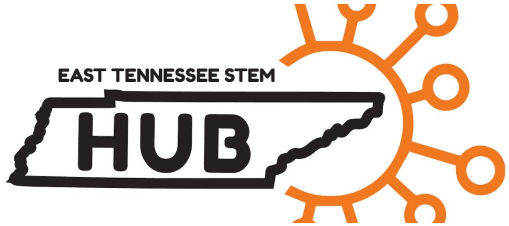
Computer Science, Code and Hummingbirds

Matthew Lane

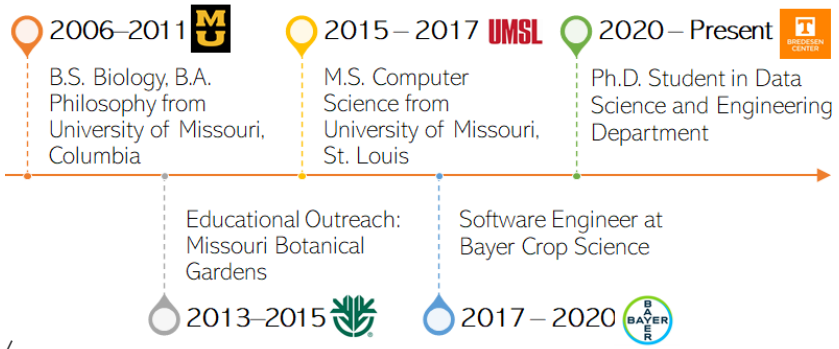


whoami

- Matthew Lane
- Graduate Research Fellow at Oak Ridge Innovation Institute

- Research Interests:
  - Explainable AI
  - Network and Node Embedding
  - Scientific Software Engineering
  - Topological Network Perturbation

- GitHub: `github.com/lanematthewj`
- LinkedIn: `linkedin.com/in/lanematthewj/`
- ORCID: `0000-0002-7750-6822`



Coding Vs. Computer Science

COMPUTER SCIENCE, CODE, AND HUMMINGBIRDS

What is a Computer?

Computers Execute Instructions

- $Instructions \iff code$

What is a Computer?

Computers Execute Instructions

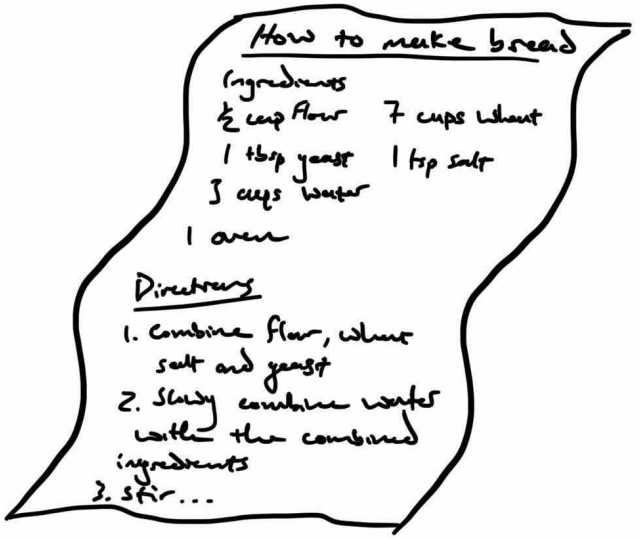
- Instructions can be “encoded” any number of ways.
- Analog computers have existed for millennia
- Antikythera Mechanism is the oldest known computer:
  - Dated to 1st-3rd Century BCE
  - Astronomical Calculator

Antikythera Mechanism: https://en.wikipedia.org/wiki/Antikythera_mechanism#/media/File:NAMA_Machine_d'Anticyth%C3%A8re_1.jpg

What is Code?

Coding is Like Writing A Recipe

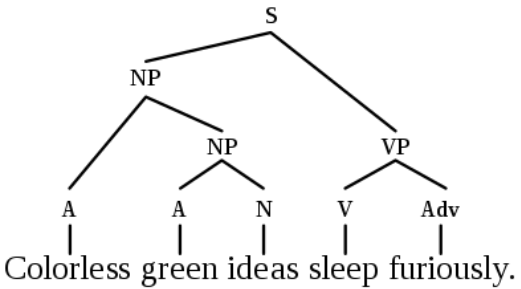
- A recipe consists of a series of steps that a chef needs to do in a specific order in order to create a particular dish.

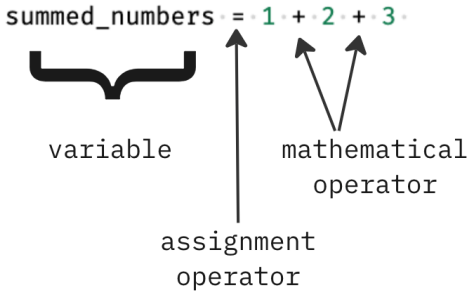


What is Code?

Lines of Code Have Syntax

- Just like a natural language sentence, code has syntax!



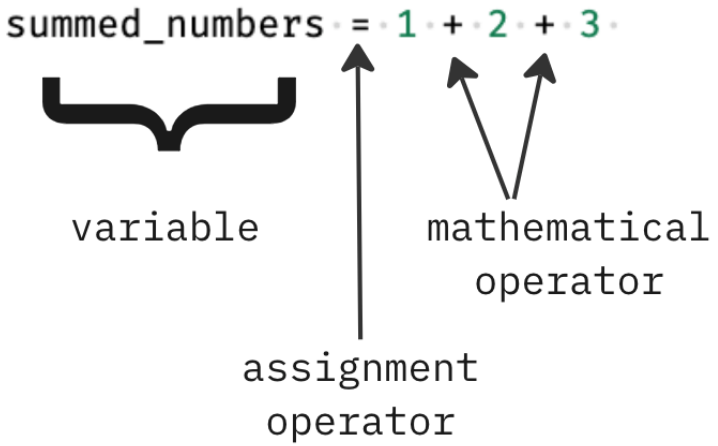


Sentence Syntax: https://en.wikipedia.org/wiki/Syntactic_Structures#/media/File:Cgisf-tgg.svg

What is Code?

Code Syntax is Simple

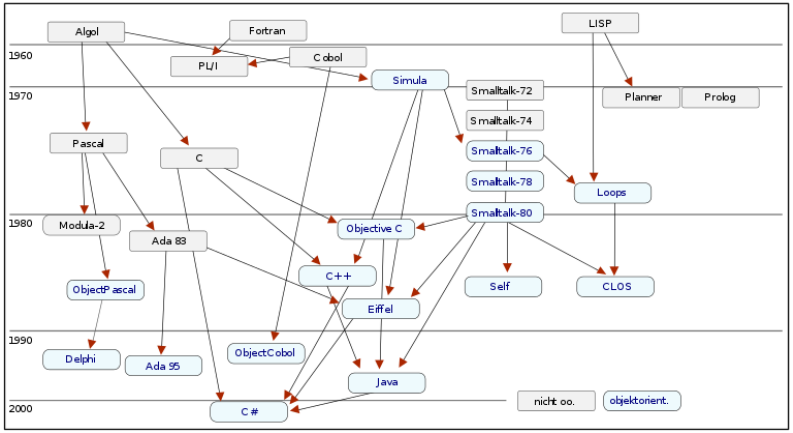
- Unlike natural languages, code is direct and simple.
- No ambiguities at all.



What is Code?

Code can be written in any number of different languages

- Each coding language acts as a way to represent our ideas for our instruction sets.
- Originally written in binary, coding languages have built upon each other and evolved over the past century.



What is Code?

- Text Based Programming Languages
  - C, C++, Rust
  - Java, Scala, C#, Dart, Kotlin, Swift
  - JavaScript, PHP, GO
  - Python, R, MATLAB

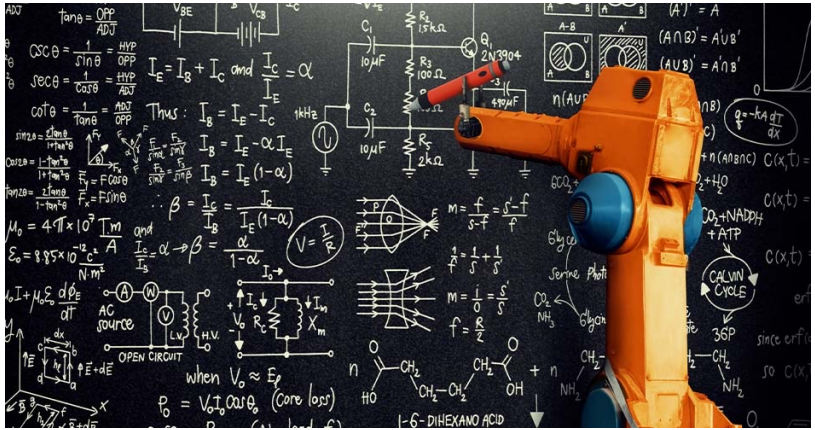
- Visual Programming Languages (VPLs)
  - Automator
  - Blockly
  - Scratch

Illustrations:
Text Based Coding Languages: https://gowithcode.com/top-programming-languages
Automator: https://support.apple.com/guide/automator/welcome/mac
Blockly: https://blockly.games/
Scratch: https://en.scratch-wiki.info/w/images/Scratch_logo.svg

What is Computer Science (CS)?

Coding vs CS vs Software Engineering vs Robotics

- Coding: Writing Code
- Computer Science:
  - The Scientific and mathematical exploration of algorithms, software, hardware, and the theory of how computers work.
- Software Engineering:
  - The building, testing, and deployment of software
- Robotics:
  - The software and hardware engineering of a machine that interacts with the physical world.



Robot building circuit: https://stock.adobe.com/cz/images/robot-arm-ai-analyzing-mathematics-for-mechanized-industry-problem-solving-concept-of-robotics-technology-and-machine-learning-for-automated-manufacturing-process/466071005

Coding: The Basics!

COMPUTER SCIENCE, CODE, AND HUMMINGBIRDS

What is a Program?

Programs are text files that are read line by line.

- A program has a definite start at the beginning of the file and a definite end at the end of the file.
- Programs are executed line by line.

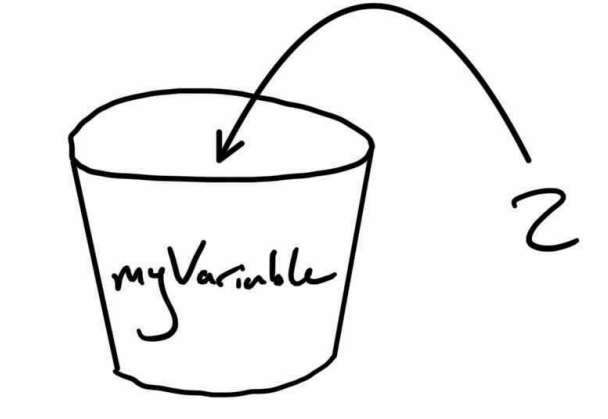
```
1  #!/bin/python
2
3  print("this is the beginning of the program")
4
5  summed_numbers = 1 + 2 + 3
6
7  print("this is the end of the program!")
8
```

What is a Program?

Programs have “variables”

- Think of variables as “containers” for information

```
myVariable = 2
```



What is a Program?

Programs have “variables”

- We can print our variables to see what’s inside them

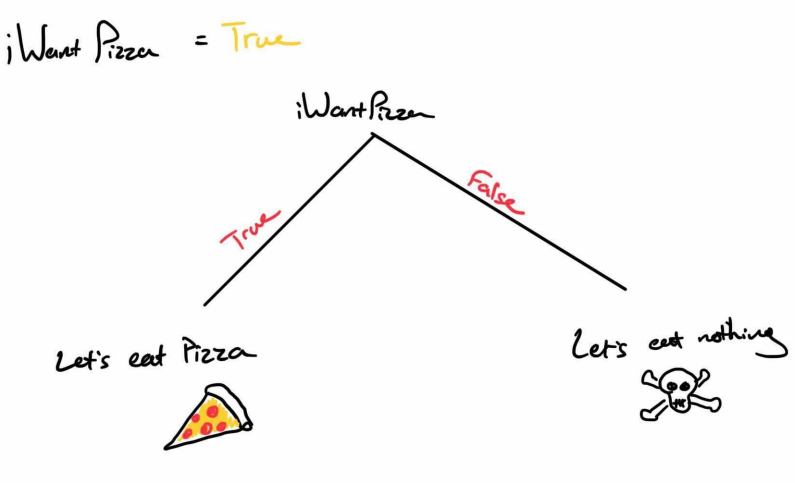
```
print(myVariable)
```

What is a Program?

Programs have “conditionals”

- When we think of programs making decisions, that’s done with a conditional!
- Think of conditionals as a flow chart!

```
iWantPizza = True
if iWantPizza:
  print("Let's eat pizza!")
else:
  print("Let's eat nothing... ☠")
```



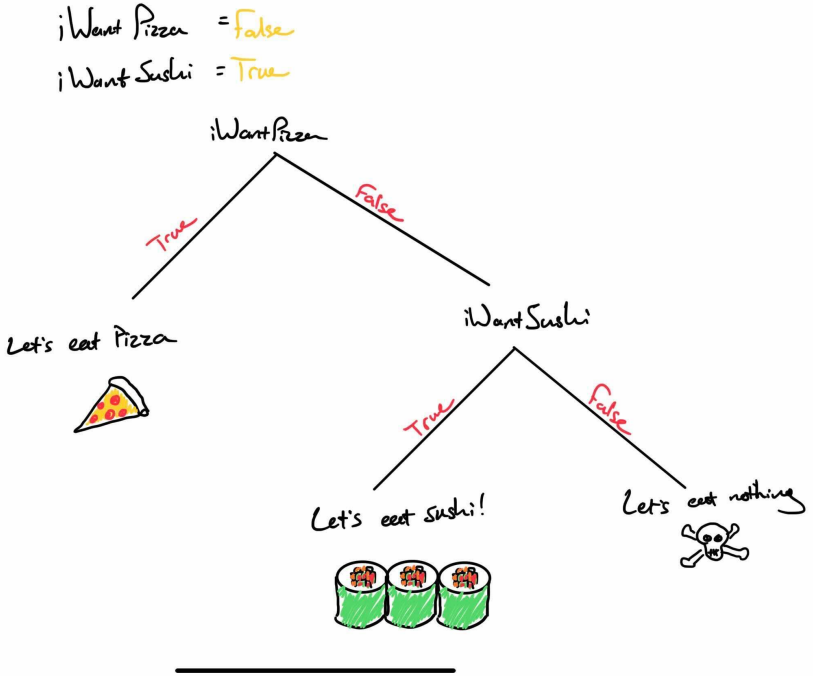
What is a Program?

Programs have “conditionals”

- We can “nest” conditionals to make things even MORE complicated

```
iWantPizza = False
iWantSushi = True

if iWantPizza:
  print("Let's eat pizza! 🍕")
elif iWantSushi:
  print("Let's eat sushi! 🍣")
else:
  print("Let's eat nothing... ☠️")
```



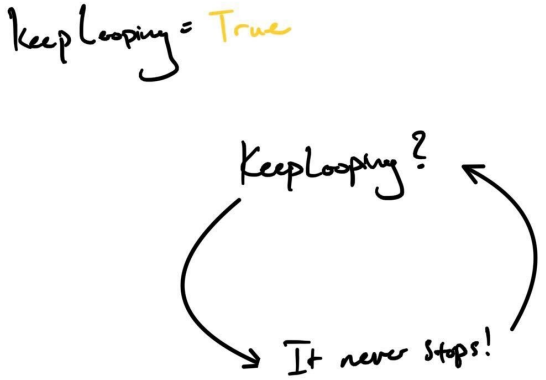
What is a Program?

Programs can have “loops”

- Inside of the code, the program can run in a “loop”
  - That is: The same code gets run over and over and over and over and over and over and over and over… until a stopping point is reached (or possibly forever: “An Infinite Loop”)

```
keepLooping = True

while keepLooping:
    print("It never stops!")
```



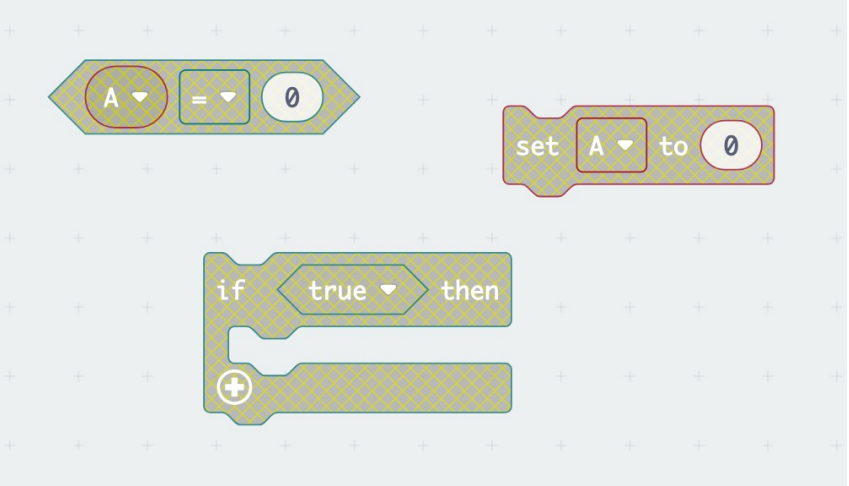
Code Blocks && Virtual Microbits!

COMPUTER SCIENCE, CODE, AND HUMMINGBIRDS

Coding with Blocks

Code Blocks make coding easy

- Text based coding is exceptionally powerful, but extremely brittle.
- A single type can ruin an entire program.
- Code Blocks hold our hands while we code to make sure we don't make silly mistakes!

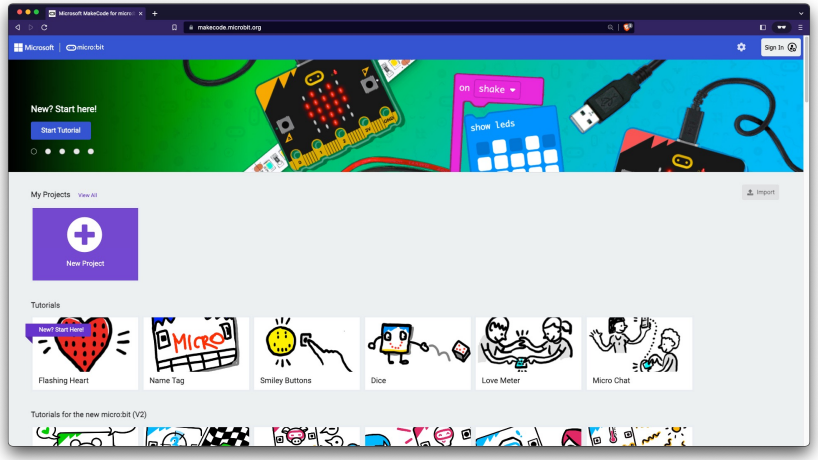


Coding with Blocks

Open a web browser and go to:

https://makecode.microbit.org/

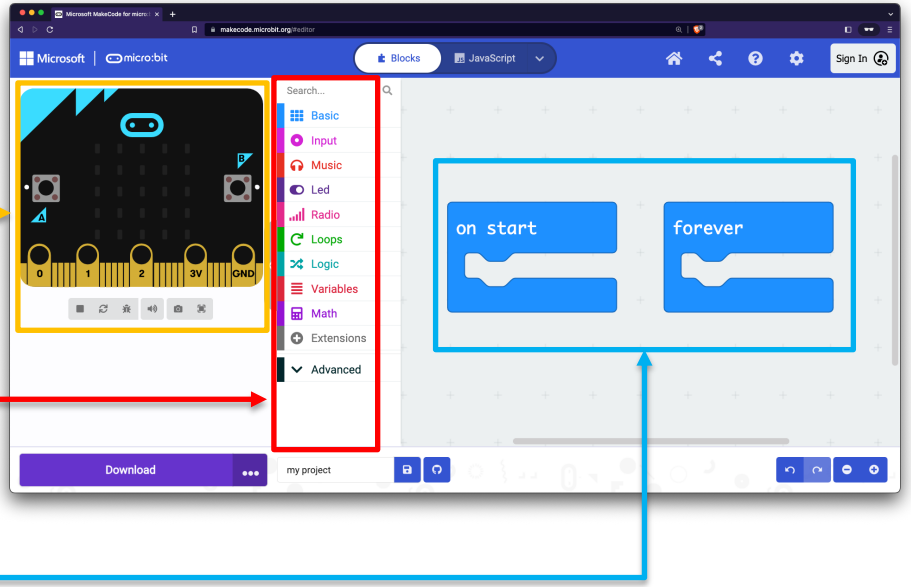
- Makecode offers us a platform to use code blocks quickly and easily!
- Click the “New Project” Button and give your project a name to begin coding!



Coding with Blocks

Makecode's block code editor shows us:

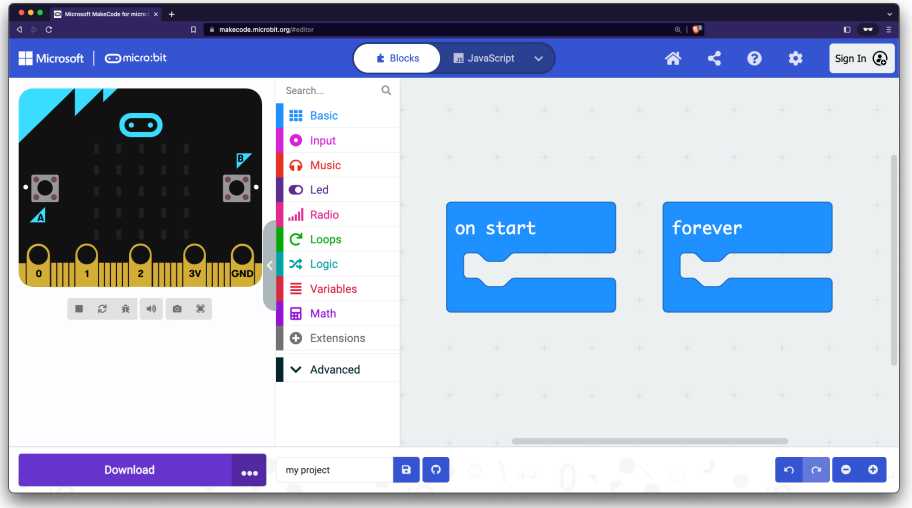
- Virtual Microbit
- Code Block Menu
- Code Block Editor



Coding with Blocks

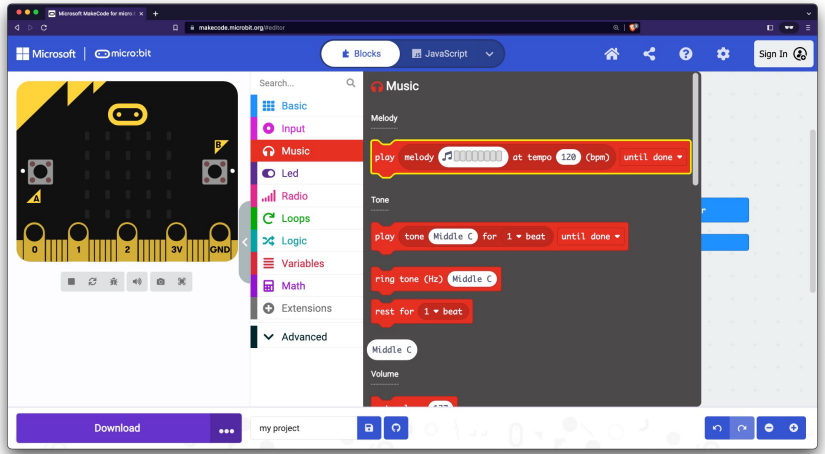
Unlike text based code, your code editor has two main blocks:

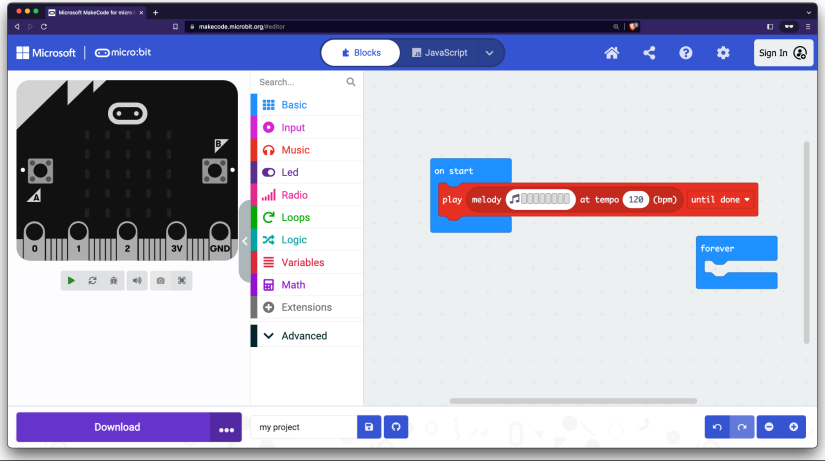
- On Start
  - Runs code sequentially from start to end.
- Forever
  - Loops code over and over.



Coding with Blocks

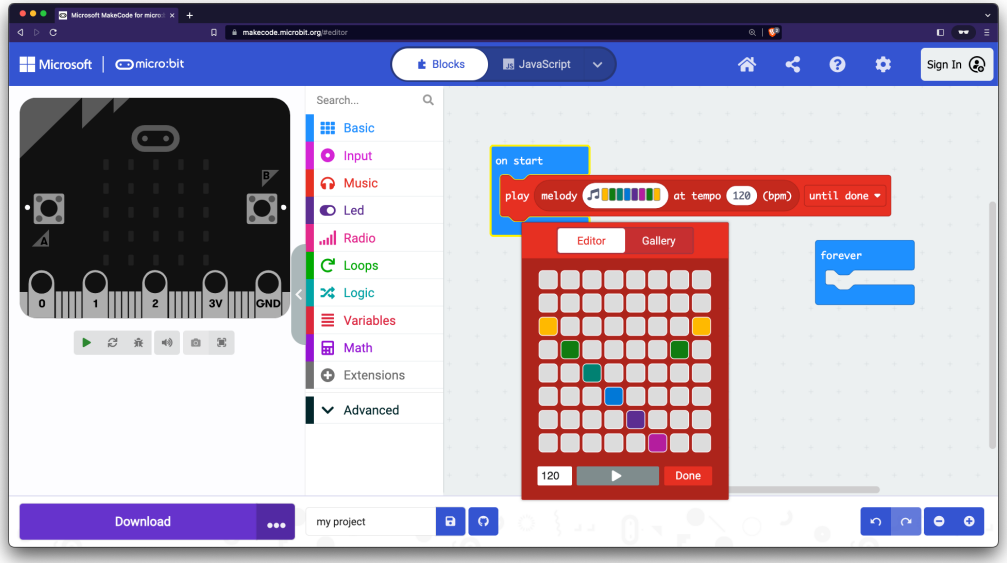
- Click the “Music” tab and select “play”
- Drag and drop “play” into “on start”





Coding with Blocks

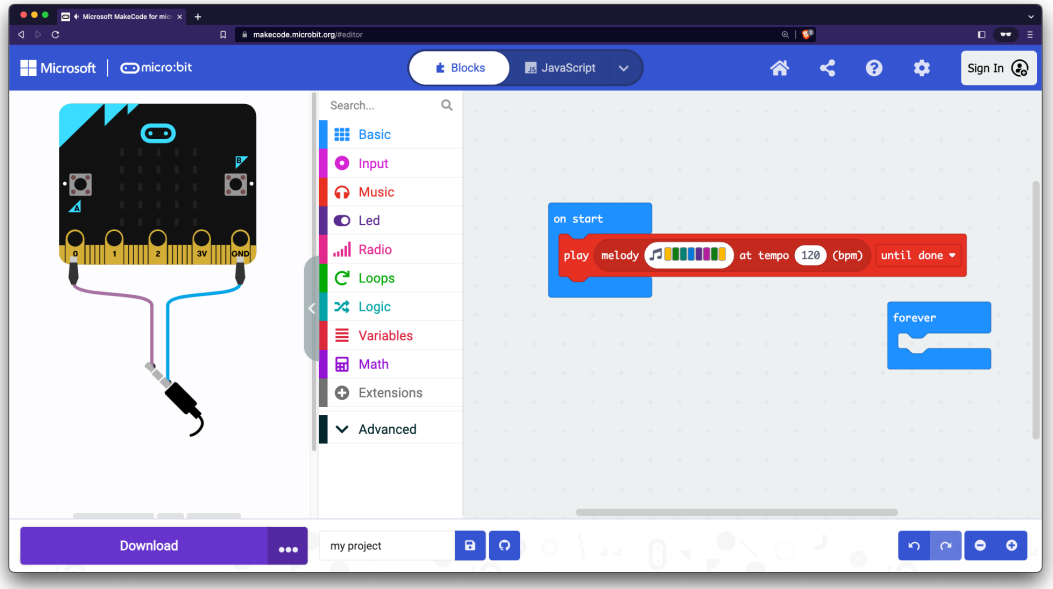
Select a tune to play by clicking the music nodes near “melody”



Coding with Blocks

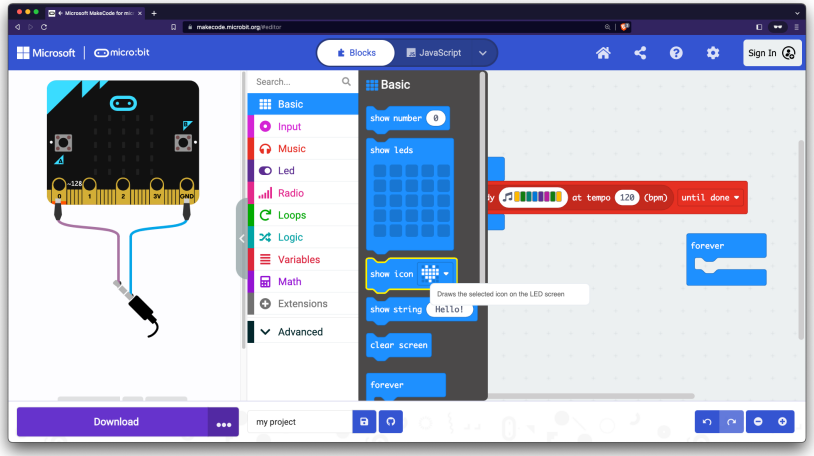
The Virtual microbit is now:

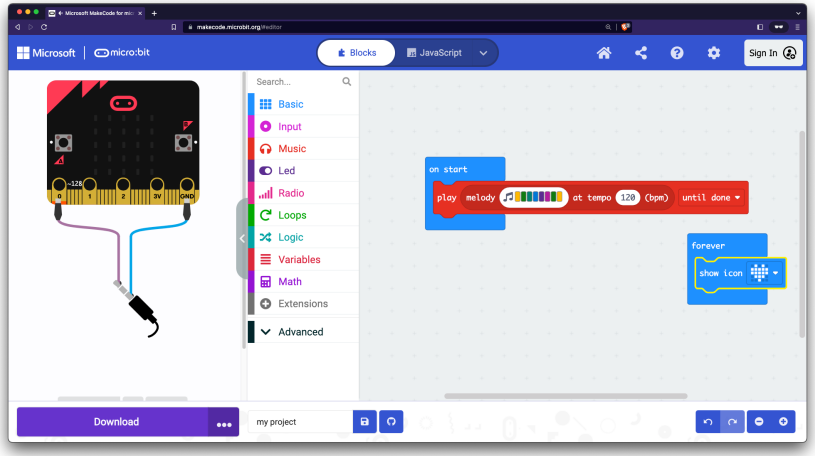
- Playing music (only once) through your computer's speakers
- Illustrating the pins required to physically play music



Coding with Blocks

- In the “Basic” tab, select “show icon”
- Drag and drop “show icon” into “forever”

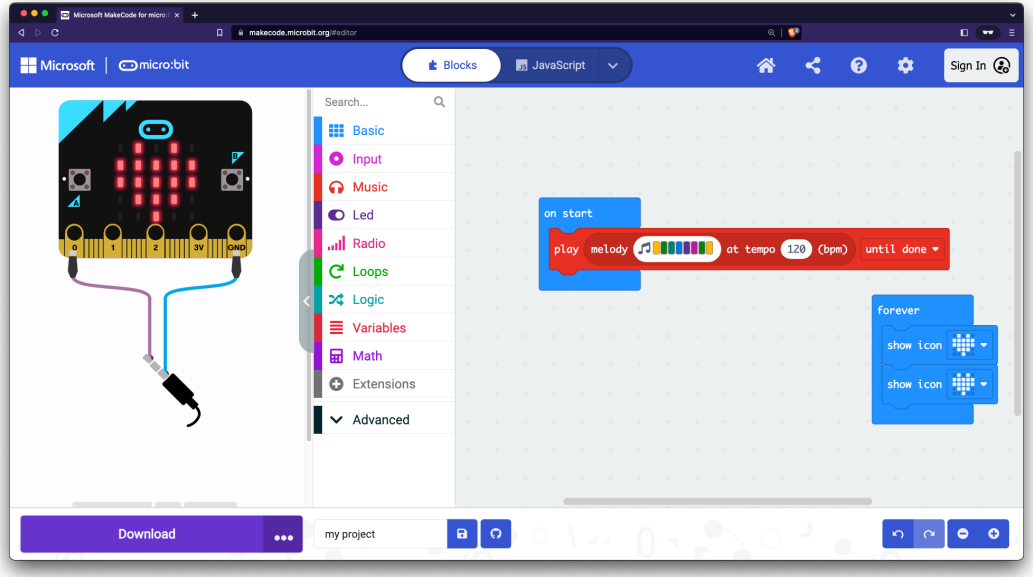




Coding with Blocks

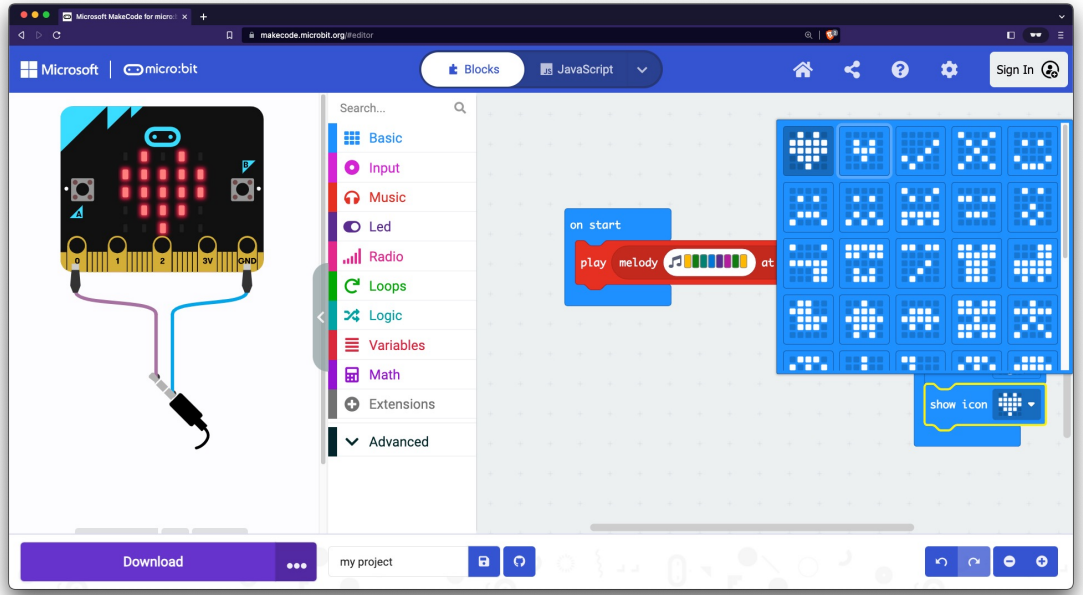
Add another “show icon” to your forever block.

- You can do this by
  - Selecting another show icon block
  - Right clicking the show icon block you already have and selecting “duplicate”



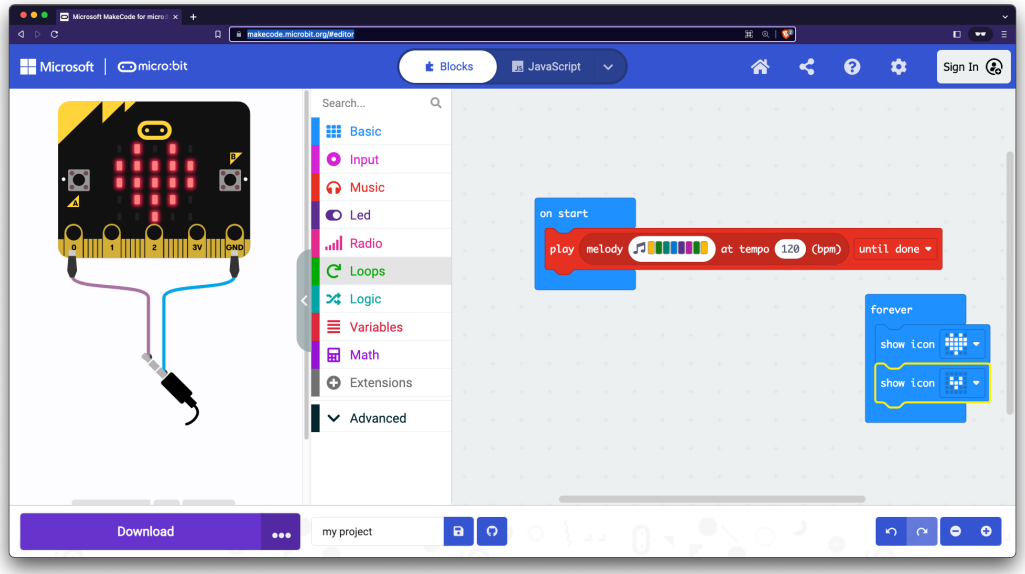
Coding with Blocks

Click the dropdown on the second “show icon” and select another icon.



Coding with Blocks

Now, after your virtual microbit plays the melody only once, it will alternate forever between your two icons.

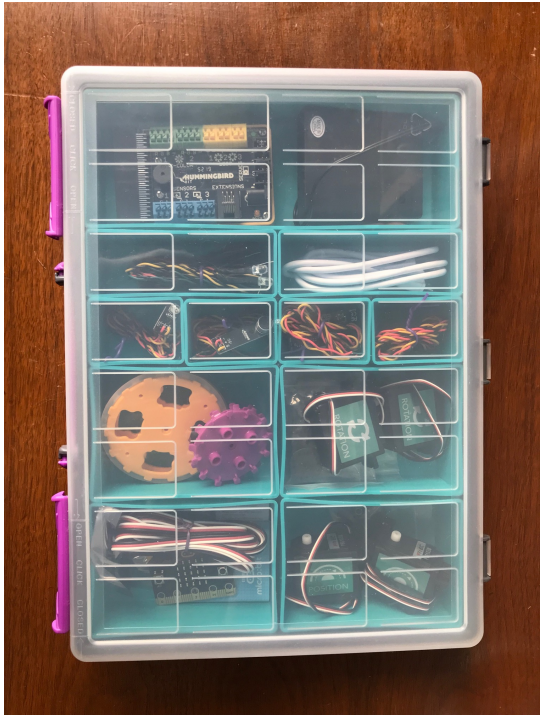


Hummingbird Kits

COMPUTER SCIENCE, CODE, AND HUMMINGBIRDS

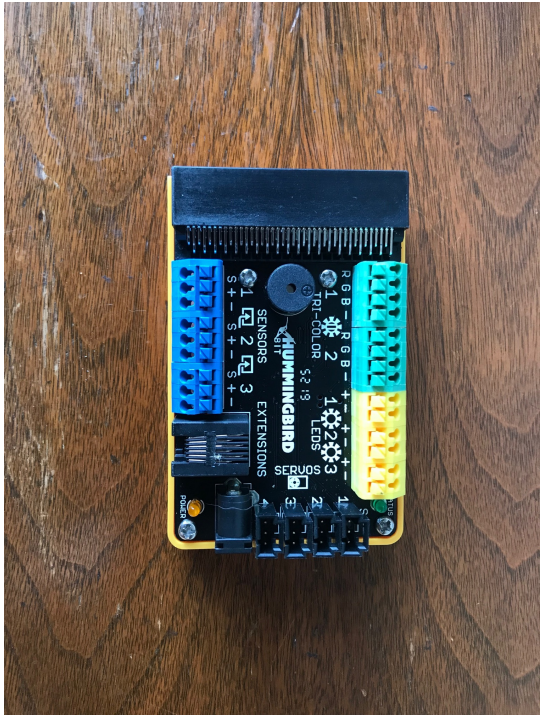
Hummingbird Kits

Birdbrain technologies offers the hummingbird bit and accessories as a straightforward introduction to coding and robotics



Hummingbird Kits

- Each hummingbird kit contains:
  - Hummingbird bit board



Hummingbird Kits

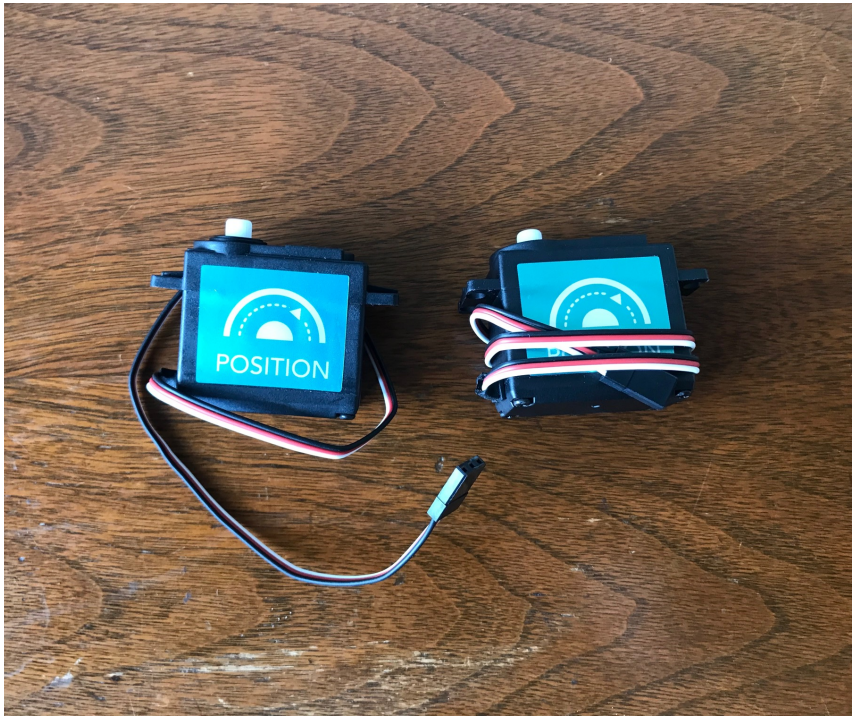
- Each hummingbird kit contains:
- Battery Pack

Hummingbird Kits

- Each hummingbird kit contains:
- Microbit controller

Hummingbird Kits

- Each hummingbird kit contains:
- 2 Positional Servos

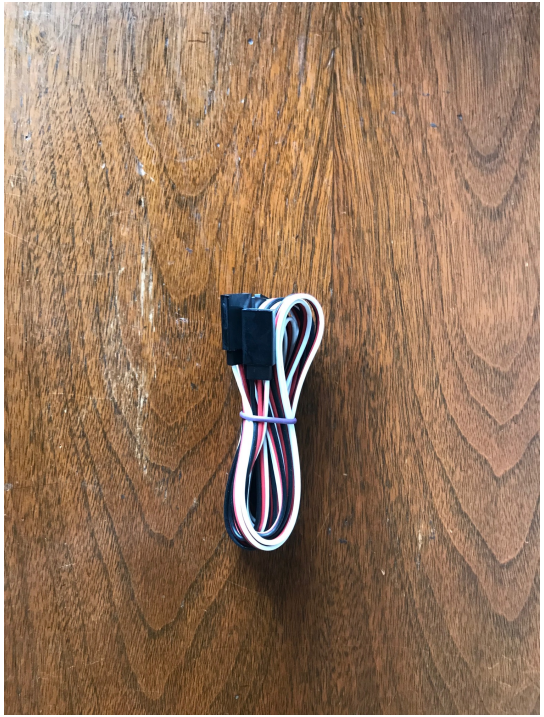


Hummingbird Kits

- Each hummingbird kit contains:
- 2 Rotational Servos

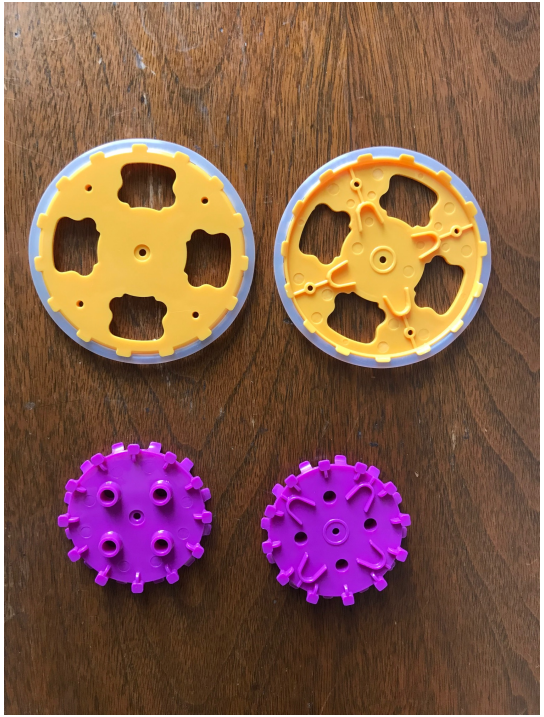
Hummingbird Kits

- Each hummingbird kit contains:
- Extension Cables



Hummingbird Kits

- Each hummingbird kit contains:
- Wheels



Hummingbird Kits

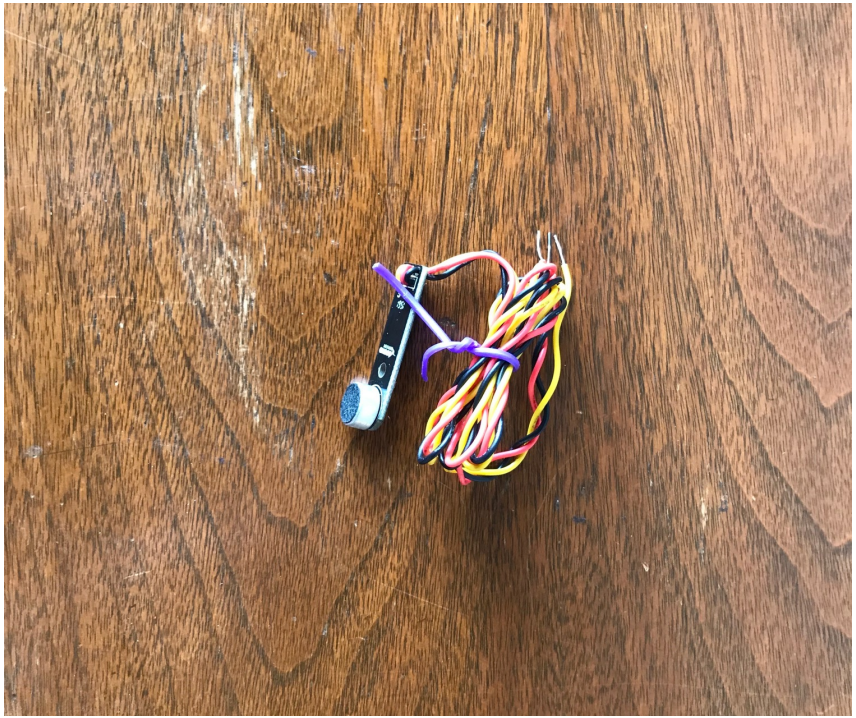
- Each hummingbird kit contains:
- 3 Single Color LEDs:
  - Red
  - Green
  - Yellow
- 2 Tri Color LEDs

Hummingbird Kits

- Each hummingbird kit contains:
- Light Sensor

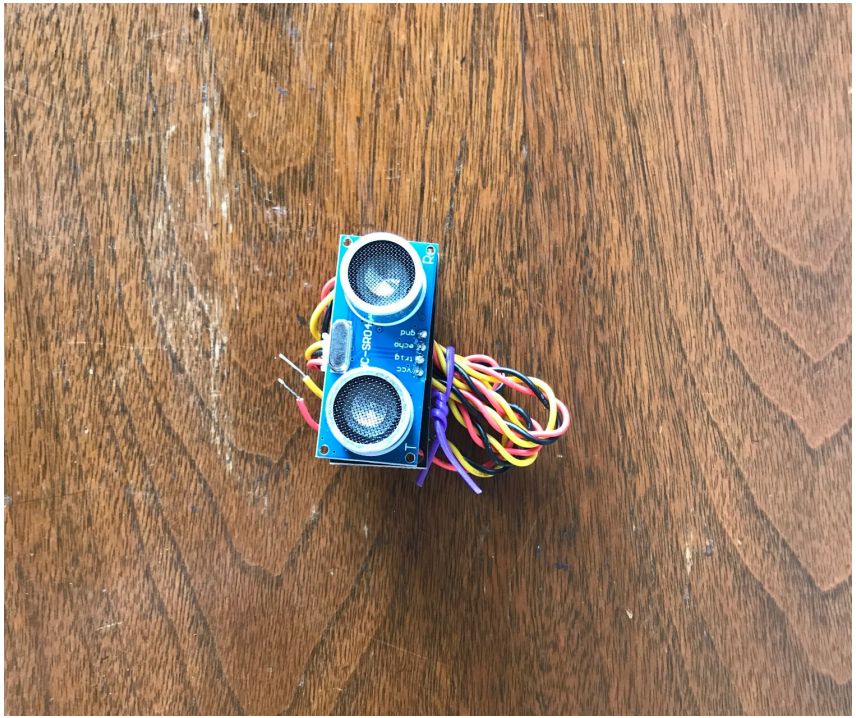
Hummingbird Kits

- Each hummingbird kit contains:
- Sound Sensor



Hummingbird Kits

- Each hummingbird kit contains:
- Distance Sensor



Hummingbird Kits

- Each hummingbird kit contains:
- Dial Sensor

Hummingbird Kits

- Each hummingbird kit contains:
- Dial Sensor

Hummingbird Kits

Each hummingbird kit contains:

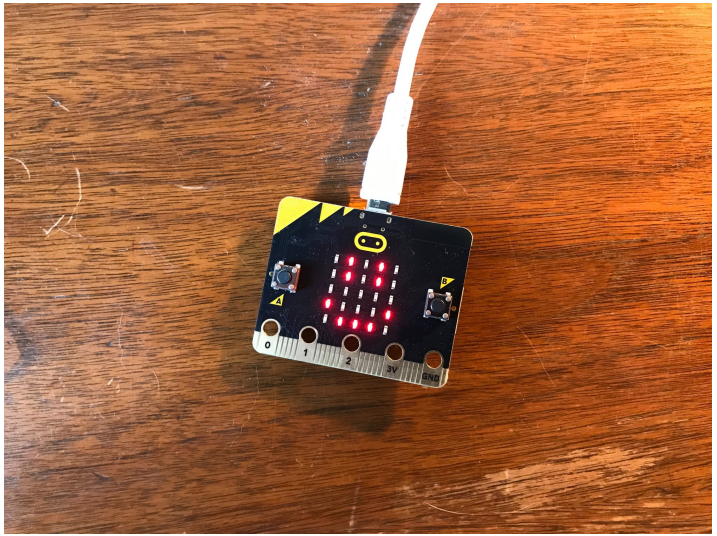
- Placement Tool

Code Blocks && Microbits

INTRODUCTION TO:
CODING AND COMPUTER SCIENCE

Hummingbird Kits: Coding

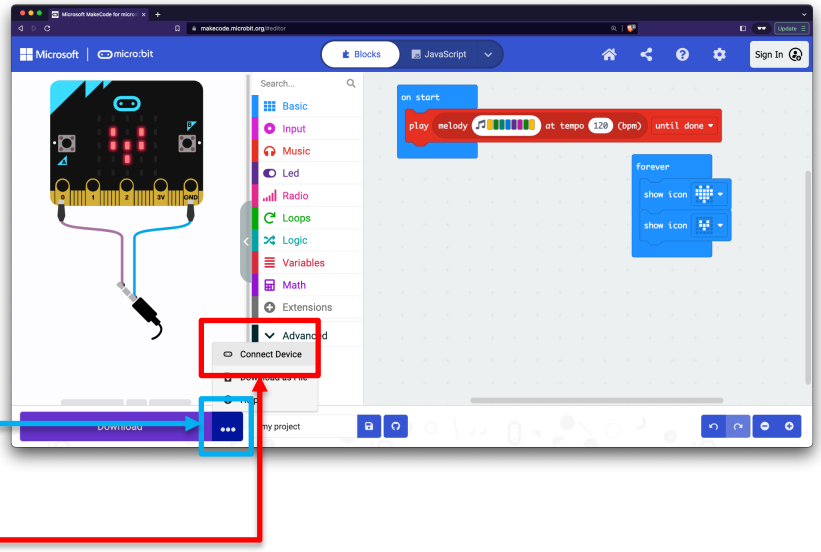
Connect the microbit to your computer with the firewire to usb cable



Hummingbird Kits: Coding

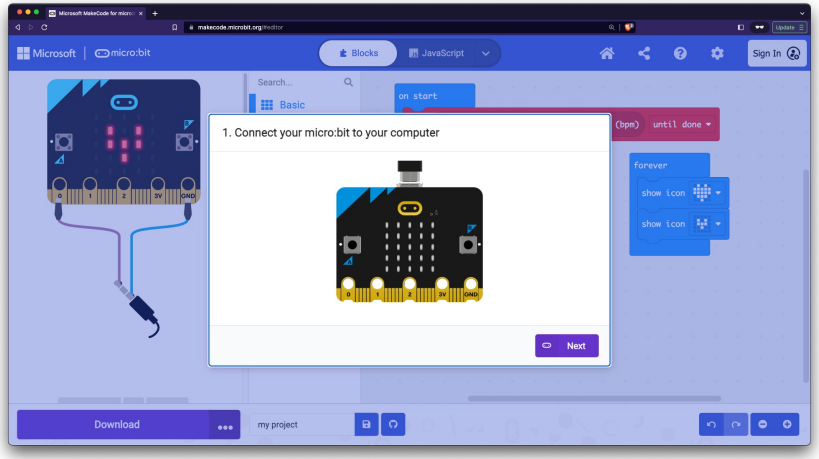
Connecting the device:

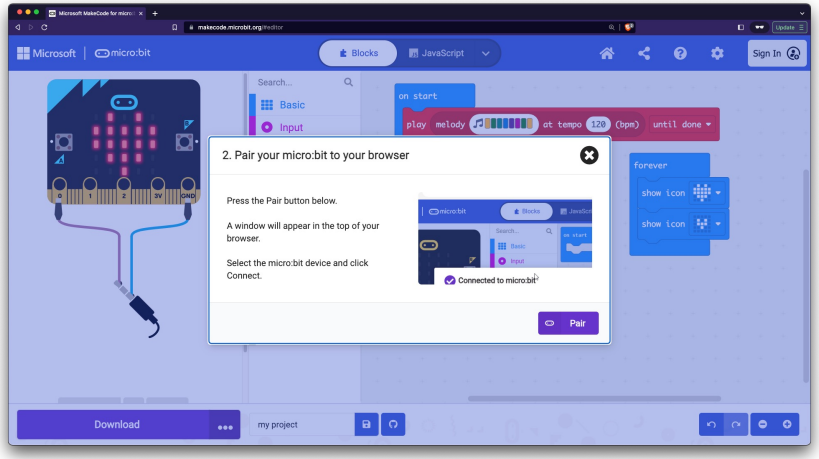
1. Click the ellipses under the virtual microbit
2. Click “connect the device”



Hummingbird Kits: Coding

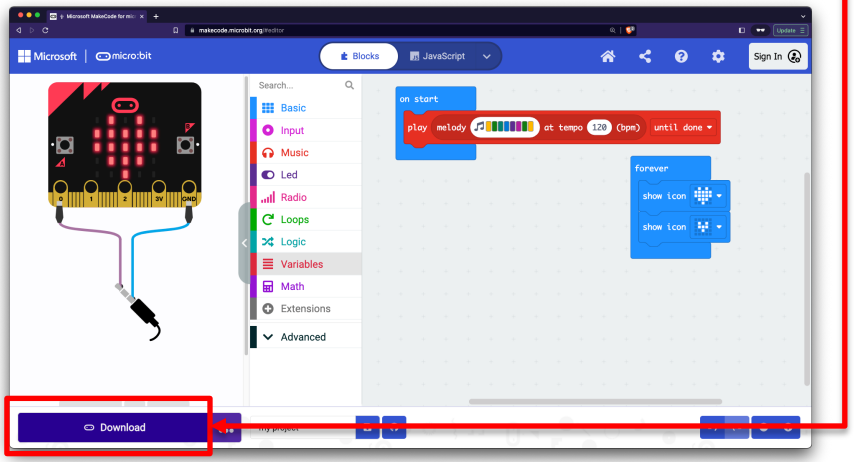
- Ensure your microbit is connected
- Follow the prompt and pair.



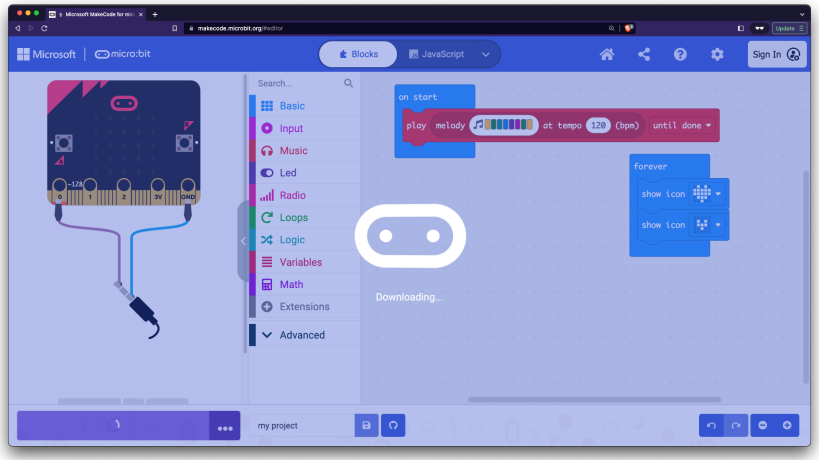


Hummingbird Kits: Coding

- To download your blocks to your bit, click download!



- It may take a second, be patient!

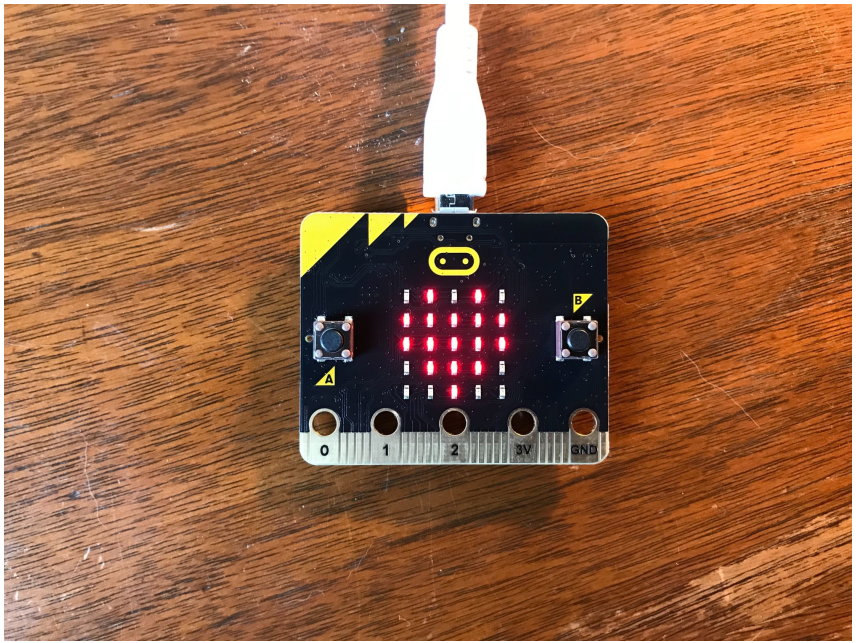


Hummingbird Kits: Coding

Now our microbit's LEDs reflect our virtual microbit!

…but why no sound?

- microbits do not contain a speaker! (but don't worry, we have one!)

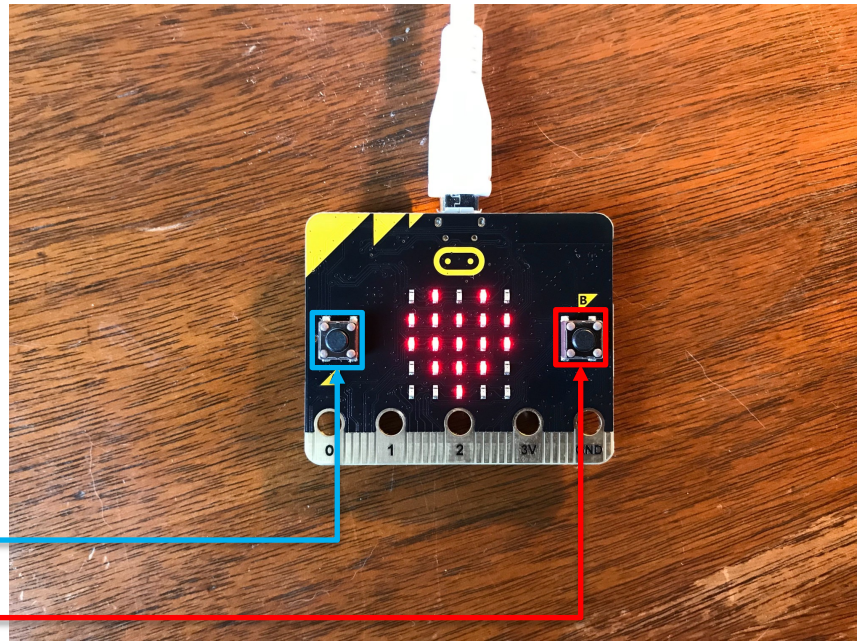


Hummingbird Kits: Coding

BUT FIRST! Let's explore conditionals!

Microbit controllers have built in buttons that can act as input!

- A
- B

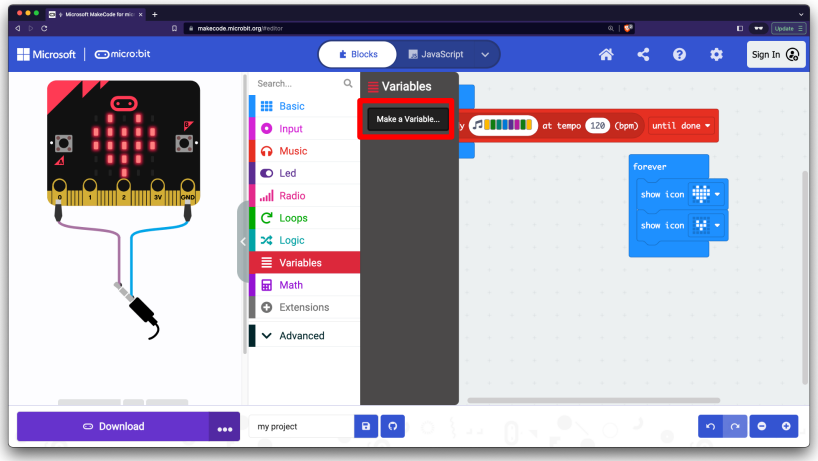


Hummingbird Kits: Coding

To use these buttons we will need to use:

- Variables
- Conditionals

So: Click the variables tab and select “make variable” and call it ”A_button”

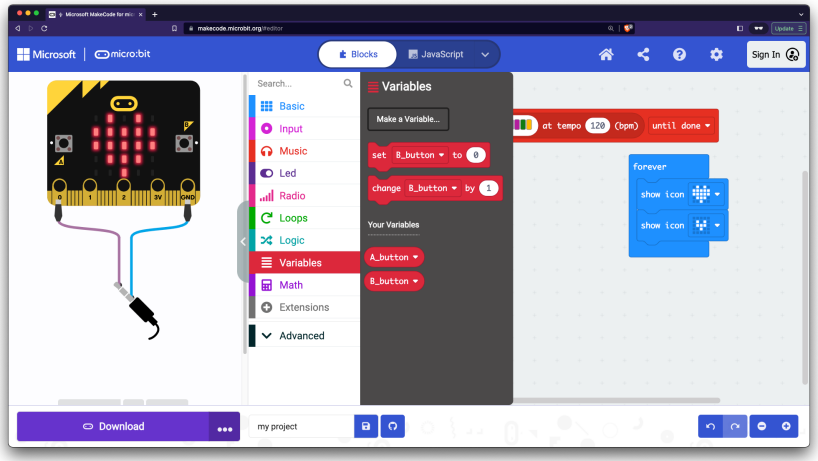


Hummingbird Kits: Coding

While we are here, click ”make variable” again and create a variable “B_button”

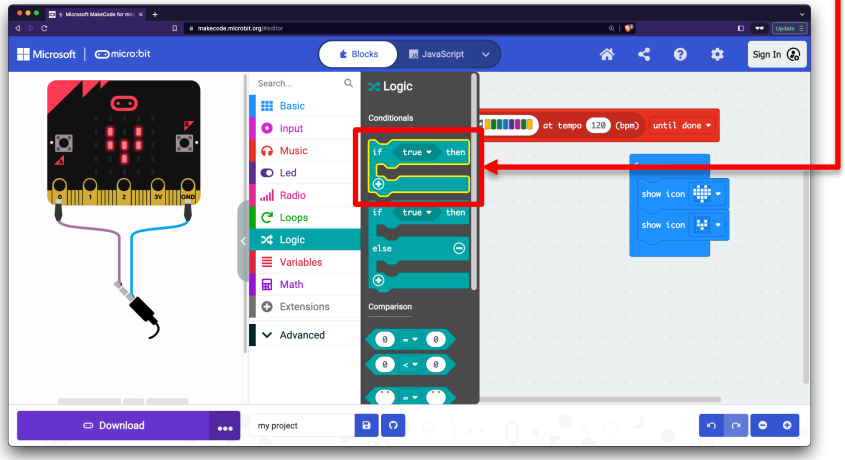
We should now have two variables:

- A_button
- B_button

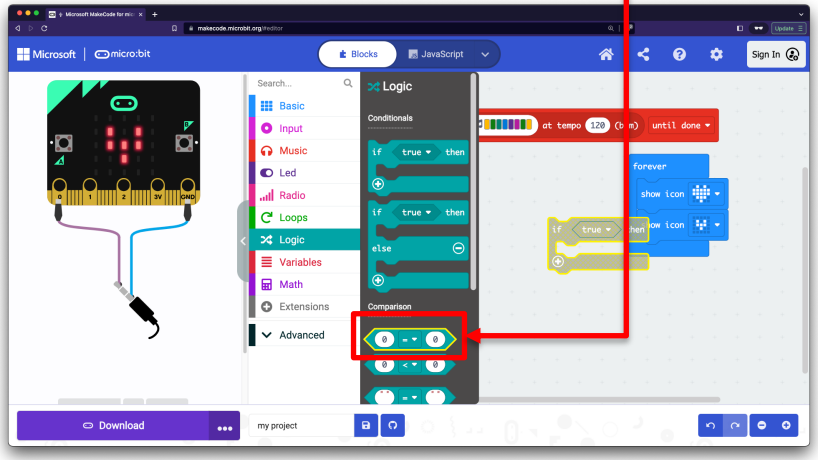


Hummingbird Kits: Coding

- Click the “logic” tab and select the first “if” block

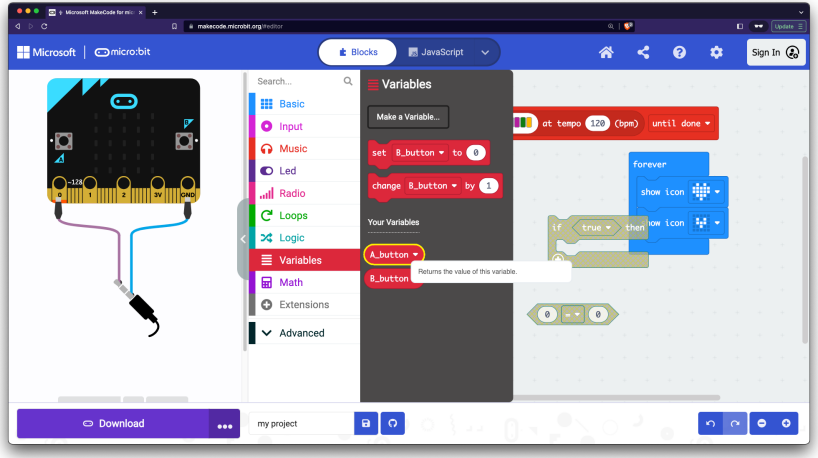


- Click the “logic” tab and select the first “comparison” block

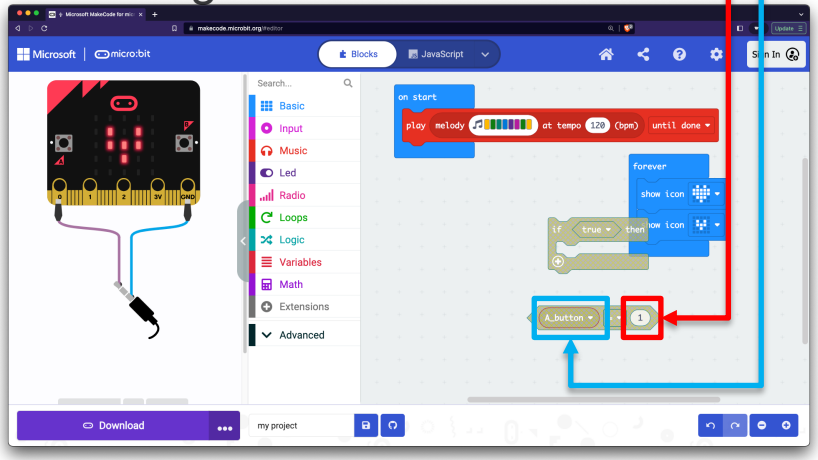


Hummingbird Kits: Coding

- Click the variable tab and select the A_button variable

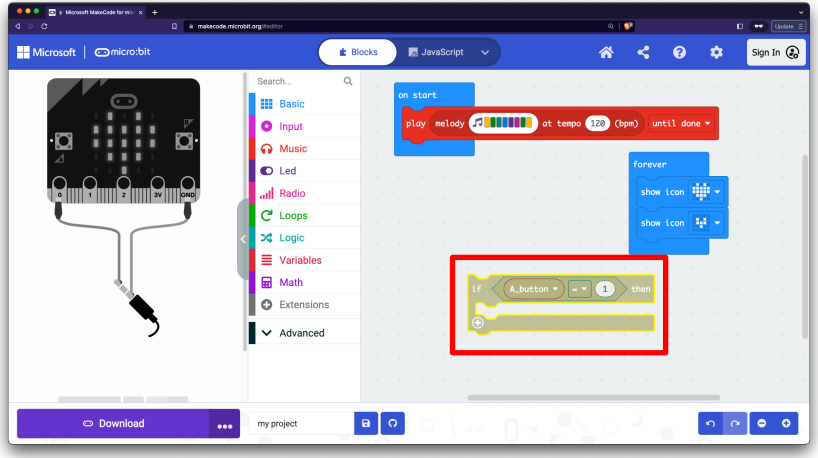


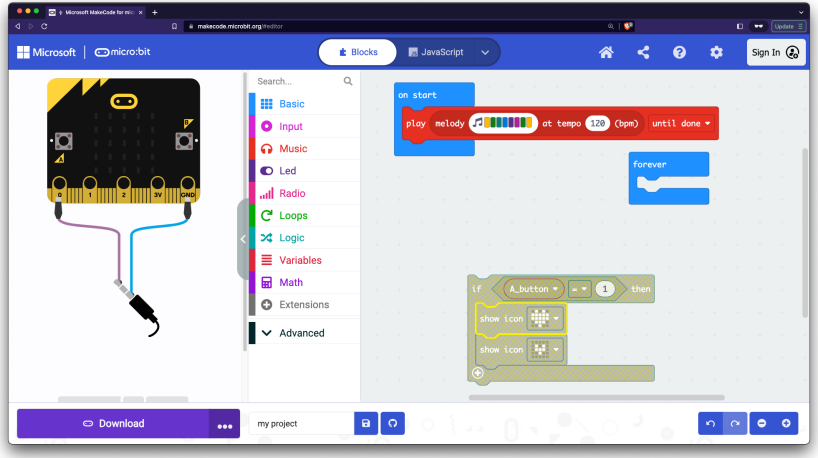
- Drag the A_button variable into the first circle of the comparison block
- Change the second value to 1.



Hummingbird Kits: Coding

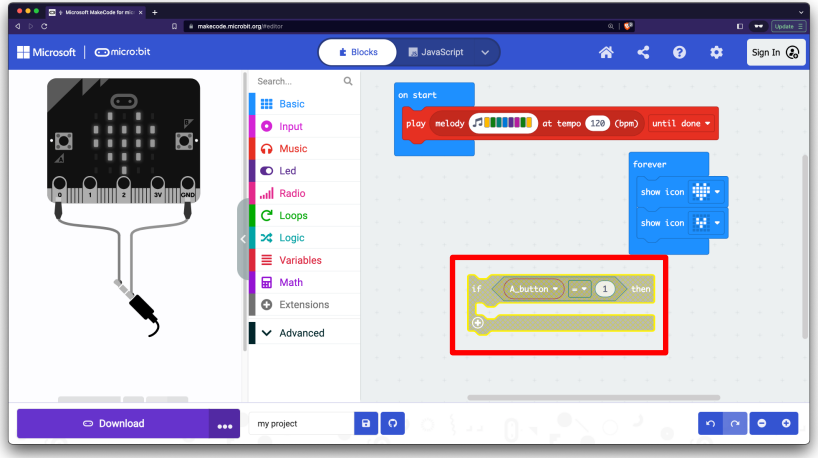
- Drag the comparison block into the matching space in the if block
- Drag the icons from the forever loop into the if block



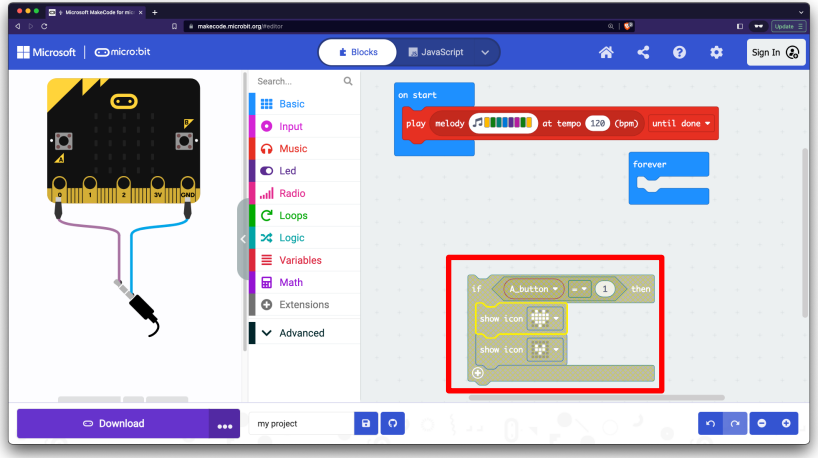


Hummingbird Kits: Coding

- Drag the comparison block into the matching space in the if block



- Drag the icons from the forever loop into the if block

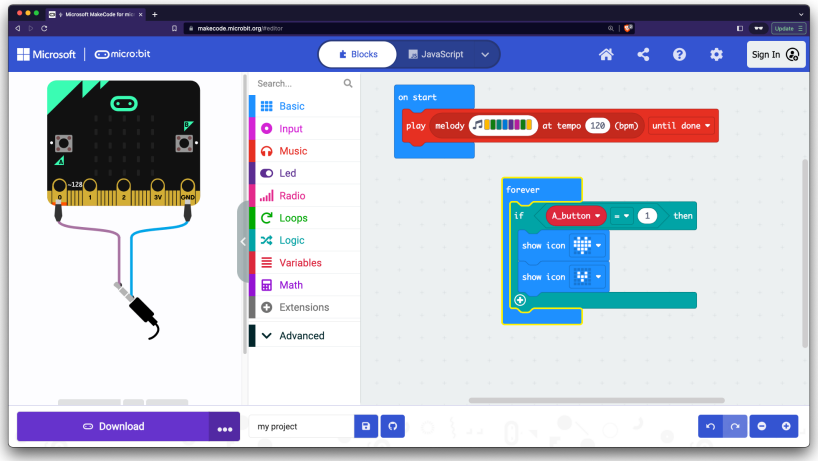


Hummingbird Kits: Coding

Finally! Drag the if block into the forever loop

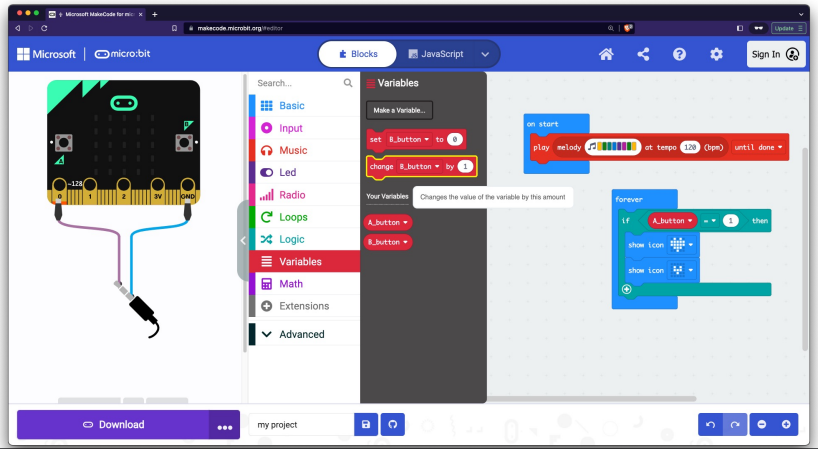
Now our virtual LEDs won't turn on until we set A_button to 1!

But how do we set our A_button to 1?

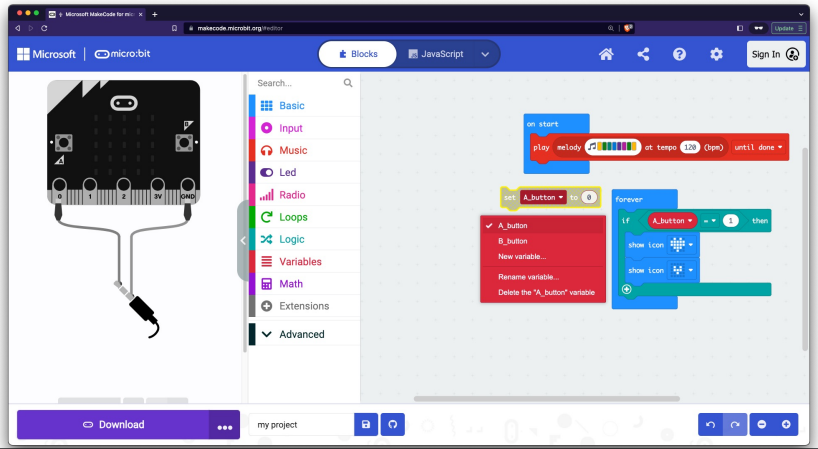


Hummingbird Kits: Coding

- We need to initialize A and B.
- Click the “Variables” tab and select “set ‘B_button’ to ‘0’” twice (one for A_button and one for B_button)

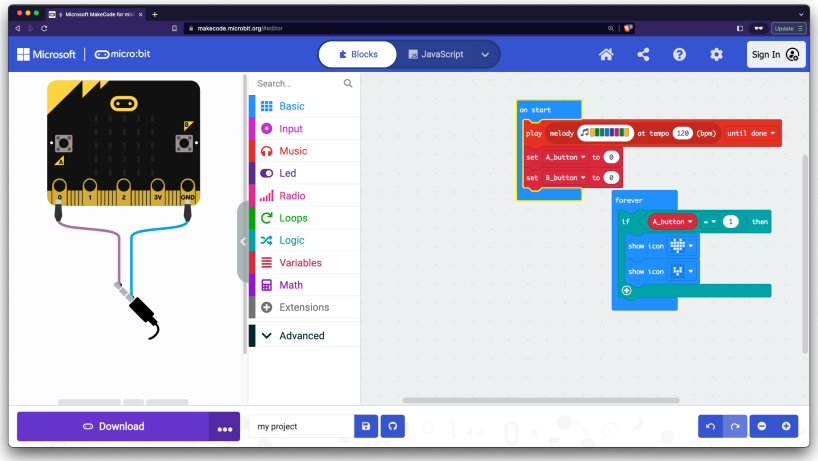


- Set one of your buttons to A_button.



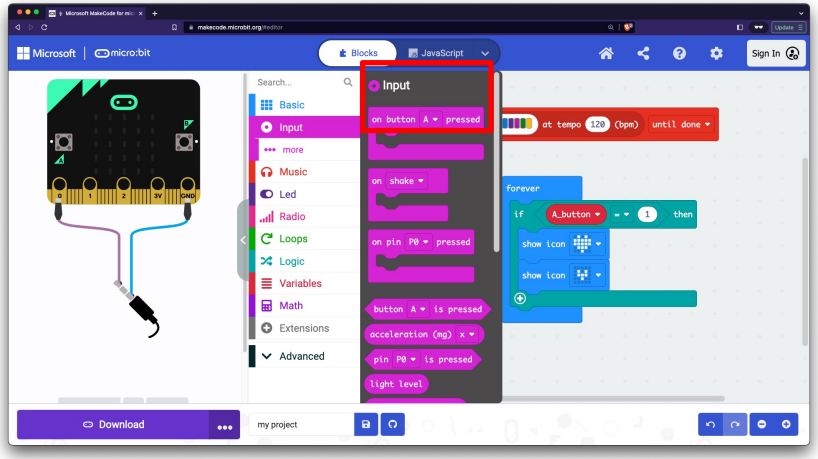
Hummingbird Kits: Coding

Drag and drop your buttons into your “on start” block!

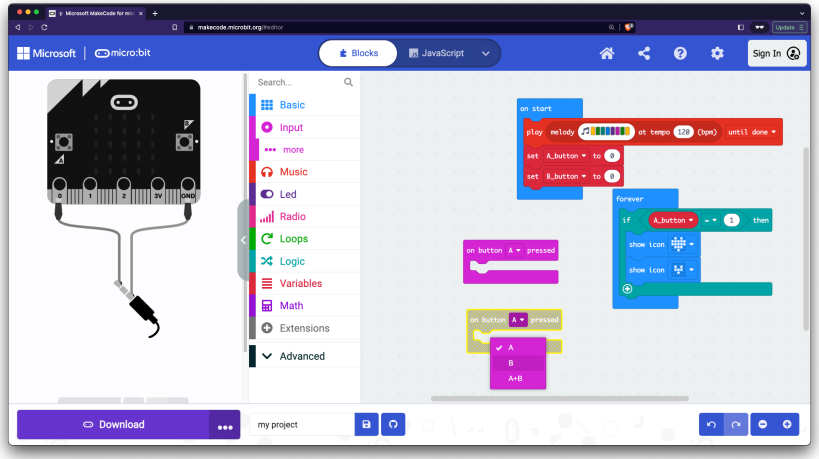


Hummingbird Kits: Coding

- Next, click the “input” tab and select the “On button ‘A’ pressed” block twice (one for A_button and one for B_button)

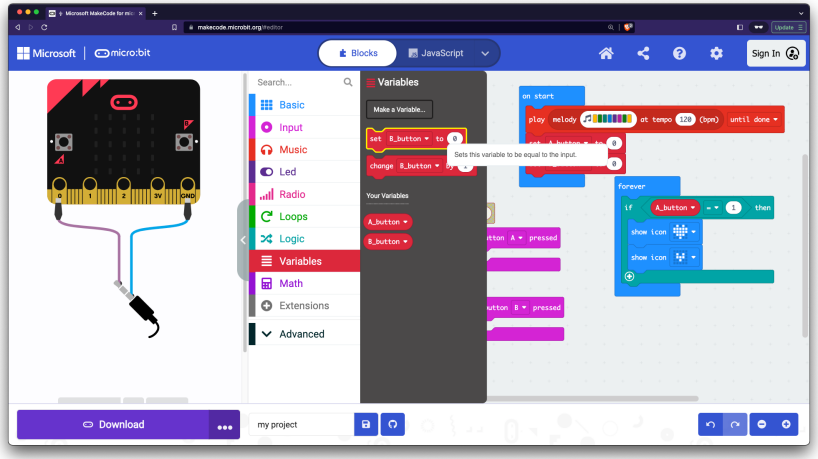


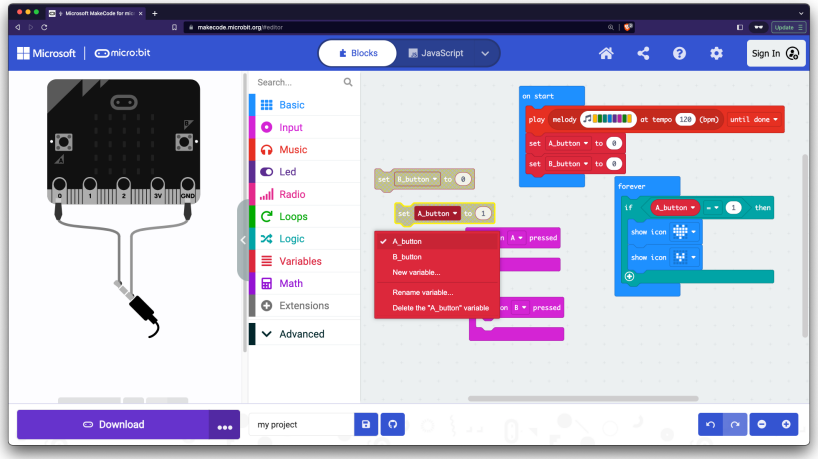
- Set one button block to “B_button”



Hummingbird Kits: Coding

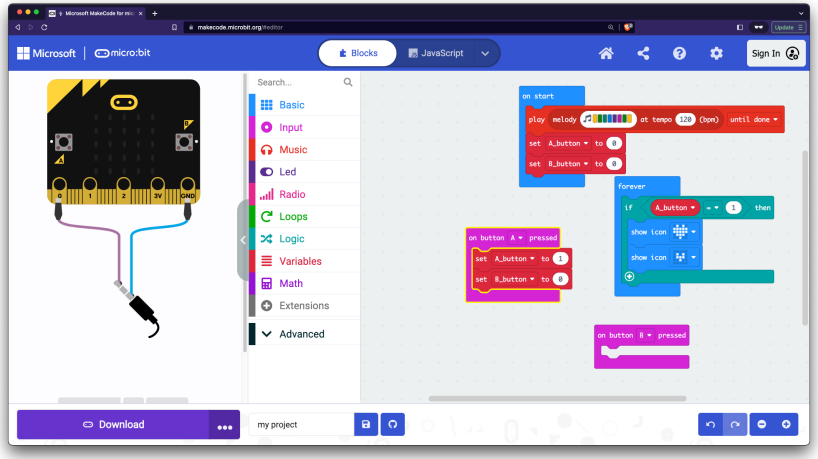
- Click the “variable” tab and select the “set ‘B’ button to 0” twice.
- Change one “set ‘B’ button to 0” block to “set ‘A’ button to 1”



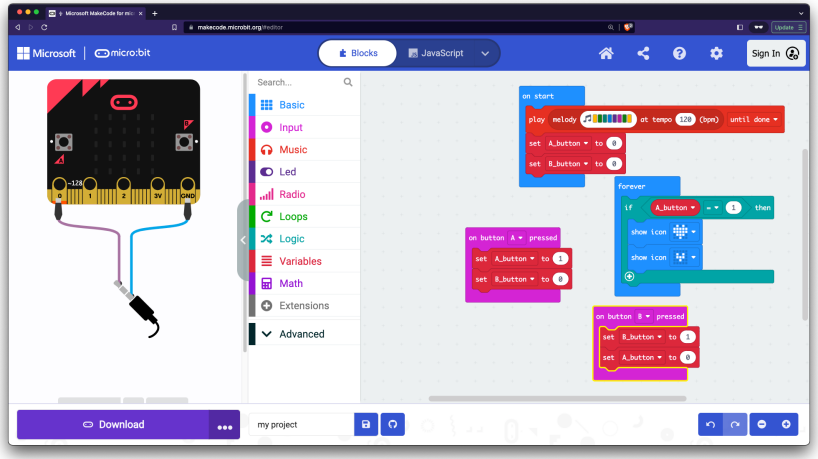


Hummingbird Kits: Coding

- Drag both “set * button to *” blocks into the “On button ‘A’ pressed” block:



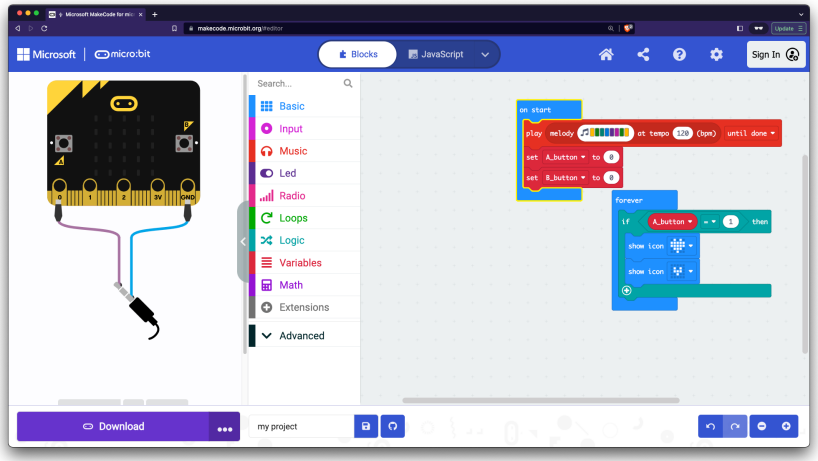
- Do the same thing, but in “on ’B_button’ pressed” set:
  - “set ‘A_button’ to 0”
  - “set ‘B_button’ to 1”



Hummingbird Kits: Coding

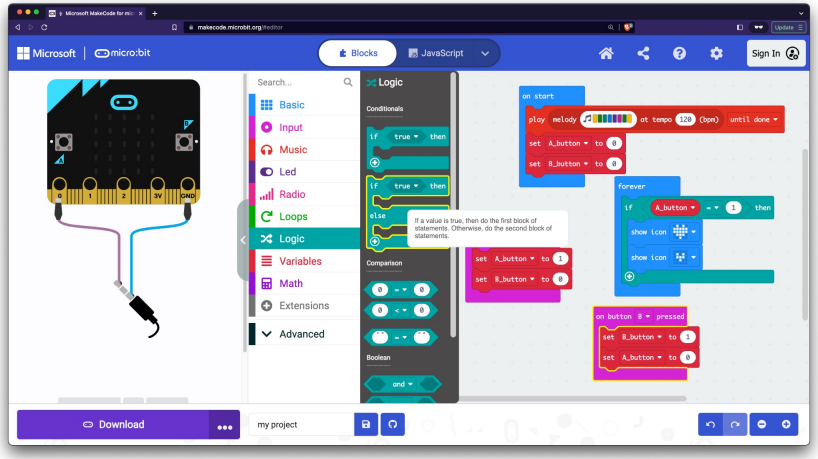
Now, when we press our A button, our LEDs turn on and cycle

But, when we press our B_button, our LEDs don't turn off, they just stop cycling

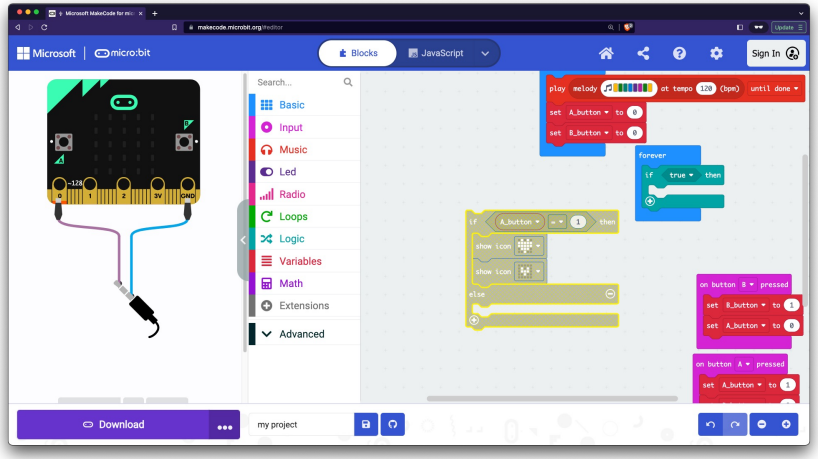


Hummingbird Kits: Coding

- Click the ”logic” tab and select the “if else” block

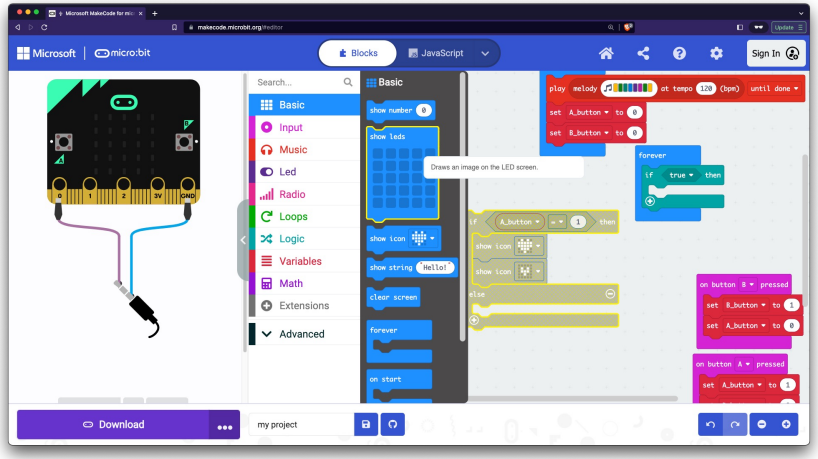


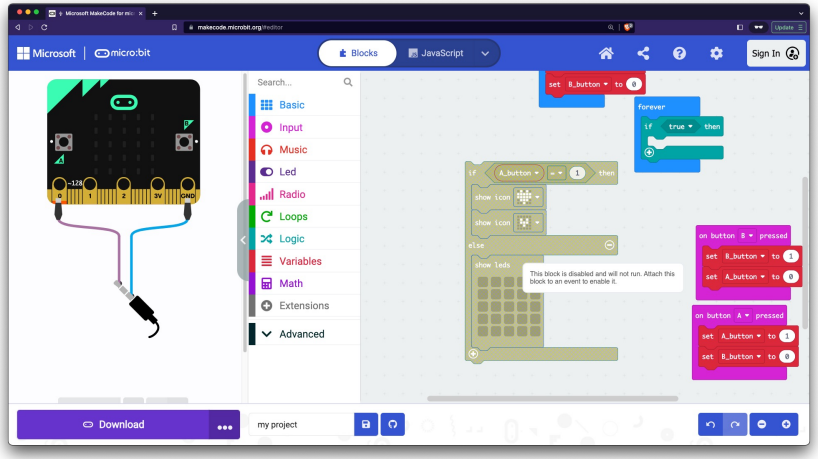
- Drag the “show icon” and the “A button = 1” comparison block to the “if else” block.



Hummingbird Kits: Coding

- In the “basic” tab, select the “show leds” block
- Drag the empty “show leds” block into the else section of the “if else” block

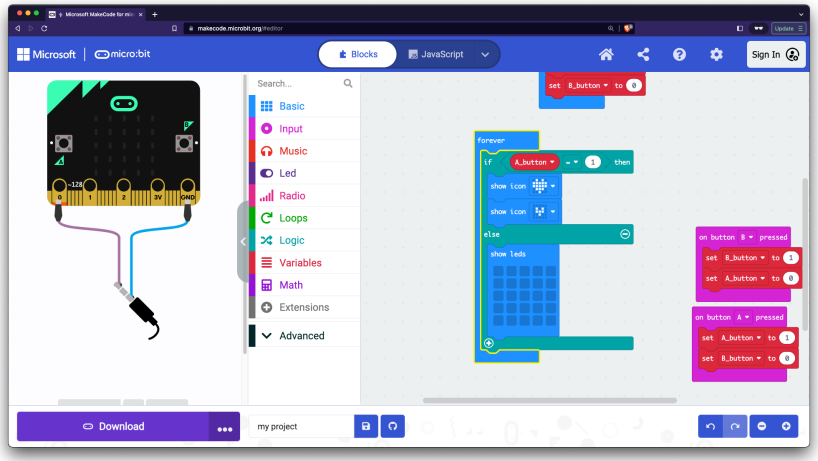




Hummingbird Kits: Coding

Replace the old “if” block with the new “if else” block!

Now you can turn your LEDs on and off with the press of a button!



Code Blocks && Hummingbird Bits

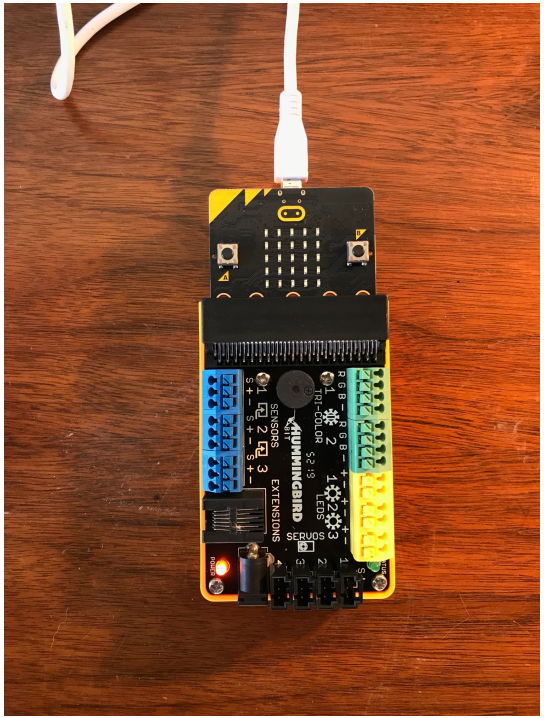
INTRODUCTION TO:
CODING AND COMPUTER SCIENCE

Hummingbird Kits: Setup

Now let's connect our microbit to our hummingbird bit!

Insert the microbit LED side up into the hummingbird bit's slot

(You might hear a sound!)

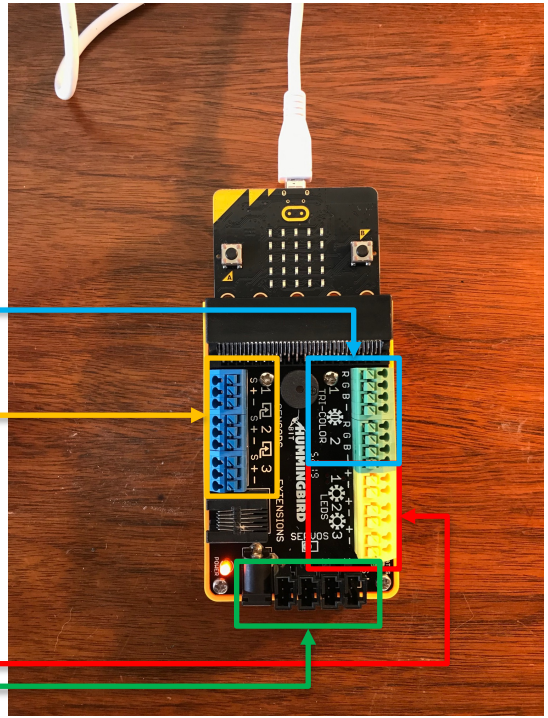


Hummingbird Kits: Setup

The hummingbird bit allows for us to interface with the components in the hummingbird kit!

The pins are labeled on the hummingbird bit for:

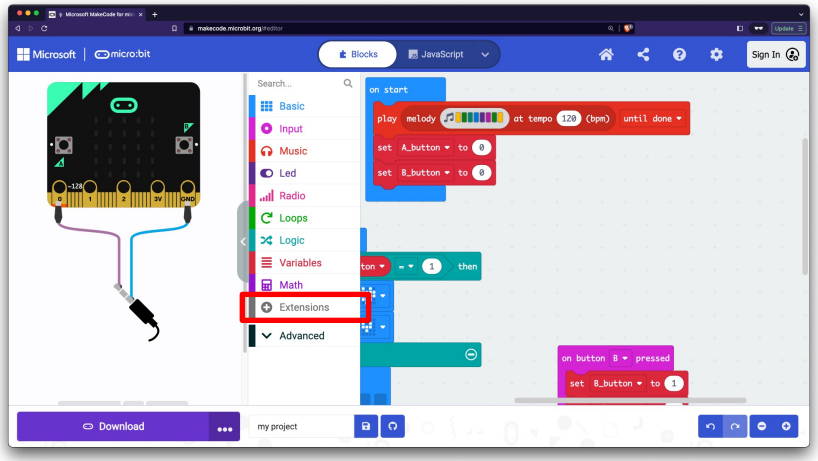
- Sensors
- Tri-color LEDs
- Single Color LEDs
- Servos



Hummingbird Kits: Setup

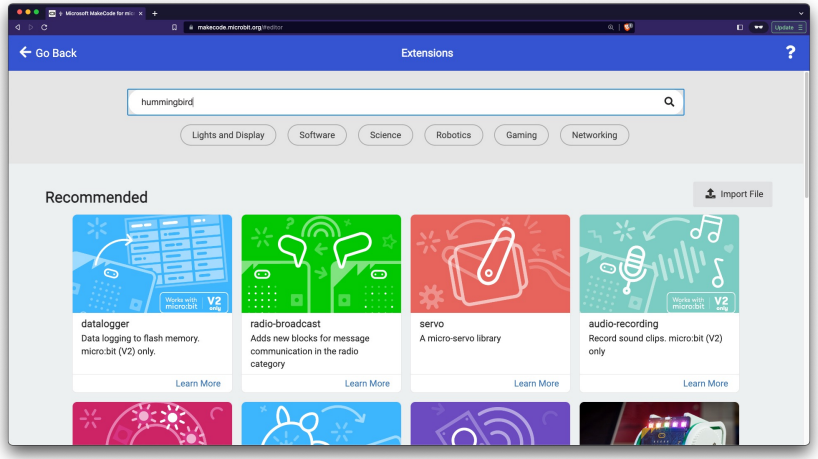
For the microbit controller to interact fully with the hummingbird bit, we need to bring in a special block

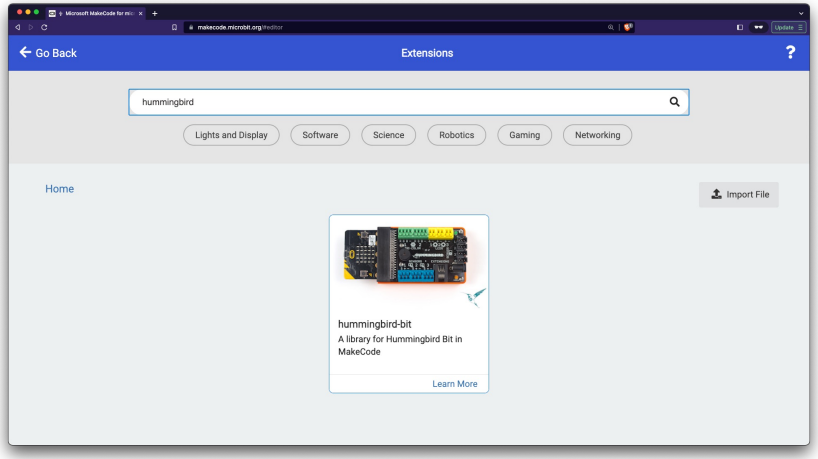
Click the Extensions tab to load into the extensions page.



Hummingbird Kits: Setup

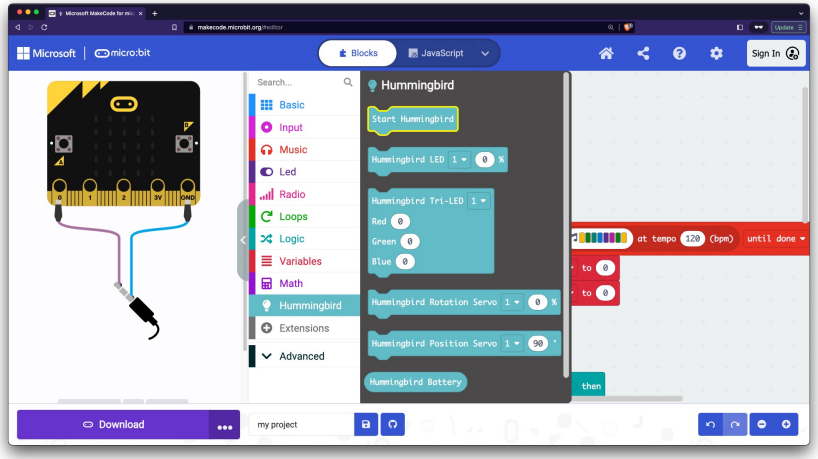
- Type ”hummingbird” into the extensions page search bar
- Select the “hummingbird-bit” extension

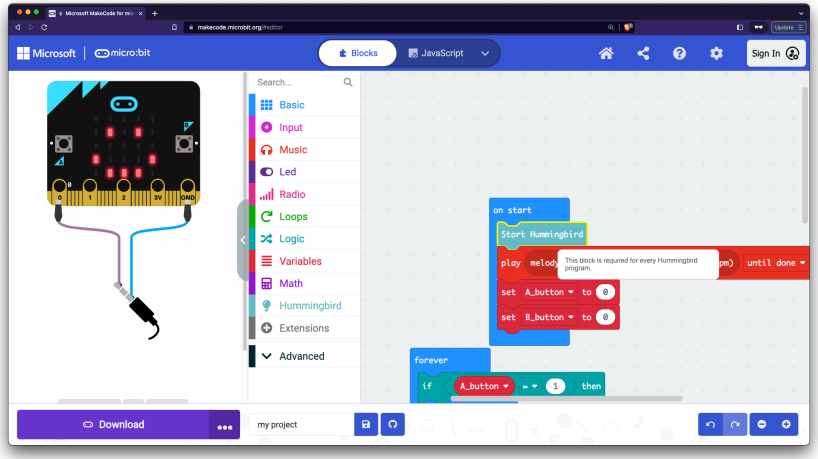




Hummingbird Kits: Setup

- In the hummingbird tab, select “start hummingbird”
- Drag your “start hummingbird” block into the ”on start” block

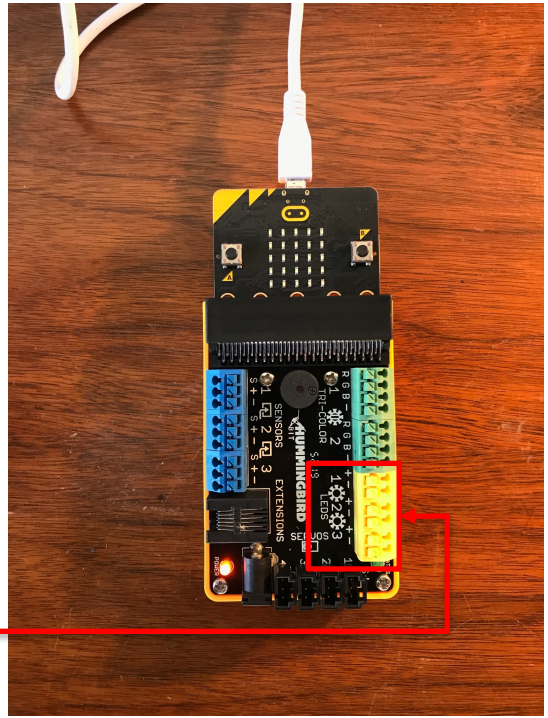




Hummingbird Kits: LED

Now we can use hummingbird specific blocks!

Let’s add a single colored LED to our board

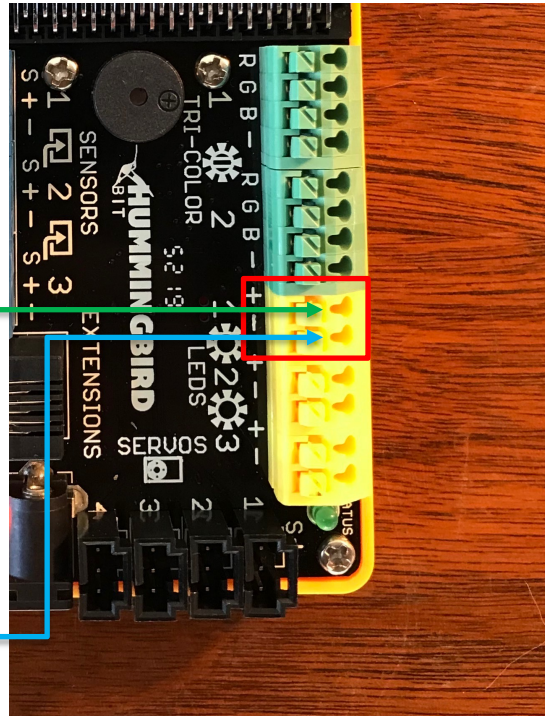


Hummingbird Kits: LED

Select a single color LED (Red, Yellow, or Green) defined by its 2 wires (Color, and Black)

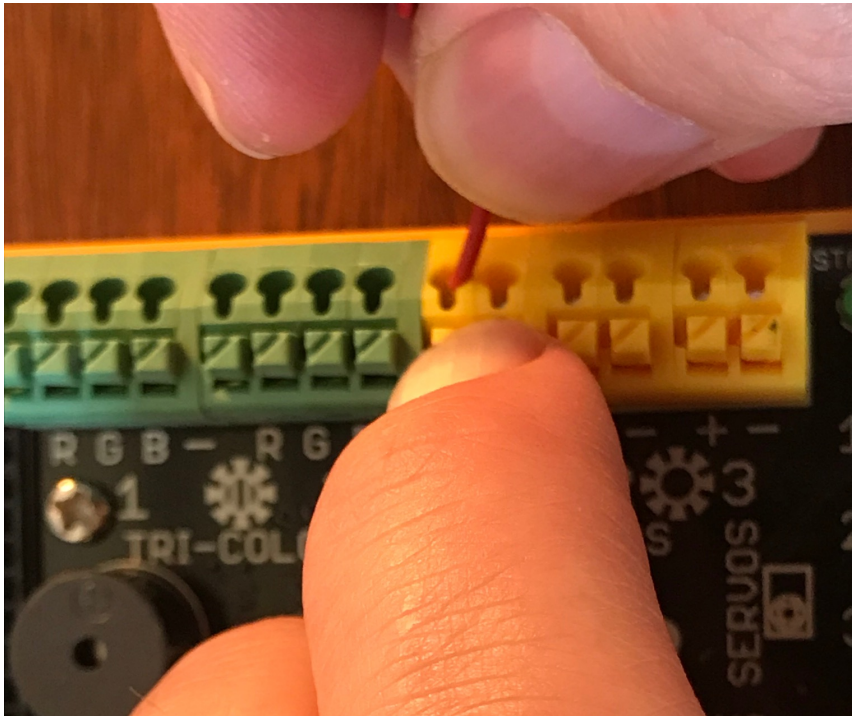
The colored wire will go in the positive (+) terminal.

The black wire will go in the negative (-) terminal



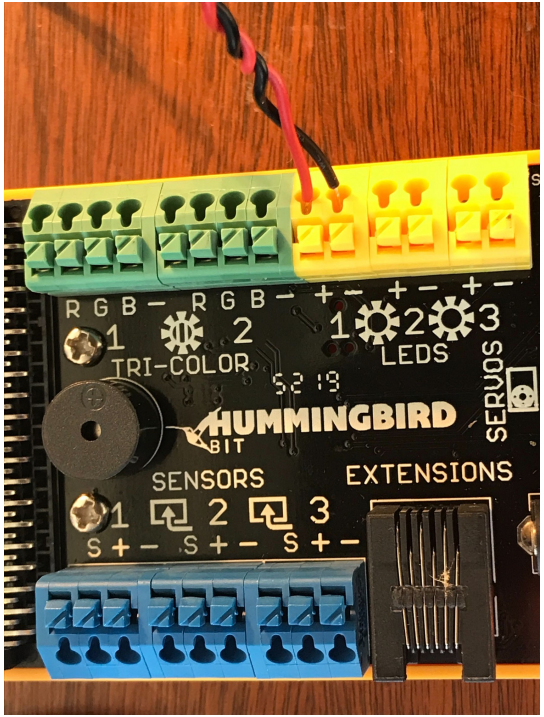
Hummingbird Kits: LED

Depress the terminal with a finger and guide the wire in with your other hand



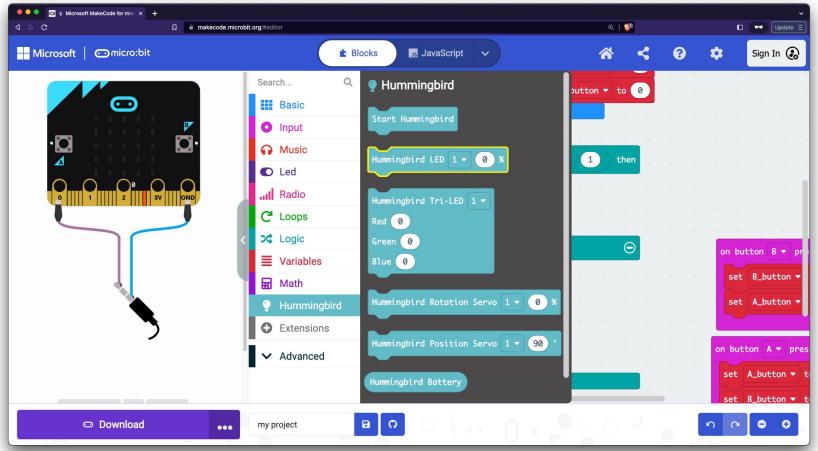
Hummingbird Kits: LED

- Do the same for the black wire in the negative terminal

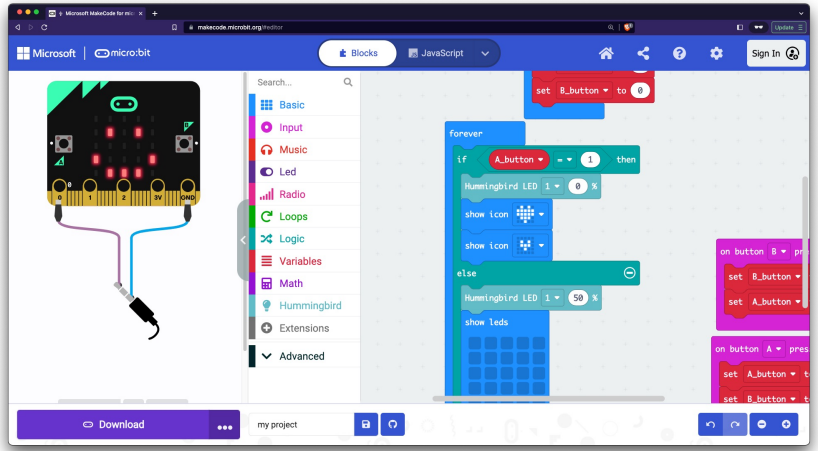


Hummingbird Kits: LED

- In the hummingbird tab, select “Hummingbird LED 1 (0%)” twice.

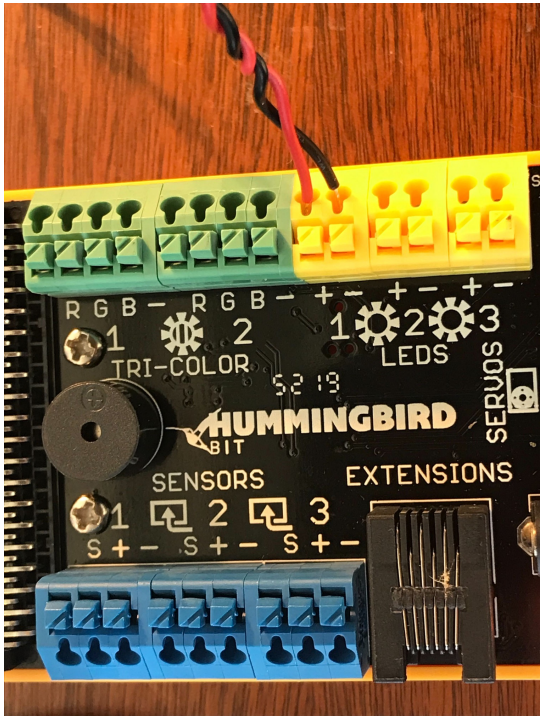


- In our if block:
  - Place Hummingbird LED 1 to 0
- In our else block:
  - Place Hummingbird LED 1 to 50



Hummingbird Kits: LED

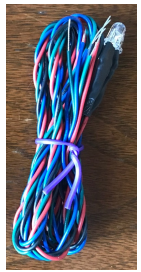
Now, when our LED grid is off, our singular LED is on!

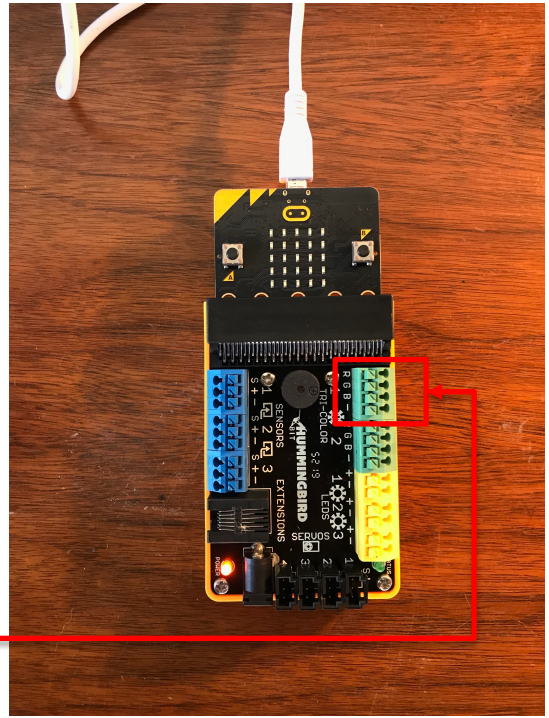


Hummingbird Kits: Tri-LED

The Tri-Color LED requires slightly more effort.

Select a Tri-color LED (the LED with 4 wires)

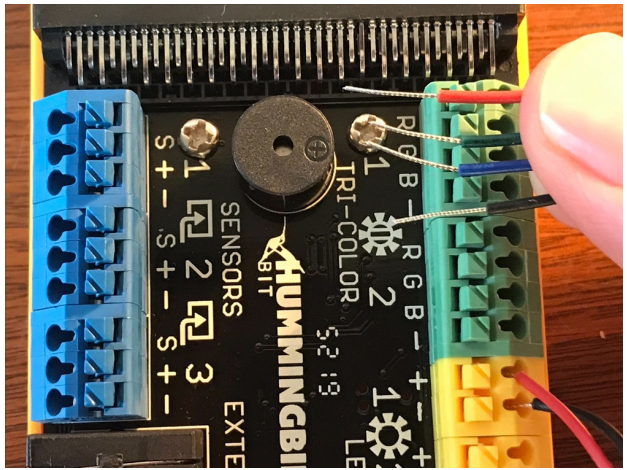




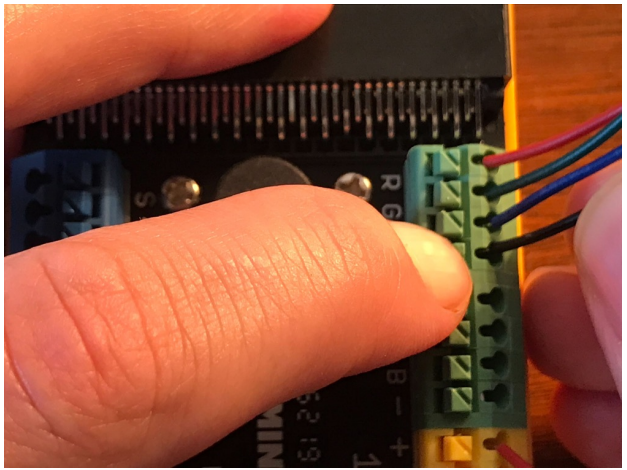
Hummingbird Kits: Tri-LED

Match up the 4 wires to their respective terminals:

- R: Red
- G: Green
- B: Blue
- -: Black

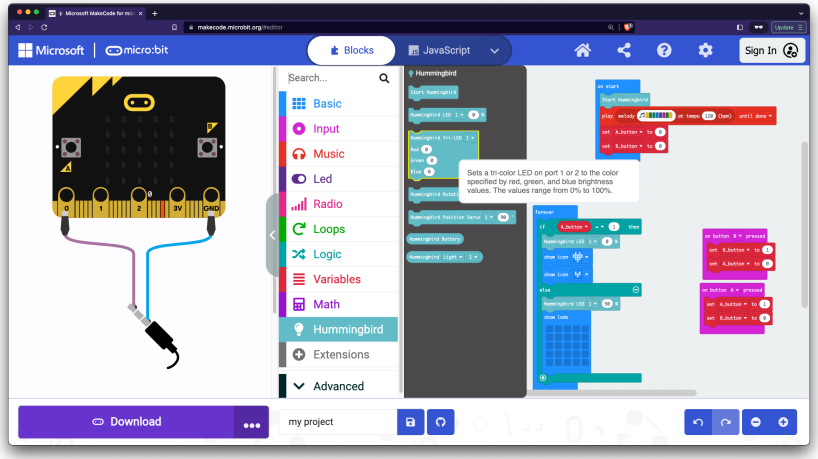


Depress the terminals and insert the wires.

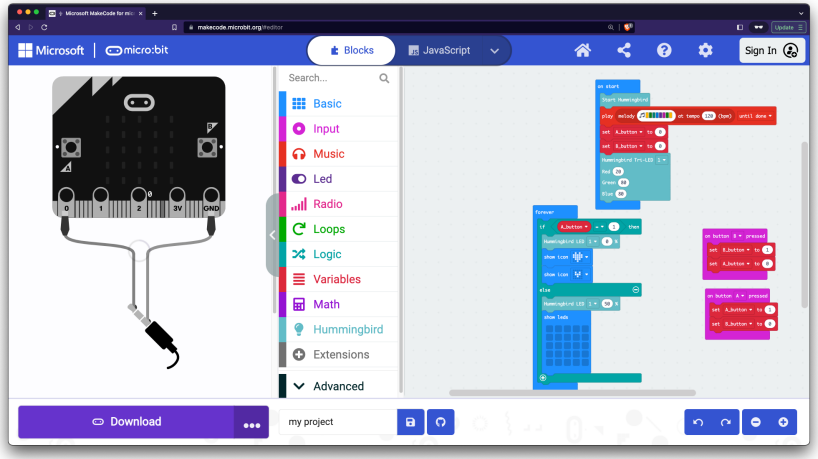


Hummingbird Kits: Tri-LED

Click the “hummingbird” tab and select the “Hummingbird Tri-LED”

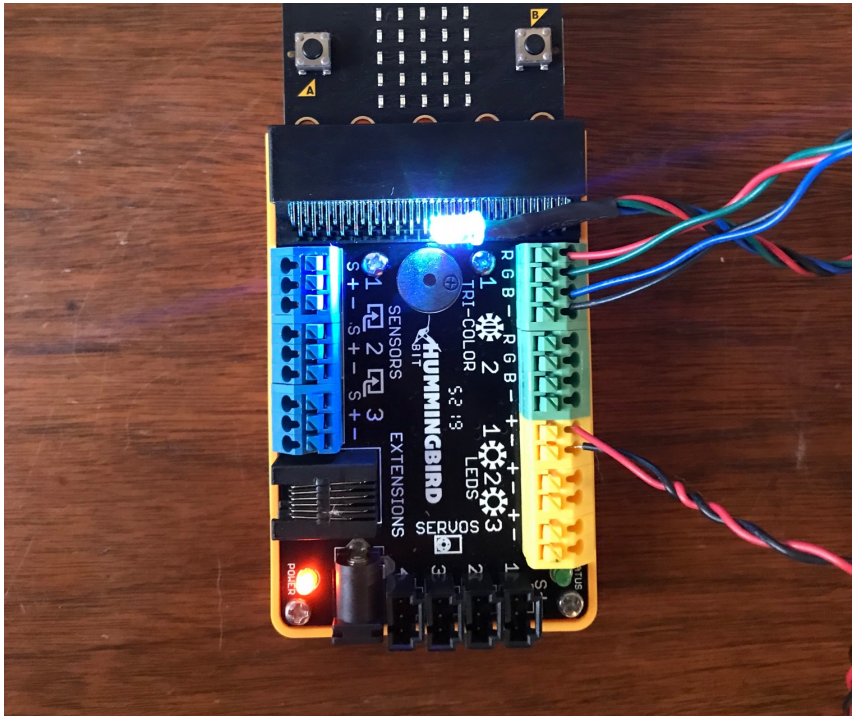


Place the “Hummingbird Tri-LED” in the ”on start” block with any RGB values you like.



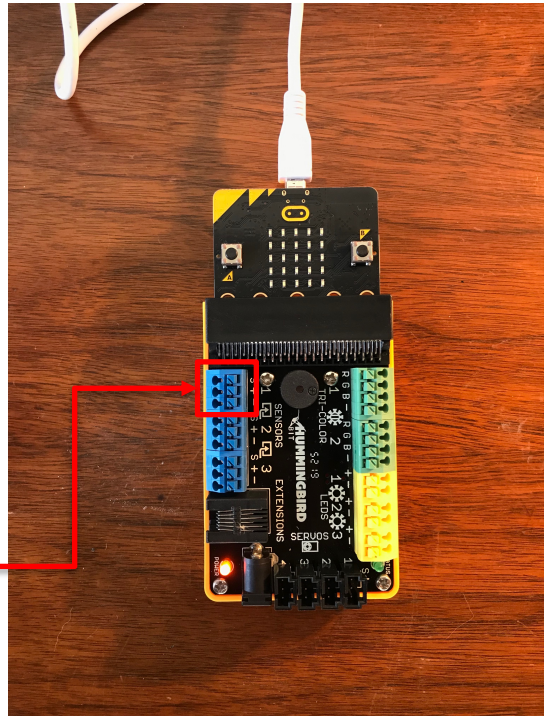
Hummingbird Kits: Tri-LED

Now we have a brightly colored LED of any color that turns on with our hummingbird bit!



Hummingbird Kits: Sensors

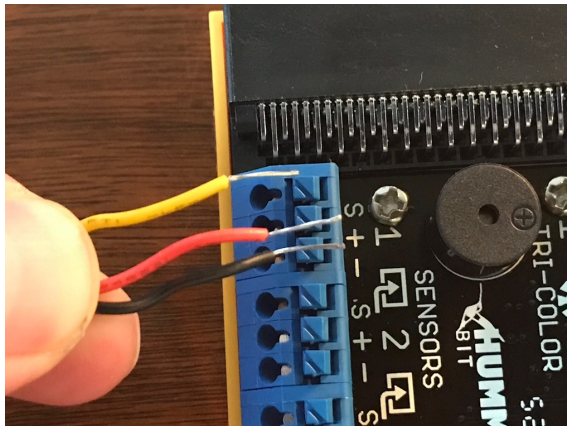
Let’s add a depth sensor to our hummingbird bit for a more interactive experience!



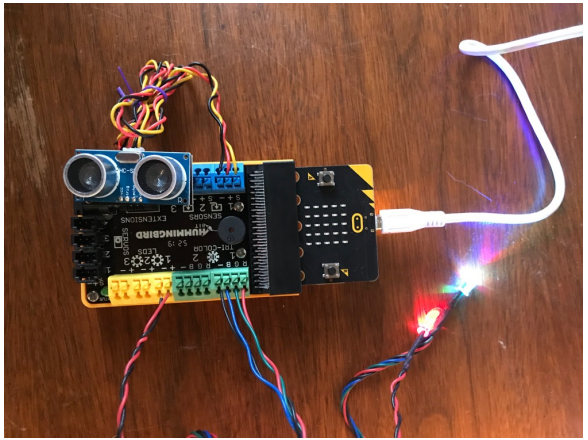
Hummingbird Kits: Sensors

Match up the 3 wires to their respective terminals:

- S: Yellow
- +: Red
- -: Black

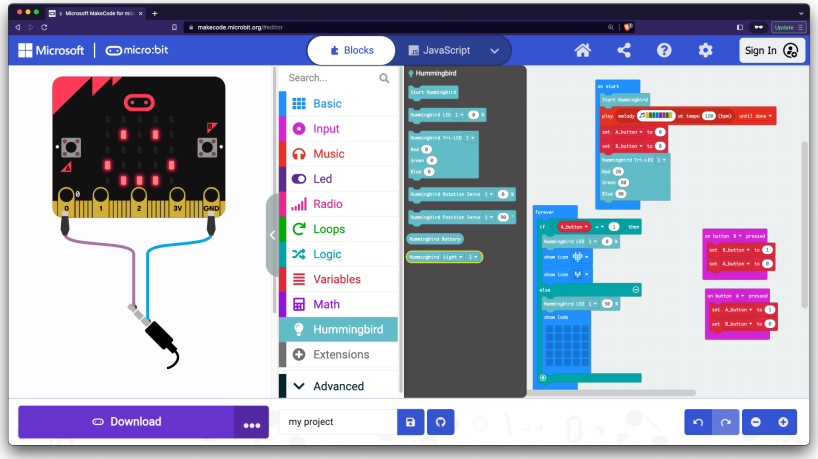


Depress the terminals and insert the wires.

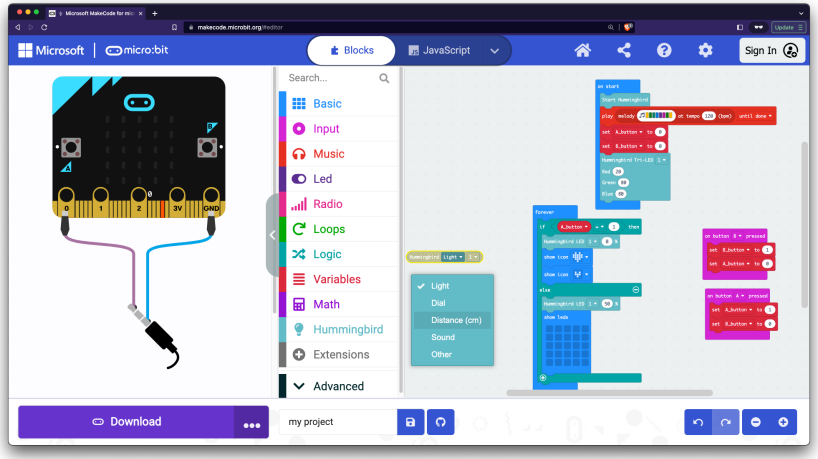


Hummingbird Kits: Sensors

Select the “hummingbird ’light’” block



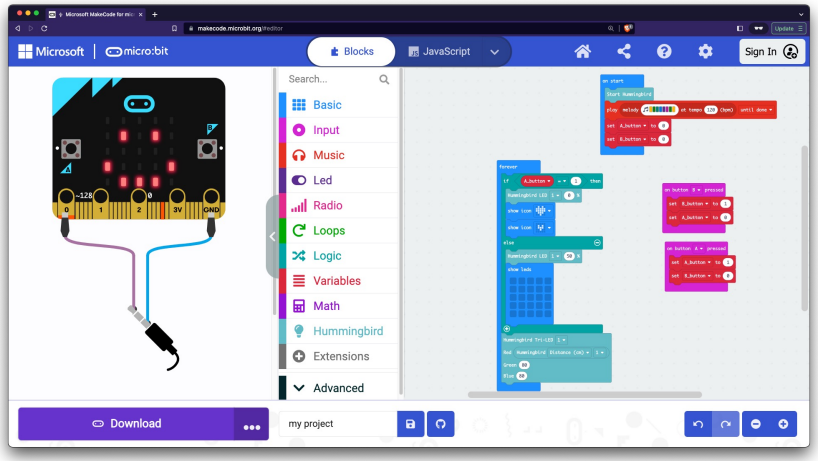
In the dropdown, select “Distance”



Hummingbird Kits: Sensors

Replace the R in RGB with the ”hummingbird distance” block, and move it to the bottom of the forever loop!

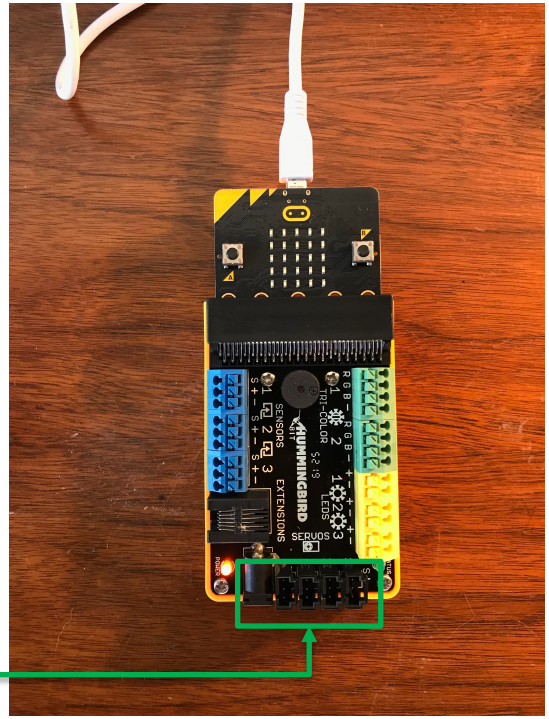
What happens when you move your hand close to the sensor?



Hummingbird Kits: Servos

When you think “robots” you probably think of a little robot that moves.

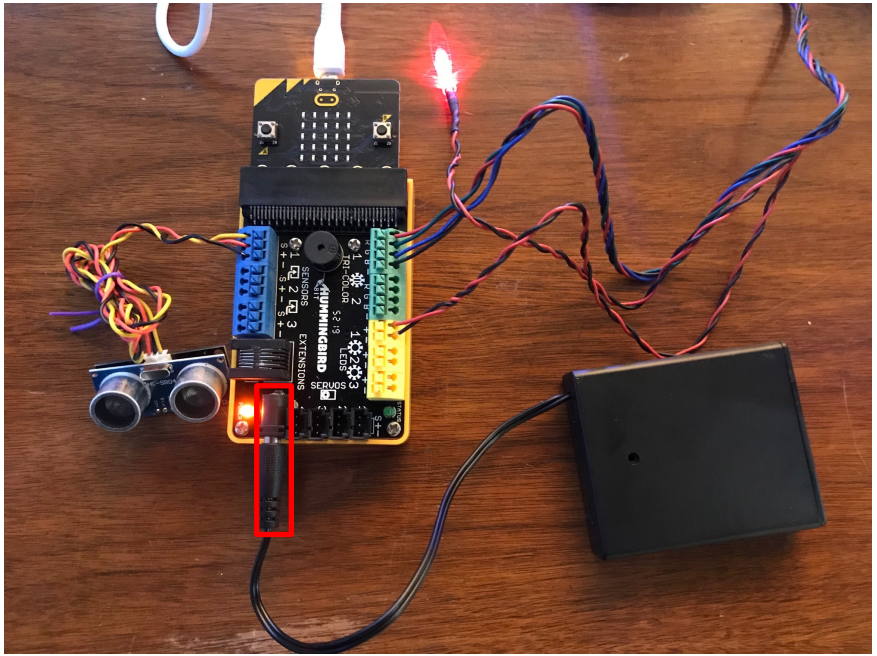
Those movements are powered by servos!



Hummingbird Kits: Servos

Servos are motors and require extra power!

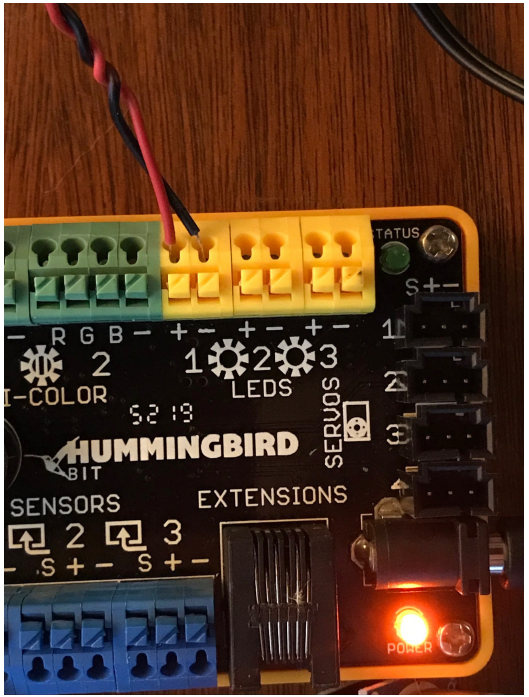
Add batteries to your battery pack and connect it to your hummingbird bit.



Hummingbird Kits: Servos

Servo ports look different than the ports we've been using.

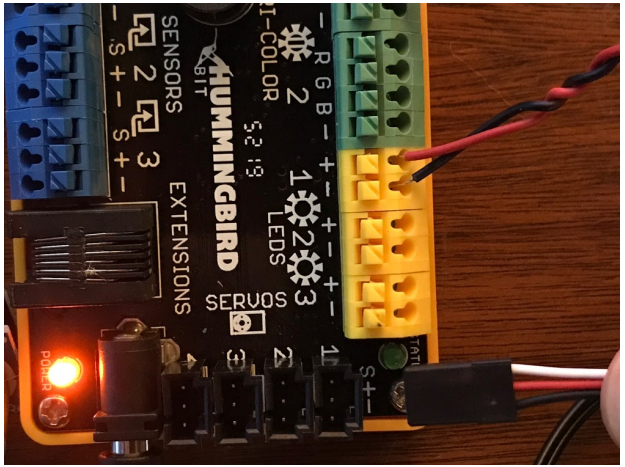
If you look inside the port, you'll notice our servos are using pins



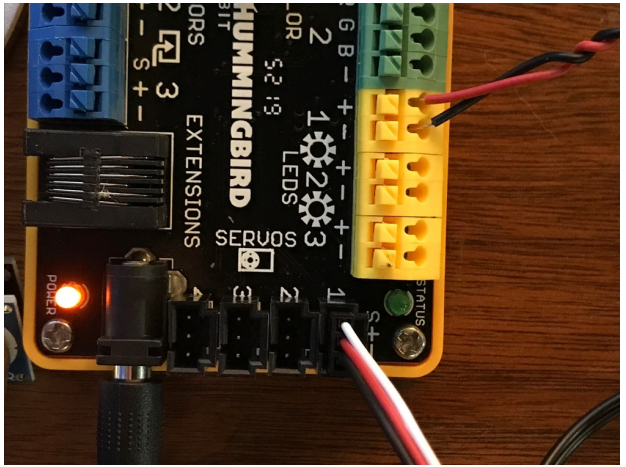
Hummingbird Kits: Servos

Align your ports:

- S: White
- +: Red
- -: Black



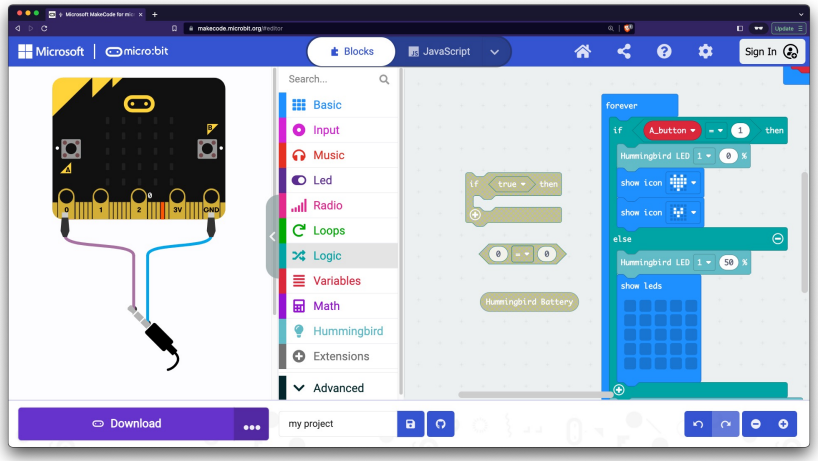
Insert the plug.



Hummingbird Kits: Servos

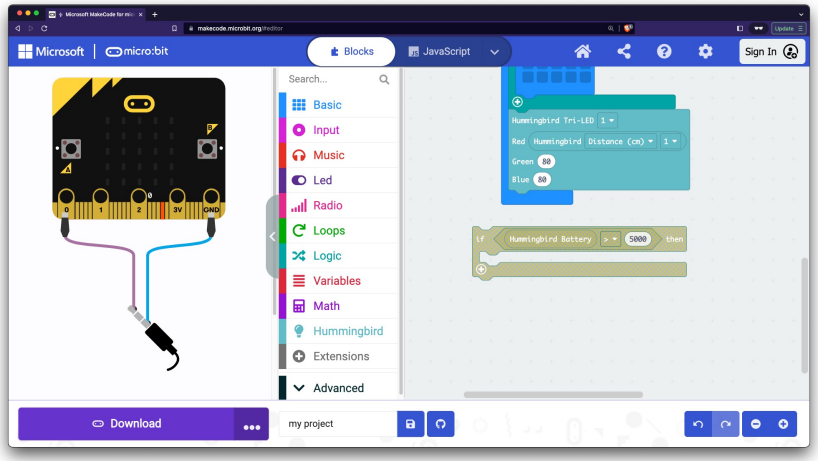
We want to be responsible with our code, so first, let's select:

- An "if" block from "logic"
- A "comparison" block from "logic"
- The "Hummingbird Battery" from "Hummingbird"



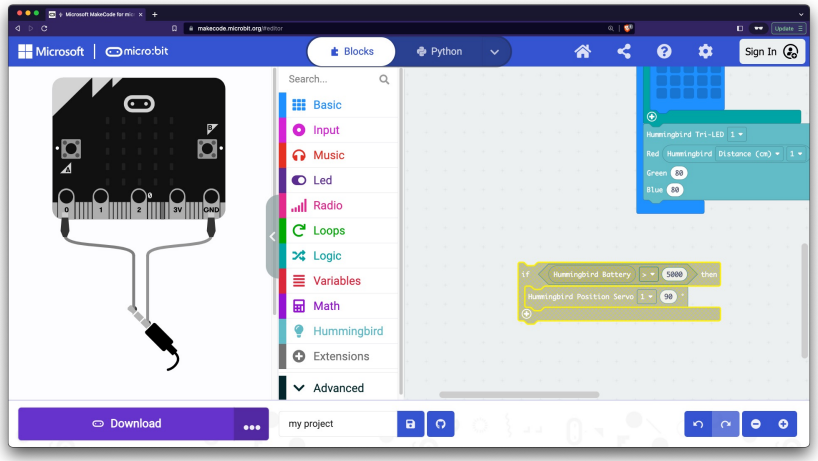
Hummingbird Kits: Servos

- Place the “comparison” block into the “if” blocks’ conditional
- Move the “Hummingbird Battery” to the “comparison” block
- Change the “=“ in the comparison block to “>”
- Change the 0 in the “comparison” block to 5000



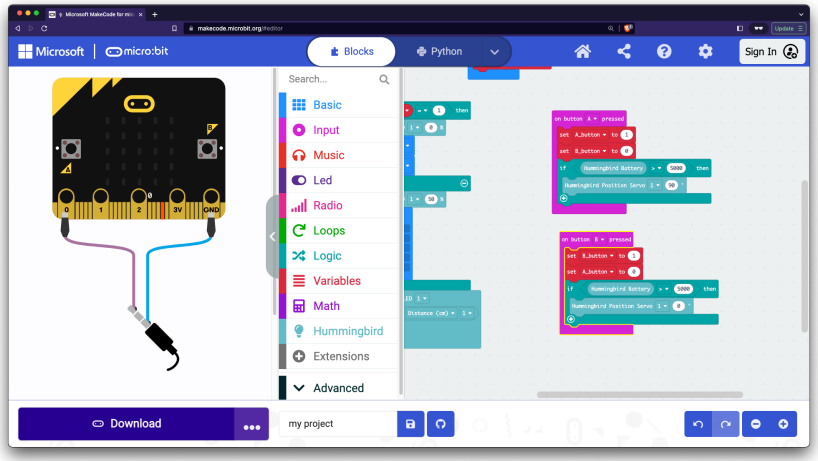
Hummingbird Kits: Servos

- In “Hummingbird” select “Hummingbird Position Servo”
- Set the Degrees to 90
- Place your “Hummingbird Position Servo” in the “if” block



Hummingbird Kits: Servos

- Add the code block to the end ”if” section of the “On Button A Pressed”
- Duplicated your code block and add it to the end of the “On Button B Pressed” with 0 degrees instead of 90
- Now try it with your battery off and then on!



Computer Science, Code and Humming Birds

Hummingbird Bit Tutorials for all devices can be found here

Hummingbird Kit robotics/craft tutorials can be found here

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Computer Science, Code and Humming Birds

QUESTIONS?

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE