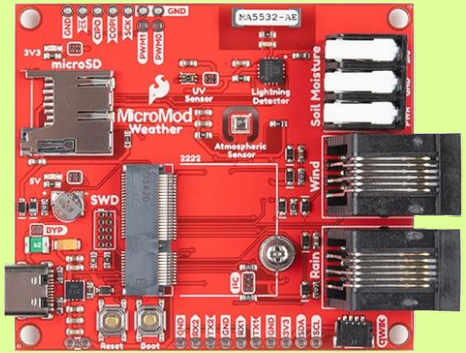
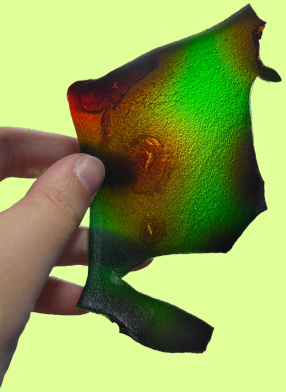
Dragtronics!

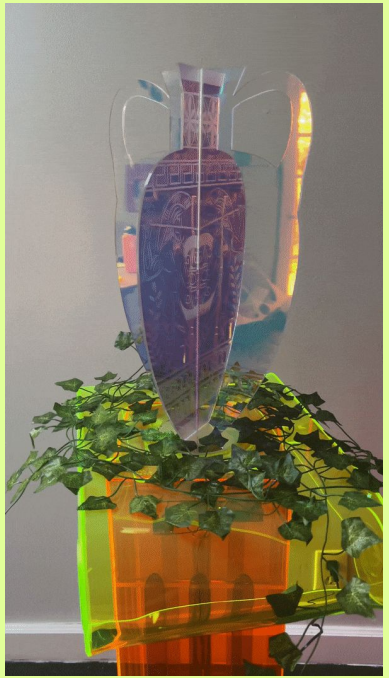
wearable electronics for drag

Priyanka Makin (stuff) she/her

nerd

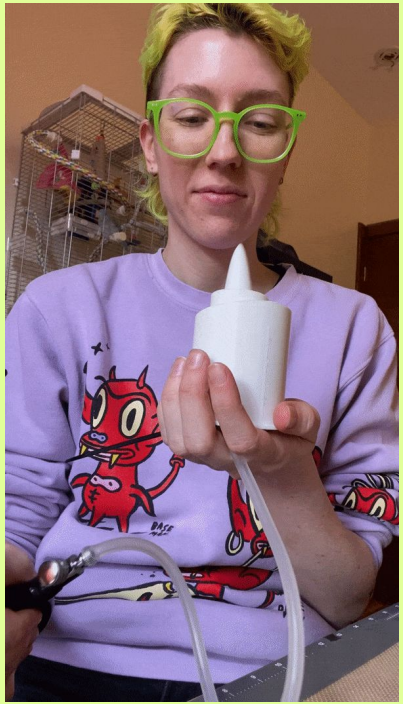






Kay (Chevy) they/them

+drag performer & producer in NYC for 5+ years
+costume designer
+2nd year grad student at ITP



Schedule

1:15-2 Concepts

- Examples
- Vocab
- Starting code concepts
- Intro to Circuit Playground
- Intro to MakeCode

2-3 Walk through project (+break)

- Walk through projects in MakeCode
  - Sound Reactive Lights
  - Wiring an external button
- Demo servos and/or bubbles (if there's time!)
- Break at some point!

3-4 Brainstorming/q&a/Ongoing projects

Cameron Hughes

LeeLee James
(Twirling Tech Goddess)

Brittany Anne Cohen

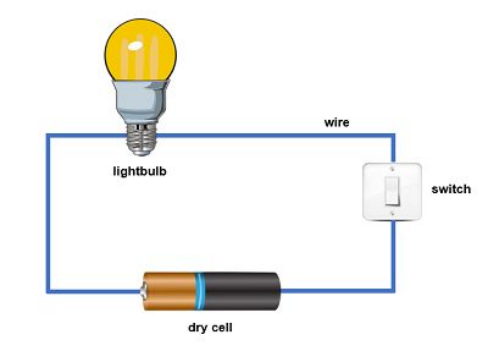
Hardware Vocab

Microcontroller = lil computer

Pin = physical part of the microcontroller that can turn on/off voltages or read incoming voltage

Circuit = physically closed loop of electronic components and wires

Sensor = a device that detects or measures a physical property in the environment



Hardware Vocab

Microcontroller = lil computer

Pin = physical part of the microcontroller that can turn on/off voltages or read incoming voltage

Circuit = physically closed loop of electronic components and wires

Sensor = a device that detects or measures a physical property in the environment

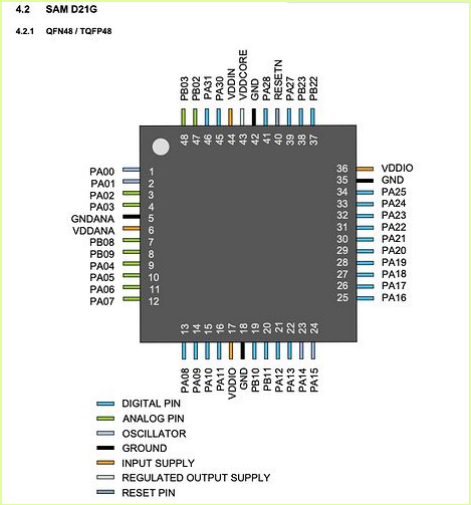
Ta da!

Physical computing = writing code that tells computer what to do with hardware

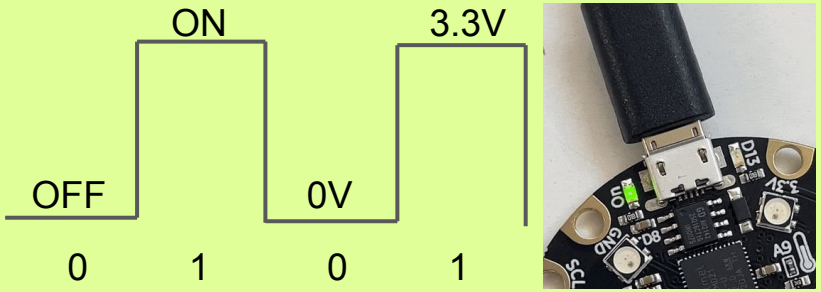
→ Electronic system that senses or responds to the world around it

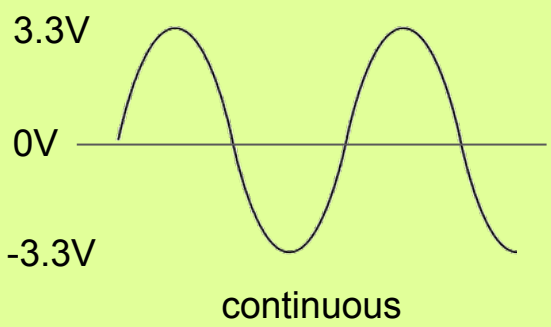
Blink, it makes the world go round

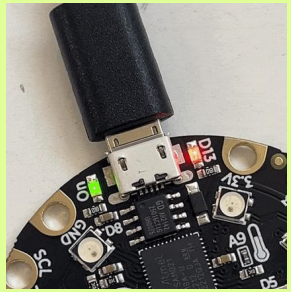
Pins can be an **input** (read incoming voltage) or an **output** (turn on/off voltage)



Digital vs. **analog**





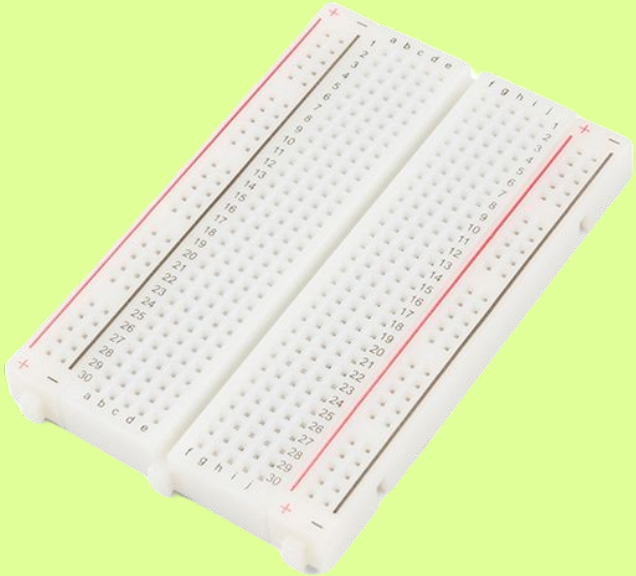


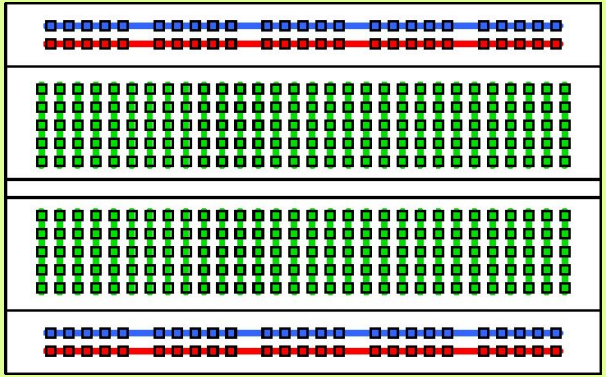
DIGITAL **ANALOG**

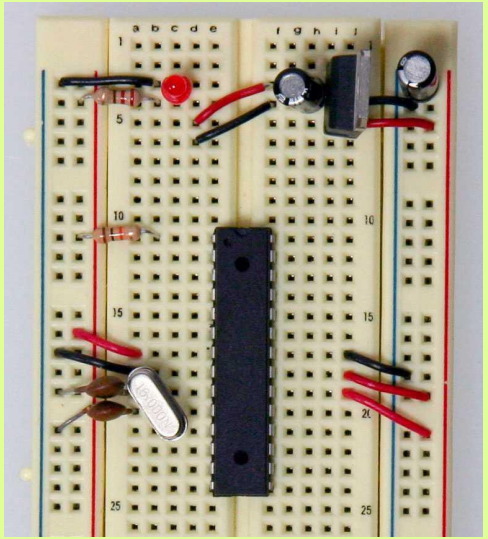
Helpful components - creating circuits!

Breadboard

Prototyping stage



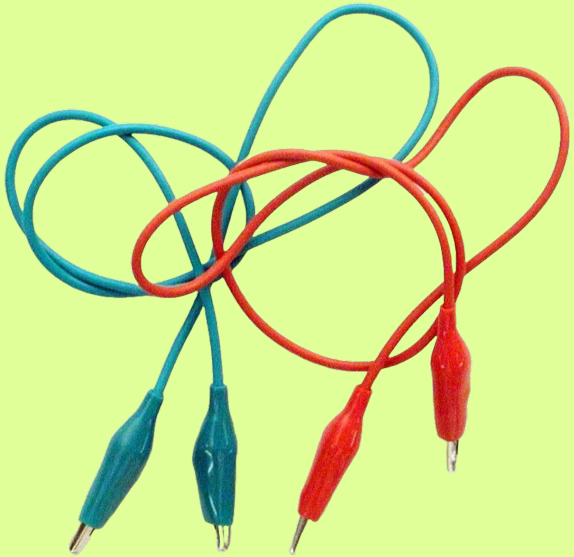


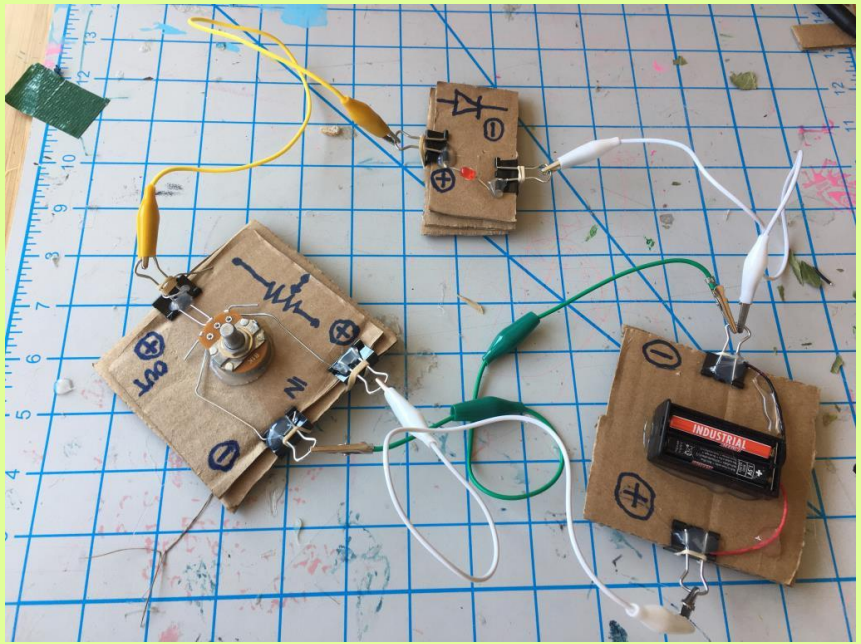


Helpful components - creating circuits!

Alligator clips

Prototyping stage

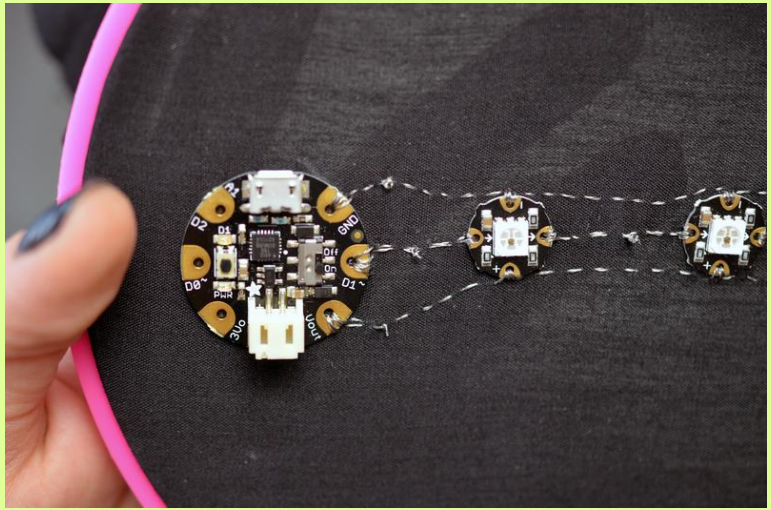




Helpful components - creating circuits!

Conductive thread

Permanent / semi-permanent

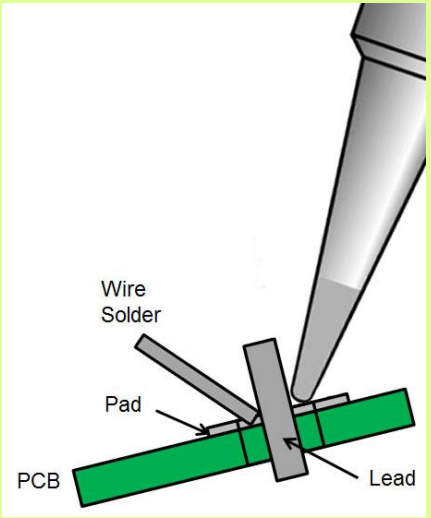


Helpful components - creating circuits!

Soldering

Permanent

Wires, through-hole, and surface mount



Say hello to the Circuit Playground Express (CPX)!

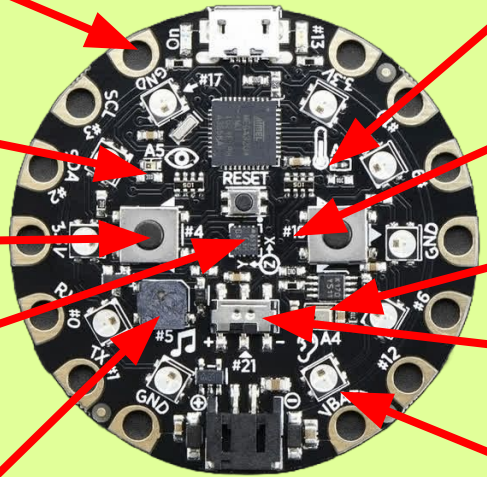
Contacts for alligator clips or conductive thread

Light sensor

2 push buttons

Motion sensor

Mini speaker with Class D amplifier



Temperature sensor

IR RX/TX

Sound sensor (microphone)

Slide switch

10 mini (RGB) neopixels

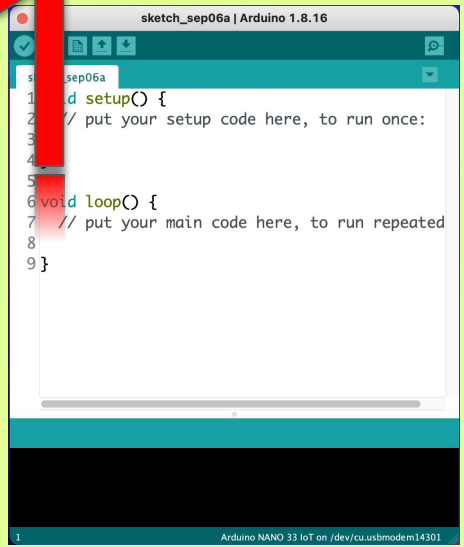
Power using USB, AAA battery pack or Lipoly battery

I2C, UART, analog input, PWM output, capacitive touch, and so much more!

Programming

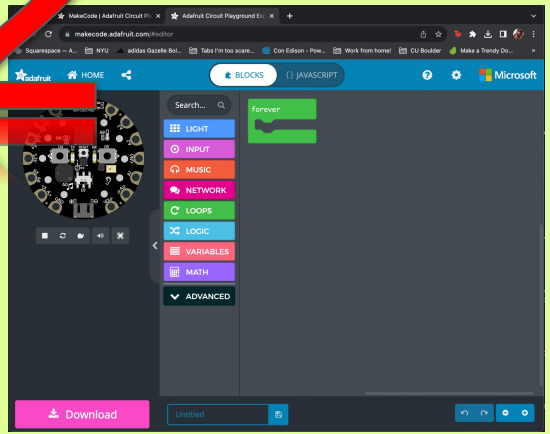
You can upload code to the CPX three different ways

1



Arduino IDE

2



MakeCode

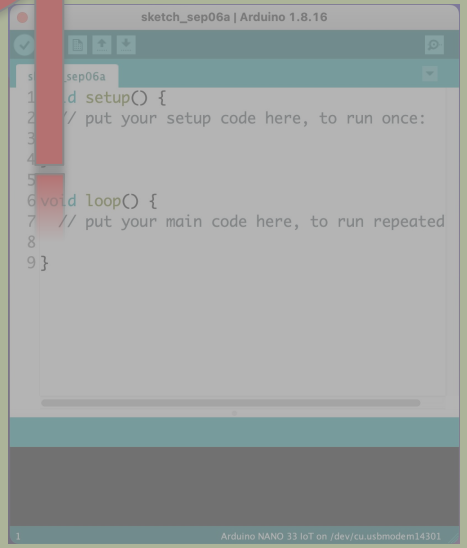
3

Circuit Python

Programming

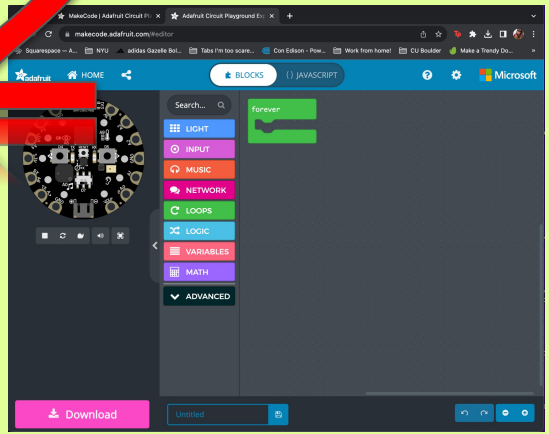
You can upload code to the CPX three different ways

1



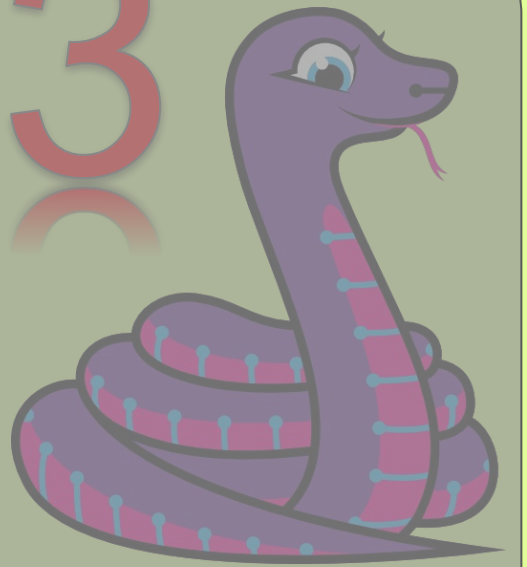
Arduino IDE

2



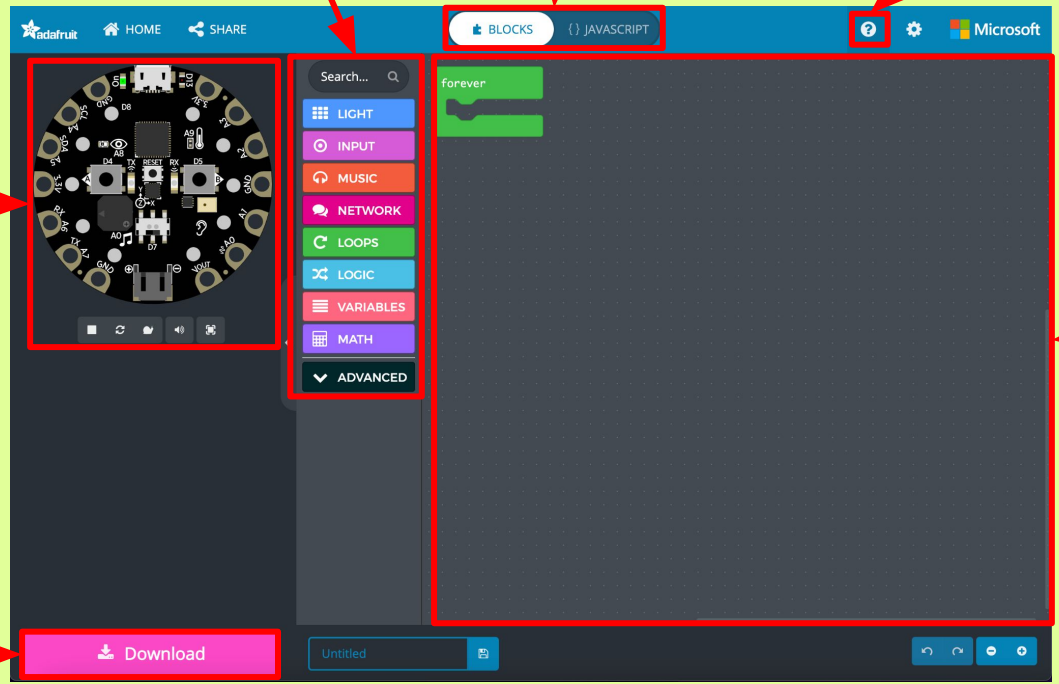
MakeCode = visual programming language

3



Circuit Python

MakeCode Tour



Programming concepts

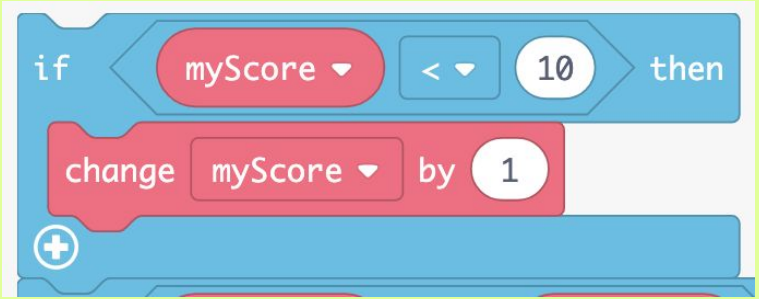
Let’s go here: **https://makecode.adafruit.com/** and create a **new project**

You should see this:

forever runs blocks in a continuous loop with a 20 ms pause in between

Programming concepts

If statement



1.
Some code

*If *some condition* is true*
Do this

Continue running code

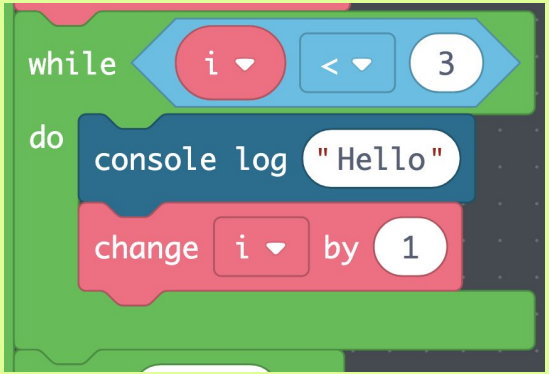
2.
Some code

*If *some condition* is true*
Do this
*Else (*some condition* is false)*
Do that

Continue running code

Programming concepts

While loop



Some code

*While *some condition* is true*
Do this

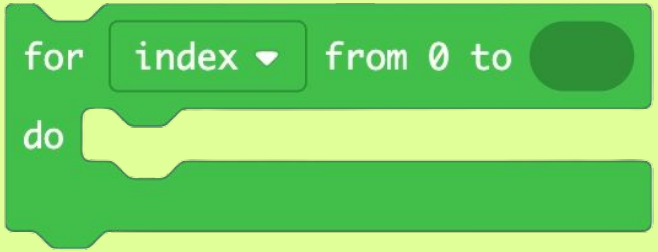
*Once *some condition* is false, continue running code*

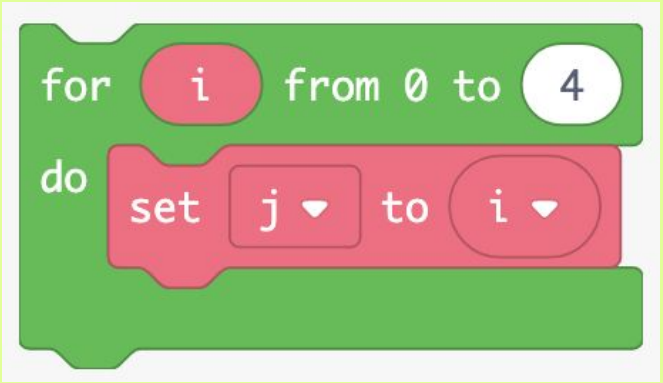
Notes:

- While loops can be ✨infinite loops✨ if *some condition* is always true
  - Kind of like the “forever” block
- While loops can be executed a specified number of times using a **variable**

Programming concepts

For loop





Some code

*For up to *certain number**
Do this
Increment count
Go back up to count

*Once *certain number* is reached, continue running code*

Notes:

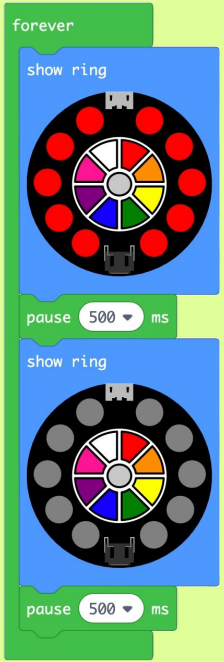
- Kinda like a while loop but more confusing

Programming concepts

For more information than you could ever want about all the code blocks, check out the **MakeCode reference manual**

Let's get started! Blink Example

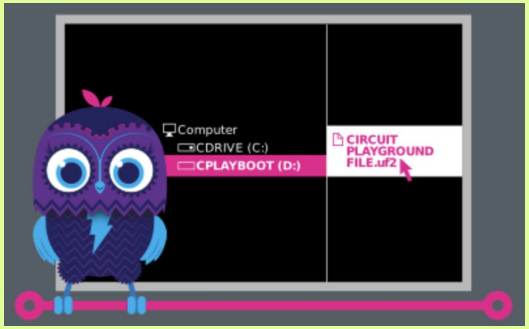
1. Open MakeCode and make this code



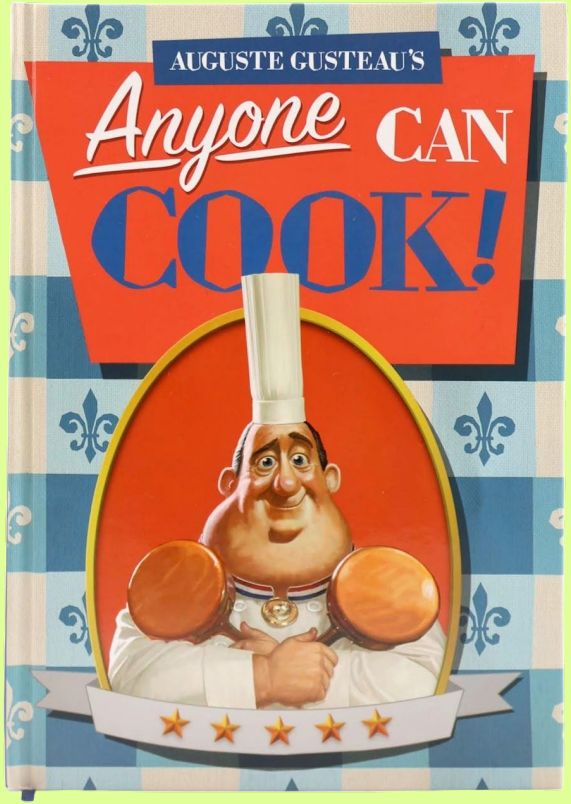
2. Connect CPX to computer using USB cable

3. Click the **RESET** button on the CPX until the neopixels turn red then green. The CPX should show up as a drive on your computer.

4. Hit the **Download** button in MakeCode and drag the .uf2 file onto the CPLAYBOOT (D:) drive

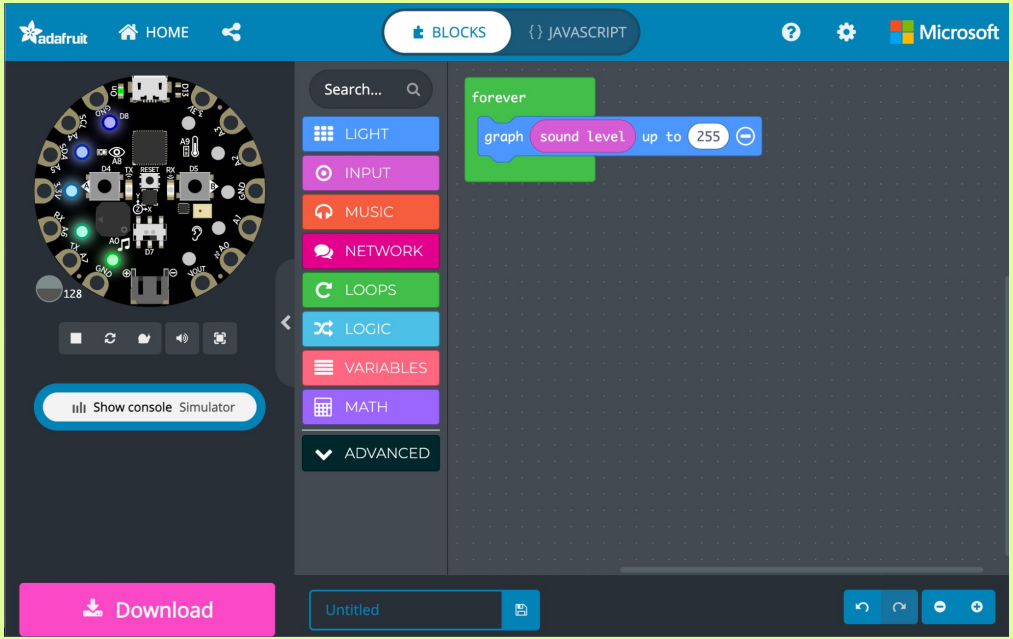


How's everyone doing?!



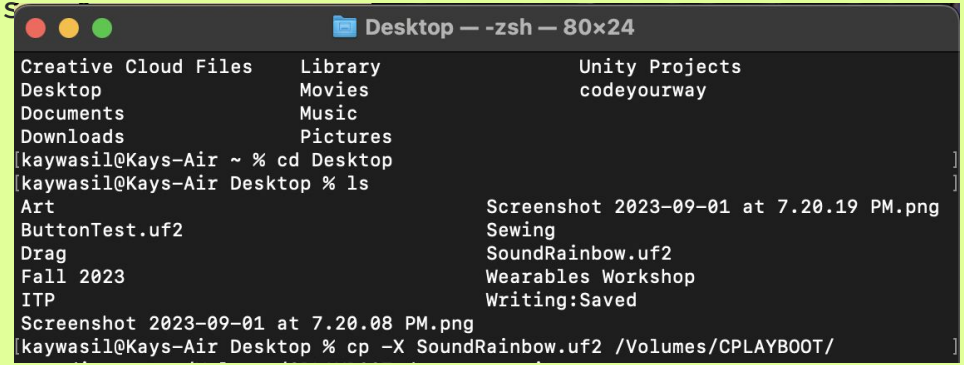
Sound-reactive LED display

- Open makecode
- Under loops, drag and drop “forever”
- Under light, drag “graph” inside of forever block
- Under input, drag “sound level” into first part of graph block; then click on the plus sign and set the max to 255;
- Download the code and put it on your circuit playground!



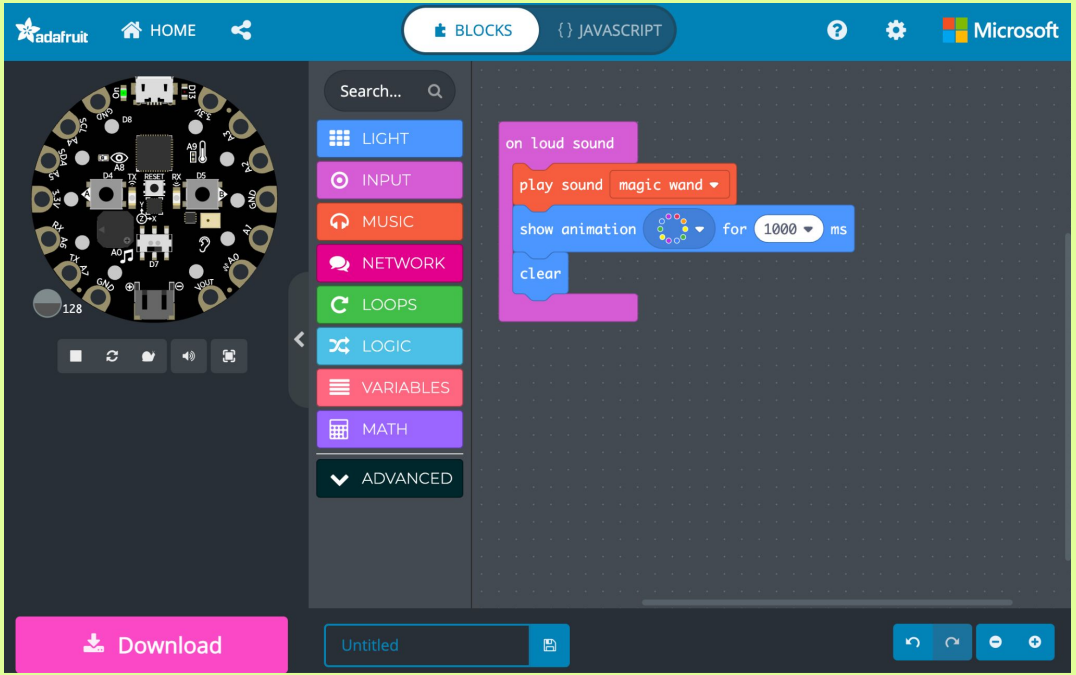
IF YOU GET ERROR 100093

Buckle up, we're going into the matrix
-make sure your uf2 file name is short
-drag your uf2 file (with your code in it) onto your desktop
-open Terminal
> capitalization, punctuation, and spacing are very important here! <
-type ls, hit enter (desktop should appear as a folder in your listed files)
-type cd Desktop
-type ls
- type: cp -X (NAMEOFYOURFILE).uf2 /Volumes/CPLAYBOOT/
-hit enter



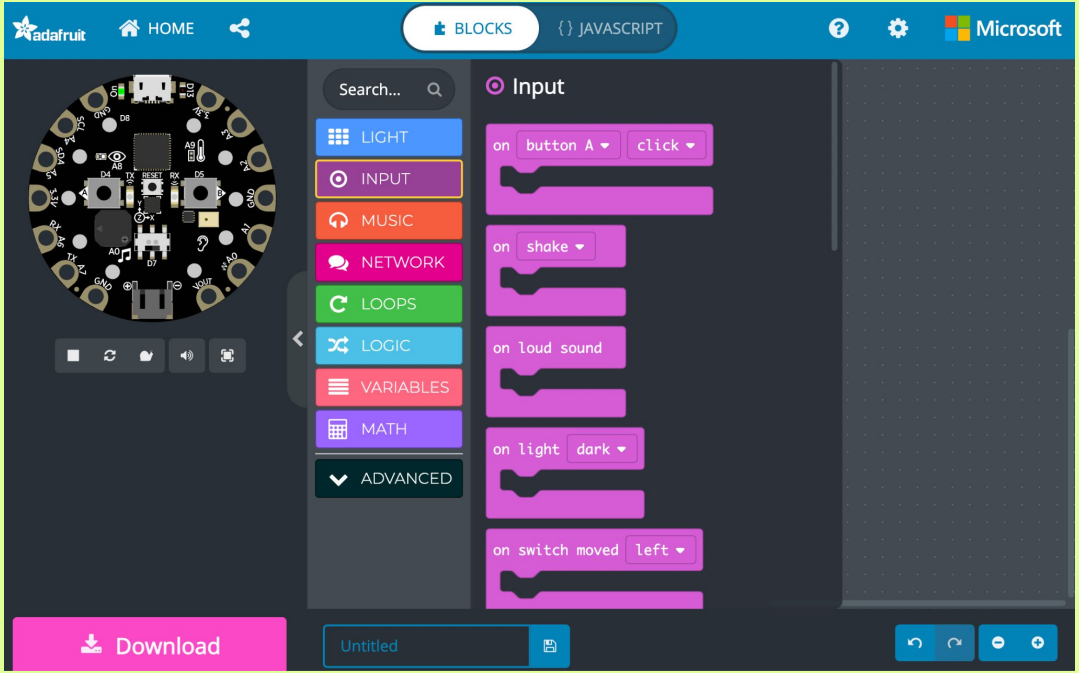
Part 2 - Playing an animation and sound in response to input

- Go to input, select “on loud sound” or “shake”
- Under sound, select and drag “play sound” and pick a sound, add this inside of the on loud sound block
- Under light pick “show animation”
- Add after sound
- Under light pick “clear” (this turns off the LEDs when the animation is done playing)
- Add after animation



Add to it!

- Go to input - pick a new input and a reaction! You could have it react to something from the sensors (sound, on shake, etc), or to one of the built in-buttons being pressed
- Have it play a sound or animation in response to the input

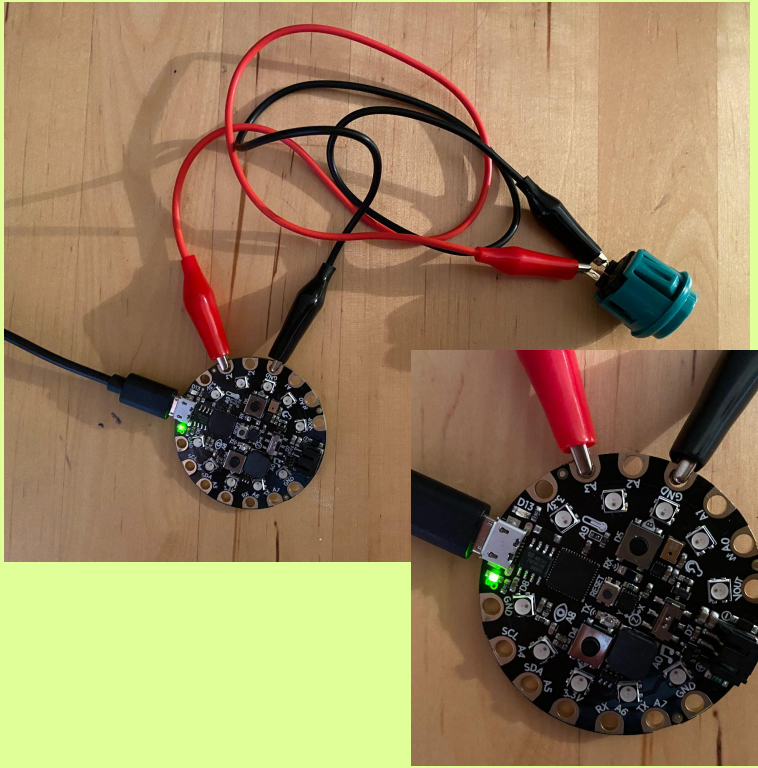


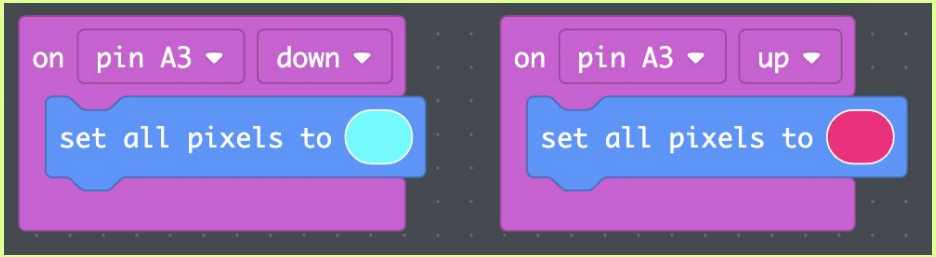
Adding an external button

Hardware Setup: connect one side of button to ground, connect other side of the button to the pin you're using (A3 for this demo)

In makecode:

- Go to input and grab first block, it's default labeled as "on button A click"
- Drag into workspace, change to "on pin A3 **down**" (when the button is pressed)
- Under light select> "change all pixels to" (pick a color)
- Copy paste this block, change to "pin A3 **up**"; change pixel color so you'll know we've changed something (when the button is released)
- Upload the code





Debugging

You will eventually run into problems! The best way to figure out what's wrong is to **break things down into very small pieces** and figure out what **is** and **isn't** working.

Good starting places:

- Is your power source working? Is everything plugged in/turned on/connected/charged
  - Is power connected to power and ground connected to ground?
- Are any components loose or connections iffy?
- Is it your code or your physical components? Can you set up tests of them separately to figure out which part is causing problems? Set up simpler code that you know works, follow along with similar projects and just change ***very*** small pieces at a time to see if you can isolate the problem(s).
- Post on forums if you get stuck! Generally people are very helpful on the adafruit and arduino ones! (we have links to them at the end of this presentation)

Powering things

Circuit Playgrounds will run on between 3.3-6 volts (other boards have more precise needs- circuit playground can manage anything in this range)

-can be powered off of an external battery with a usb port or a battery pack

-can output (on vout pin) 3.3v (enough for some lights, a small air/water pump, or a small motor/servo

Brainstorming!

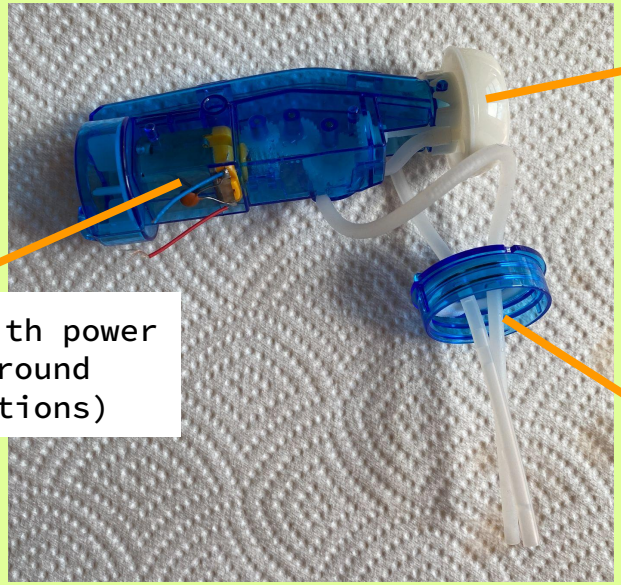
Using a bubble gun for costumes/props

Adapted from Dinalab's bubblepunk workshop!
https://www.dinalab.net/2023/04/10/bubblepunk/

You want this type of nozzle

NOT this kind

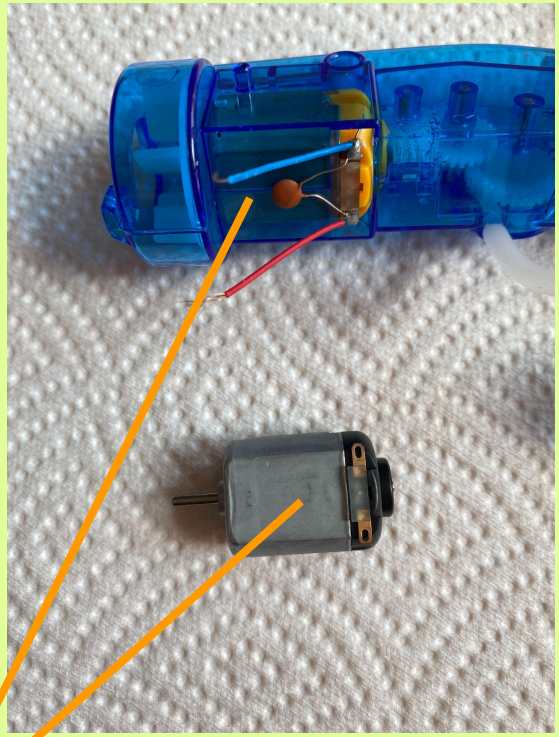
Take it all apart until you have just the pump, the nozzle, and the connection to the bubble solution bottle



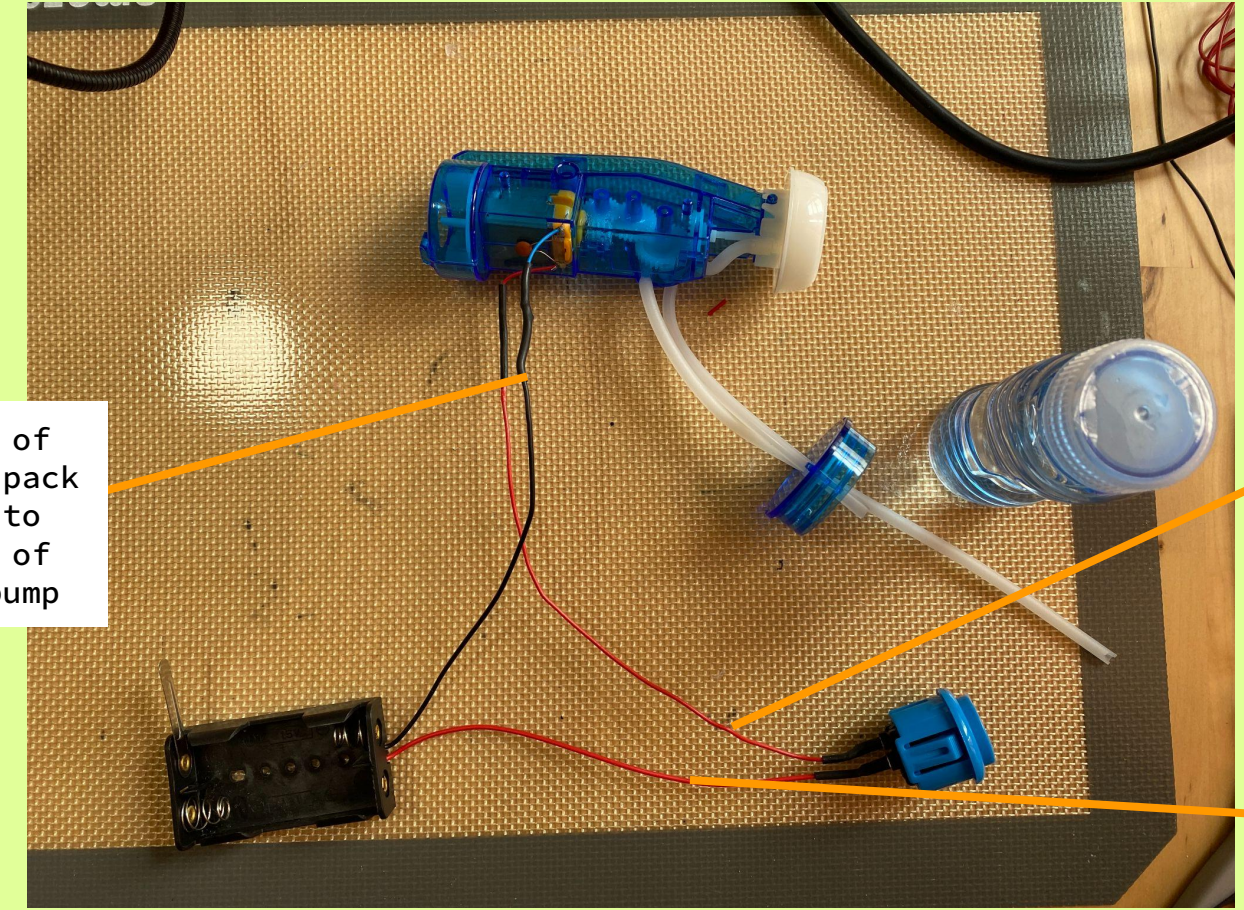
nozzle

Motor (with power and ground connections)

Bubble solution goes here



just a standard toy DC motor powering this thing!



Ground of battery pack wired to ground of motor/pump

One side of button wired to power of motor/pump

Other side of button wired to power of battery pack

Put it all inside of your prop/costume!

Two (or more) Buttons!

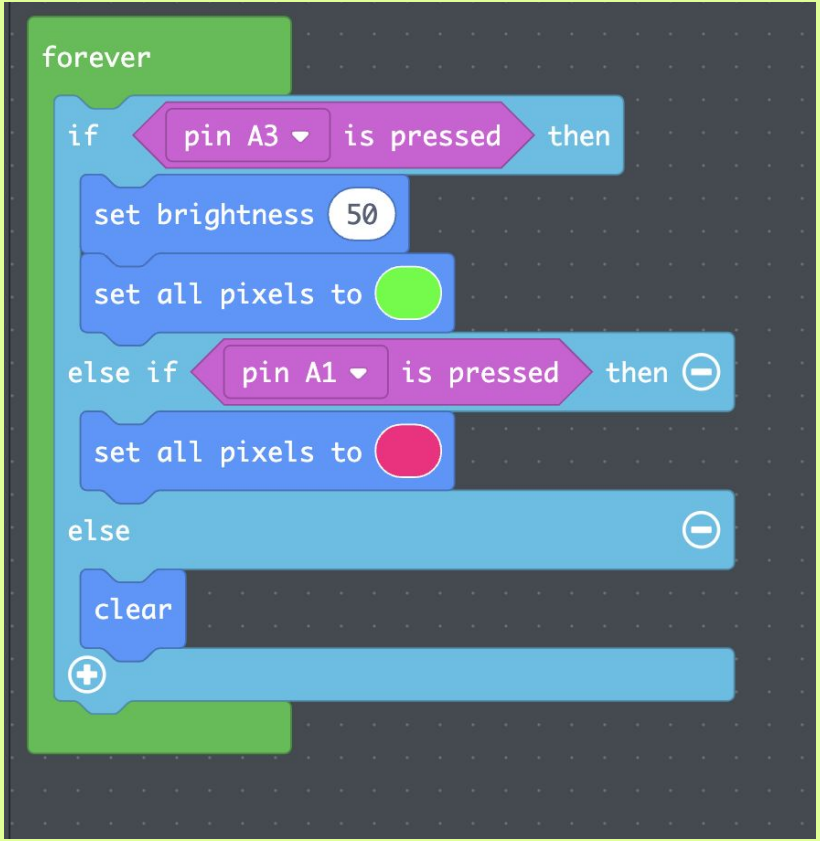
Hardware Setup: The same as one button but do it twice - one side of each to ground, the other side of each to the separate pins you're reading

In makecode:

The logic here is more complicated to avoid having conflicting commands.

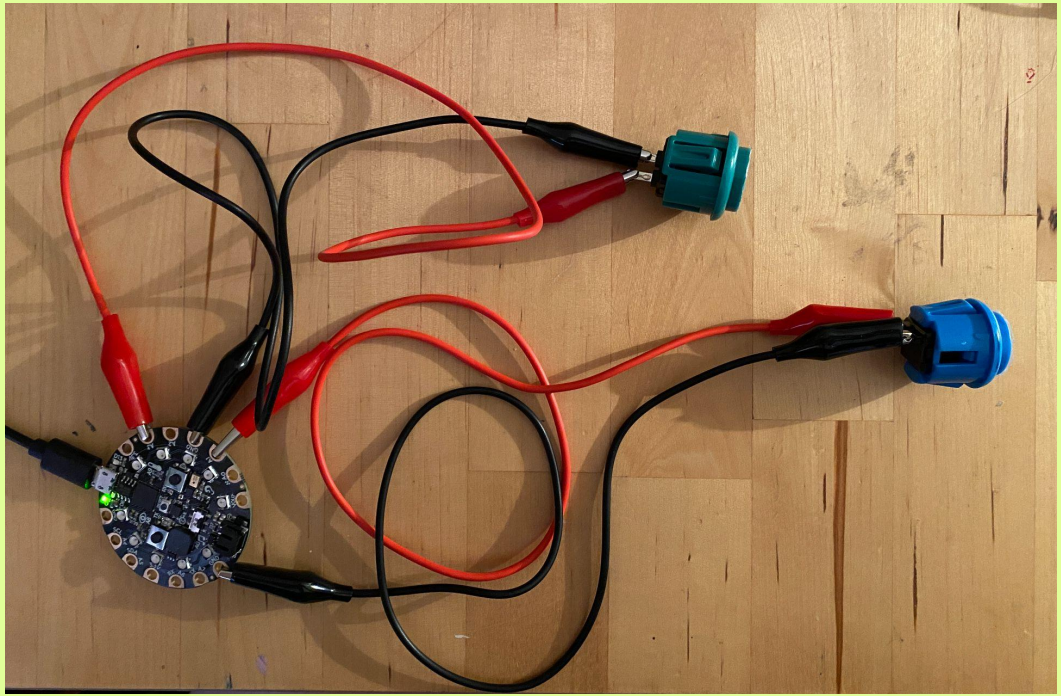
We can't say "pink when button A is pressed and blue when button A isn't pressed" bc for button B to be pressed, A isn't pressed - the commands conflict and the computer doesn't know which command to obey. The "if-else" structure lets us stop conflicts from happening

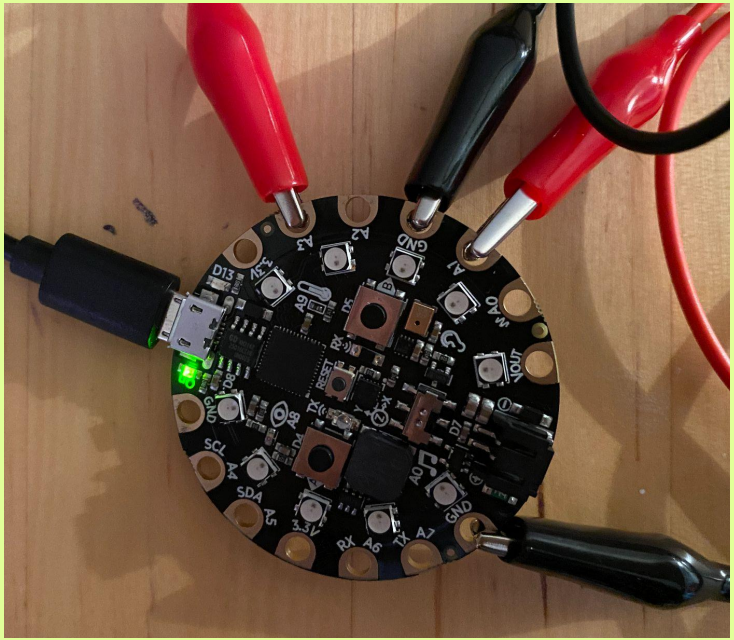
The logic written out is more like:
If button A (pin A3) is pressed, make the lights green.
If button B (pin A1) is pressed, make the lights pink.
Otherwise, turn the lights off.



Two (or more) Buttons!

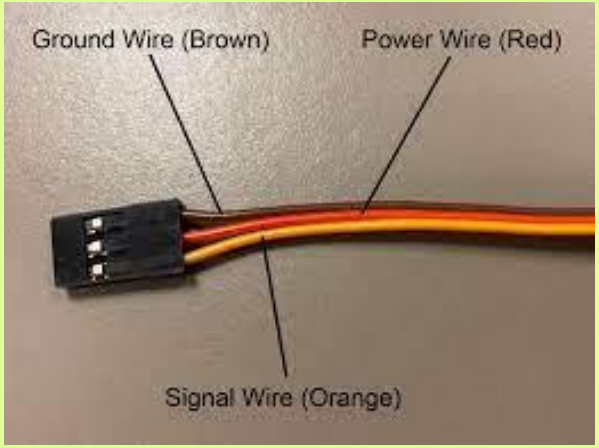
Hardware setup images:





Movement! (Setting up a servo)

VS



Motors:

On or off (rotating or not)
Can control speed with voltage

Servos:

Position of the arm can be controlled via code- have a third wire for sending data (often labeled PWM, which stands for pulse width modulation)

Continuous rotation servo - acts like a motor in this case; control speed rather than position in code (Generally speaking 180 is full speed forward, 0 is full speed backwards, and 90 is stopped)

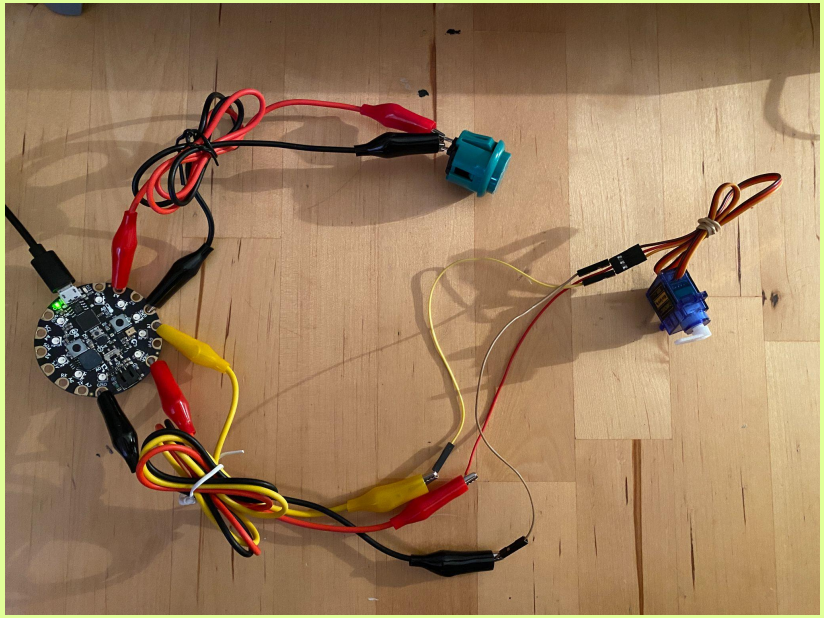
Servos work better than motors with the Circuit Playground
(you can control motors with add-ons, like the crikit)

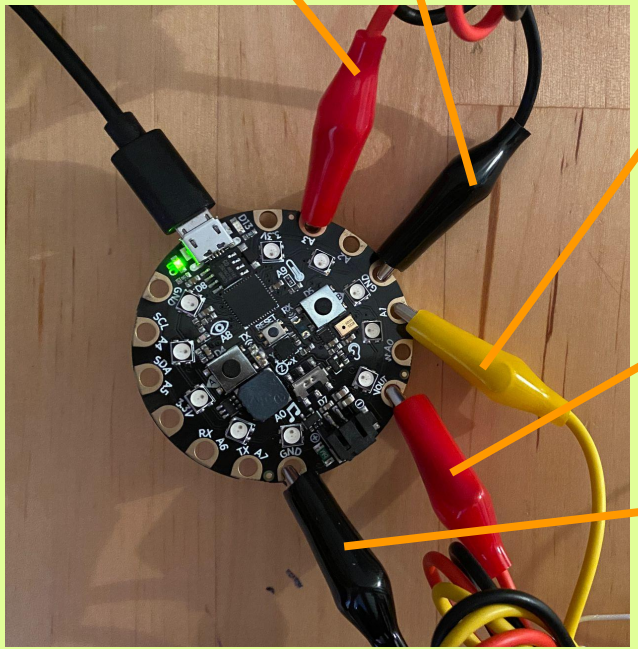
Movement! (Setting up a servo)

Hardware:

Connect power of servo to VOUT on circuit playground
Connect ground of servo to ground
Connect pwm to the data pin you're using - A1 in this case

Connect one side of button to ground
Connect other side of button to A3



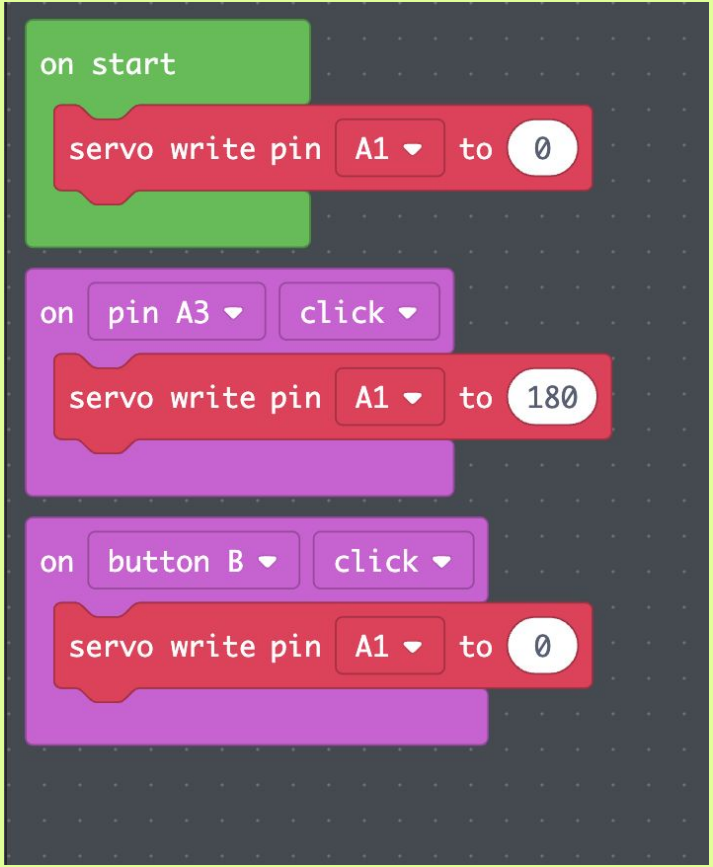


Movement! (Setting up a servo)

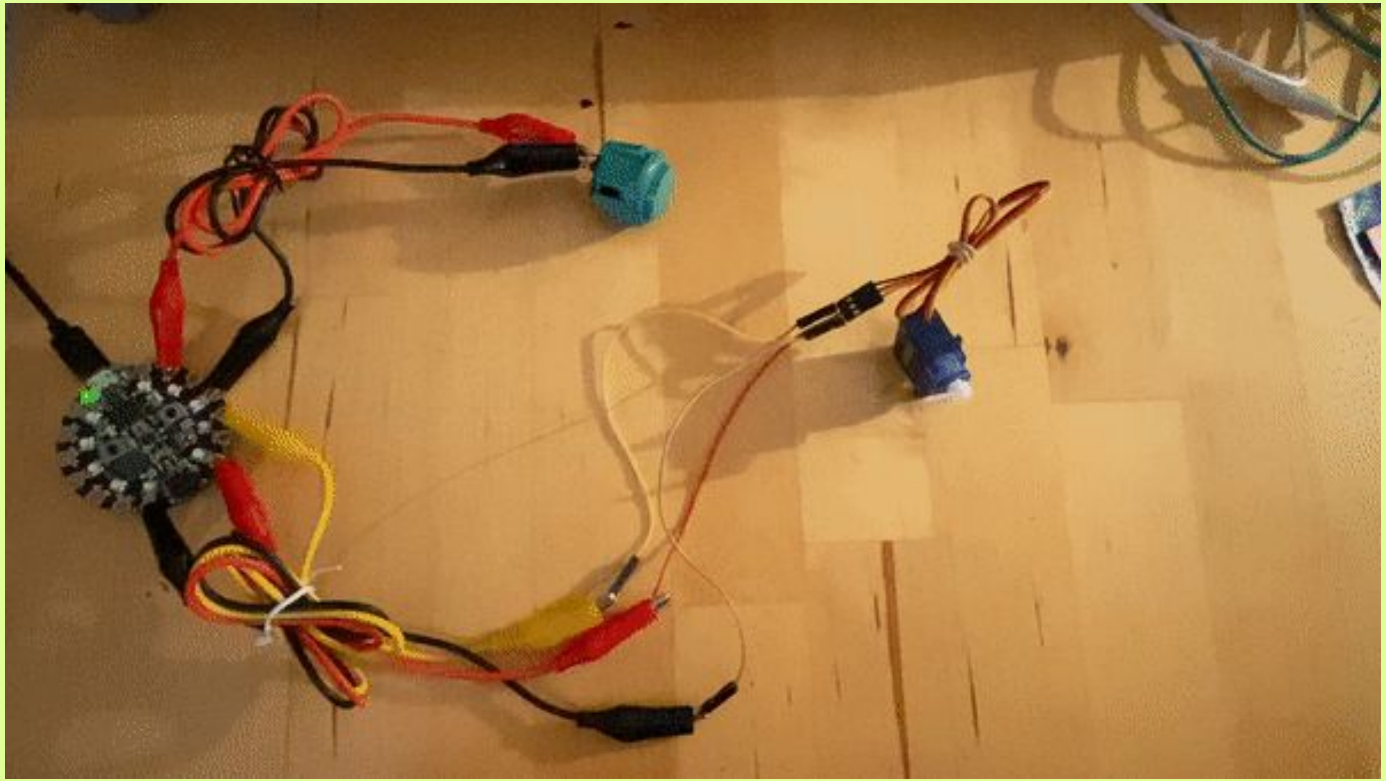
In makecode:
loops> on start
advanced> pins> “servo write pin A1 to 0” (when you turn on the circuit playground, set the servo to the starting position - zero)

input> on pin A3 click
advanced> pins> “servo write pin A1 to 180” (move the servo attached to pin A1 to position 180- the end of the range of movement)

input> “on button B click”
advanced> pins> “servo write pin A1 to 0” (move the servo back to zero when you click button B on the board- this is acting as a position reset for us)



Movement! (Setting up a servo)



Designing Electronics for Costumes Checklist

- ❏ **How durable and reliable are the components?**
  - ❏ More durable/reliable usually = more expensive
  - ❏ I try to make sure my components work 80% or more of the time; I also usually have a plan for if something goes wrong so the number isn't ruined
  - ❏ Are your connections sturdy? Can they fall out or be knocked out of place? Reinforce everything! Hot glue, zip ties, and duct tape are your friends
- ❏ **Size and weight of components and batteries**
  - ❏ Components (and batteries specifically) can be very heavy and bulky; It's worth thinking about where they need to be in order to work and how/if you're going to conceal them
- ❏ **How will your components attach to the costume and stay on?**
- ❏ **Access - can you easily access your parts to fix things, change batteries, etc?**
- ❏ **Generally speaking - lights = easy; movement = hard.** Spinning/rotation is the easiest kind of movement
- ❏ **How visible will it be in a performance space?**
  - ❏ Will it read in a large venue the same as a small one? If you have moving components make them as big as you can and the movement as big/noticeable as possible, if you have lights, make them as bright as you can and consider if the venue can control lighting or not!

Resources - All things CPX

[Product Page]

[CPX Getting Started Guide]

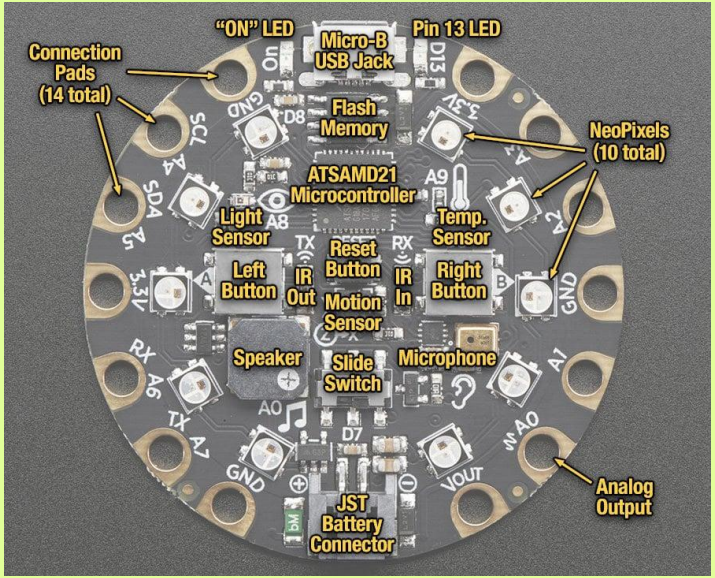
[MakeCode for CPX]

[What is MakeCode]

[MakeCode Reference Manual]

[Download Arduino]

[CircuitPython]



Resources - Learning

- Open source hardware association! OPEN SOURCE is about sharing our knowledge with the community. Knowledge is power, share what you've learned! OSHWA site - https://www.oshwa.org/
  - Specific project guides - https://certification.oshwa.org/directory.html

- Kate Hartman - http://www.katehartman.com/
- Social Body Lab- https://www2.ocadu.ca/research/socialbody/projects
- Bodies in Play - https://bip.dmg.to/
- Neilhiem mechatronics - http://www.nilheim.co.uk/
- Bubblepunk! - https://www.dinalab.net/2023/04/10/bubblepunk/
- Twirling Tech Goddess - https://www.youtube.com/channel/UC58LFxfWi4b2_AI2z9-Z0RQ
  - She's so great! And explains stuff in a really easy to understand way, not bogged down with all the stupid technical details.
- UCD Beta Lab on youtube - great circuit playground videos - https://www.youtube.com/@ucdbetalab3690

- Arduino Forums - https://forum.arduino.cc/
- Adafruit Learning - https://learn.adafruit.com/
- Sparkfun Learning - https://learn.sparkfun.com/

Resources - Equipment

Adafruit - https://www.adafruit.com/
Sparkfun - https://www.sparkfun.com/
Arduino - https://www.arduino.cc/

- Circuit Playground Express - https://www.amazon.com/dp/B0764MCBNR?psc=1&ref=ppx_yo2ov_dt_b_product_details
- Crikit board for Circuit (allows you to work more easily with motors etc) - https://www.adafruit.com/product/3093
- Battery pack (with connector for circuit playground specifically)- https://www.adafruit.com/product/727
- Motor kit - https://www.amazon.com/gp/product/B0776ZPP7V/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
- Micro servo (180 degrees of movement) - https://www.amazon.com/Dorhea-Arduino-Helicopter-Airplane-Walking/dp/B07Q6JGWNV/ref=sr_1_5?crid=3U1YKMFU1P8FV&keywords=micro%2Bservo%2Bmotor&qid=1694529024&sprefix=micro%2Bsev%2Caps%2C125&sr=8-5&th=1
- Micro servo (continuous rotation) - https://www.amazon.com/FEETECH-Continuous-Rotation-Helicopter-Airplane/dp/B097SZQ4CH/ref=sr_1_6?crid=2TBO82ZCWUXG2&keywords=micro%2Bservo%2Bmotor%2Bcontinuous%2Brotation&qid=1694529073&sprefix=micro%2Bservo%2Bmotor%2Bcontinuous%2Brotation%2Caps%2C92&sr=8-6&th=1

Resources - Equipment (continued)

- Bubble guns - https://www.amazon.com/dp/B0BX5BVCPX?psc=1&ref=ppx_yo2ov_dt_b_product_details
- Air pump -https://www.adafruit.com/product/4700
- Liquid pump - https://www.adafruit.com/product/4547
- Noods - flexible, super-bright LED filament - https://www.adafruit.com/product/5506 ; guide here: https://learn.adafruit.com/noods-uberguide/overview?gclid=Cj0KCQjwgNanBhDUARIsAAeIcAuiz0d_1qv5OXLOX1M5d5QLj820Buc-zjo9lXGgdi9tMdbAtkIphhcaAiSqEALw_wcB
- Alligator clips - https://www.amazon.com/Alligator-Electrical-21-5inch-Connection-Experiment/dp/B0995KJWR5/ref=sr_1_3_sspa?crid=1QOBS62VXCBB4&keywords=alligator+clips&qid=1694529311&sprefix=alligator+%2Caps%2C101&sr=8-3-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1
- Breadboards - https://www.amazon.com/DEYUE-breadboard-Set-Prototype-Board/dp/B07LFD4LT6/ref=sr_1_2_sspa?crid=LI943RJF1X6Q&keywords=breadboard&qid=1694529333&sprefix=breadboard%2Caps%2C111&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1
- Conductive thread - https://www.adafruit.com/product/640

Resources - Wearable Projects

Wing tutorial

507 Mechanisms

Twirling Tech Goddess - sound reactive LEDs ❤️❤️❤️

    Sound reactive Neopixels

… DC motor shenanigans (seems advanced!)

Helpful tutorials?