayMicroseconds(19000);
```

... but there are some special functions that will make your life easier...

Coding with Servos

- There are some useful structures and functions for controlling servos
- To access them, you need the **Servo Library**

```
7 #include <Servo.h>
8
9 const int servoPin = 11;
10
11 Servo myServo;
12
13 void setup()
14 {
15     myServo.attach(servoPin);
16 }
17
18 void loop()
19 {
20     myServo.write(0);
21     delay(2000);
22
23     myServo.write(50);
24     delay(2000);
25
26     myServo.write(100);
27     delay(2000);
28 }
```



Libraries

- A **library** is a collection of code that makes it easier to operate a sensor or actuator
 - This code is not normally included in the standard sketch (to save memory, keep sketches simple)

In Standard Arduino Sketches...

Always Included:

int, float
delay()
digitalWrite()

Not Included Unless Instructed To:

Servo
attach(), write()

What's in a Library?

Libraries often come
with new **objects** as
well...

...what exactly is an
object?

```
7  #include <Servo.h>
8
9  const int servoPin = 11;
10
11  Servo myServo;
12
13  void setup()
14  {
15      myServo.attach(servoPin);
16  }
17
18  void loop()
19  {
20      myServo.write(0);
21      delay(2000);
22
23      myServo.write(50);
24      delay(2000);
25
26      myServo.write(100);
27      delay(2000);
28  }
```

LED

Variables

int
ledPin
6

int
delayTime
500

int
brightness
150

Functions

blinkSlow()

blinkRate()

setBrightness()

Objects

- A structure (“package”) of variables and functions grouped together
- Often designed to “fit” some real entity
- A template for creating copies (“instances”)

What's in a Library?

- Constants and Variables
- Functions
- Objects
 - Variables and Functions



[Reference](#) [Language](#) | [Libraries](#) | [Comparison](#) | [Changes](#)

Servo library

This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

The Servo library supports up to 12 motors on most Arduino boards and 48 on the Arduino Mega. On boards other than the Mega, use of the library disables `analogWrite()` (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins. On the Mega, up to 12 servos can be used without interfering with PWM functionality; use of 12 to 23 motors will disable PWM on pins 11 and 12.

To use this library

```
#include <Servo.h>
```

Objects

- [Servo](#)

Functions

- [attach\(\)](#)
- [write\(\)](#)
- [writeMicroseconds\(\)](#)
- [read\(\)](#)
- [attached\(\)](#)
- [detach\(\)](#)

Examples

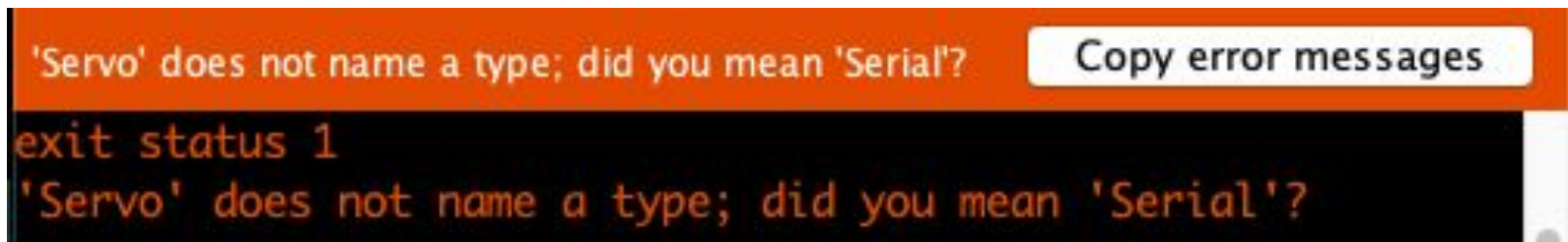
- [Knob](#): control the shaft of a servo motor by turning a potentiometer.
- [Sweep](#): sweeps the shaft of a servo motor back and forth.

How do you use a Library?

`#include <LibraryName.h>`

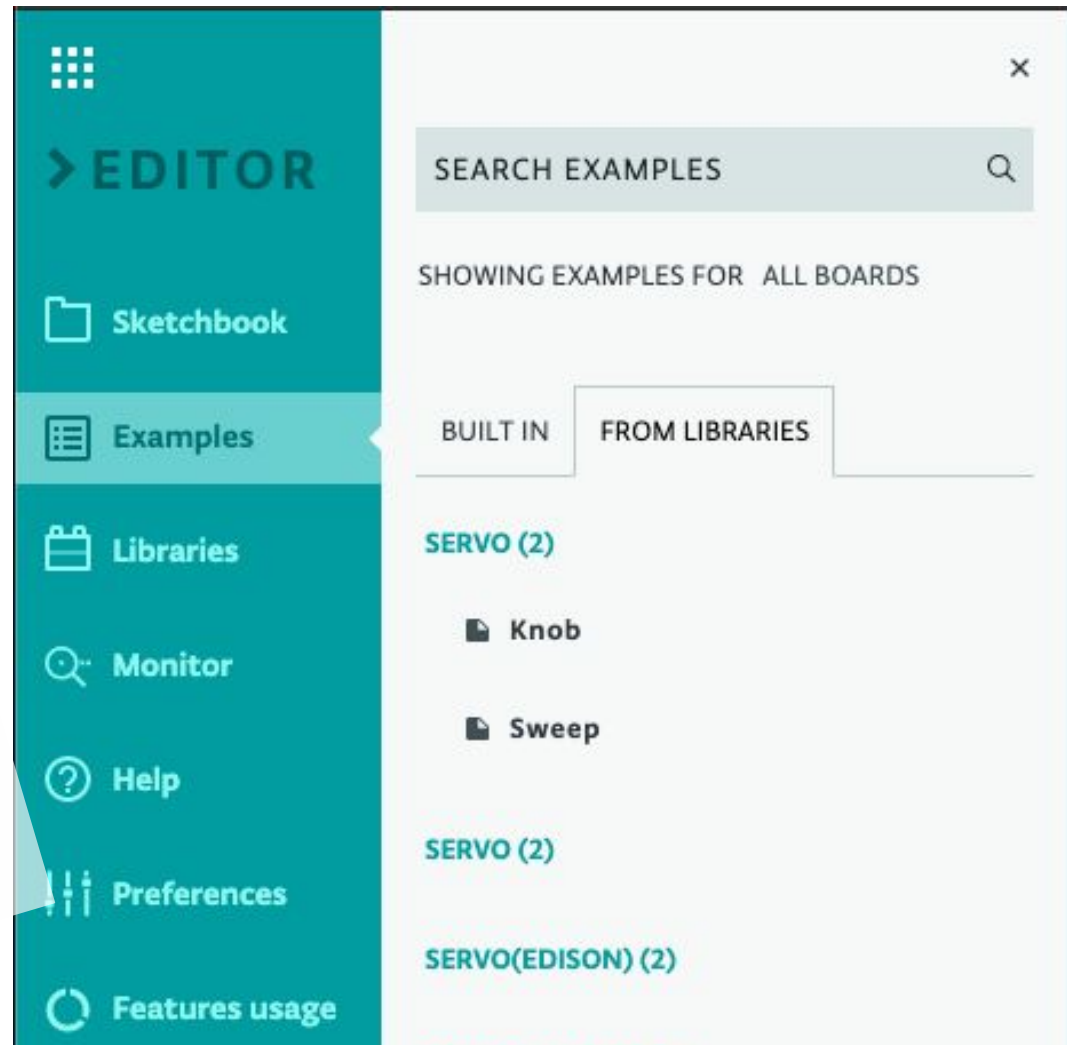
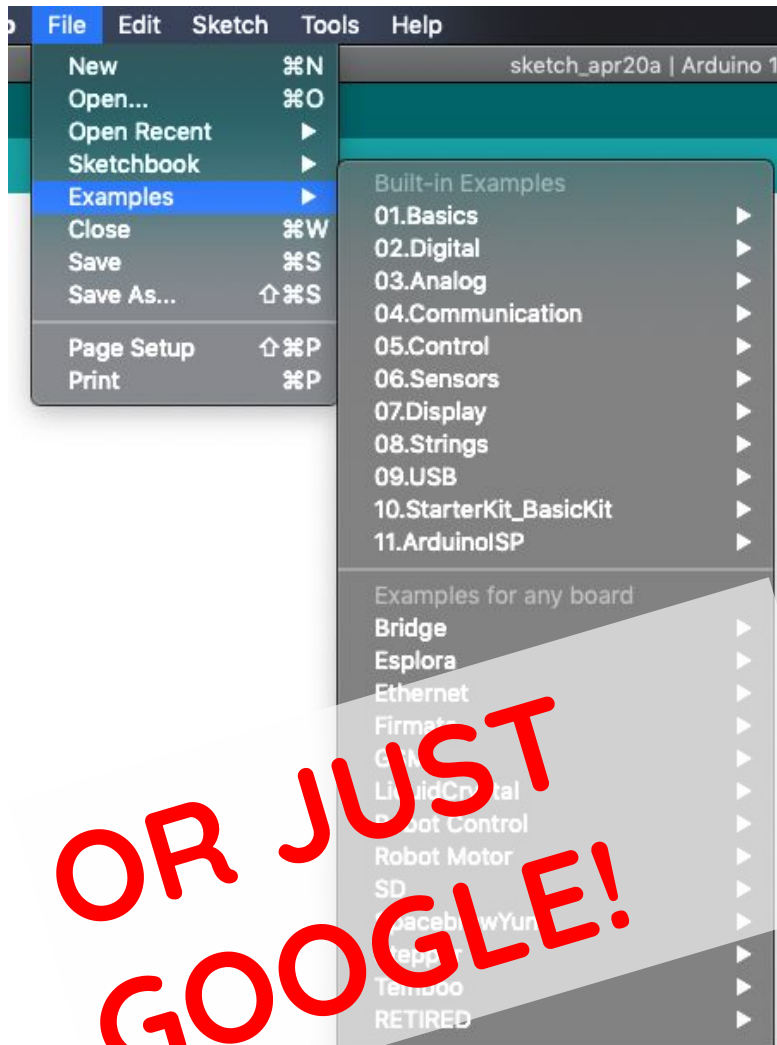
Example: `#include <Servo.h>`

Servo.h is a **header file**. It only contains declarations so that you can refer to special objects, constants, or functions without throwing an error



The header file is located in the same folder as the library, which has all of the real information. It is retrieved during the compiling process.

Finding Library examples



Always look for an example when starting out!

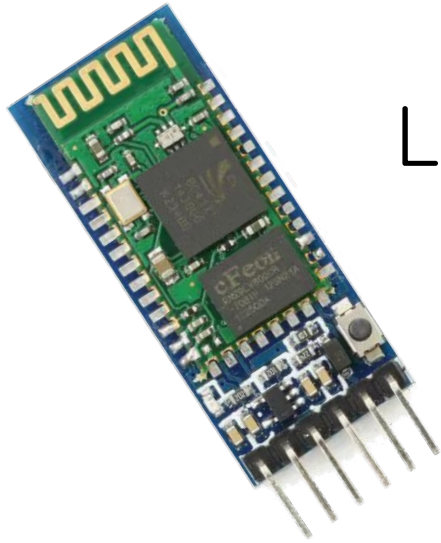
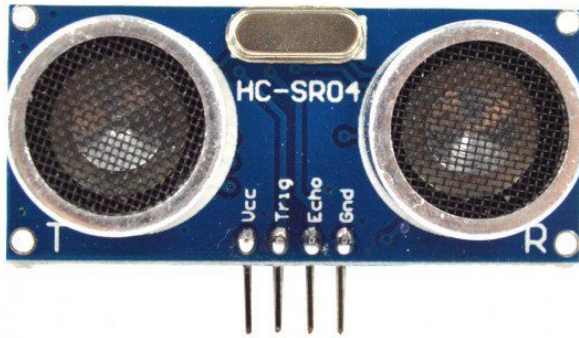
What about other libraries?

Common Arduino Libraries

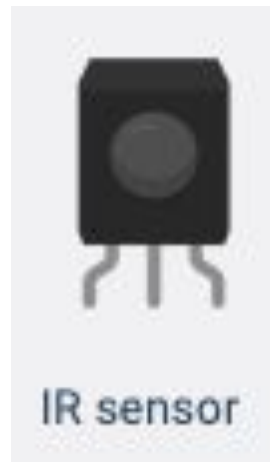
Documentation for many libraries is on Arduino.cc

Standard Libraries

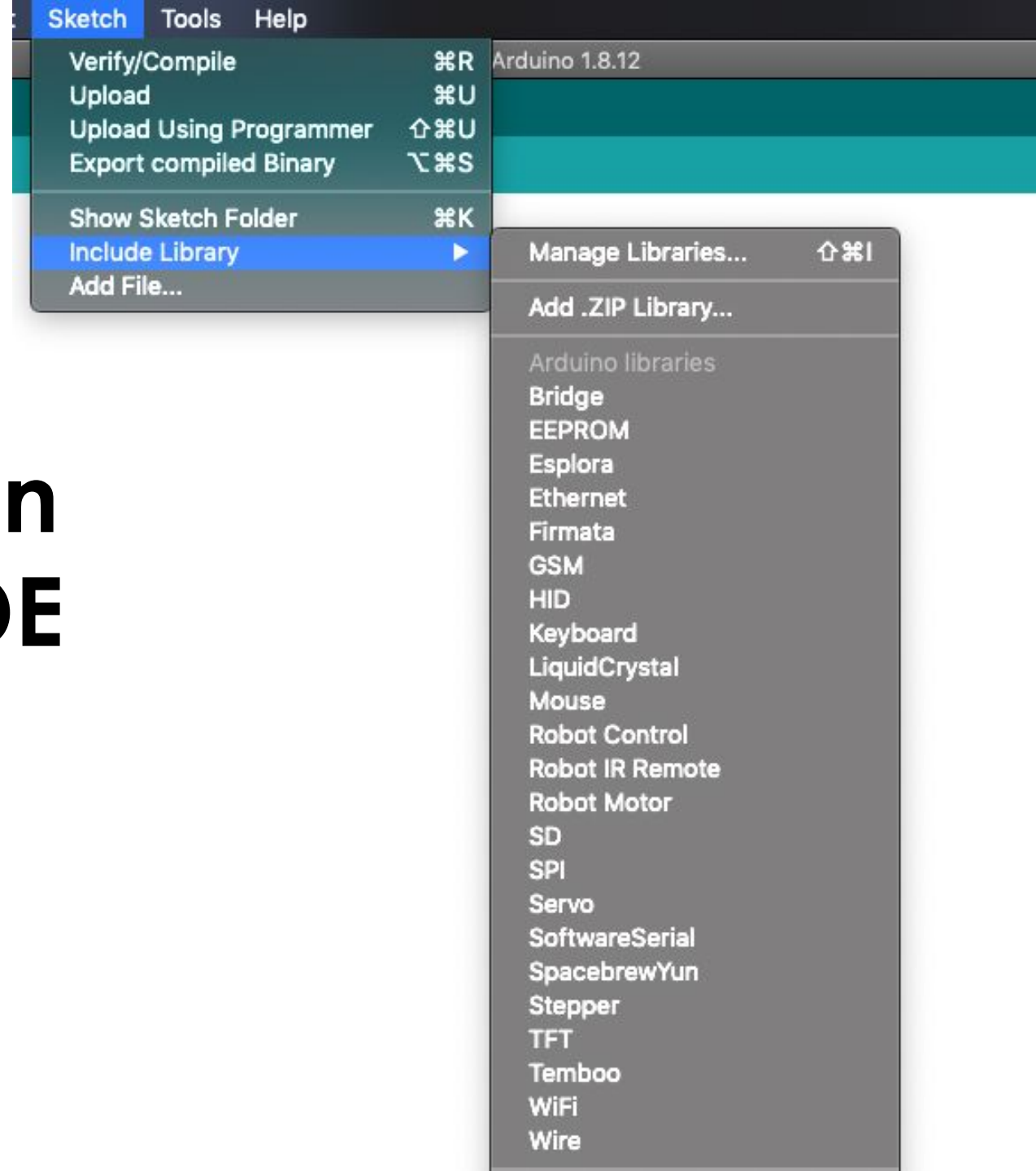
- [EEPROM](#) - reading and writing to "permanent" storage
- [Ethernet](#) - for connecting to the internet using the Arduino Ethernet Shield, Arduino Ethernet Shield 2 and Arduino Leonardo ETH
- [Firmata](#) - for communicating with applications on the computer using a standard serial protocol.
- [GSM](#) - for connecting to a GSM/GRPS network with the GSM shield.
- [LiquidCrystal](#) - for controlling liquid crystal displays (LCDs)
- [SD](#) - for reading and writing SD cards
- [Servo](#) - for controlling servo motors
- [SPI](#) - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- [SoftwareSerial](#) - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate [Mikal Hart's NewSoftSerial](#) library as SoftwareSerial.
- [Stepper](#) - for controlling stepper motors
- [TFT](#) - for drawing text , images, and shapes on the Arduino TFT screen
- [WiFi](#) - for connecting to the internet using the Arduino WiFi shield
- [Wire](#) - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.



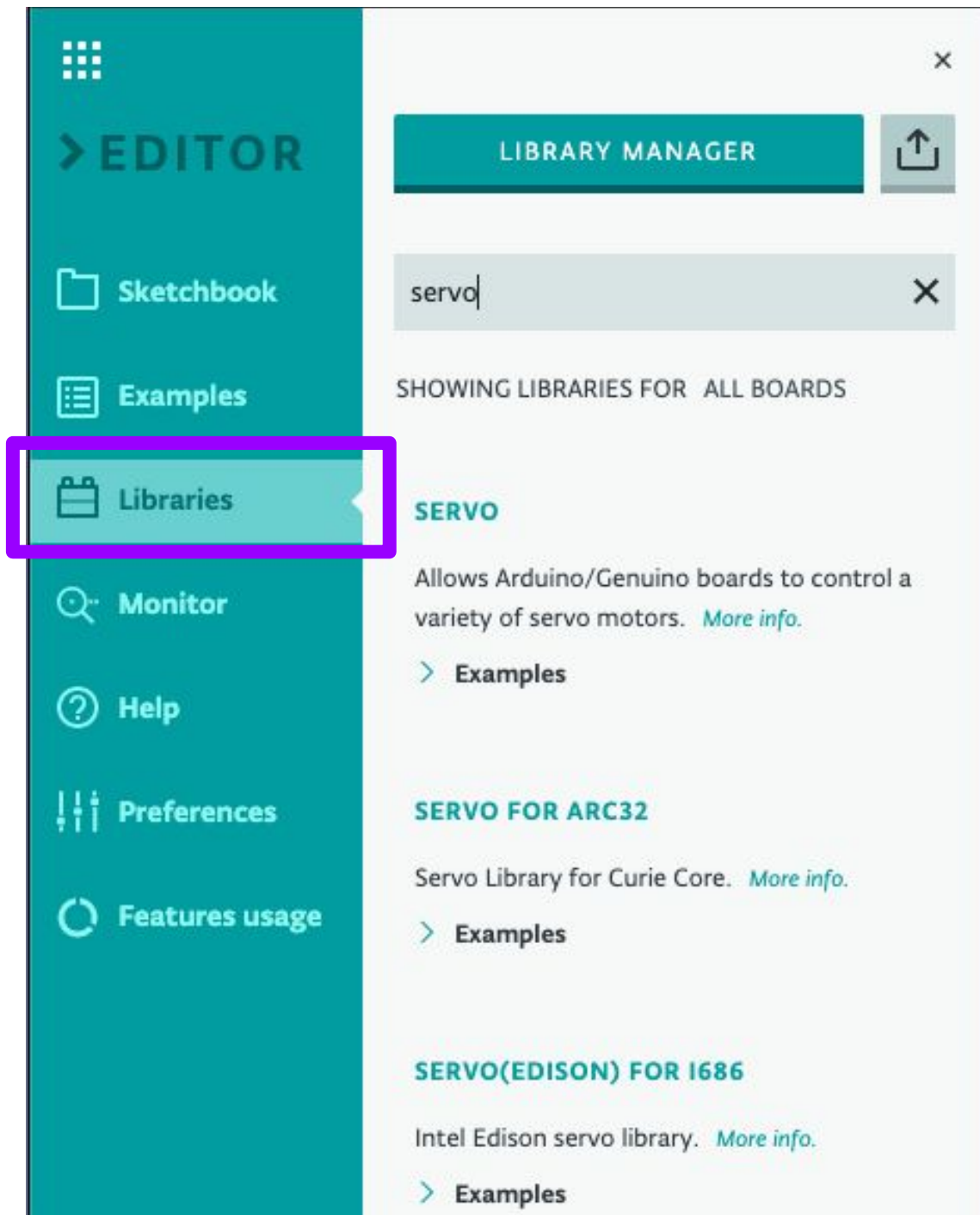
Libraries already exist
for many
components



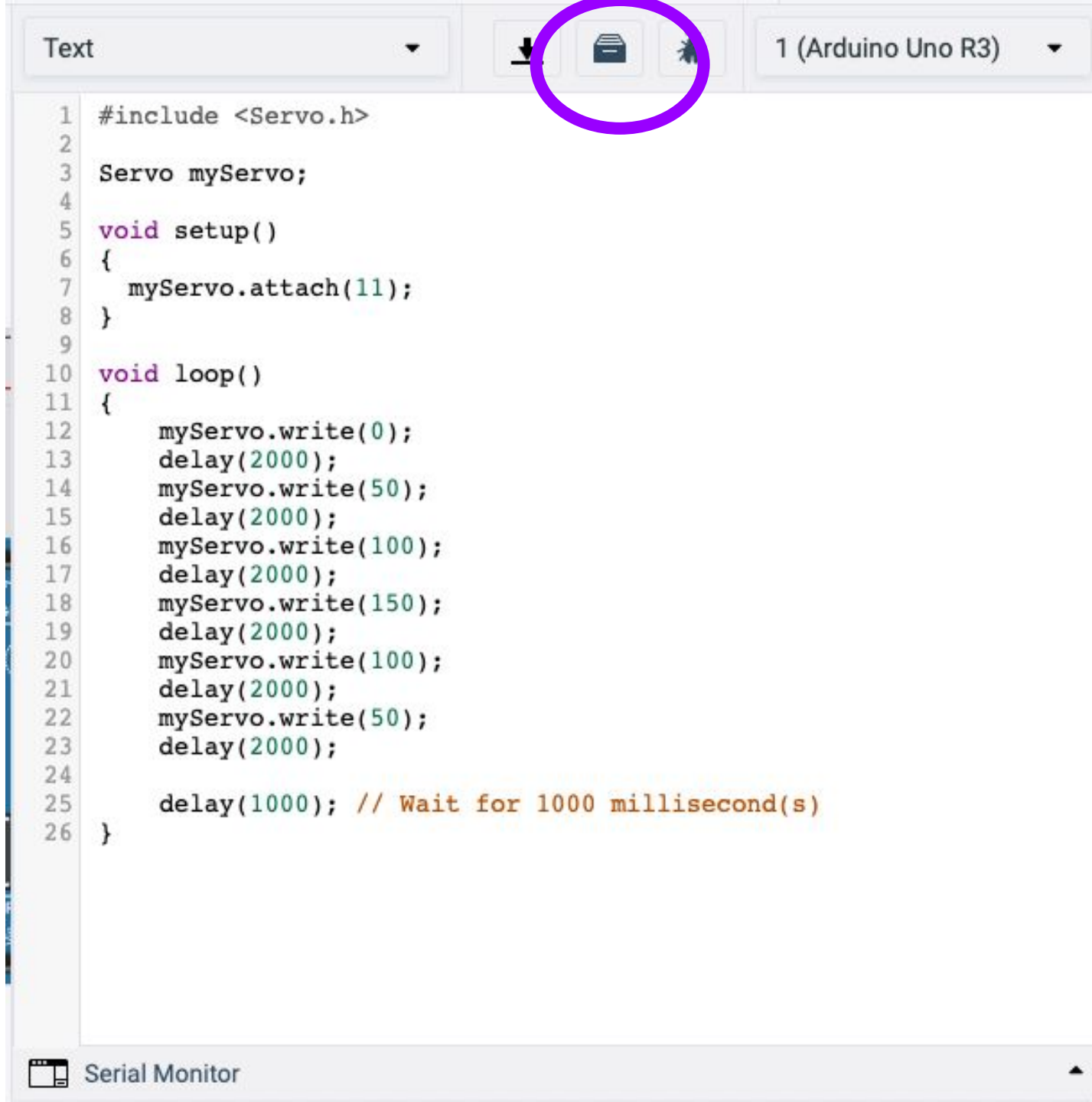
Libraries on Arduino IDE



Libraries on Arduino Create





Libraries on TinkerCad




Libraries on TinkerCad

Text



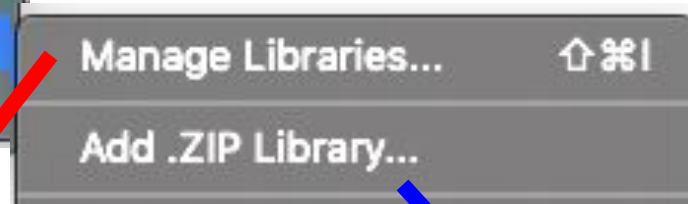
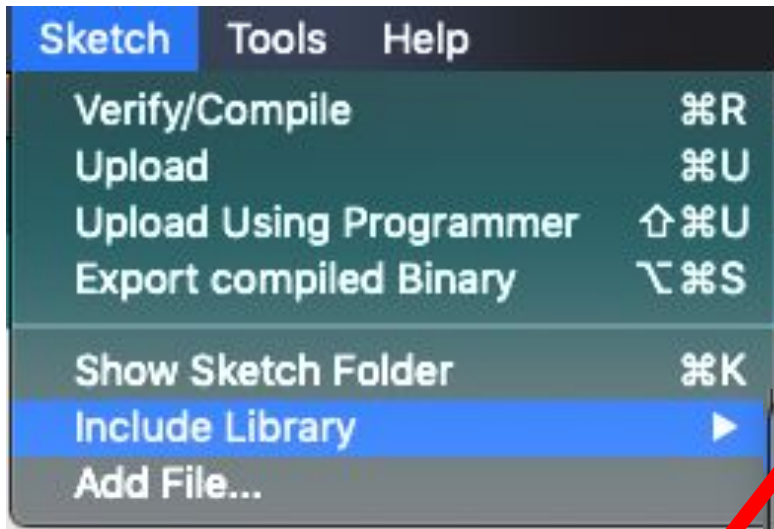




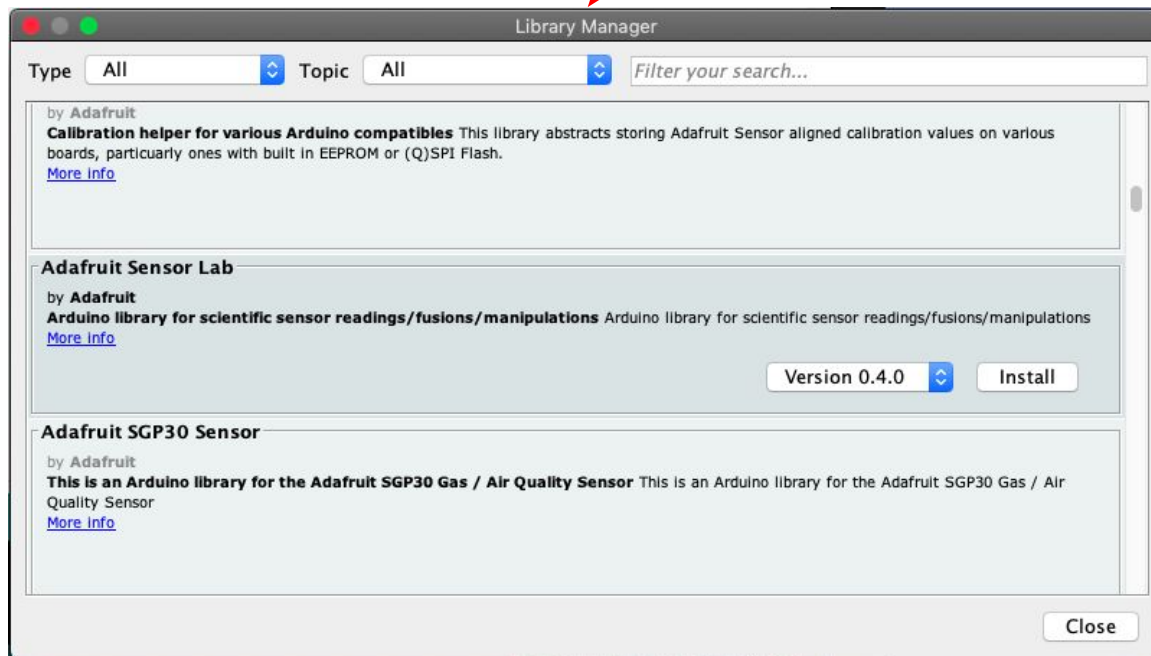
1 (Arduino Uno R3)

Include	EEPROM	Reading and writing to "permanent" storage	↗
Include	IRremote	Library to decode IR sensors	↗
Include	LiquidCrystal	Controlling liquid crystal displays (LCDs)	↗
Include	Keypad	Allows reading keypad button pushes	↗
Include	NeoPixel	Controlling NeoPixel LEDs	↗
Include	Servo	Controlling servo motors	↗
Include	SoftwareSerial	Allow serial communication on other digital...	↗
Include	Wire	This library allows you to communicate with...	↗
Include	SD	The SD library allows for reading from and...	↗

Installing Other Libraries



Library Manager

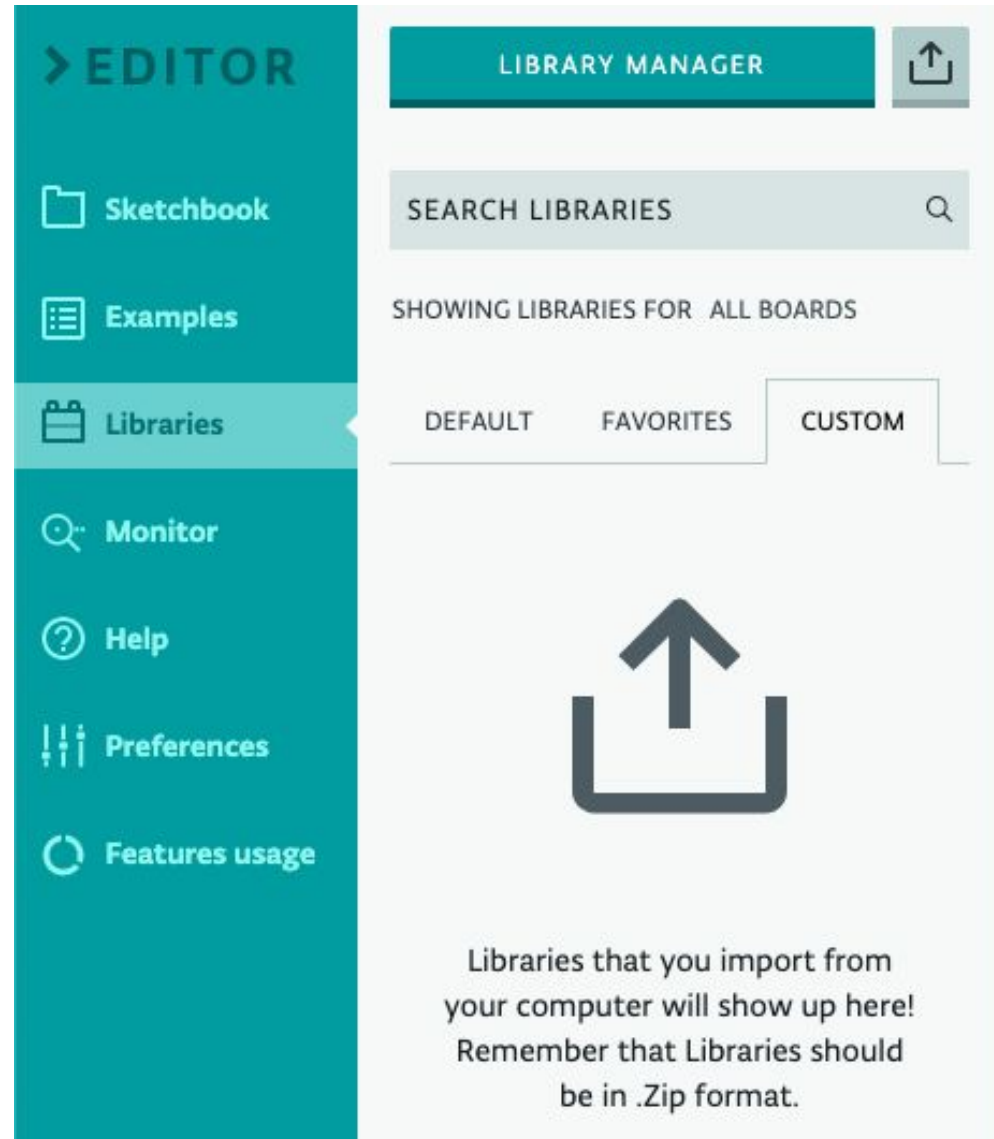


Manual Installation

Download zip file to your computer and then add it yourself

Installing Other Libraries

- Arduino Create
 - Install on your computer using Arduino IDE first, then you'll be able to upload them to Create



How do you use libraries?

So What's Happening Here?

```
7  #include <Servo.h>
8
9  const int servoPin = 11;
10
11  Servo myServo;
12
13  void setup()
14  {
15      myServo.attach(servoPin);
16  }
17
18  void loop()
19  {
20      myServo.write(0);
```

Tell the Arduino to include all of the information in the Servo Library with this sketch

Create an instance (or “copy”) of the **Servo** object and give it a name, like “myServo”

Use the functions that this instance comes with by using dot notation:
objectName . **function** (arguments)

So What's Happening Here?

```
7  #include <Servo.h>
8
9  const int servoPin = 11;
10
11  Servo myServo;
12
13  void setup()
14  {
15      myServo.attach(servoPin);
16  }
17
18  void loop()
19  {
20      myServo.write(0);
```

Among other things, this function contains:
`pinMode(servoPin, OUTPUT);`



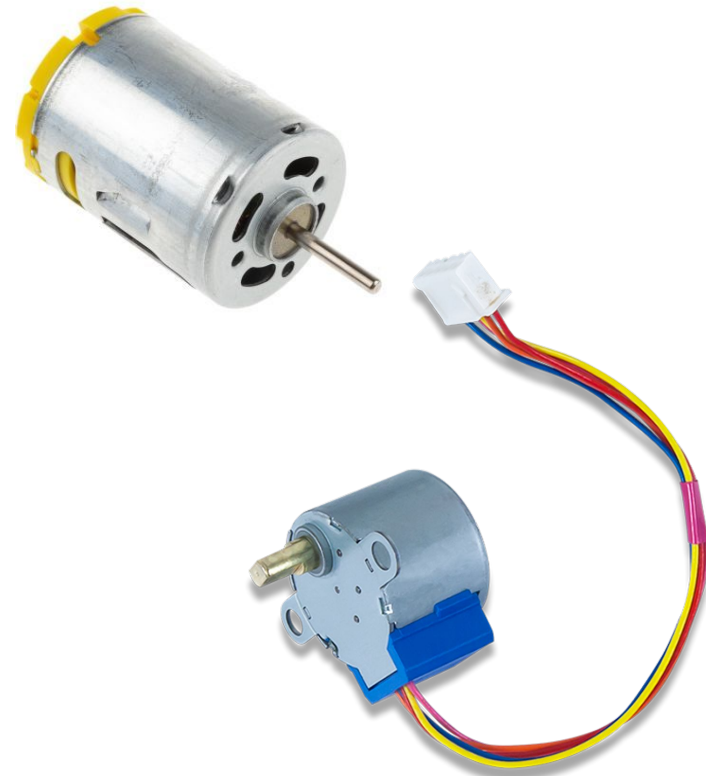
This function sets the position of the servo motor



Session 1 Servo Motors and Libraries



Session 1 Hobby Motors



Session 1 Stepper Motors

Summary

Share one thing your and your partner did

OR

One thing you learned

OR

How would you adopt/adapt this for the
classroom

Helpful Resources for Arduinos

- [Arduino.cc](http://arduino.cc)
- Examples on Arduino IDE/Arduino Create
- Arduino Tutorials on YouTube, AdaFruit
[Jeremy Blum's Arduino Tutorial Series](#)
- Google for Arduino Help
 - SparkFun has lots of resources & [tutorials](#)
- zphysics.org - Robotics Tutorial
- [TinkerCad](#) - Arduino Simulations

Where to Buy This Stuff

- AdaFruit
- SparkFun
- MyDuino
- [Amazon](#)/Ebay
- **STORES:**
 - TinkerSphere
(Allen St)
 - Microcenter
(Brooklyn, Yonkers)

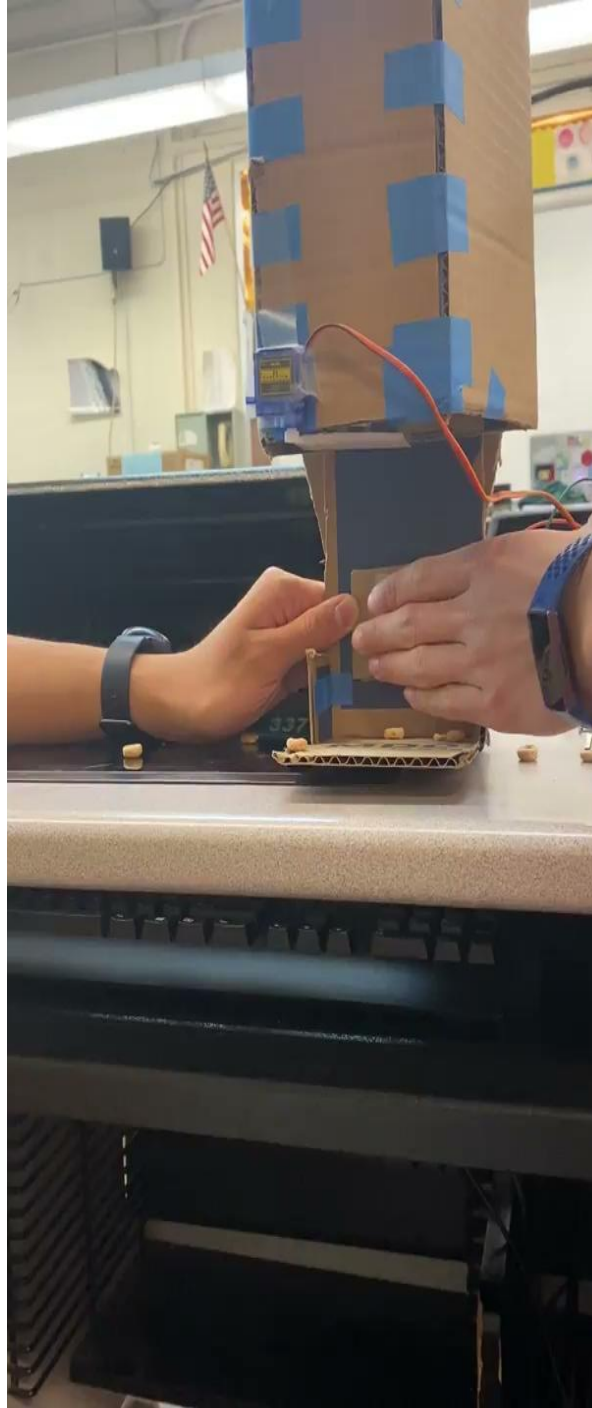


Oh the Possibilities!

**What
would
YOU
DO**



Project Example: Contactless Cereal Dispenser



Project Example: Robotic Hand

