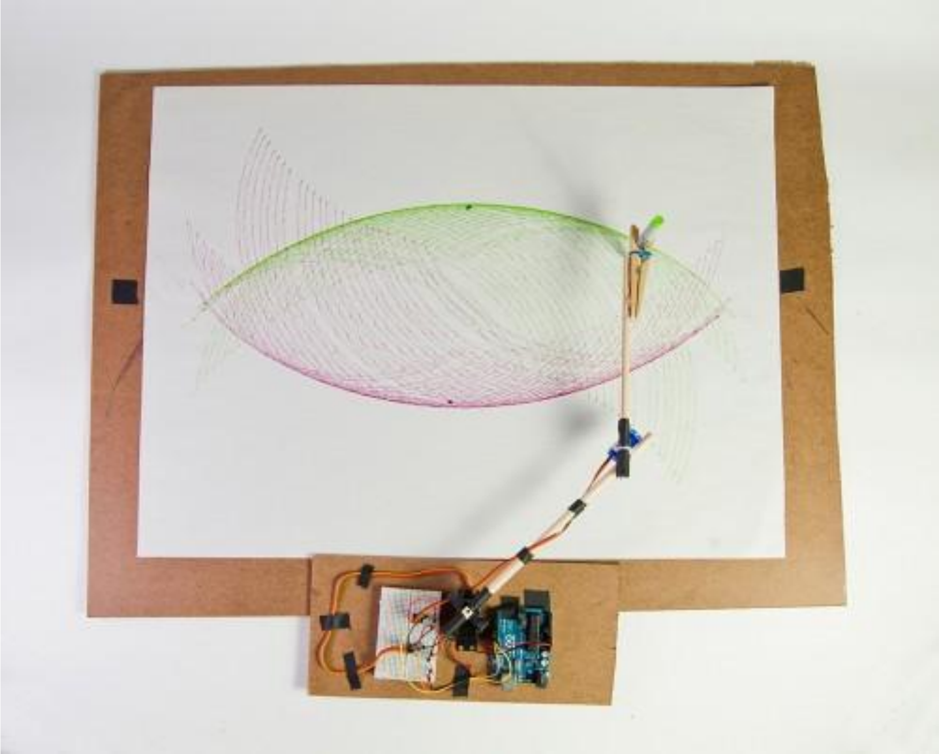
Art+CODE: Mechanized Drawing

Aaron Artrip, Research Associate I

aartrip3@gatech.edu

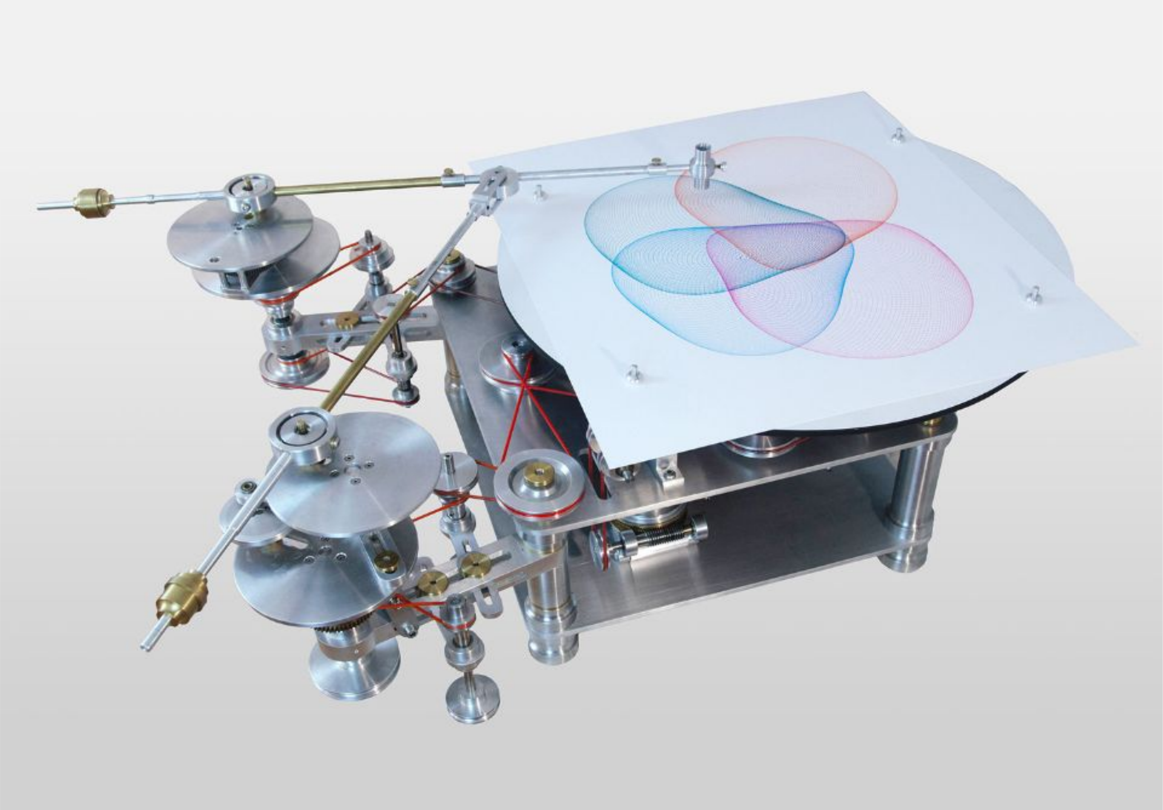
This project was inspired by the Arc-O-Matic from Pete Prodoehl of RasterWeb, a blog series.

http://rasterweb.net/raster/projects/arcomatic/

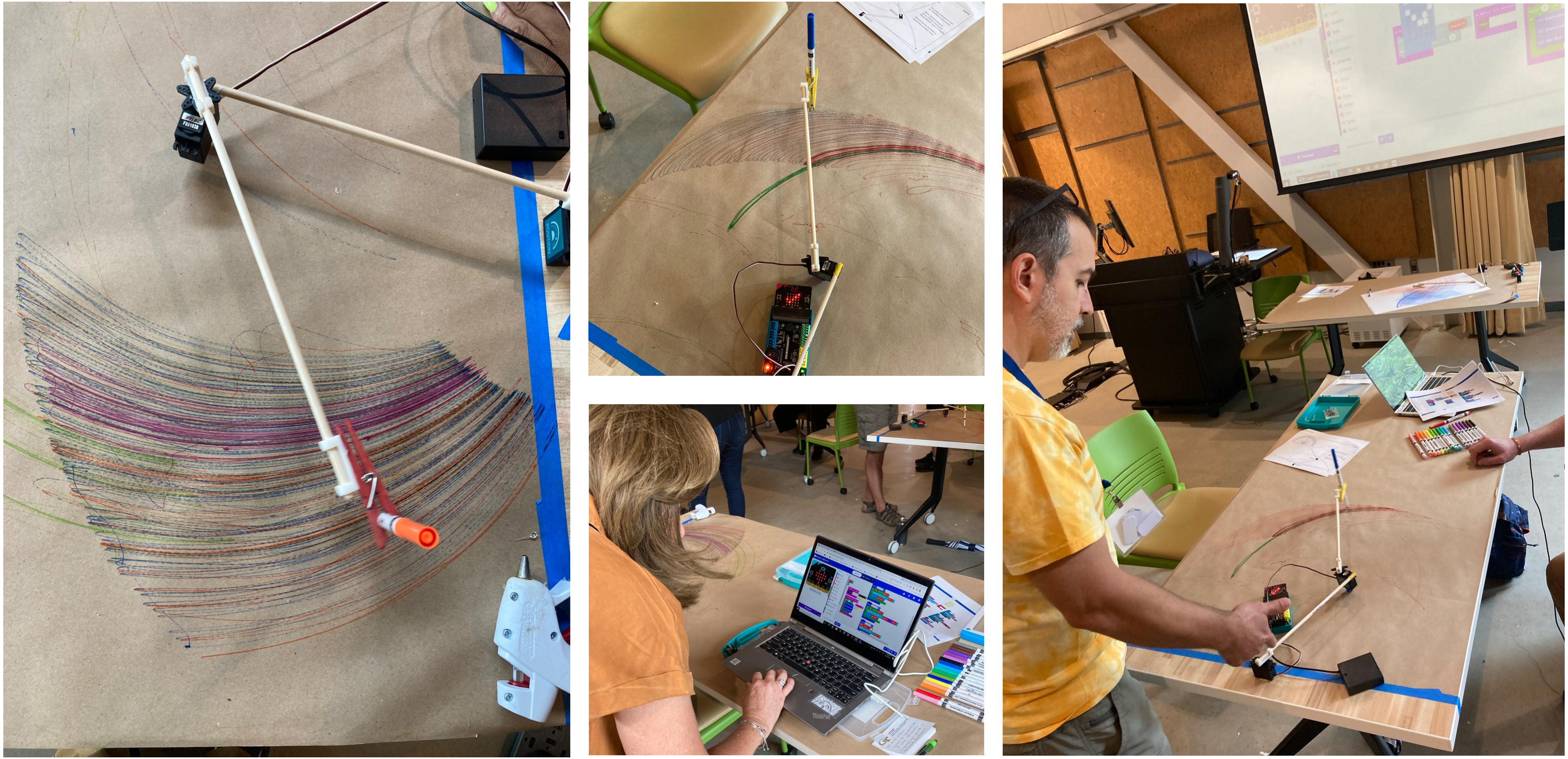


JAMES NOLAN GANDY

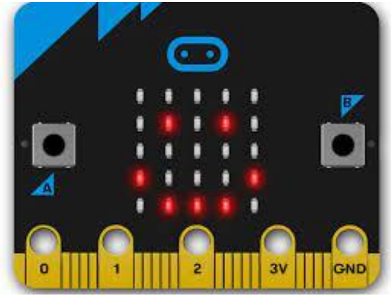
HOME DRAWINGS FOR SALE SOLD DRAWINGS GALLERY CONTACT ABOUT BLOG



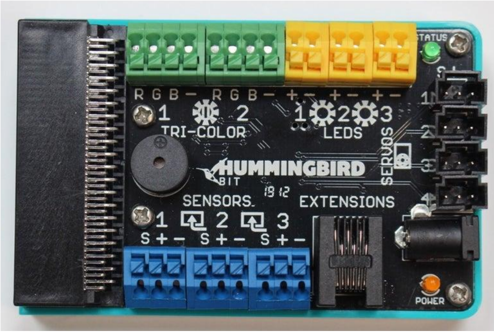
STEAM Symposium 2023, CEISMC/Georgia Tech



Microbits + Hummingbirds = Robotic Control



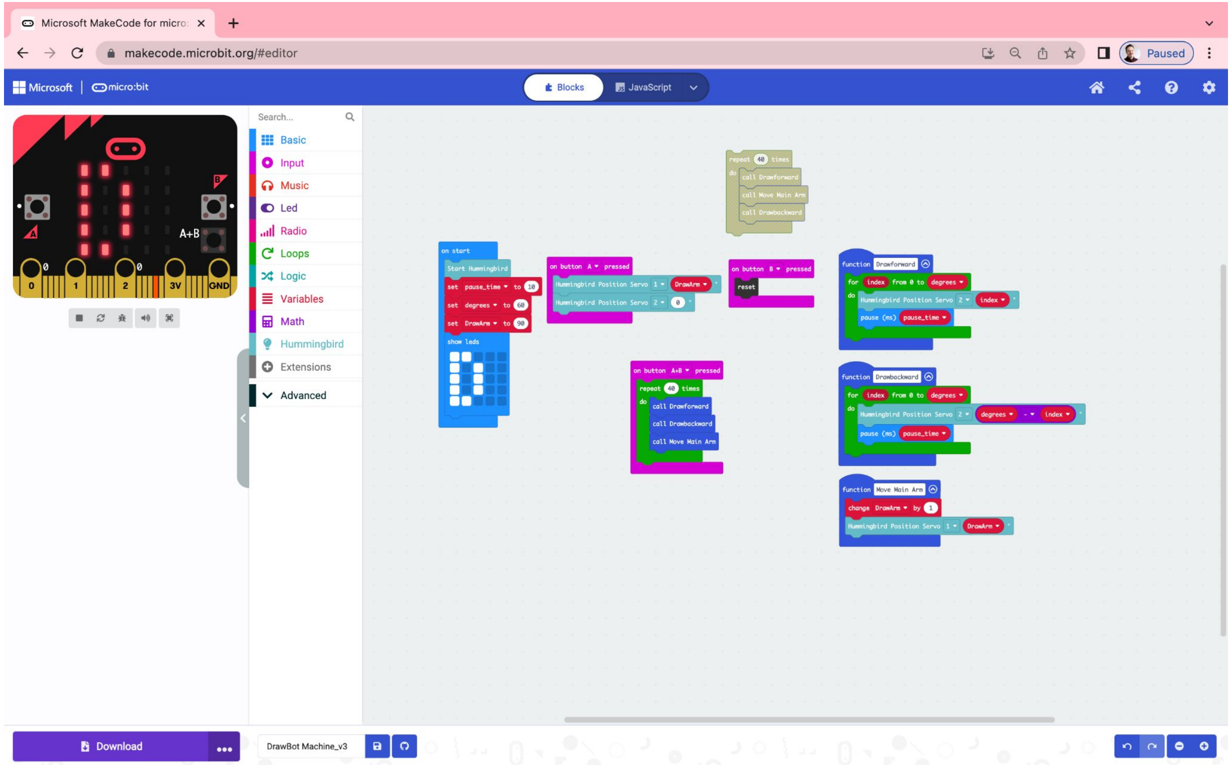
The Micro:bit is a **Microcontroller**, similar to an arduino. Multiple inputs and output types.



The Hummingbird-Bit is a **HAT.** This allows more secure connections to the micro:bit

Together, they provide many approaches to coding and hardware interaction!

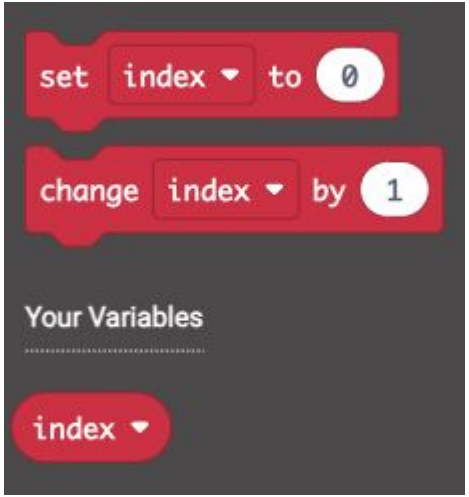
https://makecode.microbit.org



```
repeat 40 times
do
    call Drawforward
    call Move Main Arm
    call Drawbackward

on start
    Start Hummingbird
    set pause_time to 10
    set degrees to 60
    set DrawArm to 90
    show leds

on button A pressed
    Hummingbird Position Servo 1 DrawArm °
    Hummingbird Position Servo 2 0 °

on button B pressed
    reset

on button A+B pressed
    repeat 40 times
    do
        call Drawforward
        call Drawbackward
        call Move Main Arm

function Drawforward
    for index from 0 to degrees
    do
        Hummingbird Position Servo 2 index °
        pause (ms) pause_time

function Drawbackward
    for index from 0 to degrees
    do
        Hummingbird Position Servo 2 degrees - index °
        pause (ms) pause_time

function Move Main Arm
    change DrawArm by 1
    Hummingbird Position Servo 1 DrawArm °
```

Emulator Toolbox Code Window

Key Terms - Variable

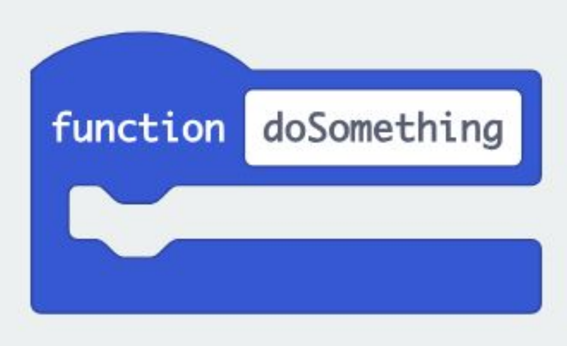


A **variable** is a place where you can store and retrieve data. Variables have a **name**, a **type**, and **value**:

- **-name** is how you'll refer to the variable
- **-type** refers to the kind of data a variable can Store (set or change)
- **-value** refers to what's stored in the variable

Example: A stereo music system has 4 speakers. When you adjust the ***[volume variable]*** knob, all 4 speakers will adjust to the ***[value]*** which is set by the volume knob. This give you the ability to change ***values*** in multiple places quickly in your code while you fine tune your settings.

Key Terms - Function

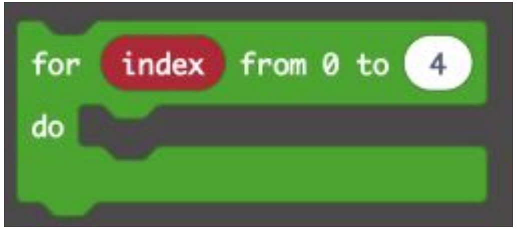


```
function doSomething
```

A **function** lets you create a portion of code that you can reuse in your program. This is handy because you can put code that you want to use over again in a **function**. Instead of copying the same code to many places in your program, you can just use a **function** you made and all the code inside is used as a single block.

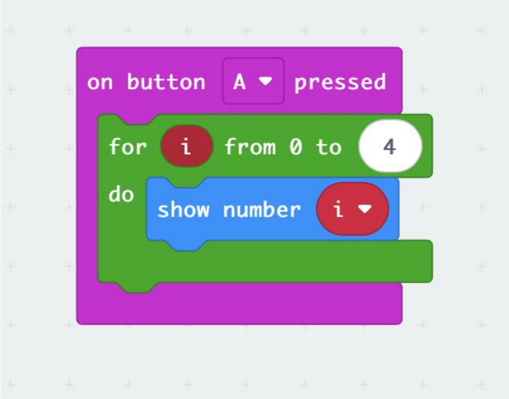
Example: Ordering a cheeseburger tells the kitchen: **Function [Cheeseburger}.** The kitchen understand that **Function [Cheeseburger}** means (Bun, cheese, beef patty, tomato, lettuce, mayo, mustard, bun)

Key Terms - Index Loop



```
for index from 0 to 4
do
```

Cycles through a list based on the value given.

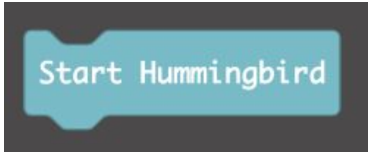


```
on button A pressed
  for i from 0 to 4
  do
    show number i
```

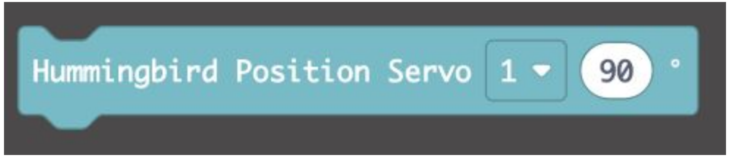
This example code will display a count from 0 to 4 on the Micro:bit LED Array on the front ONLY when you press A Button.

i is the current value based on the position of the count list

Key Terms - Hummingbird Commands

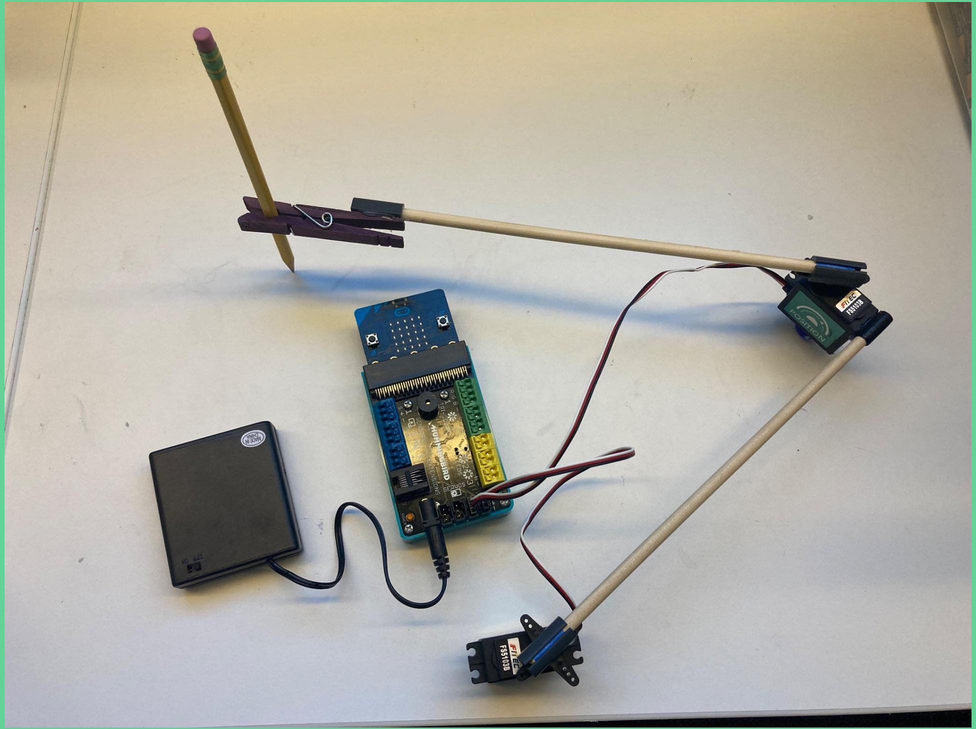

Start Hummingbird

This block engages the microbit to communicate with the Hummingbird


Hummingbird Position Servo 1 ▾ 90 °

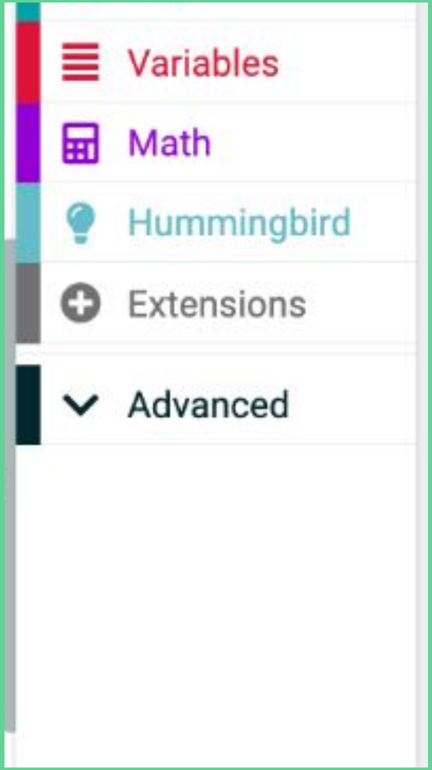
This block engages the selected Servo motor. We will use **‘Position’** Servo

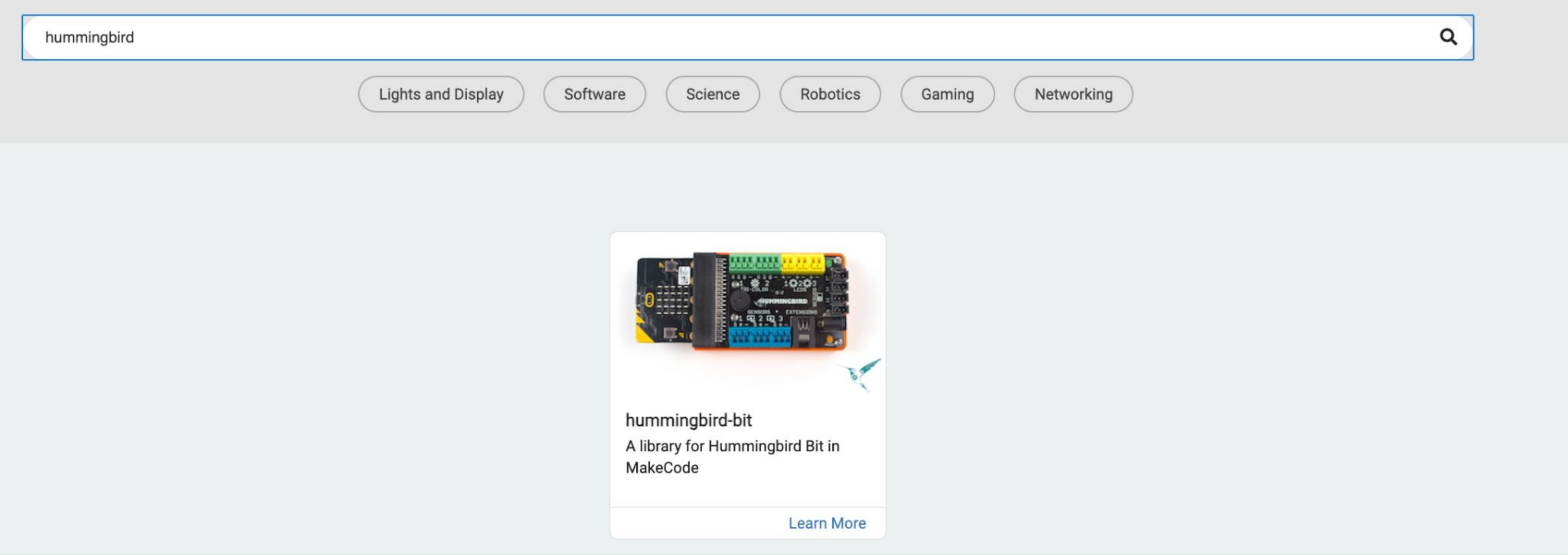
Coding the Drawing Robot



Step 1:

Add Hummingbird Extension —----------->





hummingbird

hummingbird-bit

A library for Hummingbird Bit in MakeCode

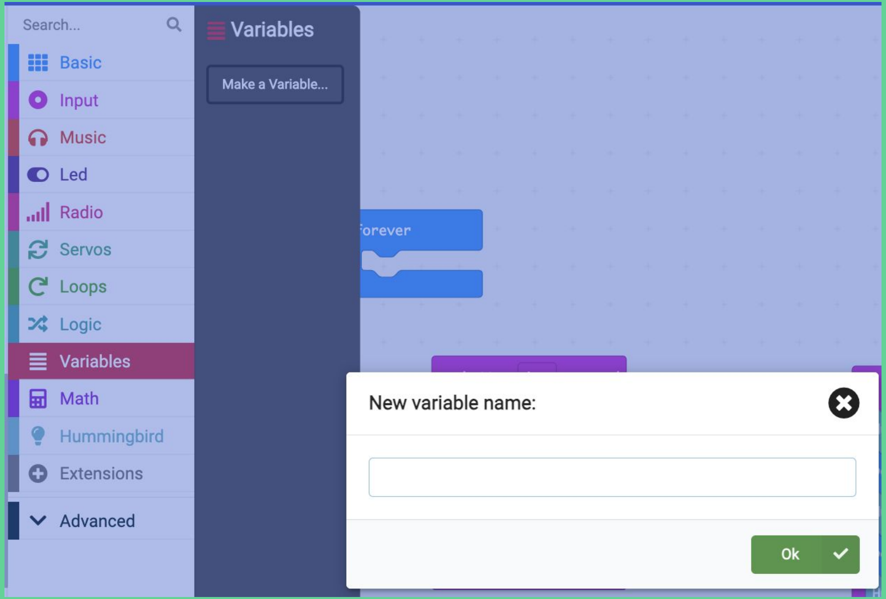
Select the hummingbird-bit and this will now be available in your TOOLKIT area

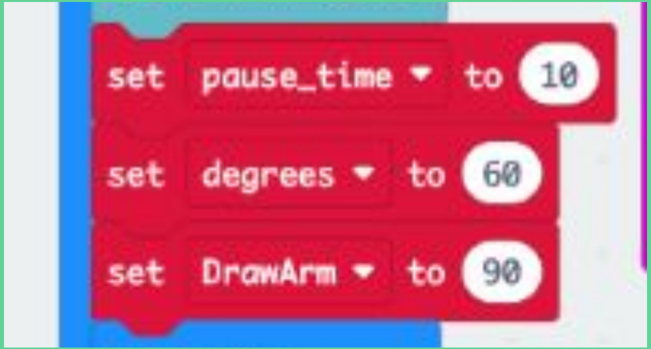
Step 2:

Name our Variables.

Make (3) variables:

- Pause_time
- degrees
- DrawArm



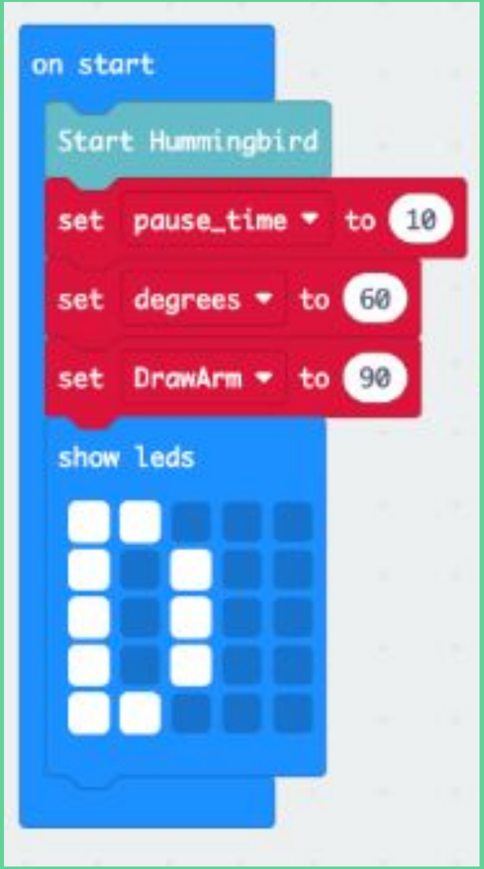


```
set pause_time to 10
set degrees to 60
set DrawArm to 90
```

Step 3:

- Build our “Start” code
- Will execute when system is powered on
- In ‘variable’ toolbox grab 3 ‘set {variable} blocks
    - Set pause_time to 10
    - Set degrees to 60
    - Set DrawArm to 90
- Add a ‘show LEDs’, I used a D for Draw. This is just to confirm the code has loaded correctly.

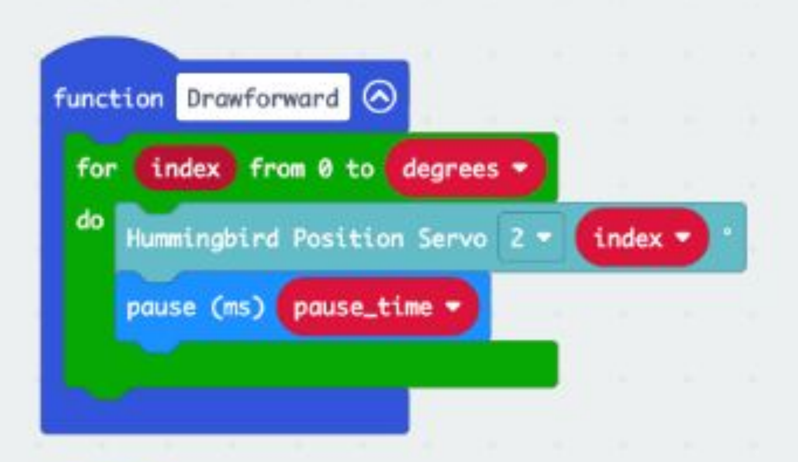
Click into the white circle of each ‘set’ variable to the values listed above



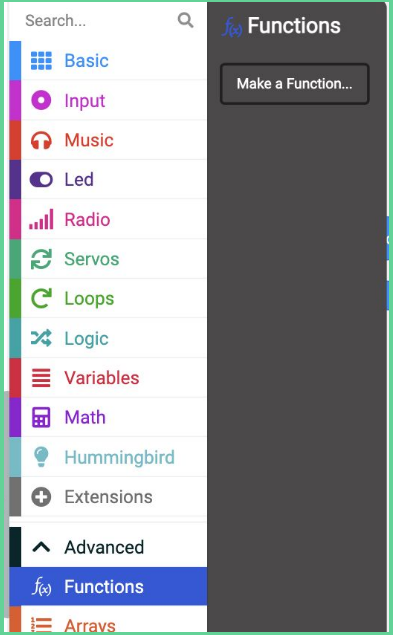
```
on start
  Start Hummingbird
  set pause_time to 10
  set degrees to 60
  set DrawArm to 90
  show leds
```

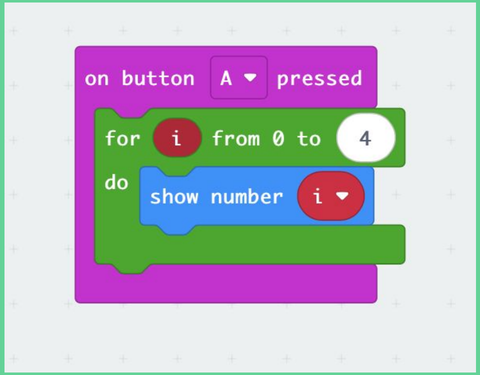
Step 4:

Build our first Function “Draw Forward”



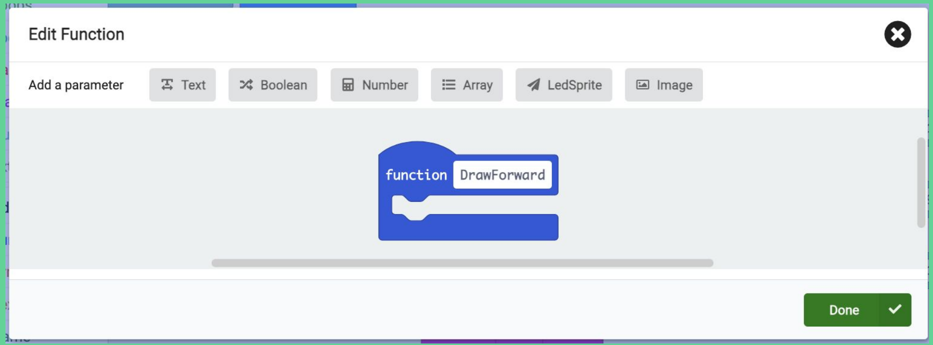
```
function Drawforward
  for index from 0 to degrees
  do
    Hummingbird Position Servo 2 index °
    pause (ms) pause_time
```





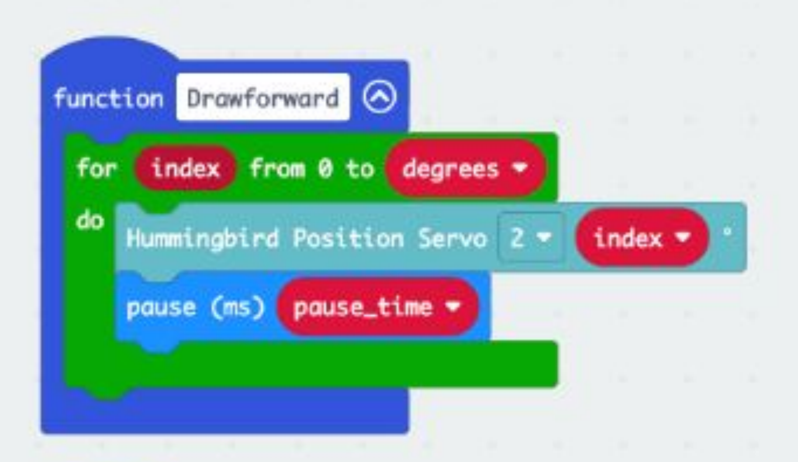
```
on button A pressed
  for i from 0 to 4
  do
    show number i
```

Index Counting Example ^



Step 4, cont:

Build our first Function “Draw Forward”



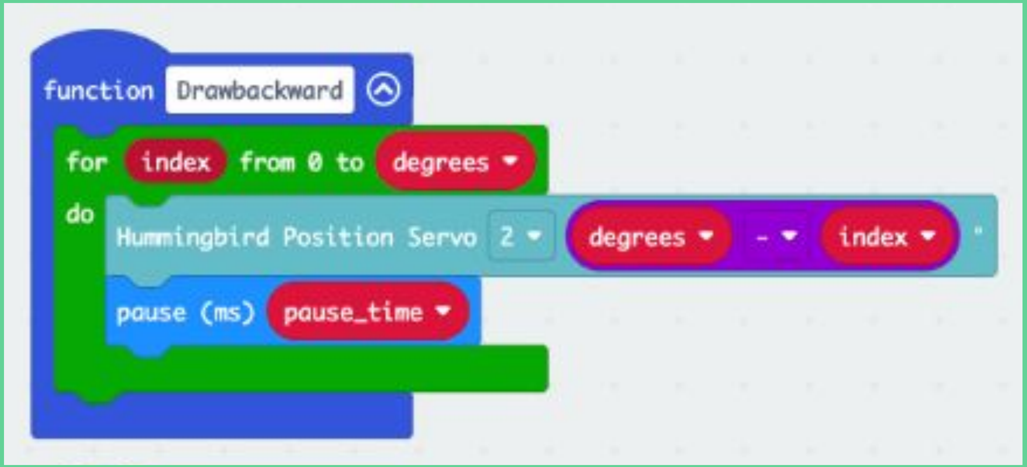
```
function Drawforward
  for index from 0 to degrees
  do
    Hummingbird Position Servo 2 index °
    pause (ms) pause_time
```

The Index will cycle a value from 0 to (degrees) and move the <u>Position Servo 2</u> to this value. It will move a value of 1 each time, pause for (pause_time), and repeat again. *make sure you are using a ‘position servo’ block and set it to ‘2’ for servo 2.*

This function will move the drawing arm 60 degrees and pause for 10 ms.

Step 5:

Build our second Function “DrawBackwards”



```
function Drawbackward
  for index from 0 to degrees
  do
    Hummingbird Position Servo 2 degrees - index
    pause (ms) pause_time
```

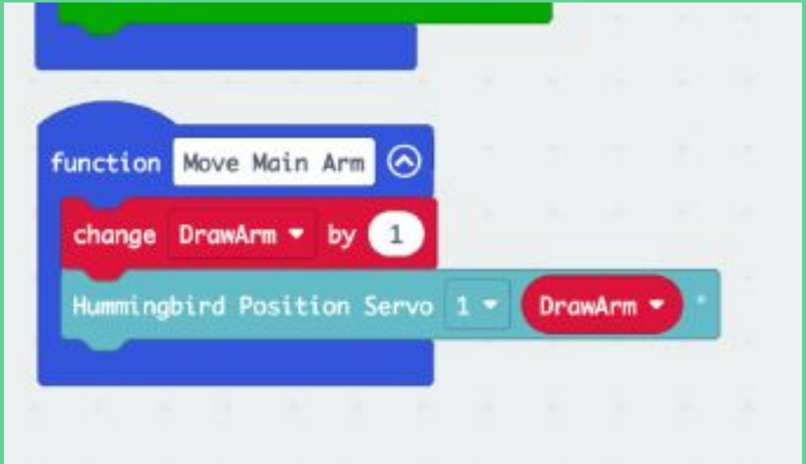
This function is exactly the same as DrawFoward, except:

We will use a MATH function to INVERSE the direction the Draw Arm is Moving by counting back to 0

Drawforward = 0 to 60 degrees / DrawBackwards = 60 to 0 degrees.

Step 6:

Build our third Function “Move Main Arm”

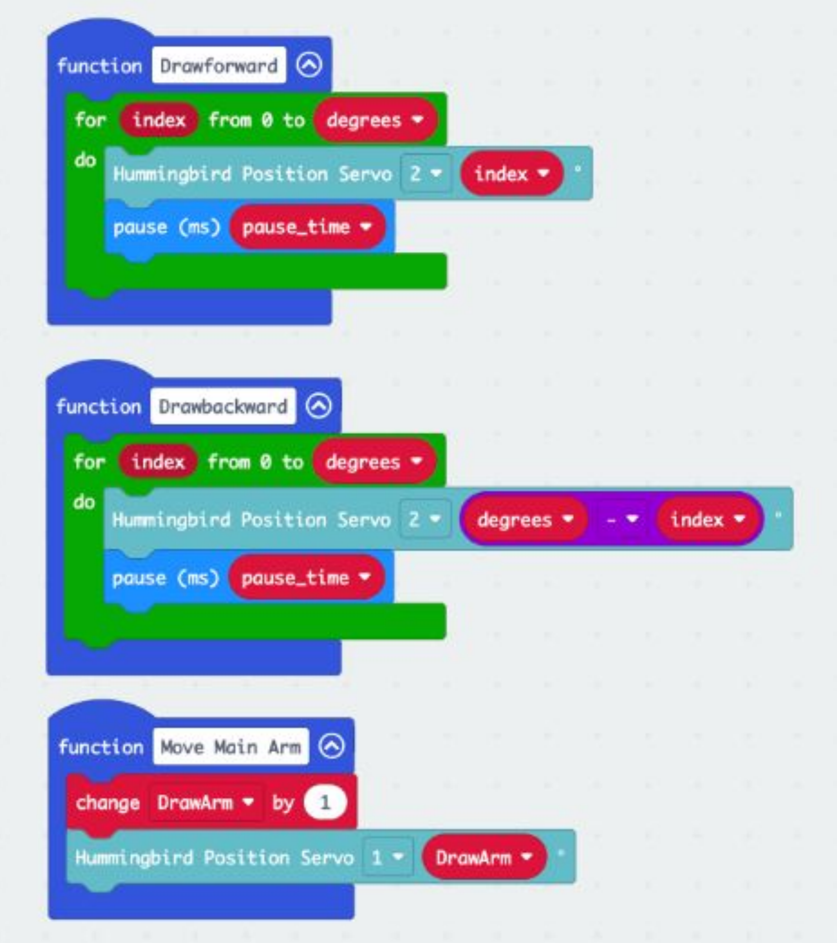


```
function Move Main Arm
    change DrawArm by 1
    Hummingbird Position Servo 1 DrawArm °
```

This will call tell the DrawArm Value (90) to change by 1, making it 91. The Servo 1 will move to position of DrawArm, which is now 91. This will move the Servo 1 Arm 1 degrees at a time.

3 Functions:

Drawfoward
Drawbackward
Move Main Arm

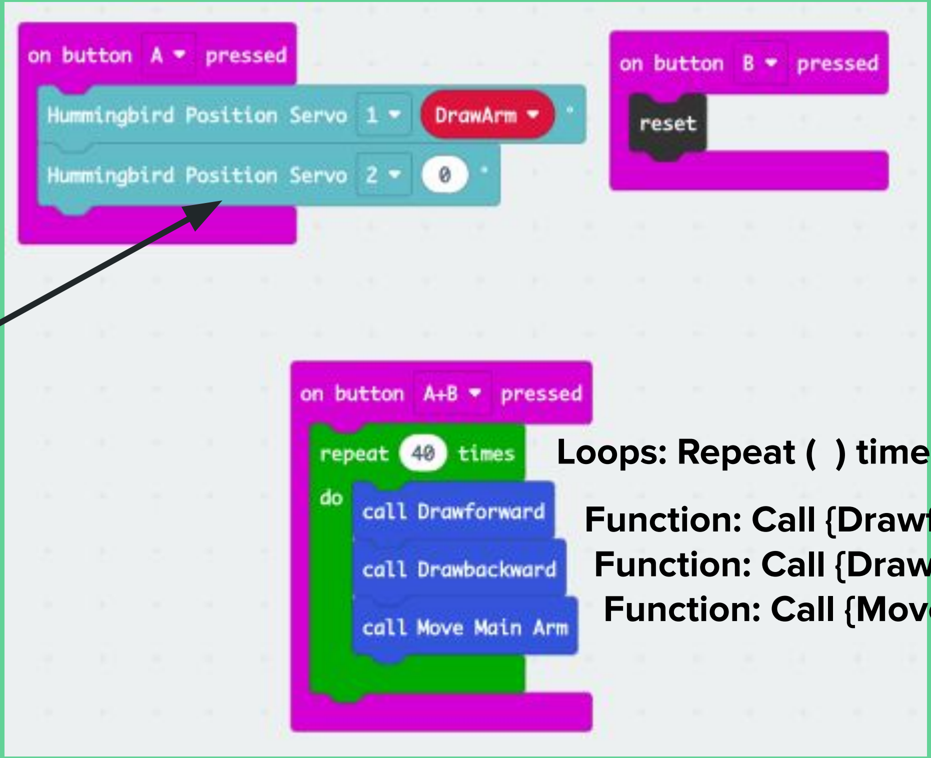


```
function Drawforward
  for index from 0 to degrees
  do
    Hummingbird Position Servo 2 index °
    pause (ms) pause_time

function Drawbackward
  for index from 0 to degrees
  do
    Hummingbird Position Servo 2 degrees - index °
    pause (ms) pause_time

function Move Main Arm
  change DrawArm by 1
  Hummingbird Position Servo 1 DrawArm °
```

Step 7: Code our Buttons

Input: Button A, B, A+B

***make sure you are using Hummingbird POSITION Servo Block**



```
on button A pressed
    Hummingbird Position Servo 1 DrawArm °
    Hummingbird Position Servo 2 0 °

on button B pressed
    reset

on button A+B pressed
    repeat 40 times
    do
        call Drawforward
        call Drawbackward
        call Move Main Arm
```

Reset:
Advanced -> Control

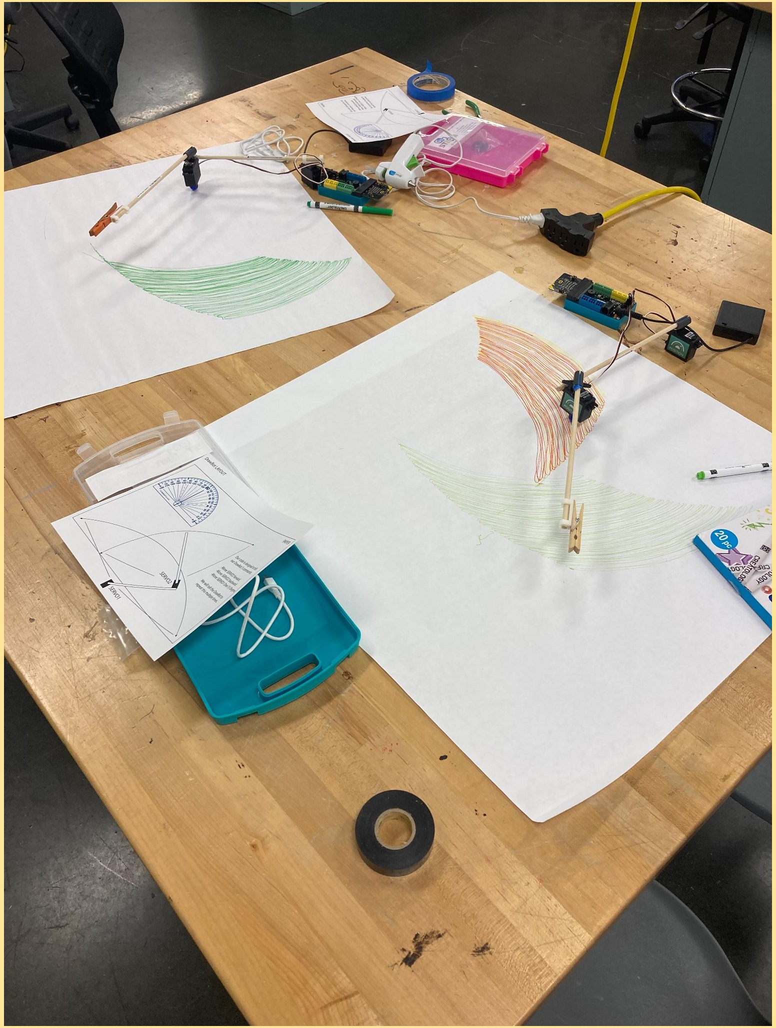
Loops: Repeat ( ) times

Function: Call {Drawforward}
Function: Call {Drawbackward}
Function: Call {Move Main Arm}

Building the Drawing Robot

Materials Needed:

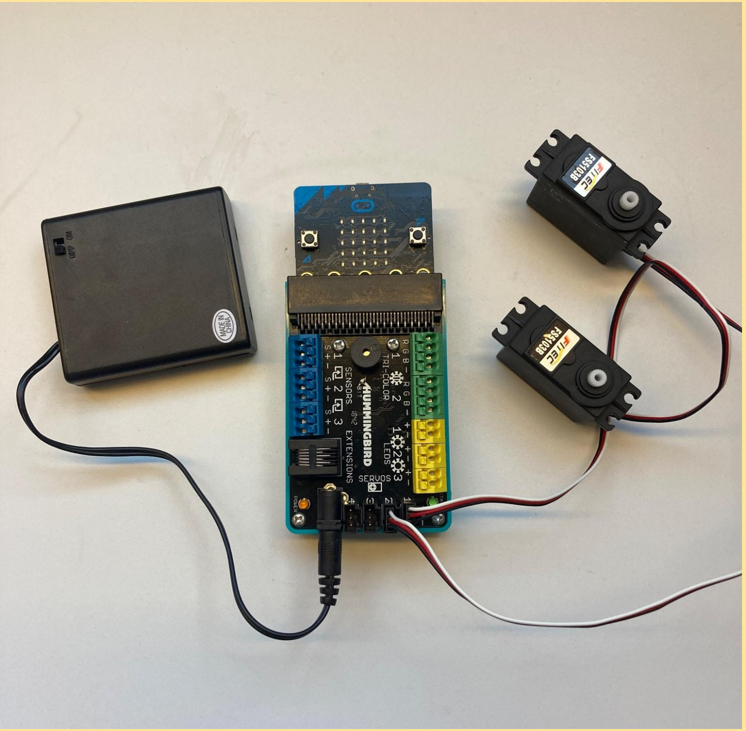
- Micro:bit + Hummingbird KIT (w/ battery pack)
- 1 Glass Marble
- 2 - ¼'' wooden dowels, around 8''-10'' lengths
- Slim Felt tip pens or markers. (Crayola's are too big :/)
- 3 - 3D Printed brackets (click here for .stl file)
- Hot glue gun + glue sticks
- Brown Paper (to cover tables)
- 18x24'' drawing pad
- Scissors
- blue tape
- Electrical Tape
- Wooden Clothespins



Step 1:

Connect all components

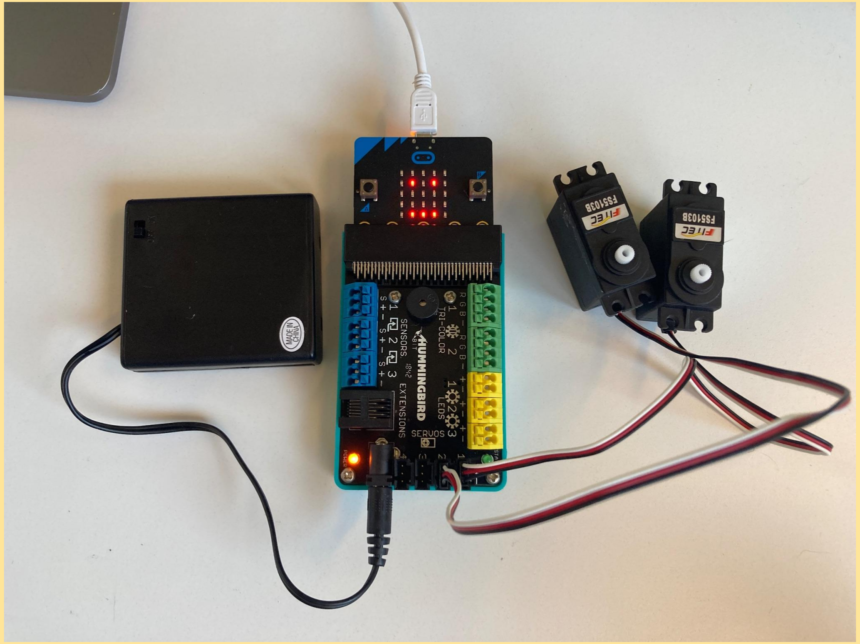
- connect Micro:bit into Hummingbird.
- Connect Servo motors to Hummingbird. Make sure the BLACK wire is on the Ground (-) side and the WHITE wire is on the Signal (S) side.
- Connect the battery pack and leave it in **OFF** position for now.



Step 2:

Connect to Computer

- Use the supplied USB cable to connect to your micro:bit to you computer
- Be sure to use GOOGLE CHROME. This will help with downloading the code in the the micro:bit.

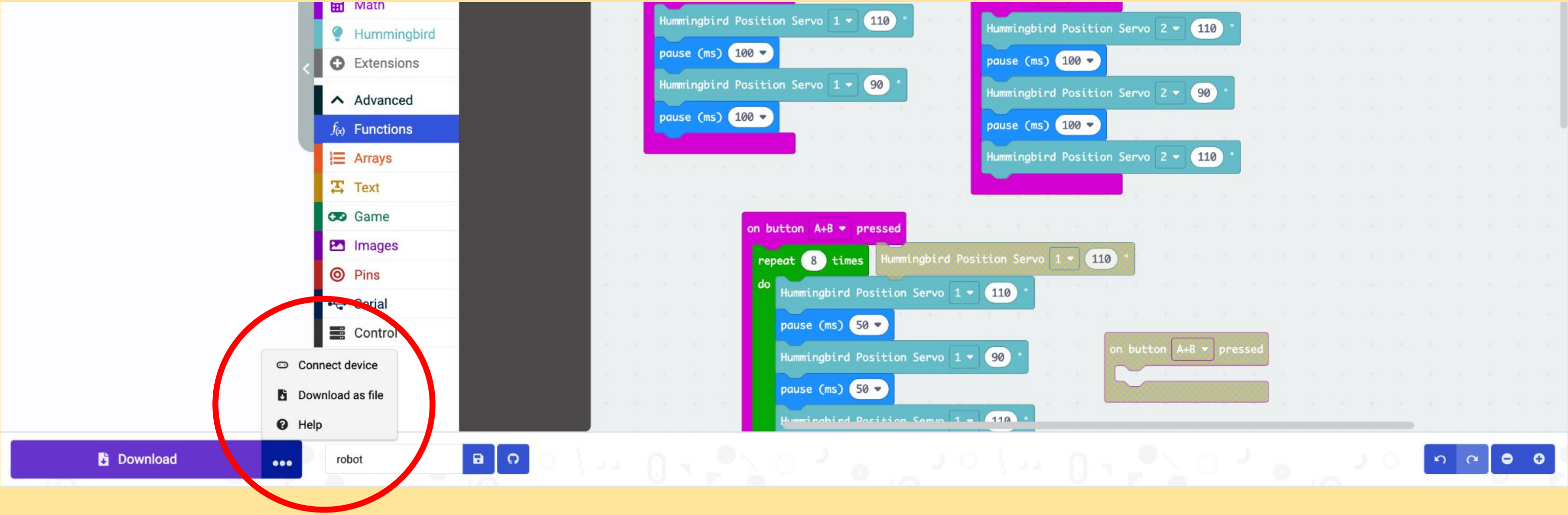


Step 3: Connect Device/download code

Google Chrome Recommended for easy downloading of code

When you are ready to test code, click download.

*If you are not using google chrome, you will need to drag **.hex file** from ‘downloads’ into the ‘MICROBIT’. It will show up similar to a USB drive.*

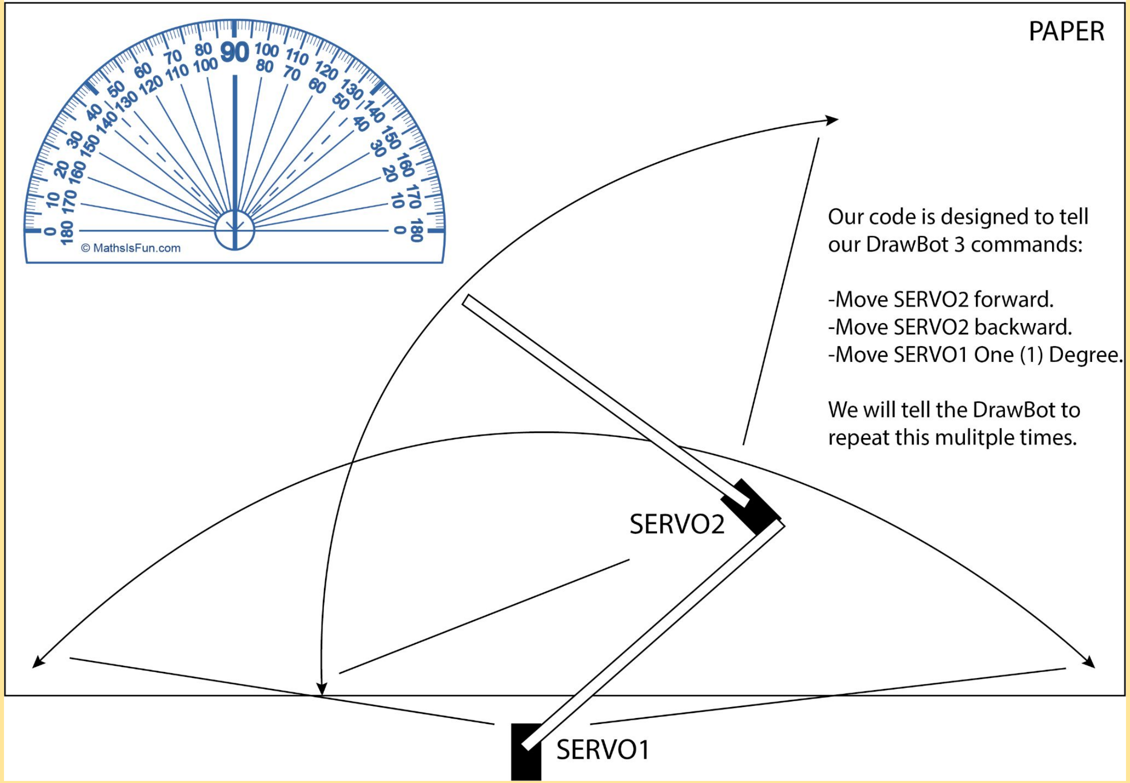


Step 4:

Add plastic horns

- Connect a 4 sided plastic horns to each of the servo motors.
- Attach a small strip of **blue tape** to each horn. We will use these to verify everything is working before we get to building the drawing arms.

DrawBot LAYOUT



Our code is designed to tell our DrawBot 3 commands:

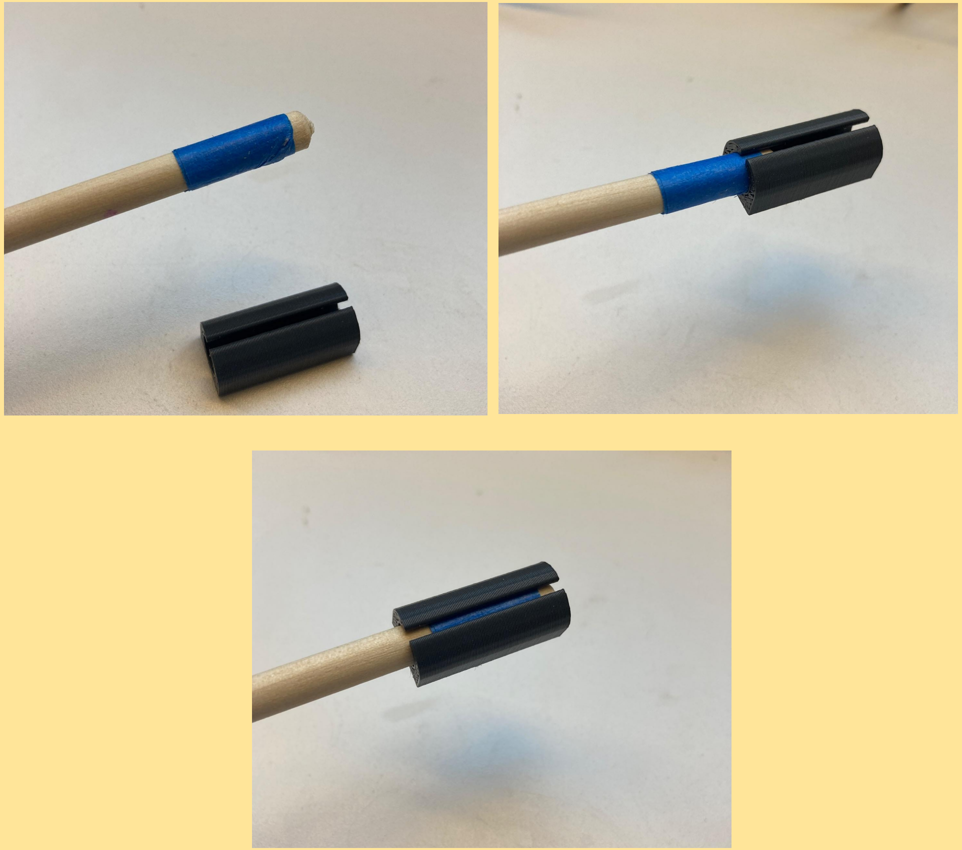
-Move SERVO2 forward.
-Move SERVO2 backward.
-Move SERVO1 One (1) Degree.

We will tell the DrawBot to repeat this mulitple times.

We are going to hot glue SERVO 1 to the table to keep it stable.

Step 5:

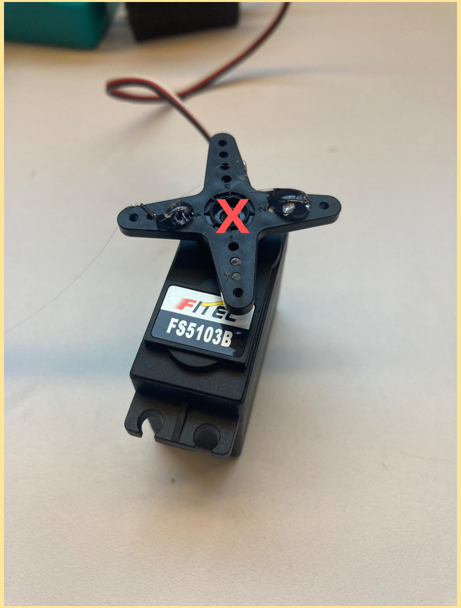
Build the arms

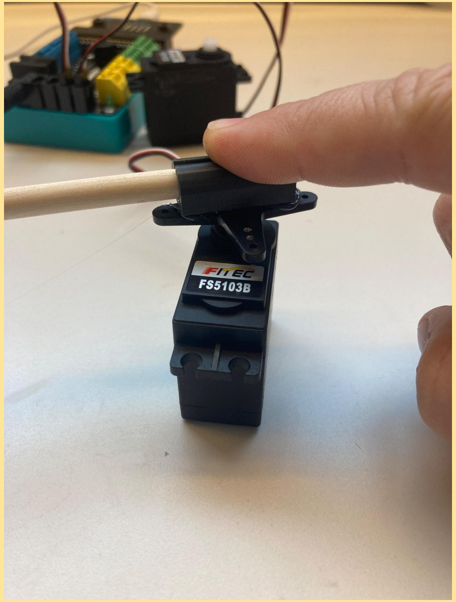


- Wrap 1 layer of blue painters tape around the end of the wooden dowel.
- Slide a wooden dowel into the bracket GENTLY It helps if you rotate the dowel as you are sliding into the bracket.

Step 5, cont:
Build the arms

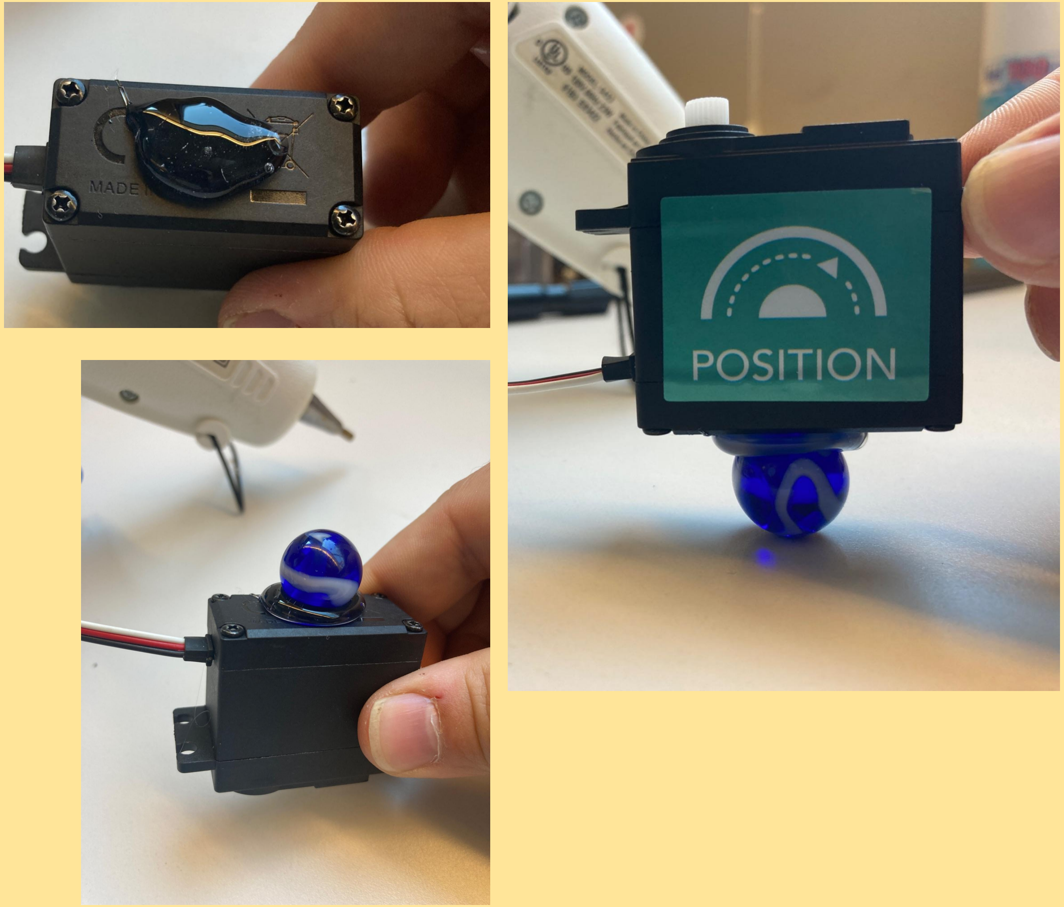
- Add hot glue to the *arms* of plastic horn. Be mindful to NOT apply hot glue in the center (X) of the horn. This will interfere with the white motor gear.
- Using the the dowel to hold the bracket over the horn, place the 3D printed bracket to the horn, making sure to keep it parallel to the horns direction.
- You'll do this for both horns and place one horn/bracket set on each motor(Servo1 and Servo2).





Step 6:
Build the arms

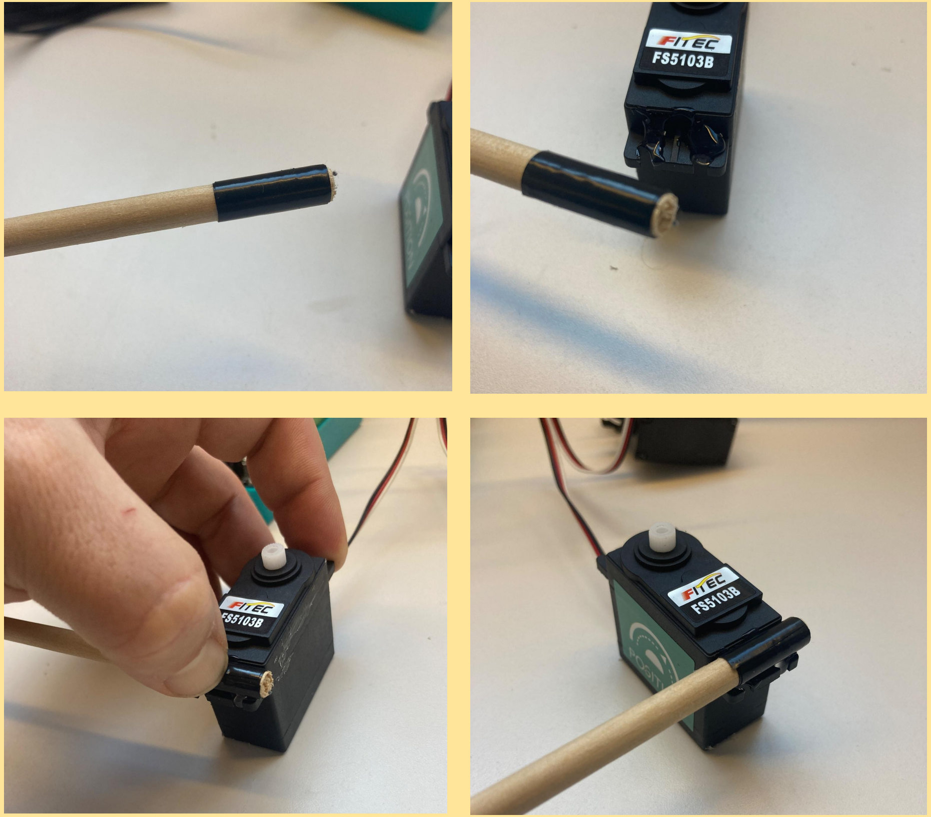
- Hot glue the marble to the bottom of the Draw Arm (servo2). This is going to help the servo glide along the paper, keep the weight distributed, and match the height of the first arm/wooden dowel combo.



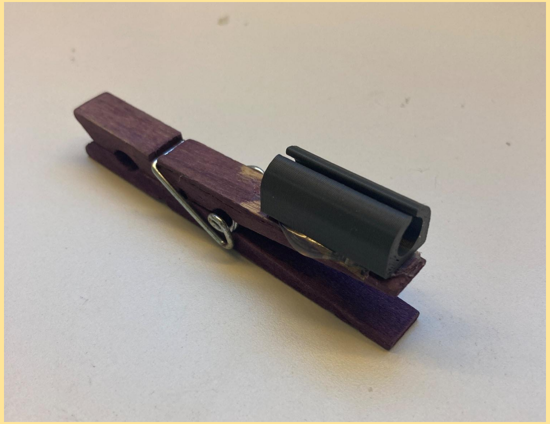
Step 7:

Build the arms

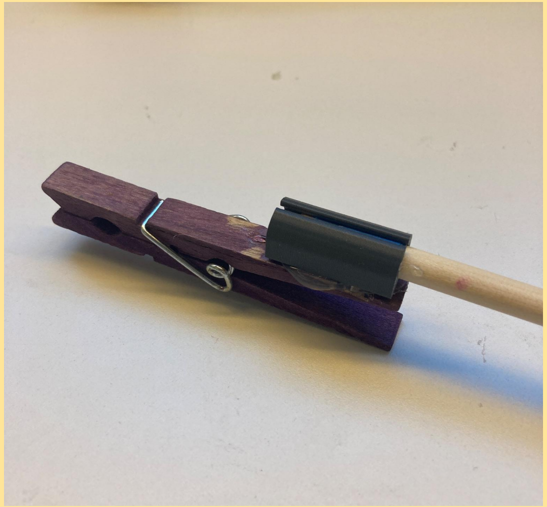
- Wrap the end of the wooden dowel in 4-5 layers of electrical tape before gluing to the servo motor. This will help with the removal of the arm.
- Hot glue the wooden dowel to the back of the Servo2 motor. Make sure you are gluing on the OPPOSITE side of the servo motor wire.



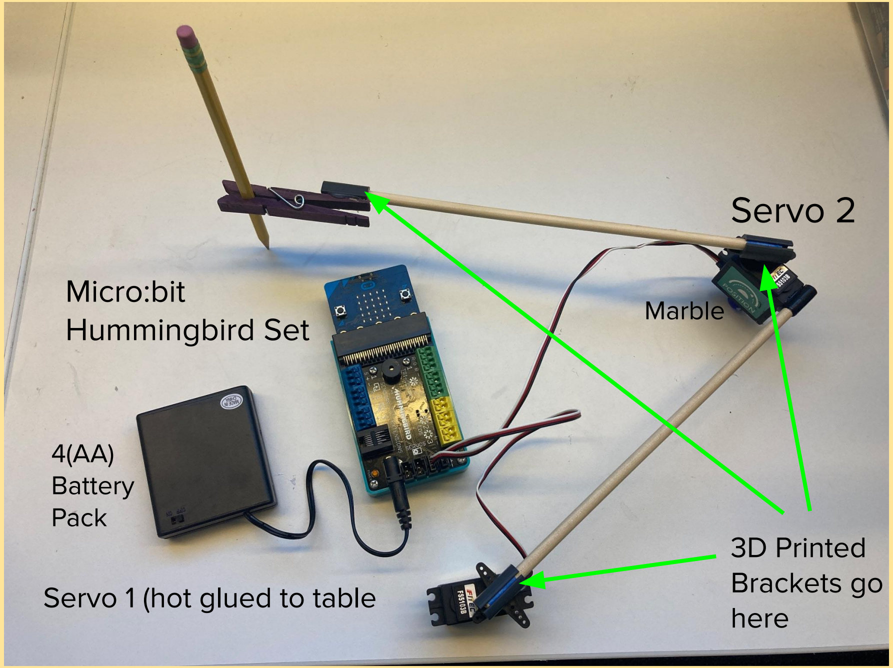
Step 8:
Hot glue Clothes Pin to wooden dowel.

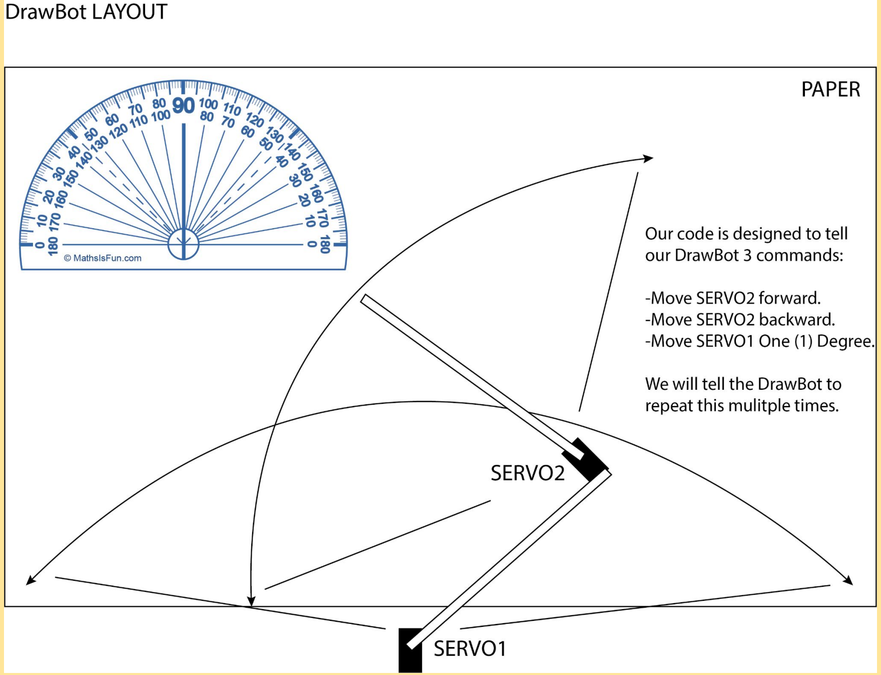


- Hot glue clothes pin to 3d printed bracket and ***slide*** on the end of the 2nd wooden dowel.
- Wooden dowel should have 1 layer of blue tape wrapped around. This may be different depending on your 3D printers results.



Final set up



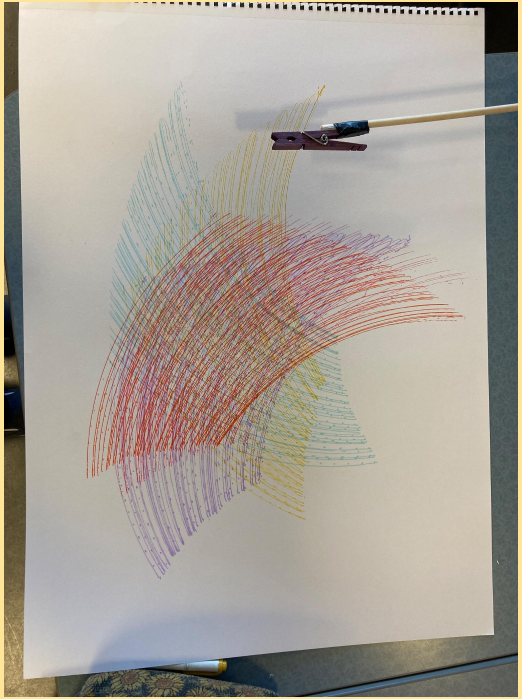


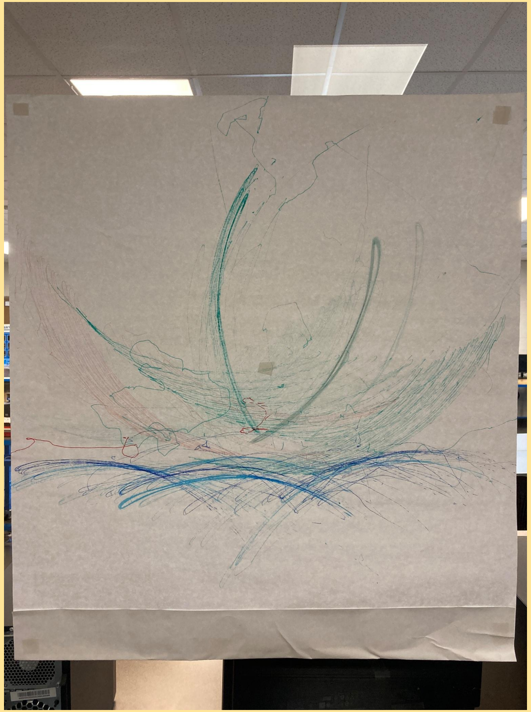
Our code is designed to tell our DrawBot 3 commands:

- Move SERVO2 forward.
- Move SERVO2 backward.
- Move SERVO1 One (1) Degree.

We will tell the DrawBot to repeat this mulitple times.

Examples





Order of Operations

1. *Make sure all components are connected to Hummingbird Kit.*
    a. *Connect servo motors into Slots 1 and 2, with the black wire facing the (-) symbol*
2. *Turn on 4(AA) battery pack, check for Orange Power Light.*
3. *Press A Button on Microbit to set the motors to their starting position.*
4. *THEN attach motor horns/3d printed bracket combo to servo motors in a position similar to DrawBOT Layout (see next slide)*
5. *Open clothes pin and place marker between clasps. This will give room for the arms to work its way across the drawing surface.*
6. *Make sure to position marker that allows the 2nd wooden dowel arm to be level with drawing surface. The marble glued to the 2nd Servo motor will help the motor slide easier on the paper.*
7. *Press A+B Button to engage the full working code.*
    a. *Its might jerk and not work the first time. This is normal, this process is a learning opportunity to make sure we follow the order of operations to make a successful drawing session.*
    b. *If things get weird, Press B button and this will reset the Micro:bit and Hummingbird.*

Troubleshooting Issues

- ***Motors are jerking/not moving consistently***
    - Make sure you are using Hummingbird coding block “Servo POSITION Servo, not Servo ROTATIONAL” Hummingbird command.
    - Double Check Variable values are correct and make sure the order of Functions called is correct.
        - Refer to “Step 7: Code the Buttons” slide

- ***Marker lines are dotted/skipping***
    - Make sure your marker is as close to 90 degree from drawing surface.
    - Try rotating the wooden dowels in the 3D printed bracket.

- ***Motors are not moving***
    - Check 4(AA) battery pack is connected and turned ON.
    - Ensure Microbit is fully seated in slot of Hummingbird.
    - Ensure the servo motor connection is secure and the correct orientation.
        - The Black wire should align with the (-) symbol on Hummingbird

Questions? Comments? Feedback?

Email Aaron Artrip:

aartrip3@gatech.edu