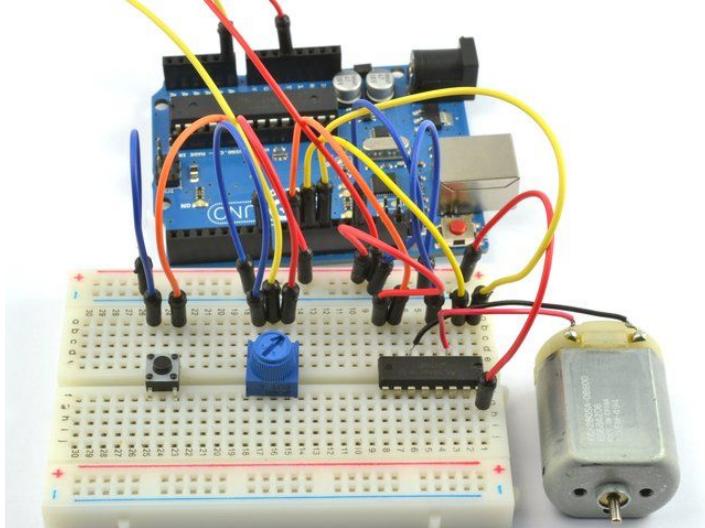
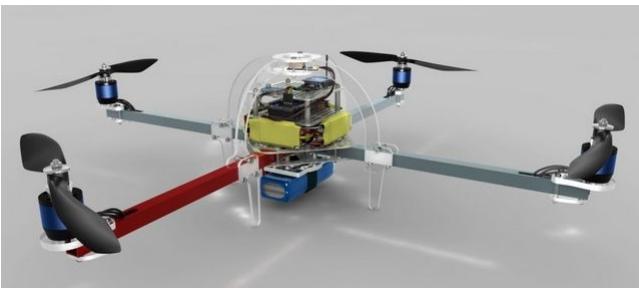


Do Now

What is the role of a magnet in a motor?

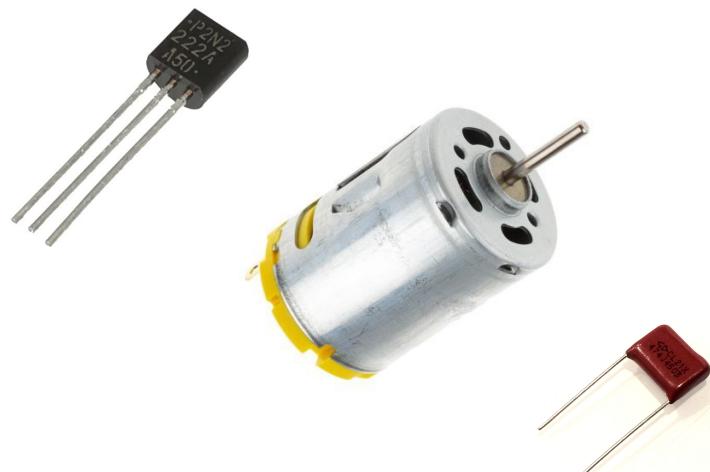


Intro to Digital Motors



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

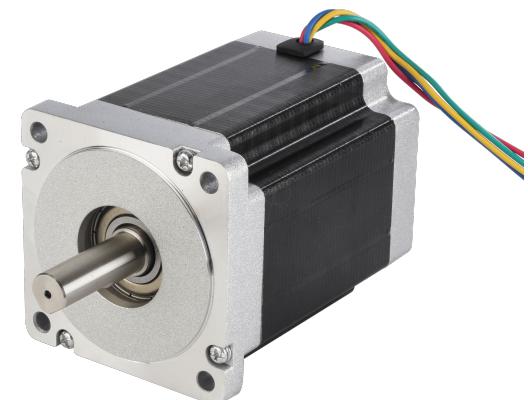
- Devices that convert electrical energy into rotational kinetic energy



Basic DC
Motors



Servo
Motors



Stepper
Motors

Standard DC Hobby Motors

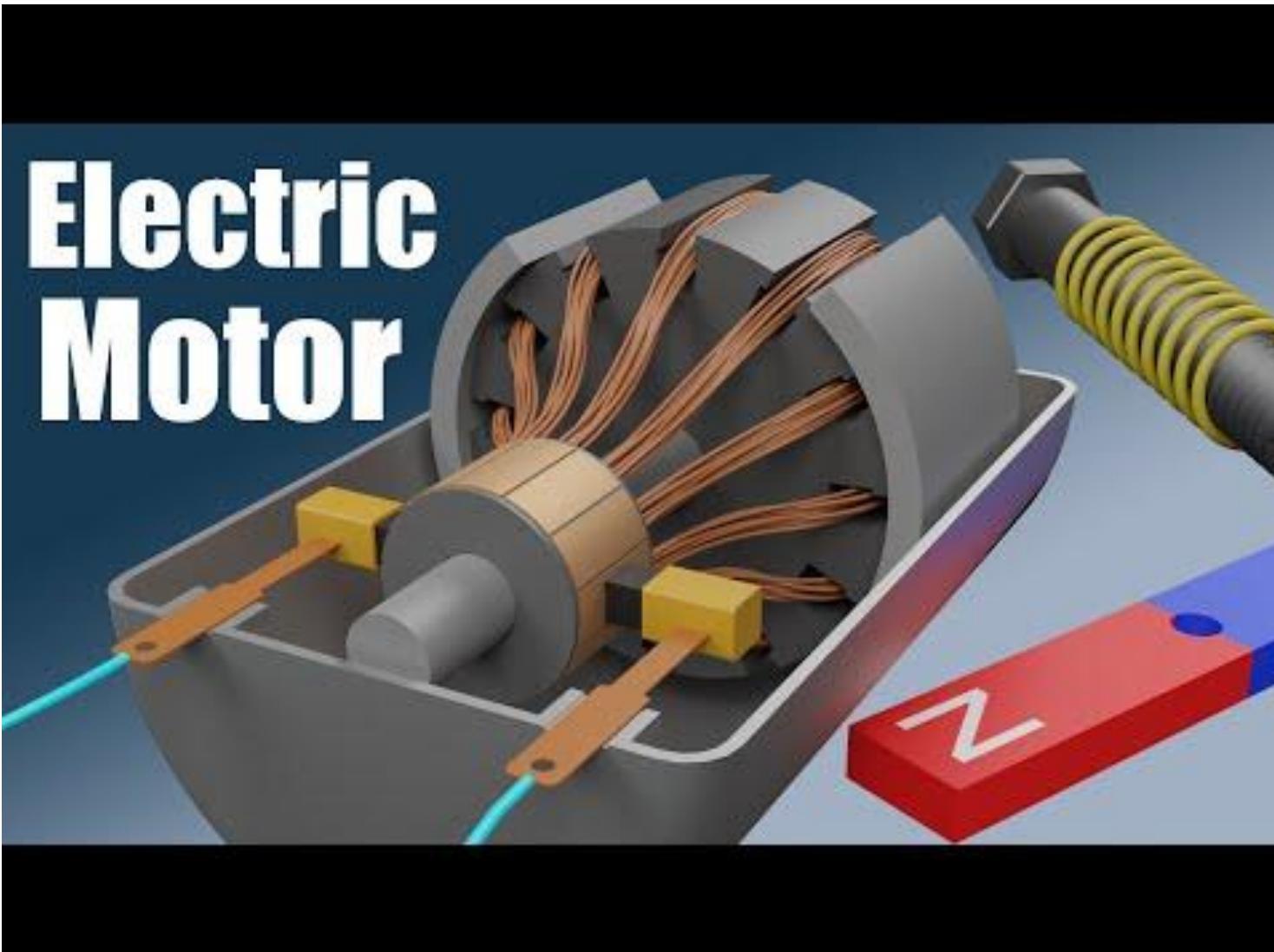
- Can spin CW or CCW
- Crude: either on or off
 - Speed determined by voltage
 - Direction determined by polarity of current
- Unlimited rotation in one direction
- What are some devices that could use hobby motors?



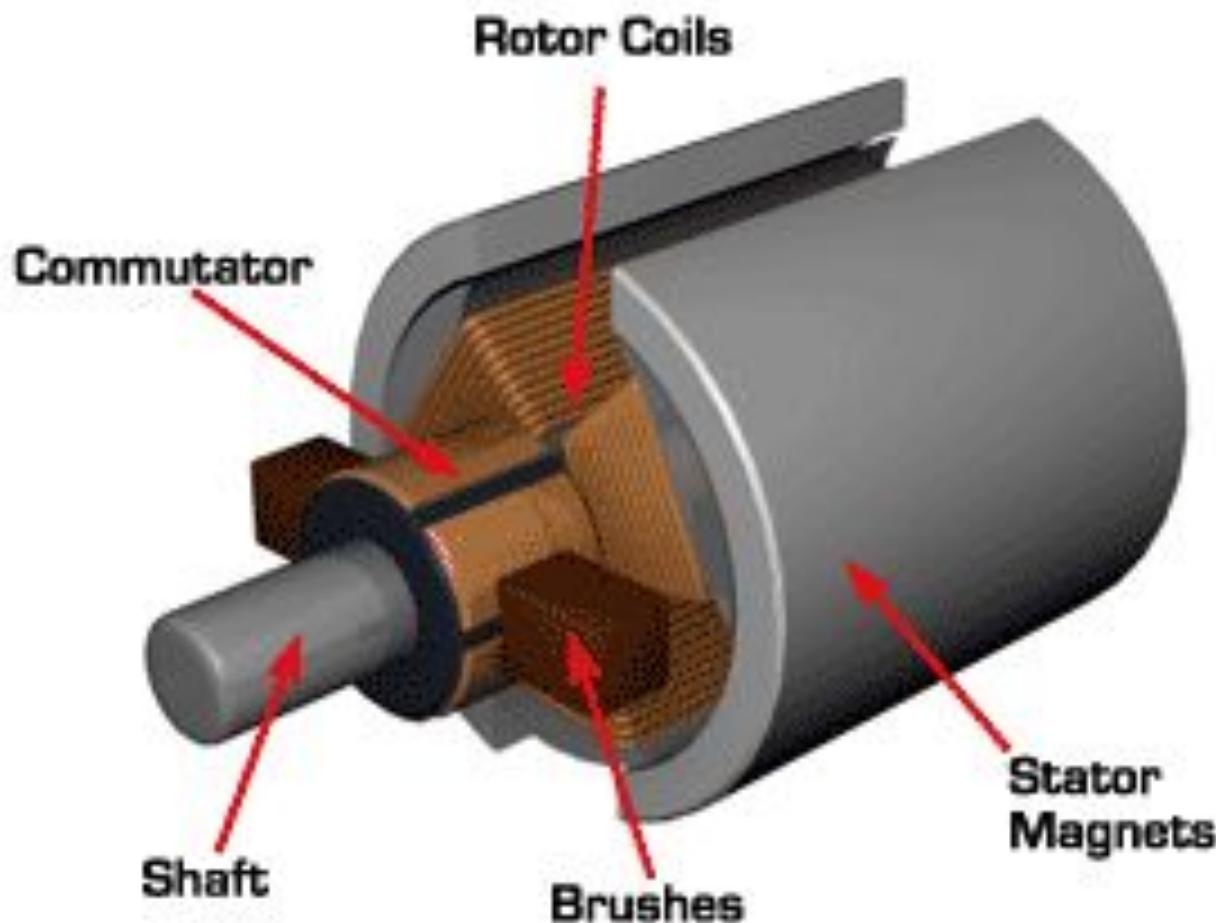
Photo by ElectroPeak

gearbox motor

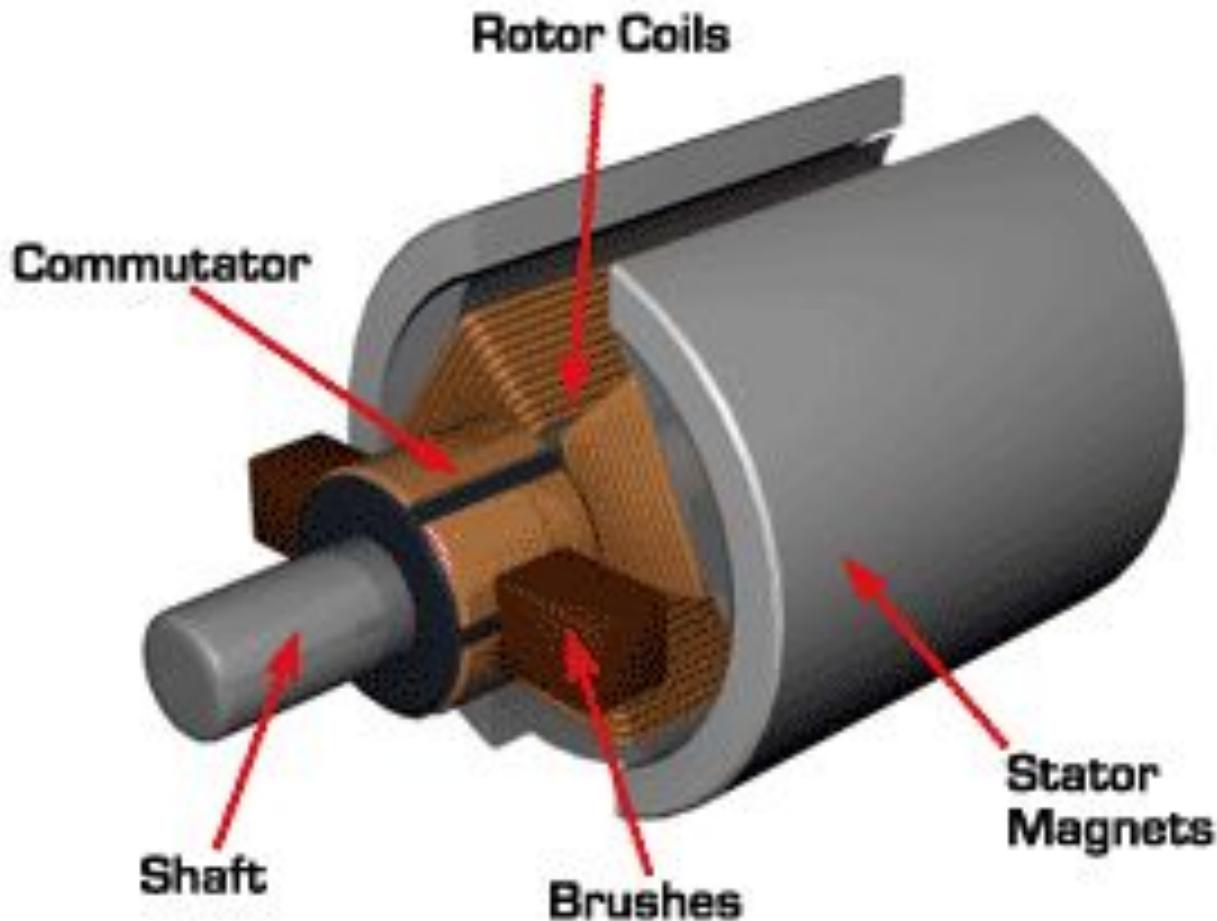
Basic Motors Video



What's Inside a DC Hobby Motor?



What's Inside a DC Hobby Motor?



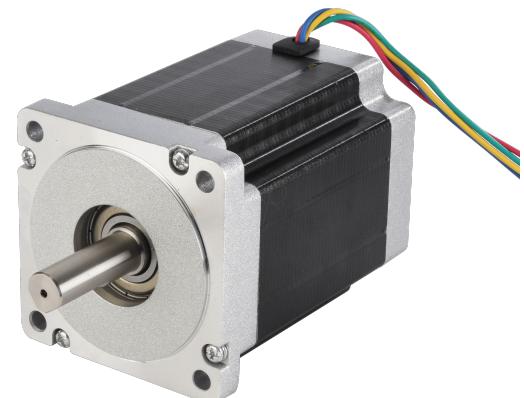
How is this design an improvement on the basic motors we built yesterday?

Motors

- Devices that convert electrical energy into rotational kinetic energy



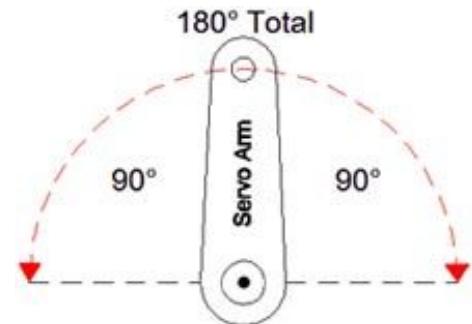
Basic DC
Motors



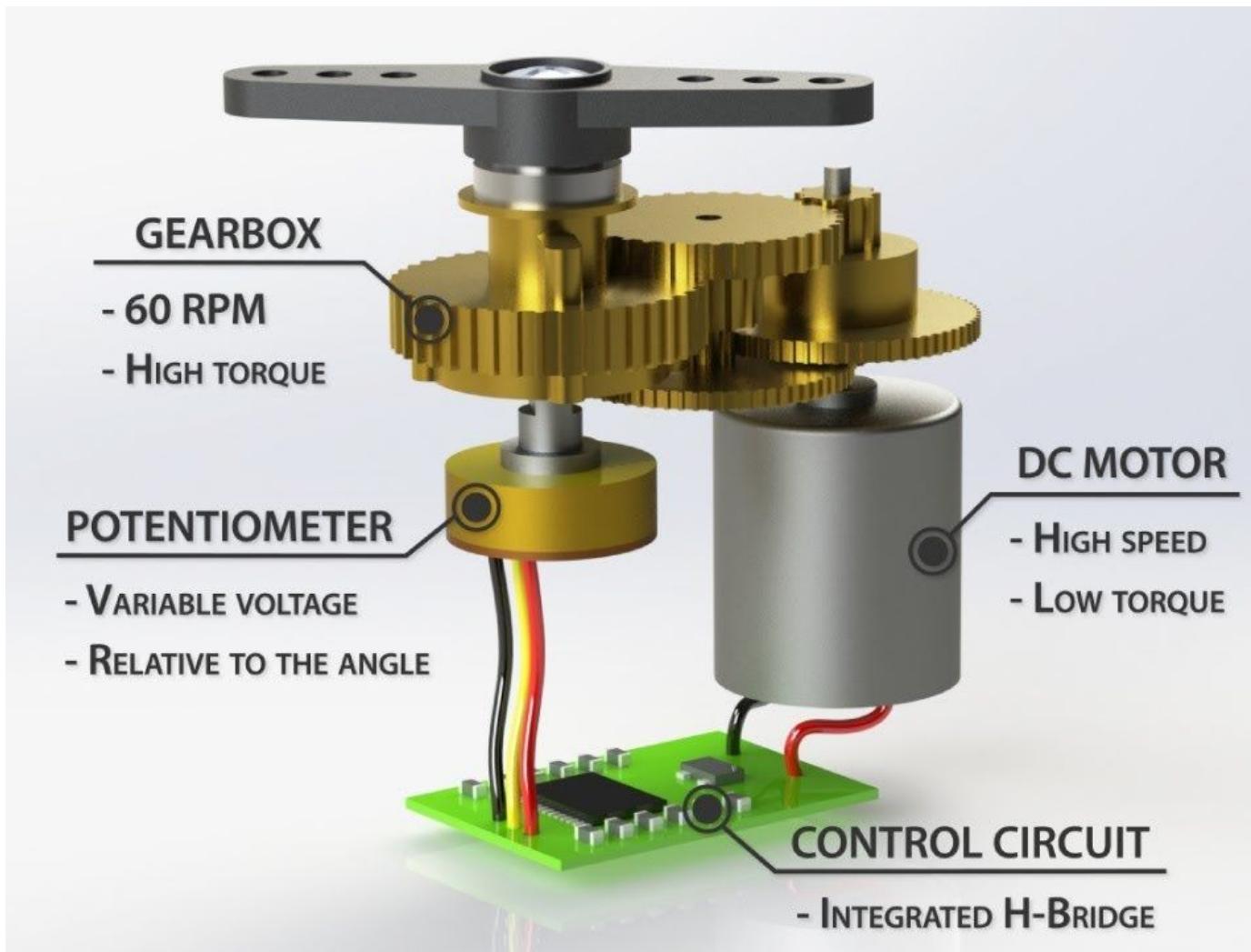
Stepper
Motors

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

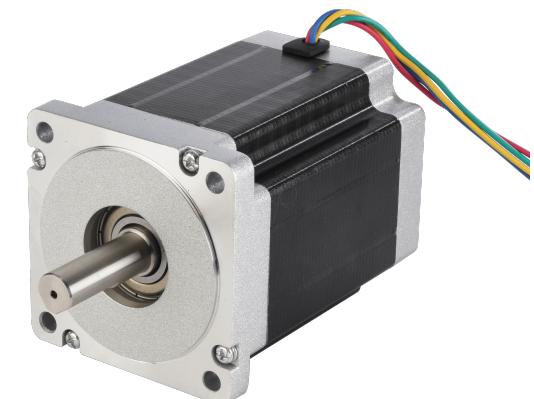
- Devices that convert electrical energy into rotational kinetic energy



Basic DC
Motors



Servo
Motors



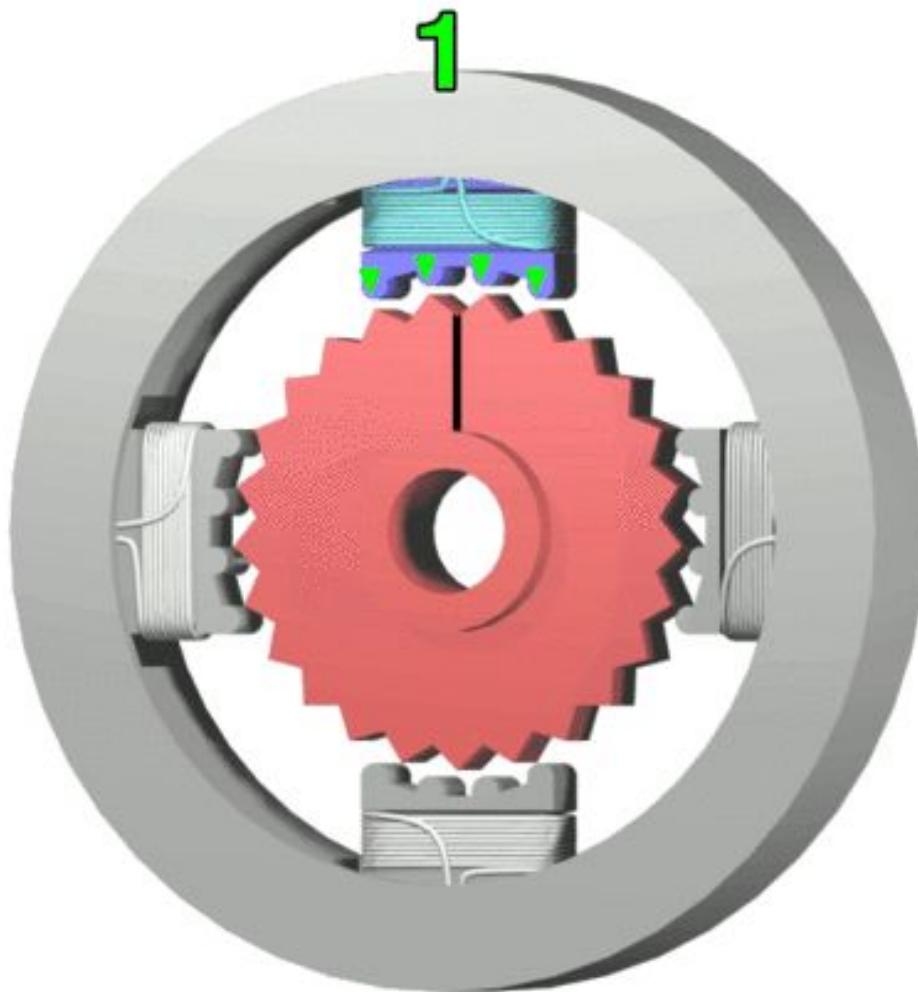
Stepper
Motors

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?



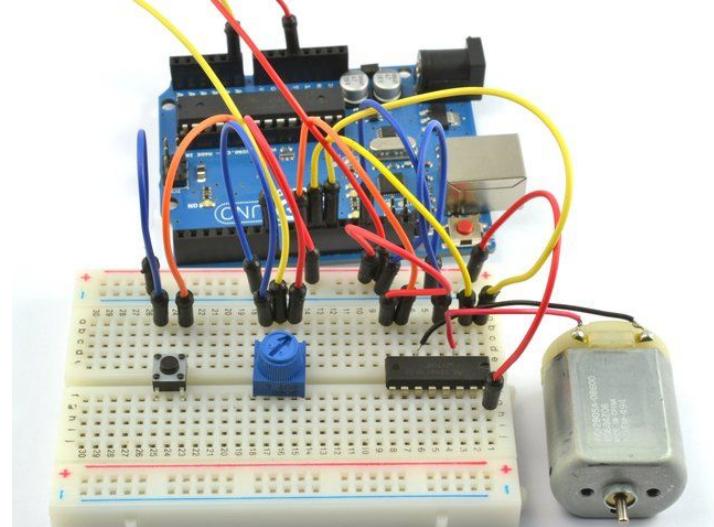
What's Inside a Stepper Motor?



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



DC Hobby Motors



Motors

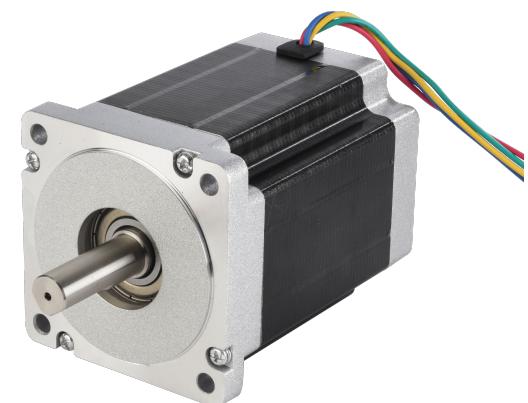
- Devices that convert electrical energy into rotational kinetic energy



Basic DC
Motors



Servo
Motors



Stepper
Motors

Standard DC Hobby Motors

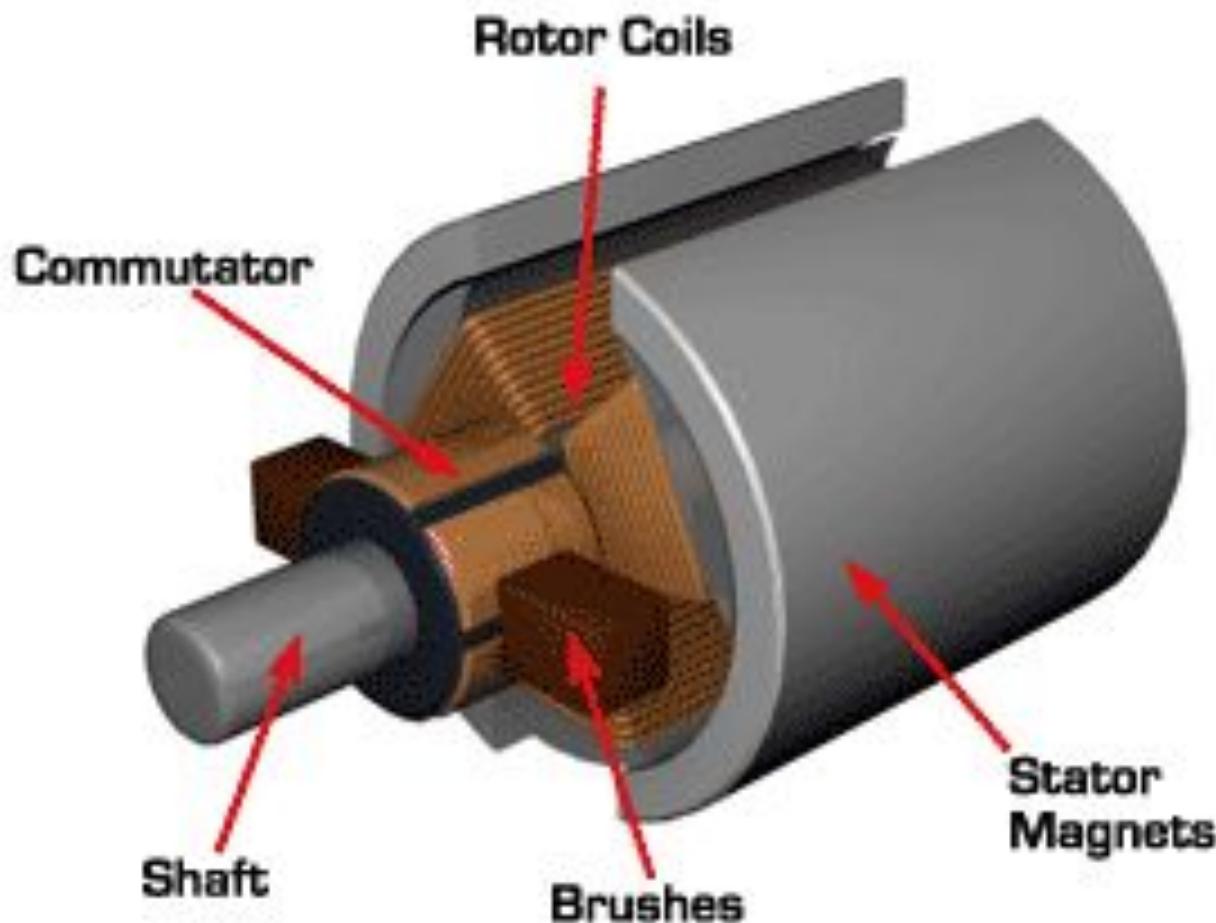
- Can spin CW or CCW
- Crude: either on or off
 - Speed determined by voltage
 - Direction determined by polarity of current
- Unlimited rotation in one direction
- What are some devices that could use hobby motors?



Photo by ElectroPeak

gearbox motor

What's Inside a DC Hobby Motor?

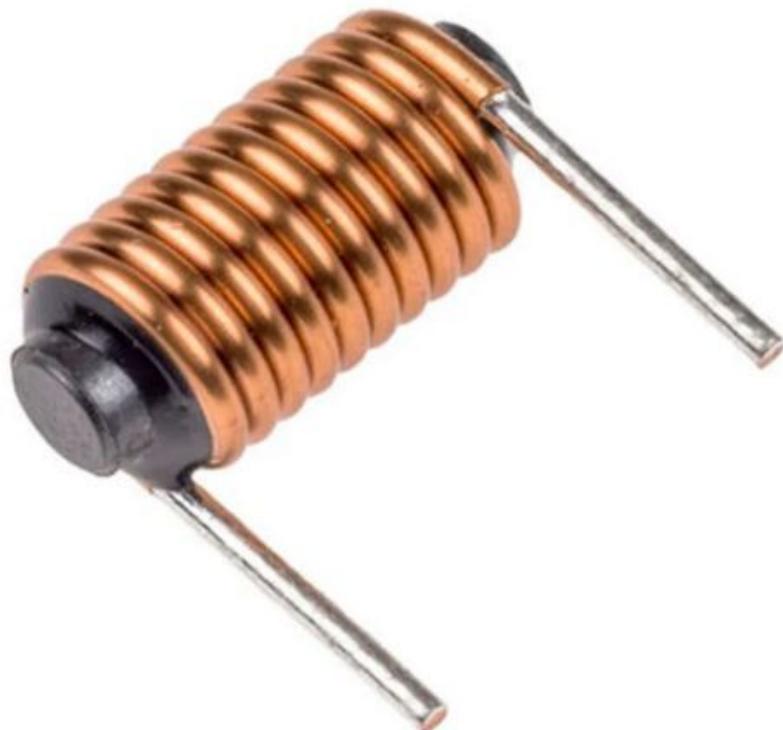


Inductors



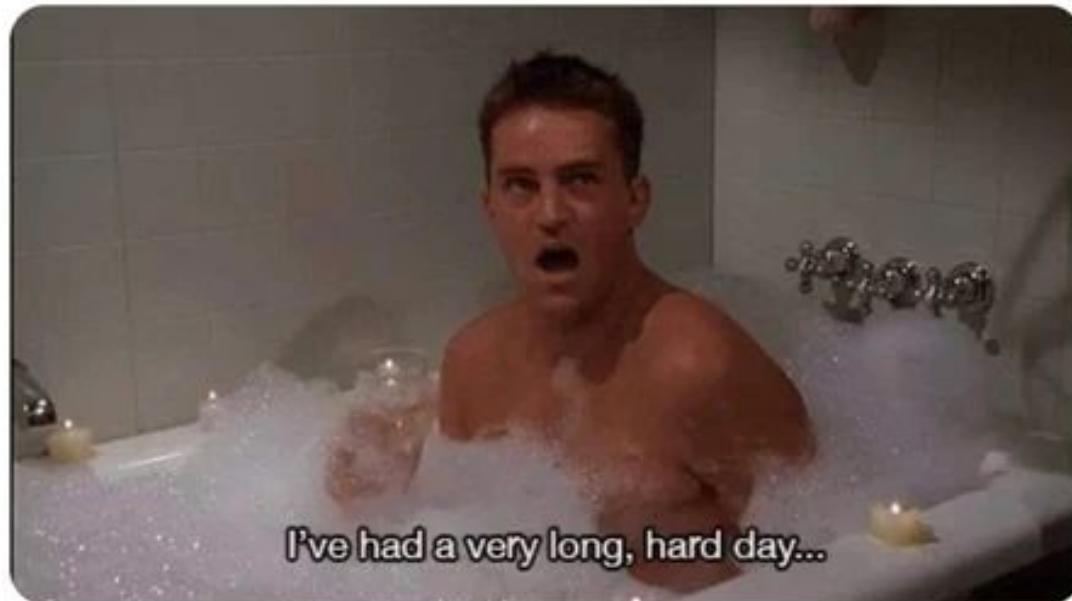
Prevent rapid changes in current flow

Made from wire coils (often around an iron core)



Work Time!

- Work with your partner to begin the Hobby Motors assignment (posted on Google Classroom)
me after working for 15 minutes



Summary

- How were you able to slow down the motor?



Hobby Motors Day



Work Time!

- Work with your partner to complete the Hobby Motors assignment (posted on Google Classroom)



Summary

- One thing you accomplished

OR

- One bug you overcame

OR

- One idea you have for how to use stepper motors in the future





Servo Motors

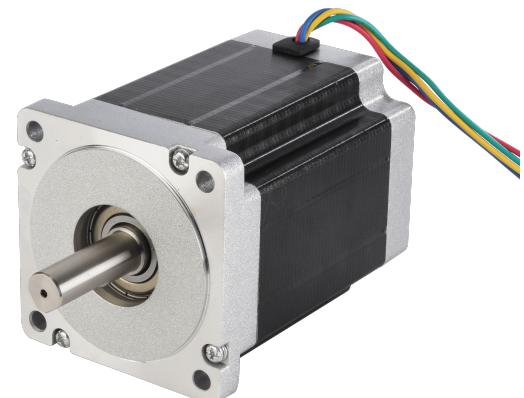


Motors

- Devices that convert electrical energy into rotational kinetic energy



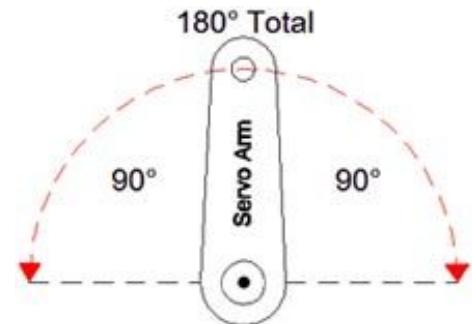
Basic DC
Motors



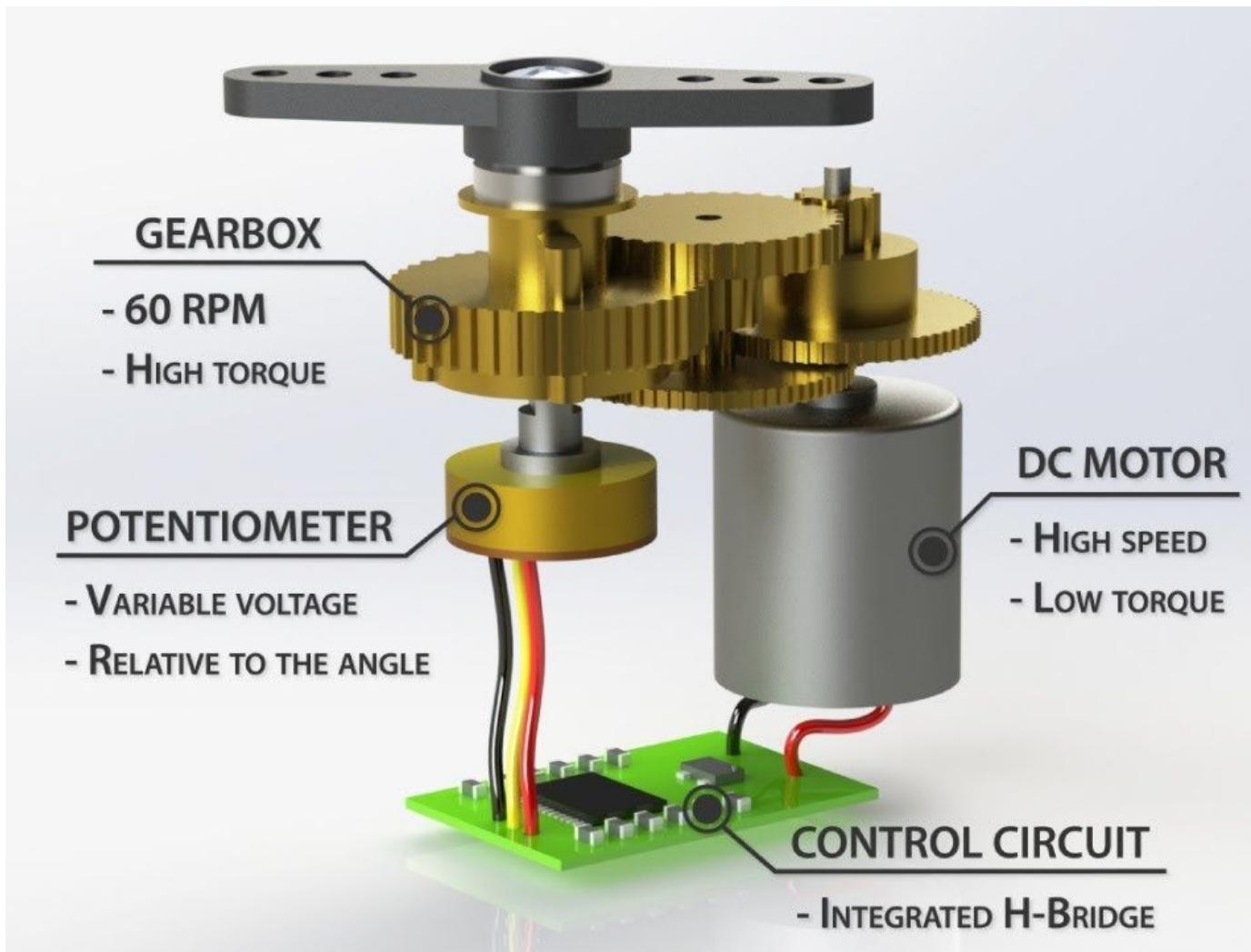
Stepper
Motors

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?

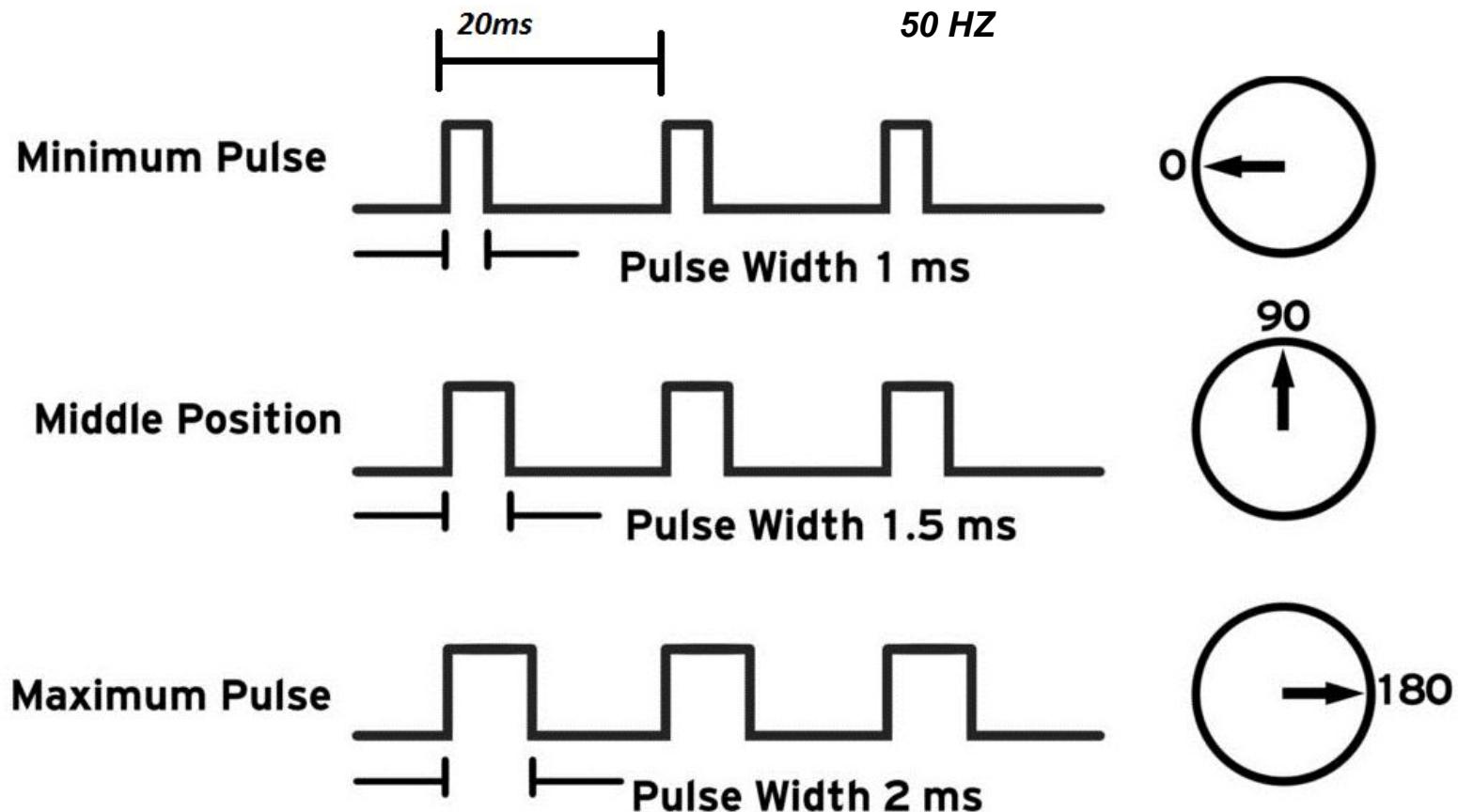


What's Inside a Servo?



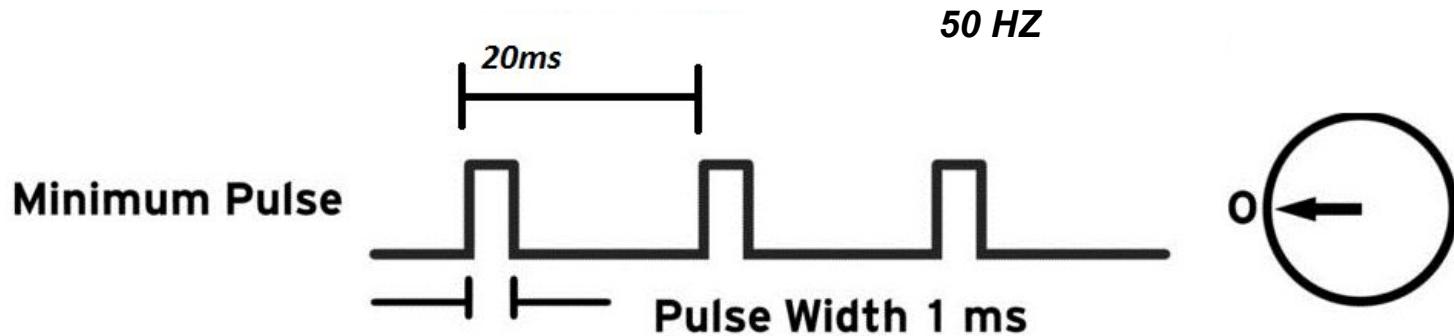
Controlling a Servo

- 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 code...



```
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?

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

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
- A template for creating copies (“instances”)



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\(\)](#)

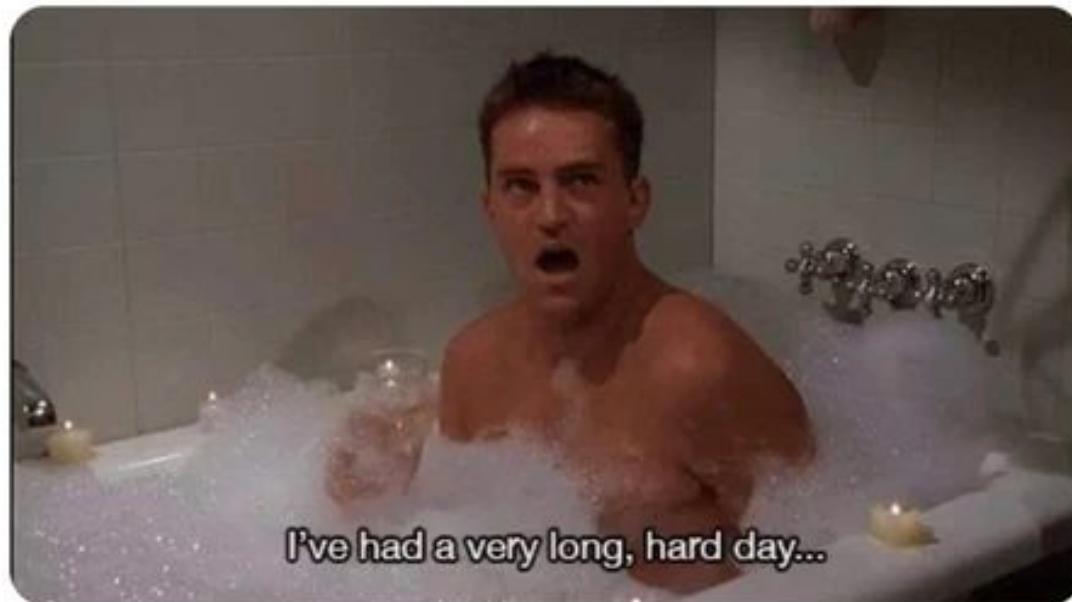
Examples

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

Work Time!

- Work with your partner to begin the Servo Motors assignment (posted on Google Classroom)

me after working for 15 minutes



I've had a very long, hard day...

Day 1 Summary

- Take a look at the [Arduino Servo Library](#).
 - Which two pins (on an Arduino Uno board) are affected by using the Servo Library? In what way are they affected?
 - What does the **write()** function do? What are the lowest and highest numbers that can be used as arguments?
- We are using a *standard* servo, but there is another popular kind called **continuous**. Look it up. How are continuous servos different from standard servos? (Write at least three sentences.)
- What error did you receive when you deleted line 7? Why?

```
'Servo' does not name a type; did you mean 'Serial'?
```

[Copy error messages](#)

```
exit status 1
```

```
'Servo' does not name a type; did you mean 'Serial'?
```

Servo Motors Day



Work Time!

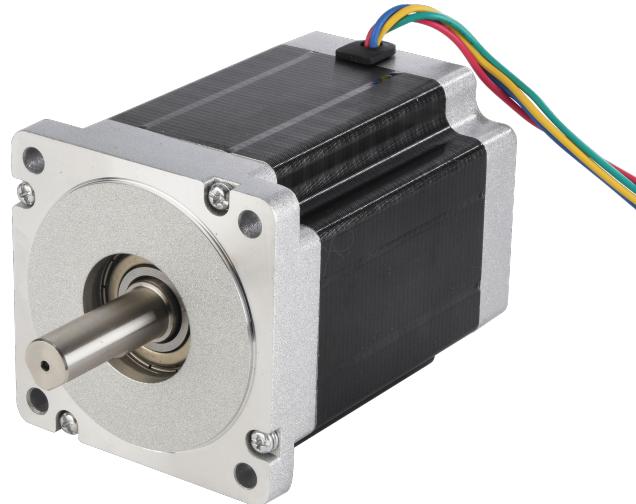
- Work with your partner to complete the Servo Motors assignment (posted on Google Classroom)



Summary

- One thing you accomplished
 - OR
- One bug you overcame
 - OR
- One idea you have for how to use stepper motors in the future





Stepper Motors



Motors

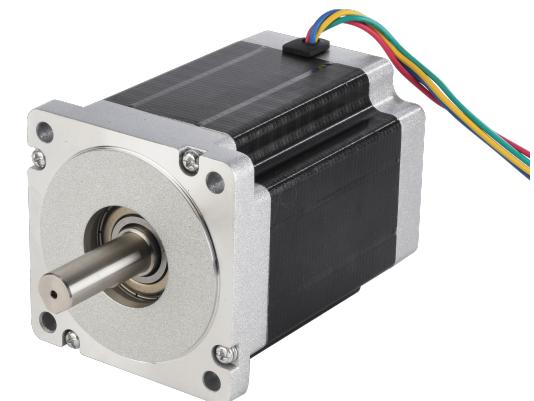
- Devices that convert electrical energy into rotational kinetic energy



Basic DC
Motors



Servo
Motors



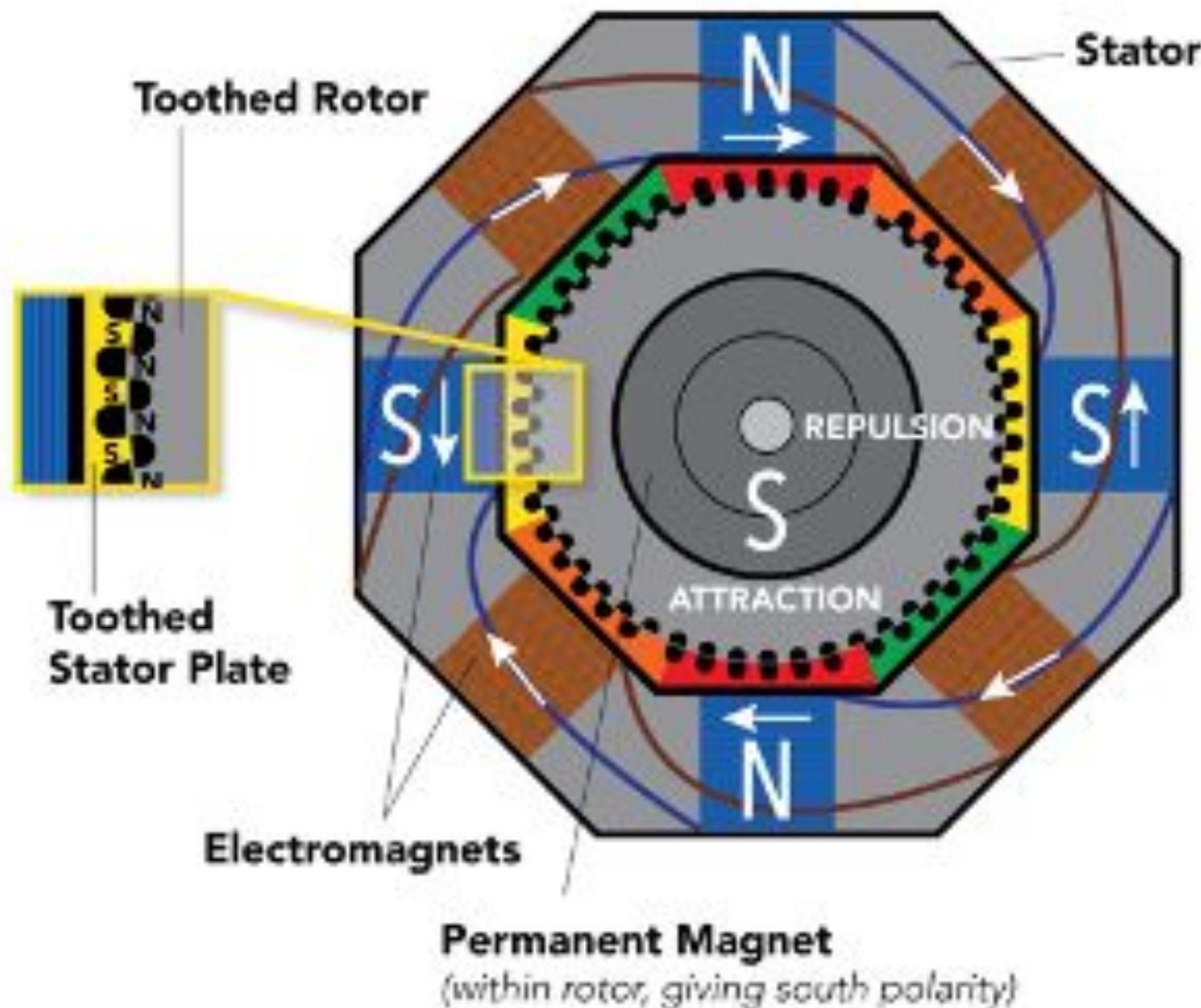
Stepper
Motors

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?

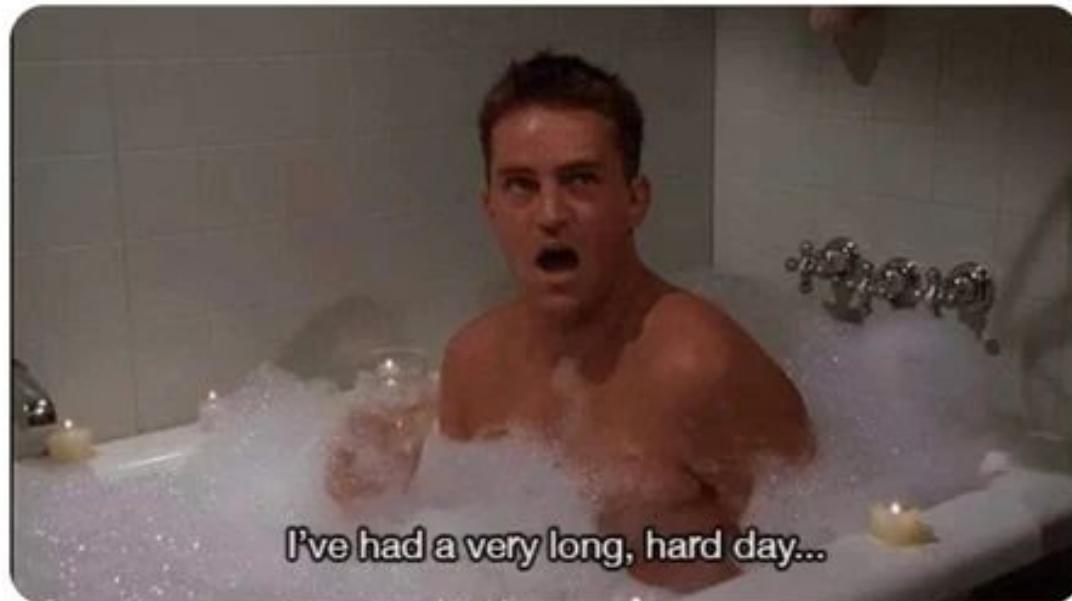


What's Inside a Stepper Motor?



Work Time!

- Work with your partner to begin the Stepper Motors assignment (posted on Google Classroom)
me after working for 15 minutes



Day 1 Summary

- How can we make the stepper motor spin faster?
- What happens if you change the number of steps?
- What does the negative sign inside of `myStepper.step(-stepsPerRevolution);` do?
- How can we make the stepper motor rotate 90°?

Stepper Motors Day



Work Time!

- Work with your partner to complete the Stepper Motors assignment (posted on Google Classroom)



Summary

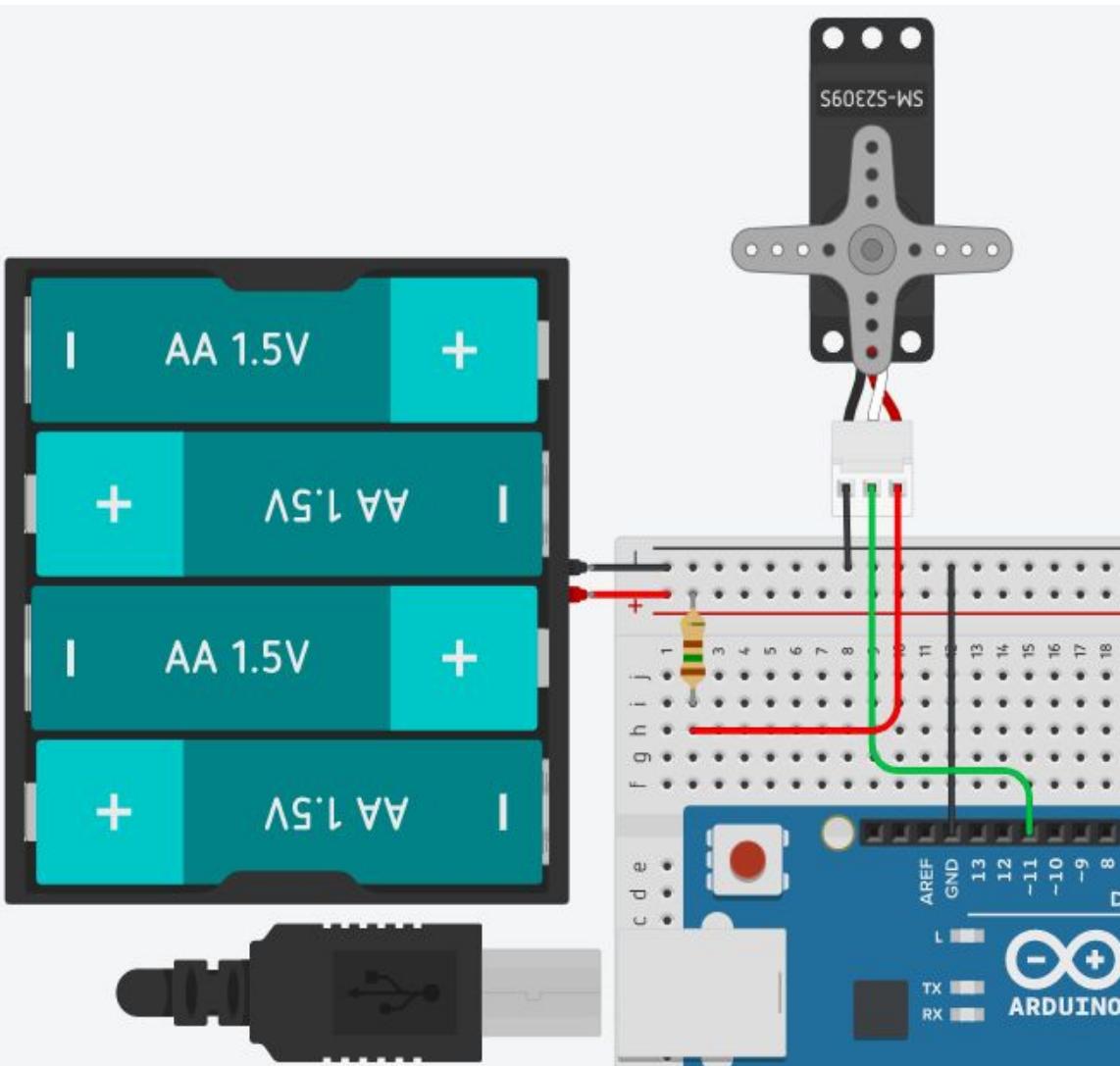
- One thing you accomplished
 - OR
- One bug you overcame
 - OR
- One idea you have for how to use stepper motors in the future



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.

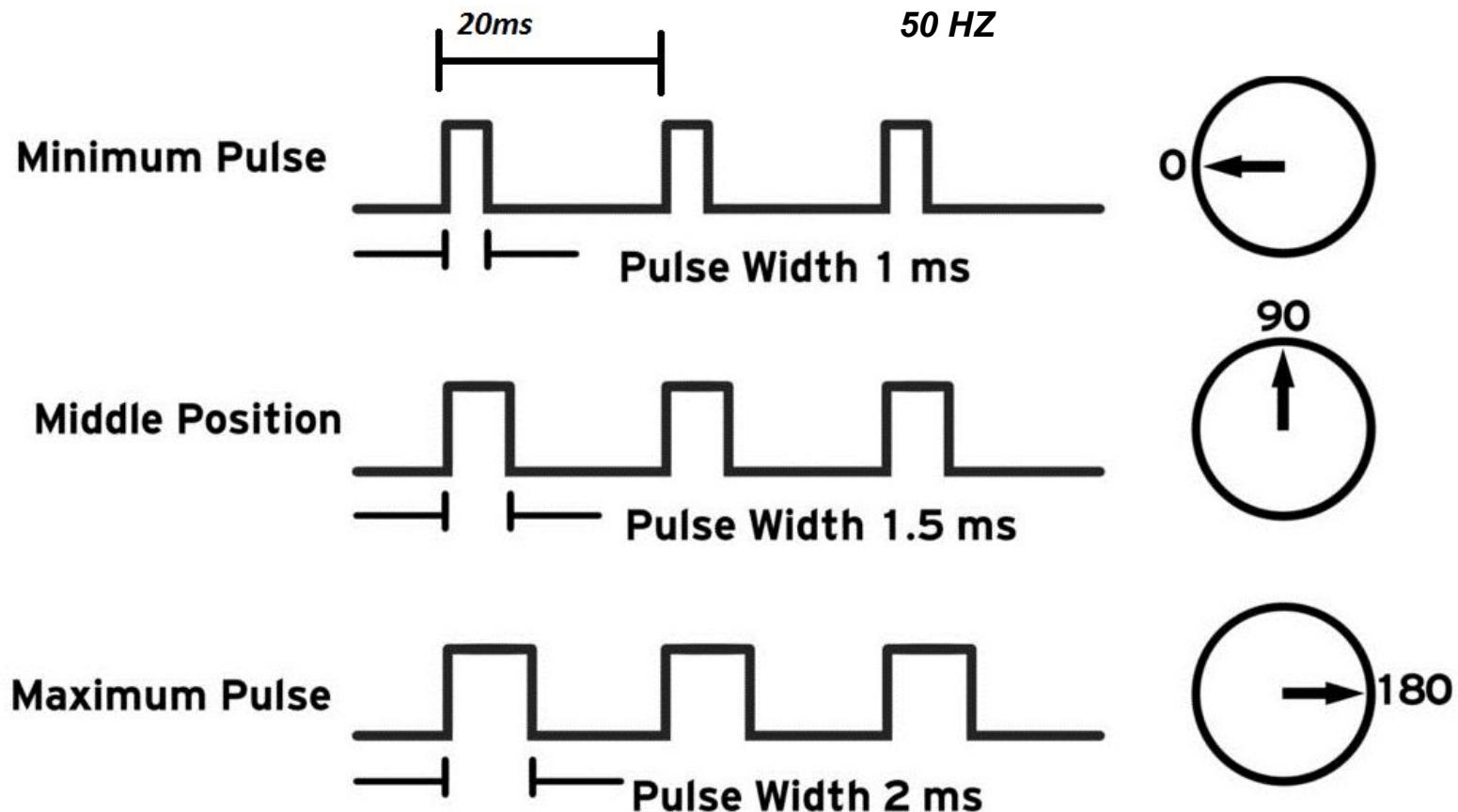
Otherwise, you will burn out your Arduino.

Session 1 Servo Motors and Libraries



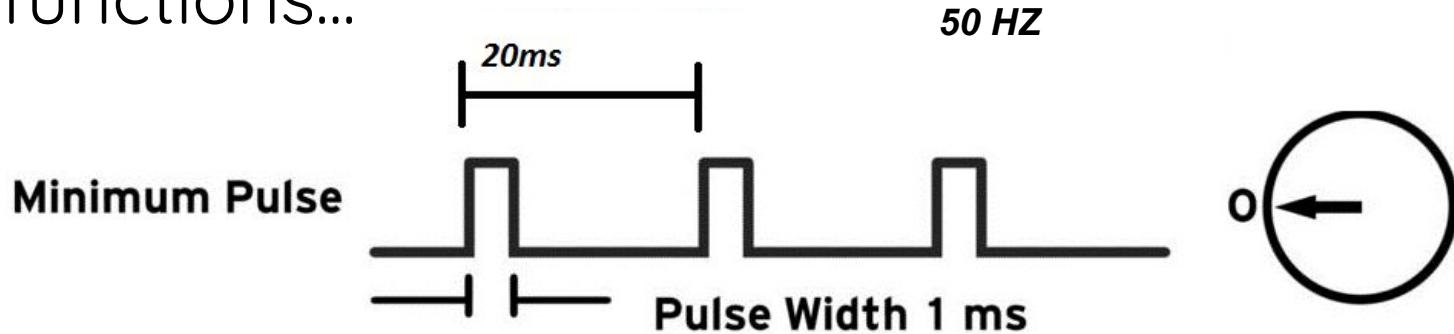
Controlling a Servo

- 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



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

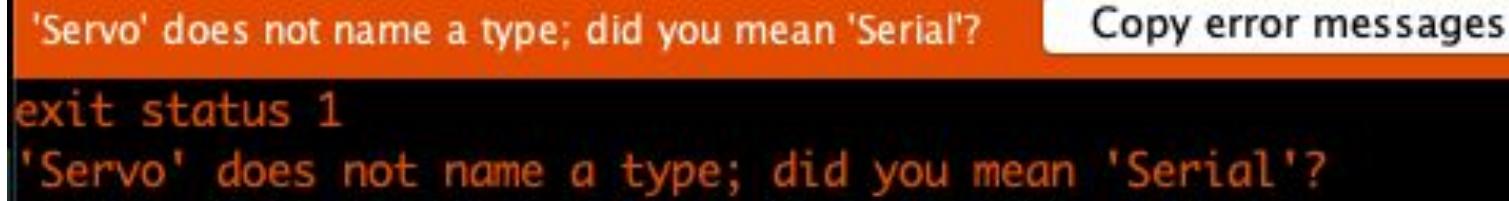
- [Knob](#): control the shaft of a servo motor by turning a potentiometer.
- [Sweep](#) : sweeps the shaft of a servo motor back and forth.

How do you use a Library?

`#include <LibraryName.h>`

Example: `#include <Servo.h>`

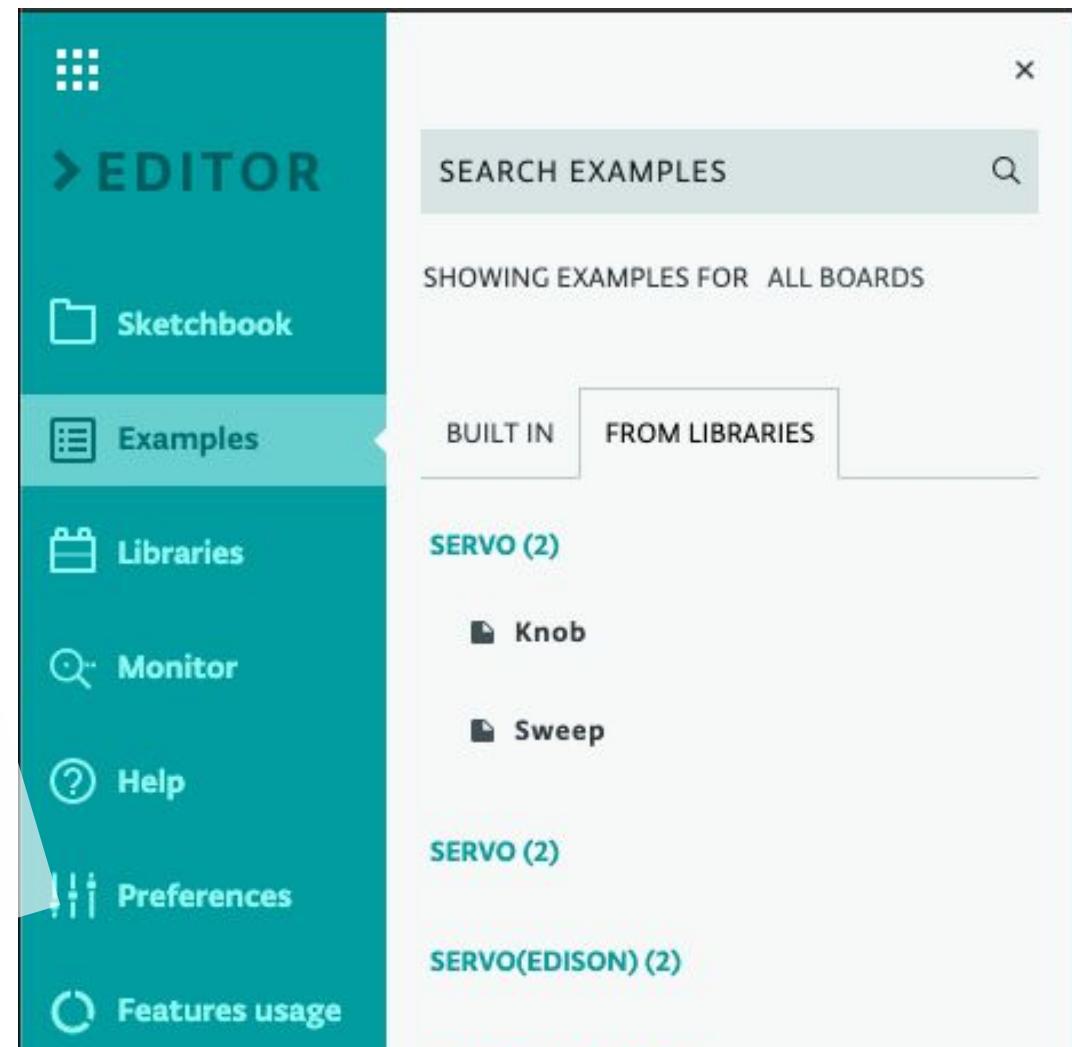
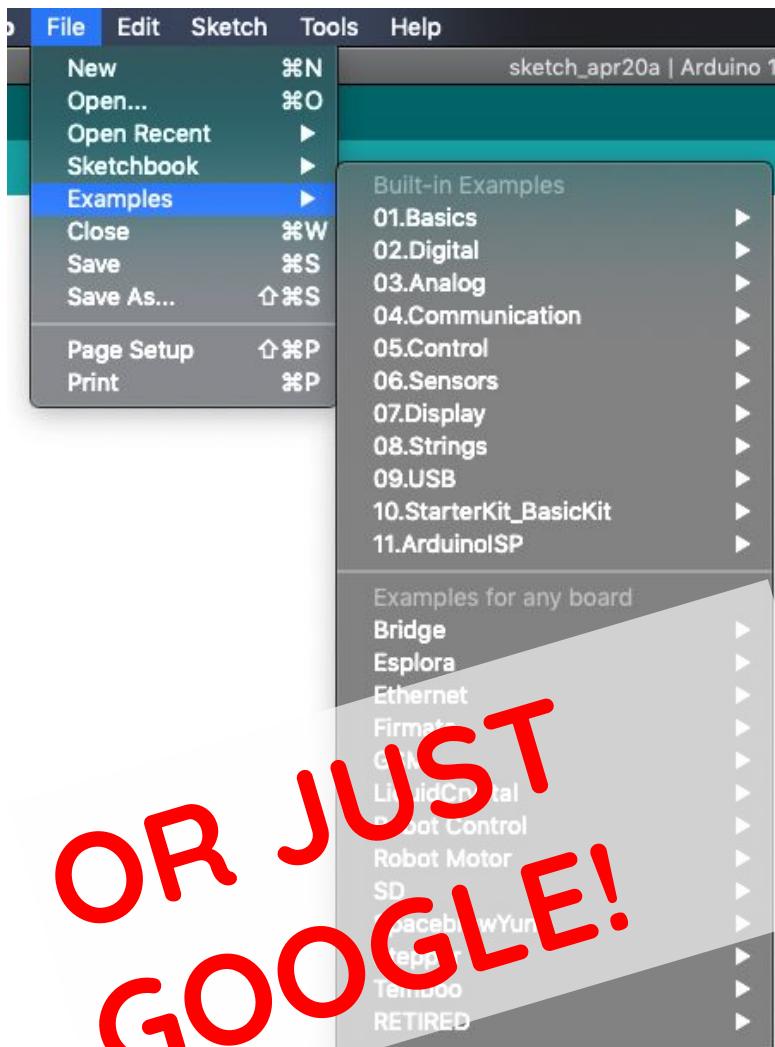
`Servo.h` is a **header file**. It only contains declarations so that you can refer to special objects, constants, or functions without throwing an error



A screenshot of a terminal window showing a compilation error. The error message is: "'Servo' does not name a type; did you mean 'Serial'?". Below the message, the text "Copy error messages" is visible. The terminal then shows the exit status "exit status 1" and repeats the error message: "'Servo' does not name a type; did you mean 'Serial'?".

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!

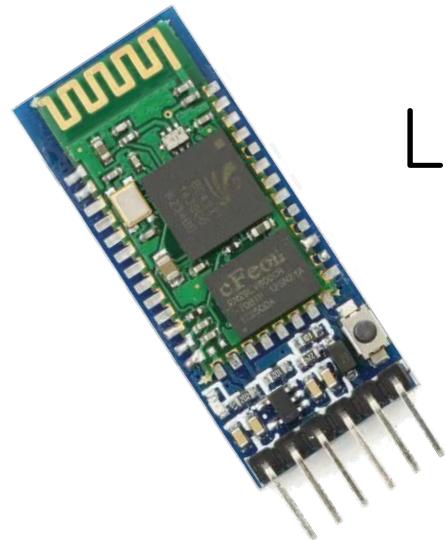
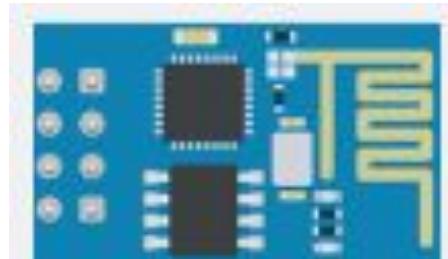
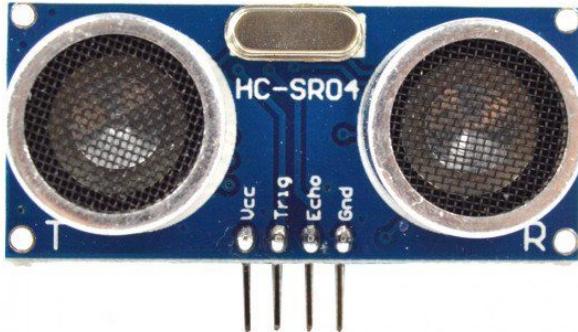
What about other libraries?

Common Arduino Libraries

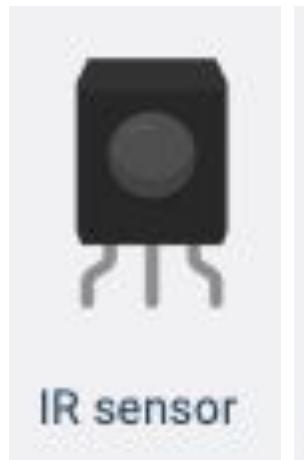
Documentation for many libraries is on Arduino.cc

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



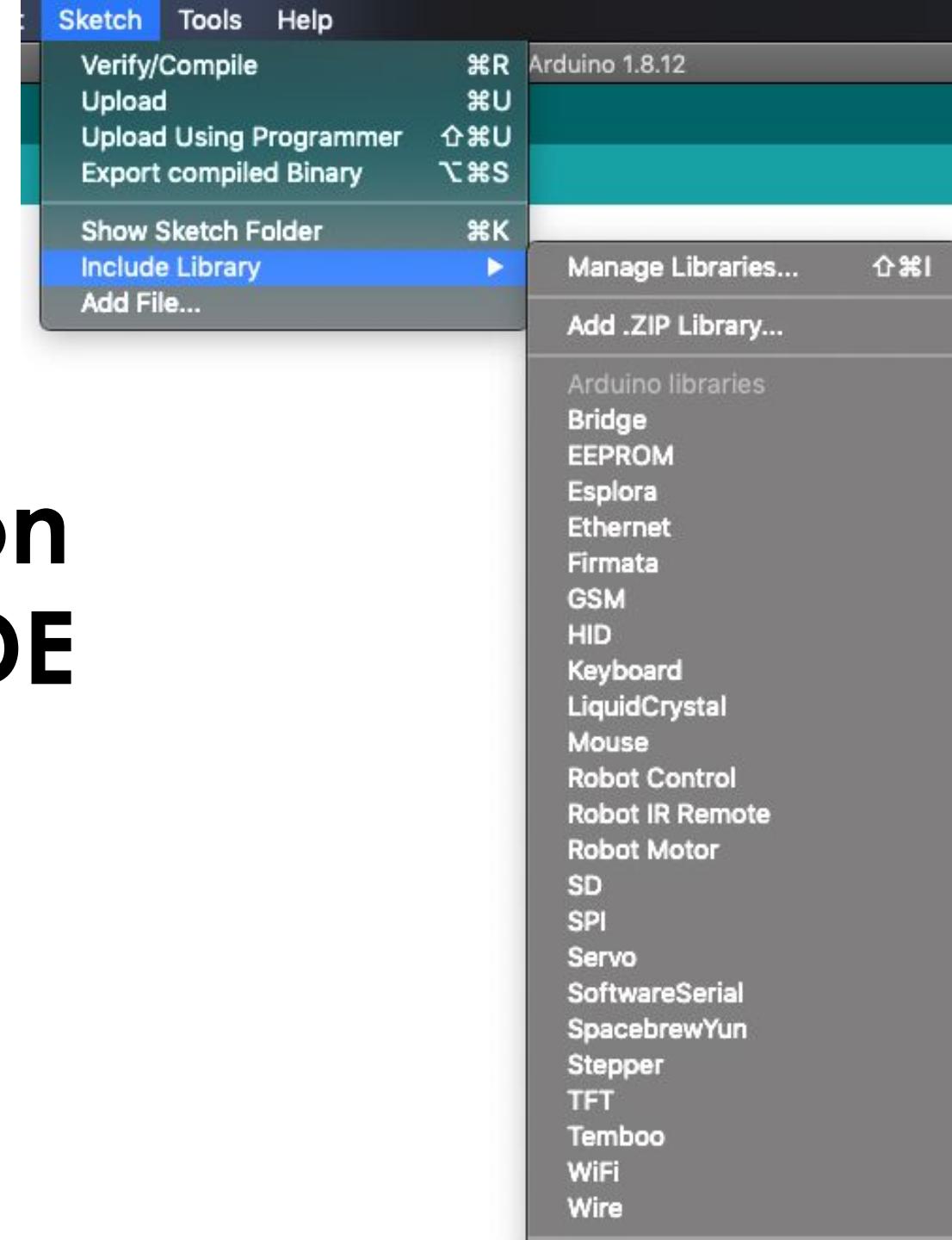
IR sensor



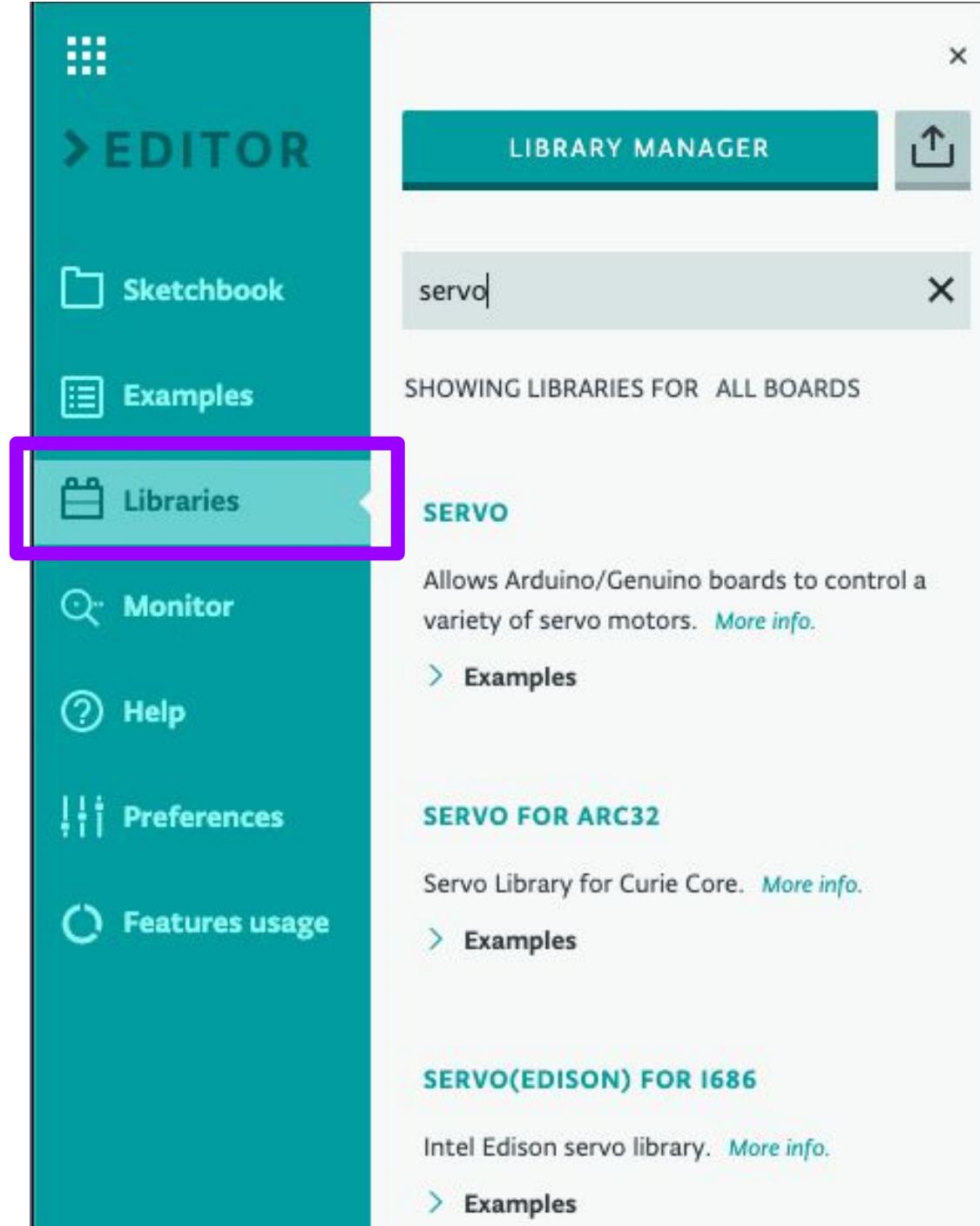
IR remote



Libraries on Arduino IDE



Libraries on Arduino Create



The image shows the Arduino Create interface. On the left, a teal sidebar menu is open, with the 'Libraries' option highlighted by a thick purple rectangular border. The menu also includes 'Sketchbook', 'Examples', 'Monitor', 'Help', 'Preferences', and 'Features usage'. To the right of the sidebar, a search bar at the top contains the text 'servo'. Below the search bar, a message says 'SHOWING LIBRARIES FOR ALL BOARDS'. The main content area displays three library entries: 'SERVO', 'SERVO FOR ARC32', and 'SERVO(EDISON) FOR I686'. Each entry includes a brief description and a 'More info.' link. The 'SERVO' entry has an additional 'Examples' link. The 'SERVO FOR ARC32' and 'SERVO(EDISON) FOR I686' entries also have 'Examples' links.

LIBRARY MANAGER

servo

SHOWING LIBRARIES FOR ALL BOARDS

SERVO

Allows Arduino/Genuino boards to control a variety of servo motors. [More info.](#)

> Examples

SERVO FOR ARC32

Servo Library for Curie Core. [More info.](#)

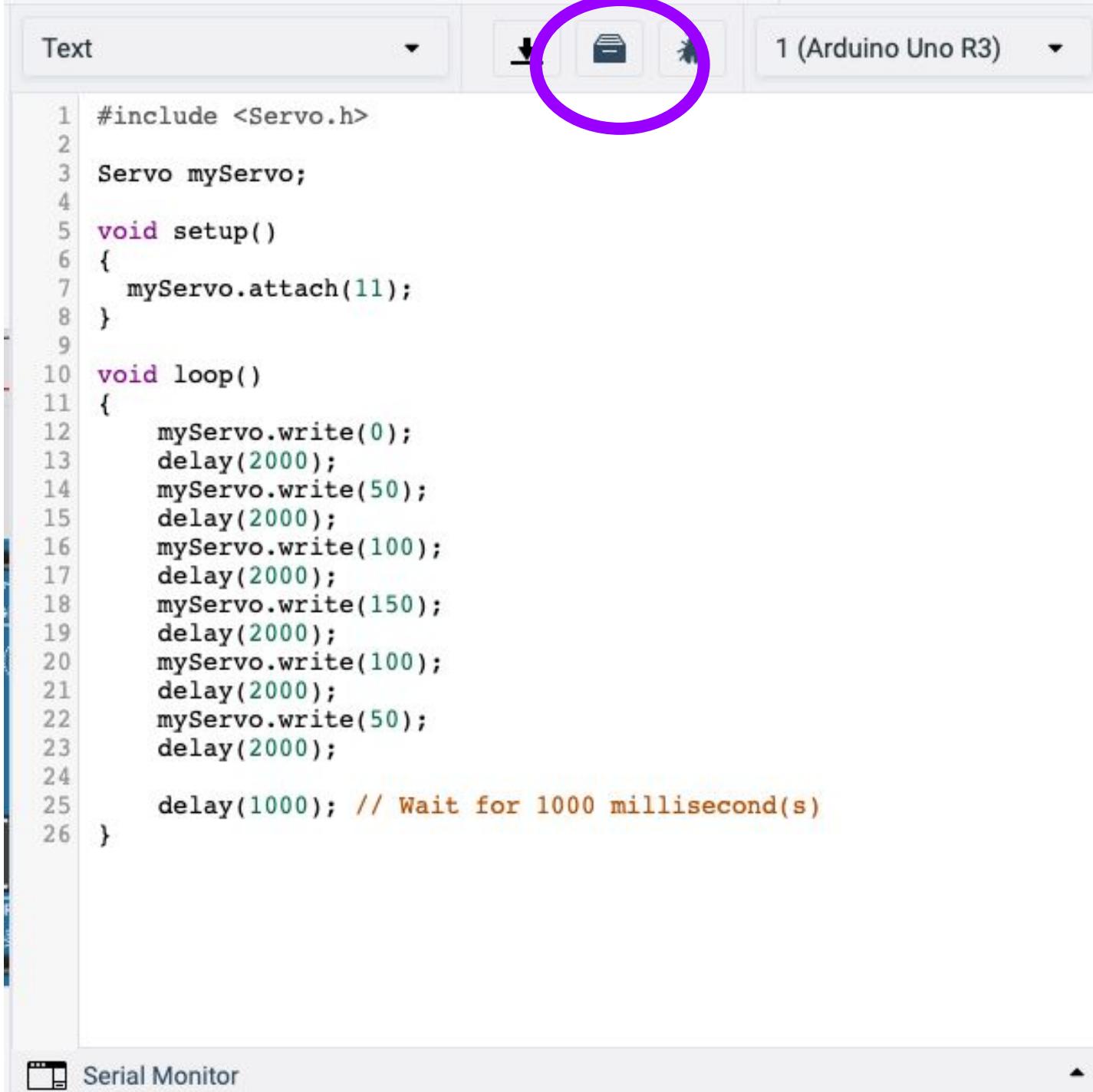
> Examples

SERVO(EDISON) FOR I686

Intel Edison servo library. [More info.](#)

> Examples

Libraries on TinkerCad

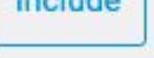
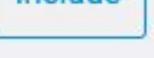
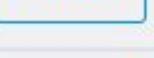
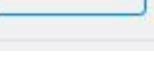


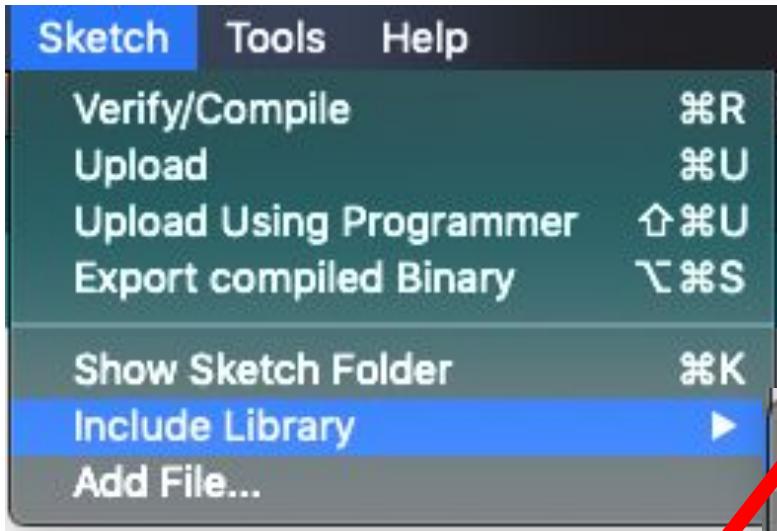
The screenshot shows the TinkerCAD Arduino IDE interface. The top bar includes a 'Text' dropdown, a download icon, a file icon (circled in purple), a settings icon, and a '1 (Arduino Uno R3)' dropdown. The main area displays an Arduino sketch for controlling a servo. The sketch includes setup and loop functions for a servo attached to pin 11, with a sequence of write and delay commands. The bottom of the screen features a 'Serial Monitor' button.

```
1 #include <Servo.h>
2
3 Servo myServo;
4
5 void setup()
6 {
7     myServo.attach(11);
8 }
9
10 void loop()
11 {
12     myServo.write(0);
13     delay(2000);
14     myServo.write(50);
15     delay(2000);
16     myServo.write(100);
17     delay(2000);
18     myServo.write(150);
19     delay(2000);
20     myServo.write(100);
21     delay(2000);
22     myServo.write(50);
23     delay(2000);
24
25     delay(1000); // Wait for 1000 millisecond(s)
26 }
```

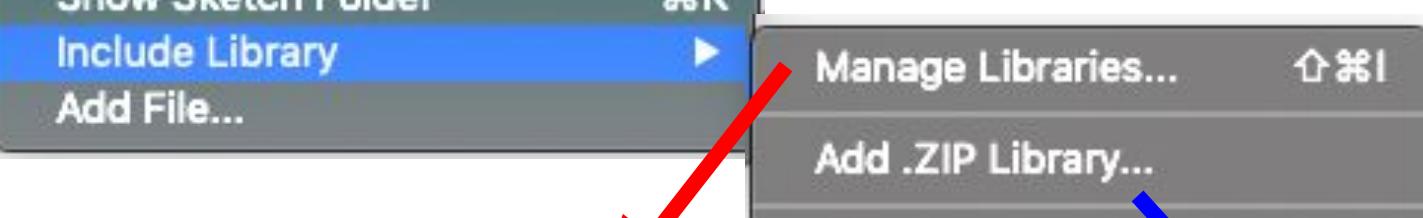
Serial Monitor

Libraries on TinkerCAD

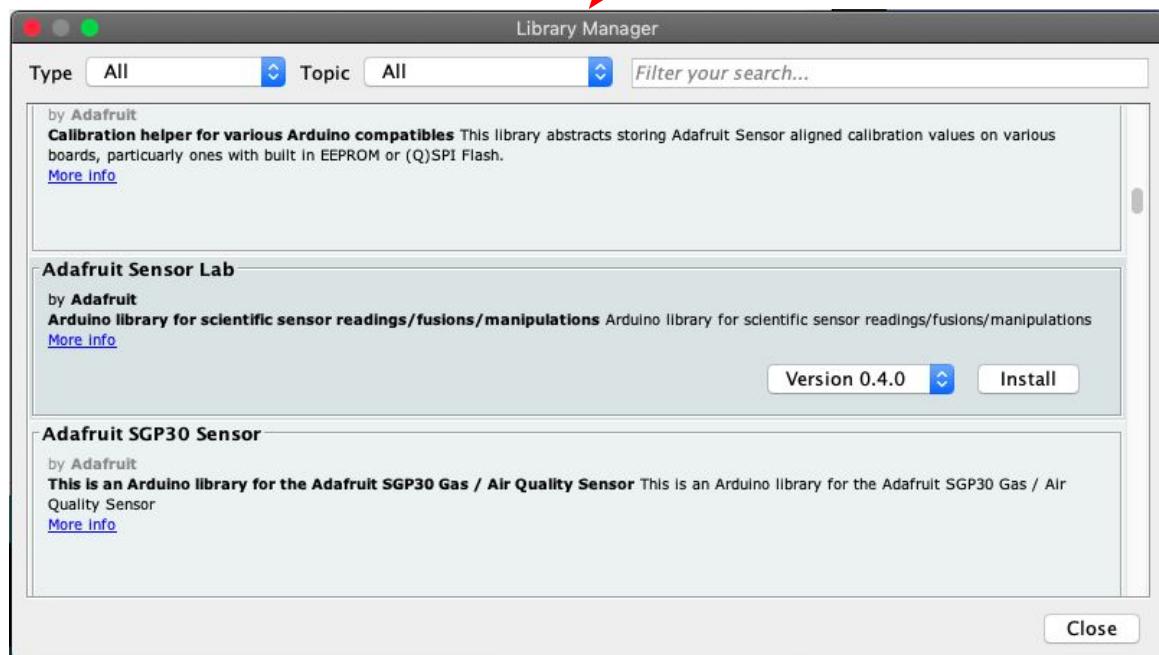
Text	  	1 (Arduino Uno R3)
 EEPROM	Reading and writing to "permanent" storage	
 IRremote	Library to decode IR sensors	
 LiquidCrystal	Controlling liquid crystal displays (LCDs)	
 Keypad	Allows reading keypad button pushes	
 NeoPixel	Controlling NeoPixel LEDs	
 Servo	Controlling servo motors	
 SoftwareSerial	Allow serial communication on other digital...	
 Wire	This library allows you to communicate with...	
 SD	The SD library allows for reading from and...	



Installing Other Libraries



Library Manager

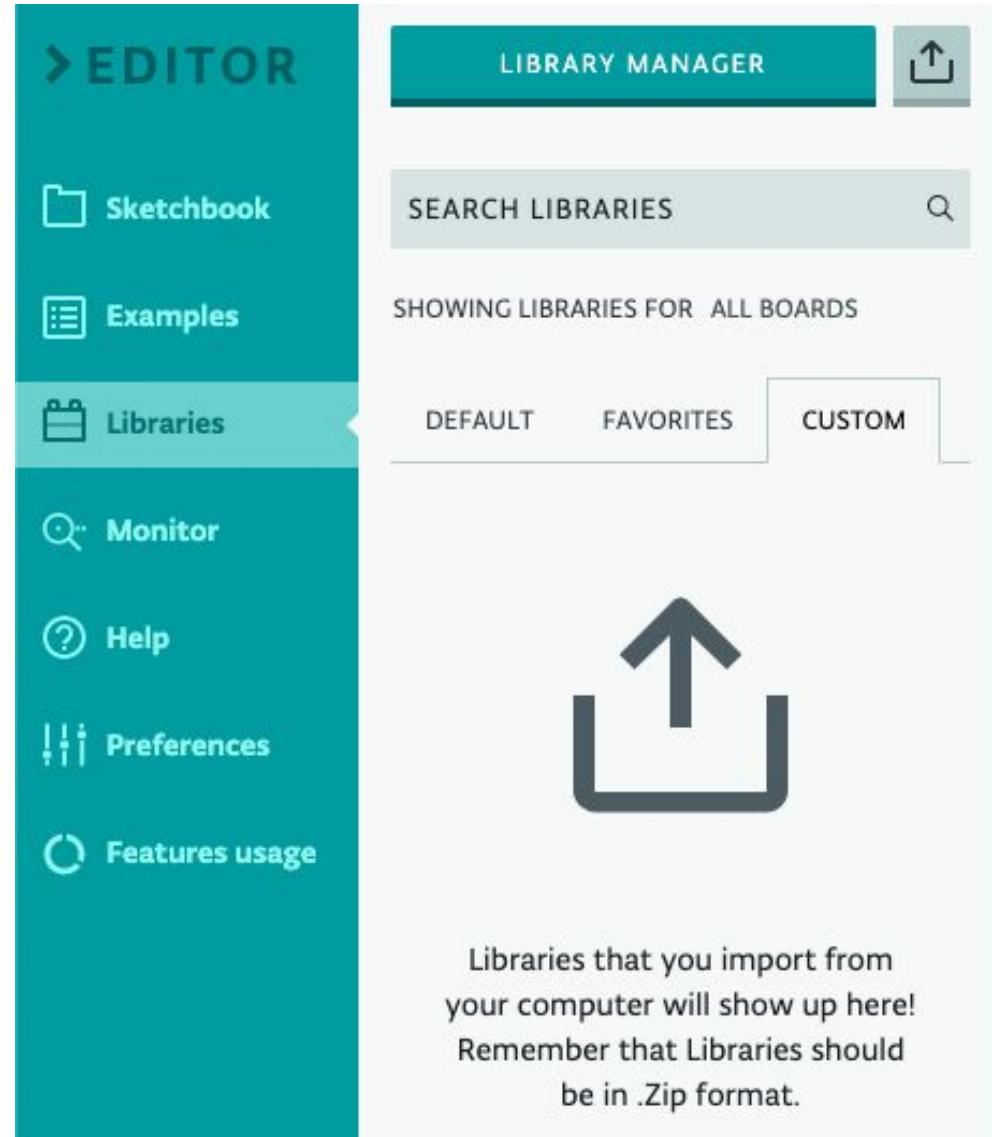


Manual Installation

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);
```

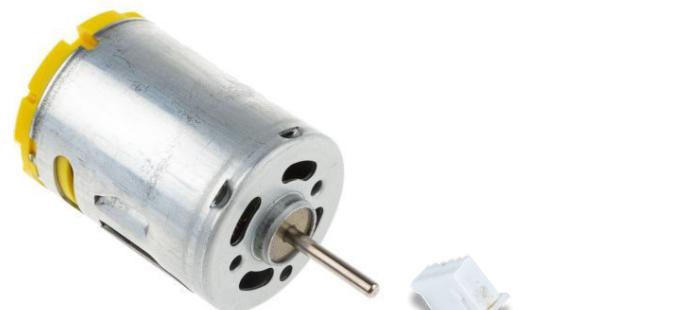
Among other things, this function contains:
`pinMode(servoPin, OUTPUT);`

This function sets the position of the servo motor

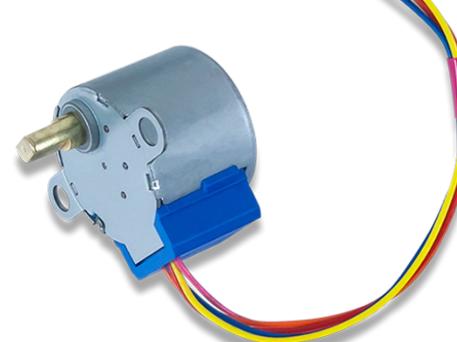
Session 1 Servo Motors and Libraries



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
- 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

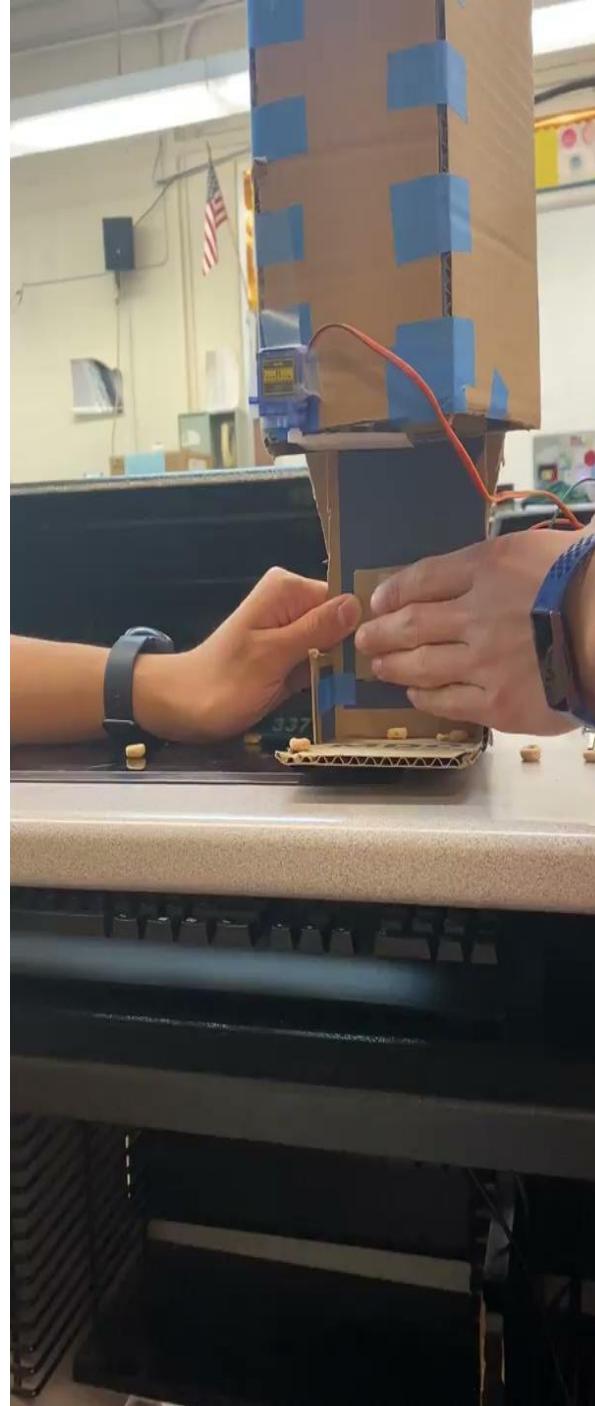
- AdaFruit
- SparkFun
- **STORES:**
 - TinkerSphere
(Allen St)
 - Microcenter
(Brooklyn, Yonkers)
- MyDuino
- Amazon/Ebay



Oh the Possibilities!

What
would
YOU?
Do

Project Example: Contactless Cereal Dispenser



Project Example: Robotic Hand

