CuteBot Instructions

IoT4Ag
The Internet of Things for Precision Agriculture
an NSF Engineering Research Center

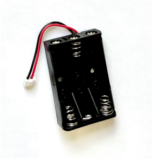
NSF

Penn
UNIVERSITY *of* PENNSYLVANIA

Objectives: SWBAT

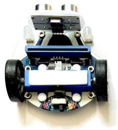
1. Program the CuteBot to follow a line or track
2. Program the CuteBot to avoid collision with objects

Your Kit

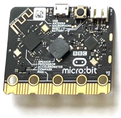


Battery Pack

USB Cable

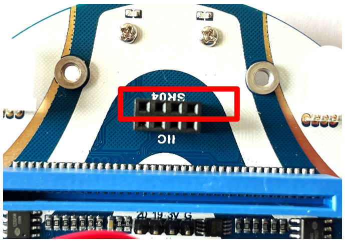


Assembled Smart Cutebot Robot *



Micro:Bit Microcontroller

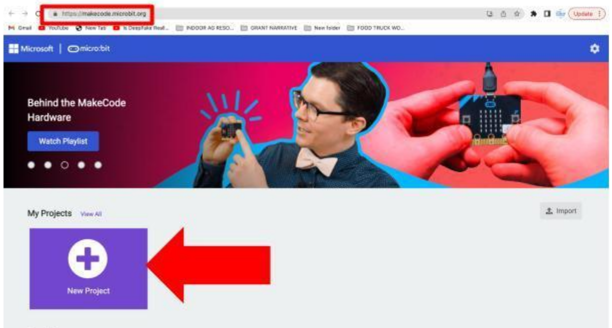
* Make sure cutebot eyes are placed in the front row of pin sockets



See your name in lights / Give your project a name

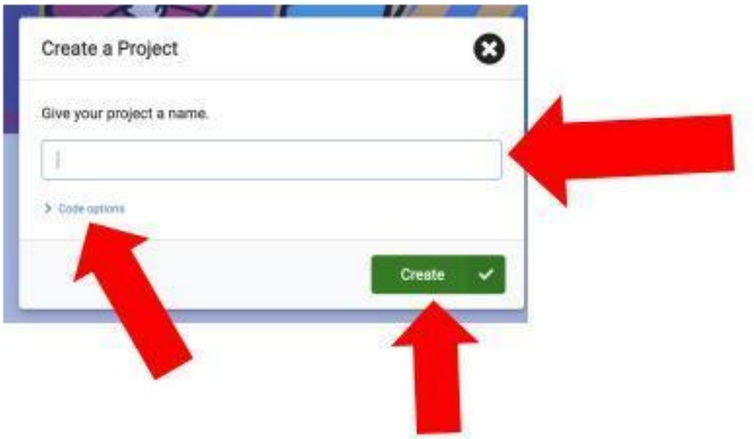
Programming your Microbit

1. You can program the microbit by going to https://makecode.microbit.org
2. Click on "+ New Project".
3. Give your project a name in the dialog box.

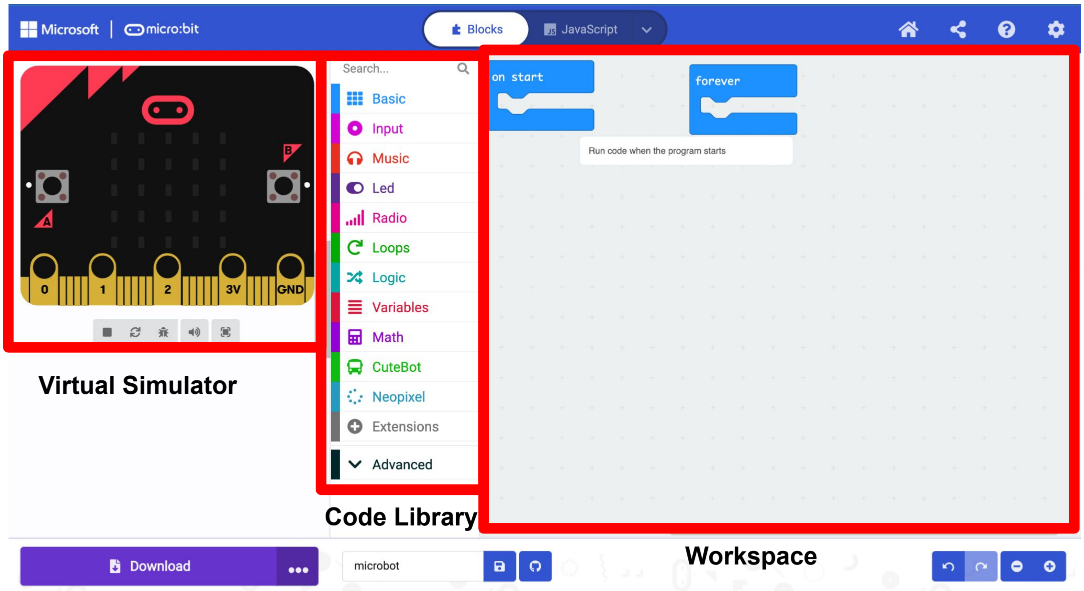


For the Code Options, you can select “Blocks, JavaScript or Python”. For this activity select “Blocks.”

Then click “Create”.



Your New Project Looks Like This



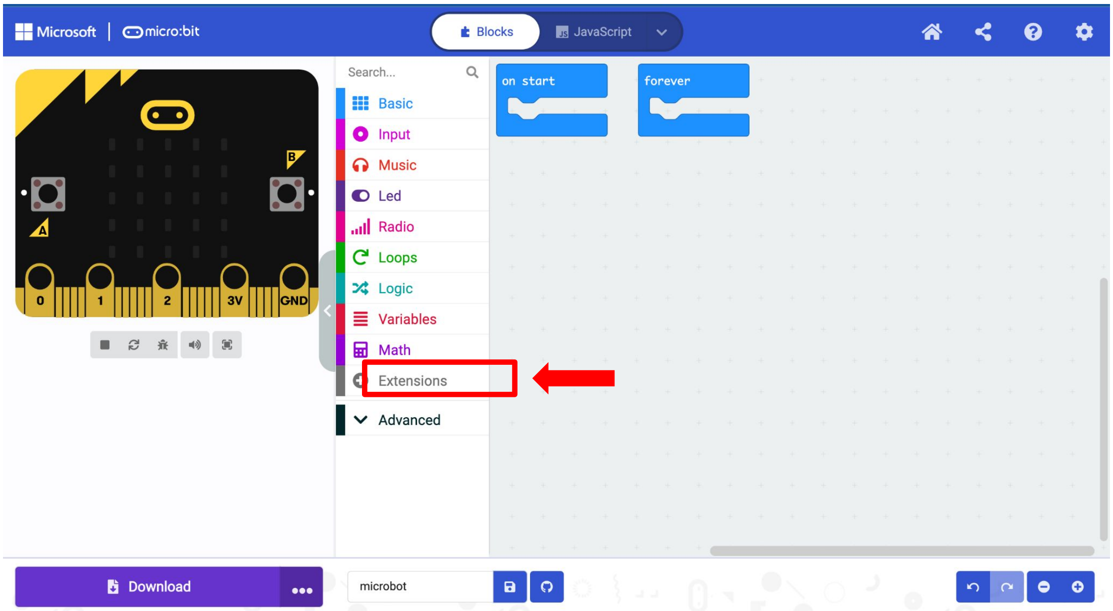
Run code when the program starts

Virtual Simulator

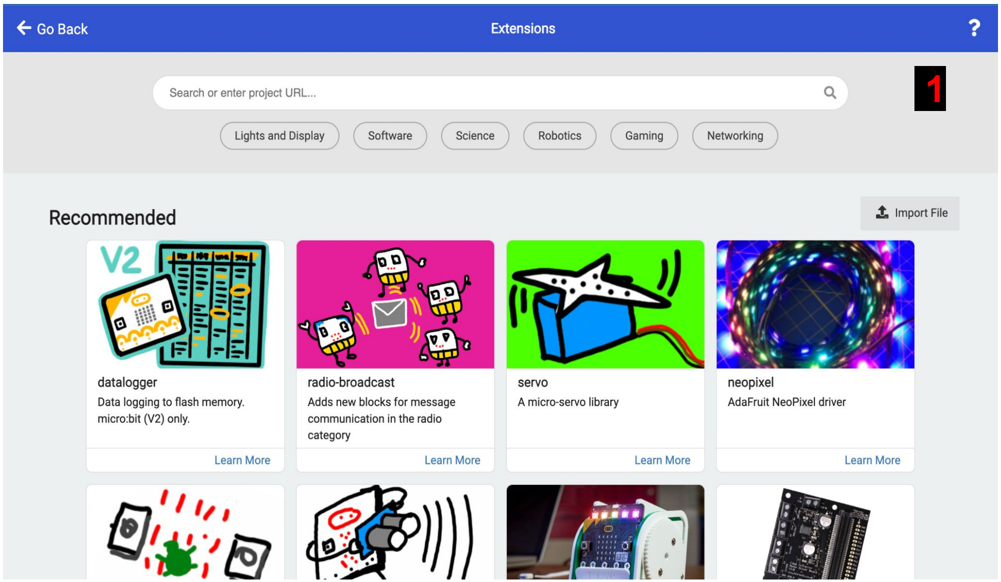
Code Library

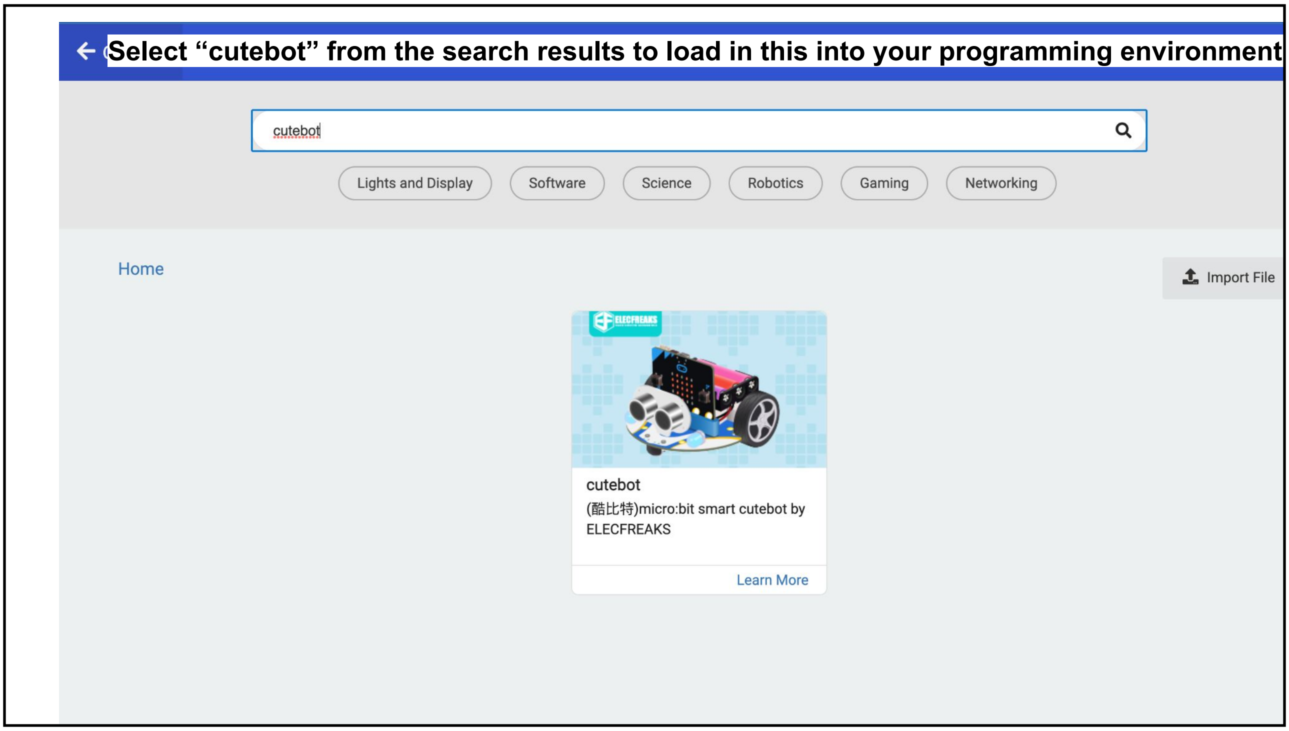
Workspace

Next, you need to add in the Extension pack for the CuteBot. To do this, click“Extensions” from this menu.

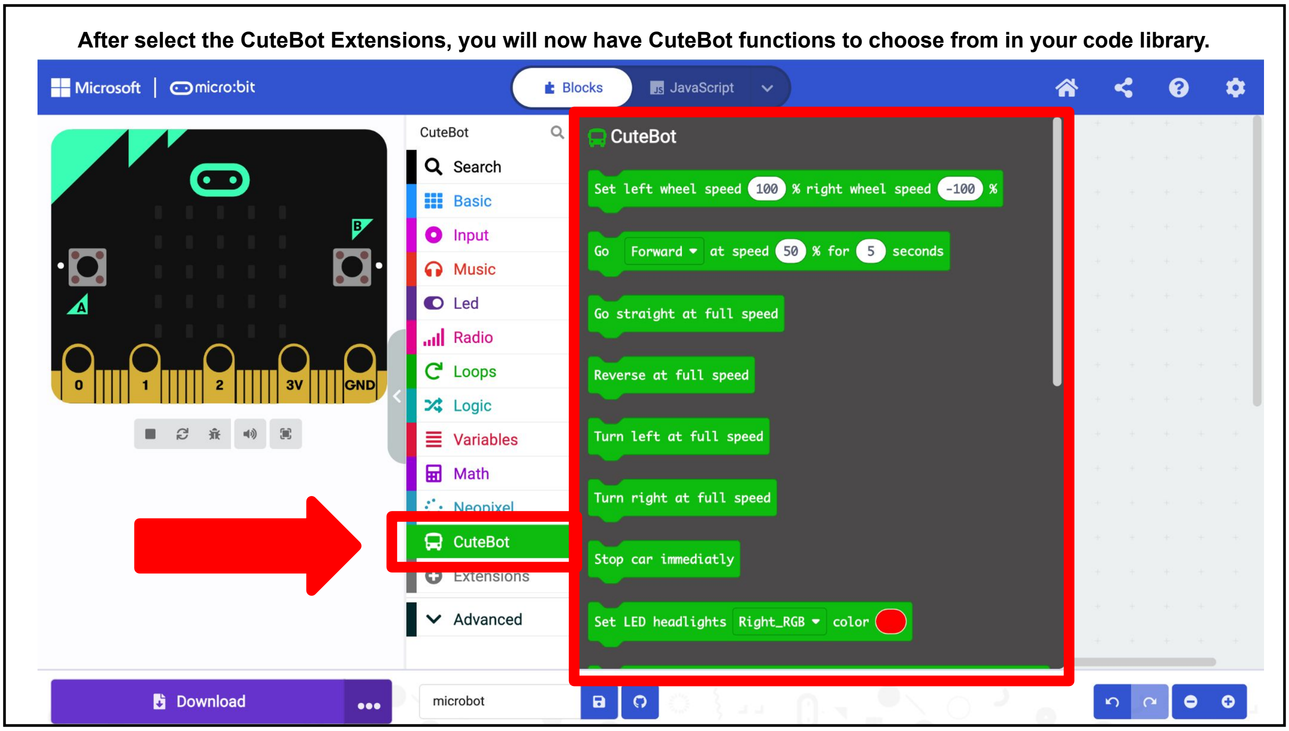


You will now be at a page where you can search for extensions for many different projects. Type in “CuteBot” in the search bar and hit return.





Select “cutebot” from the search results to load in this into your programming environment

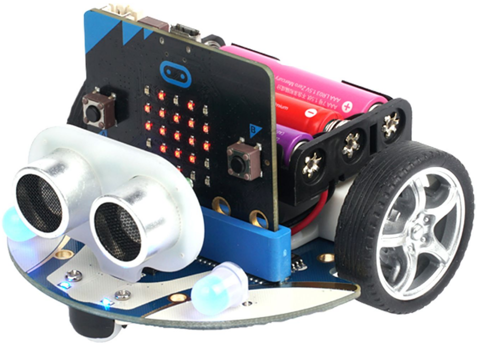


After select the CuteBot Extensions, you will now have CuteBot functions to choose from in your code library.

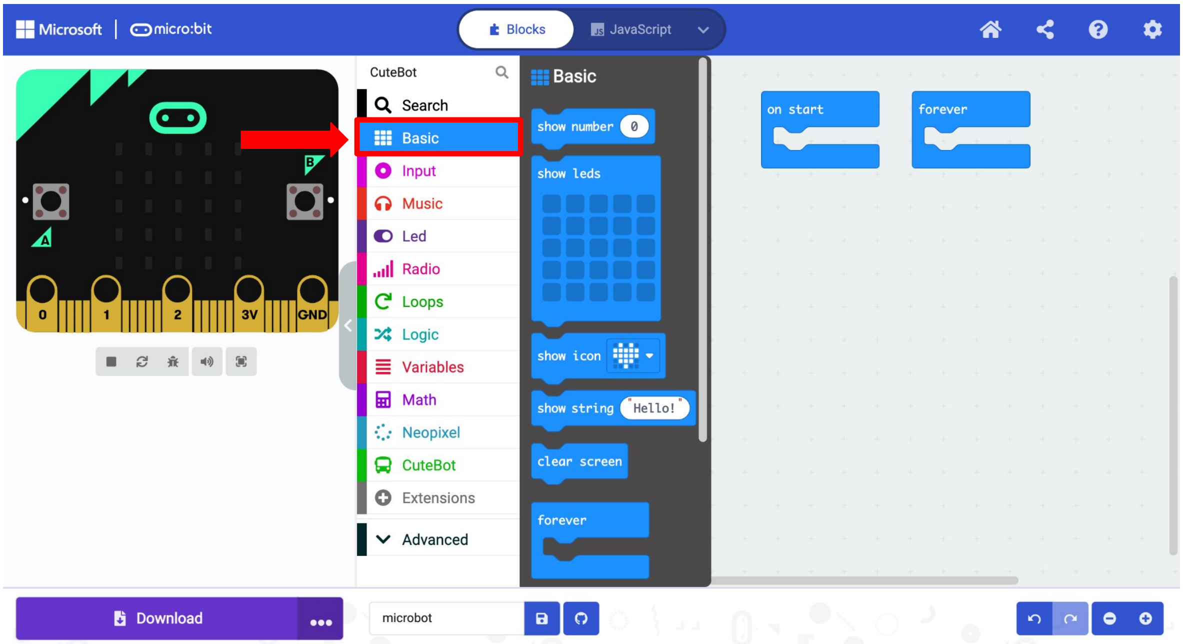
CuteBot

- Set left wheel speed 100 % right wheel speed -100 %
- Go Forward at speed 50 % for 5 seconds
- Go straight at full speed
- Reverse at full speed
- Turn left at full speed
- Turn right at full speed
- Stop car immediatly
- Set LED headlights Right_RGB color

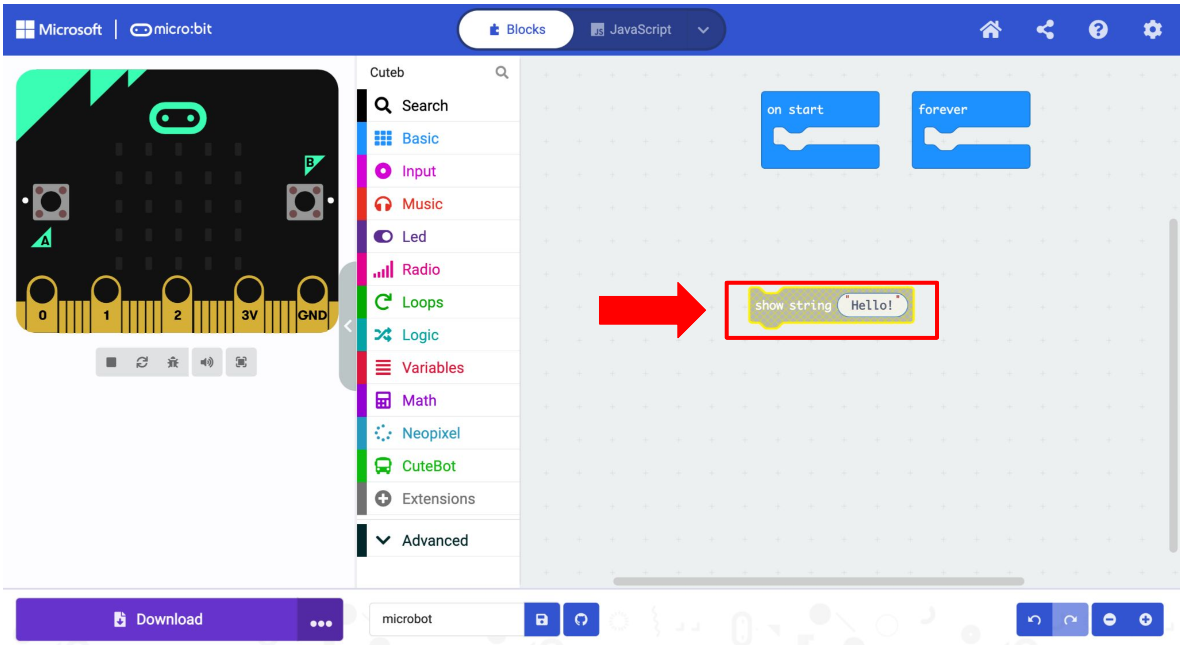
Now we can create a program to test the micro:bit and CuteBot. It displays text on the LED screen of the micro:bit at start up and drives the CuteBot forwards and backwards four times.



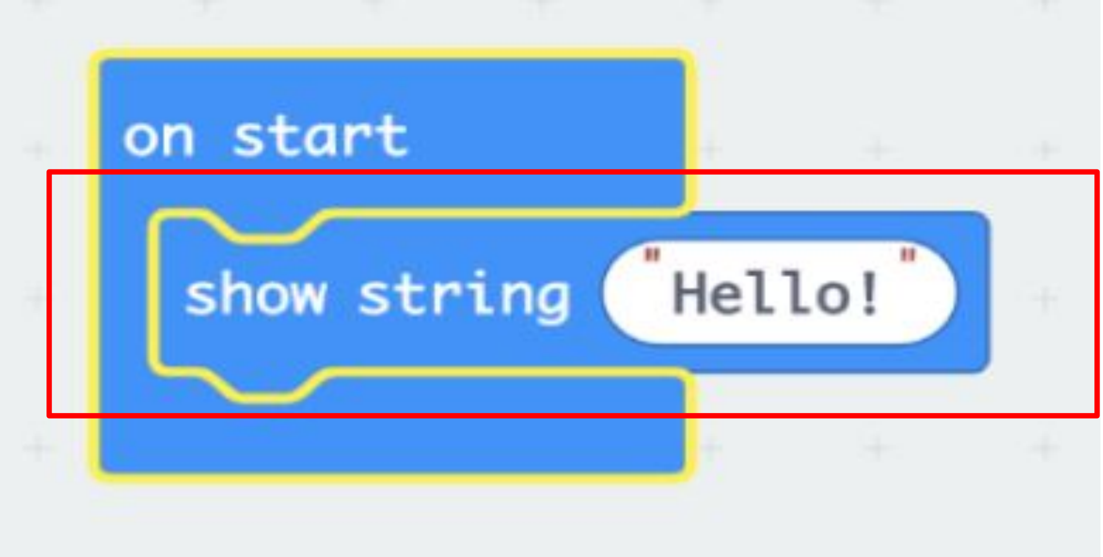
Start by clicking on the “Basic” from the code library.



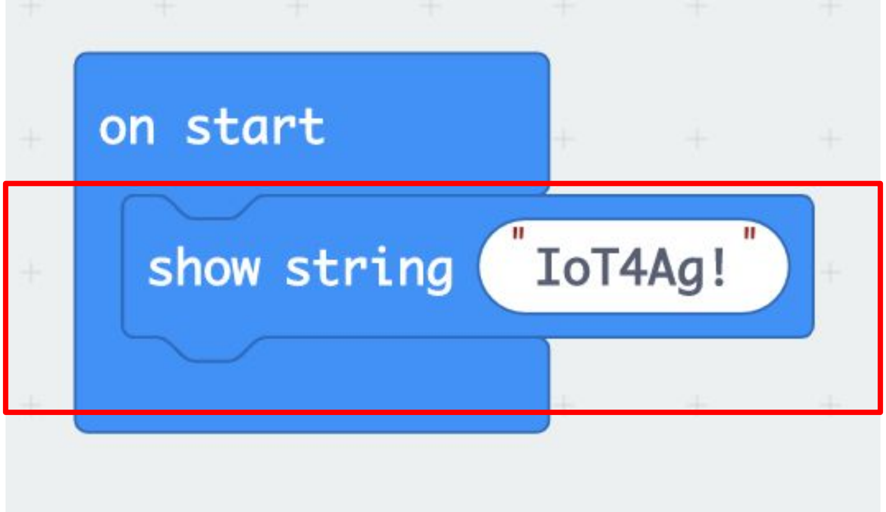
Select the “Show String” block and drag it to the program environment.



Move it inside the “On start” block and it will turn blue showing that there are no errors.

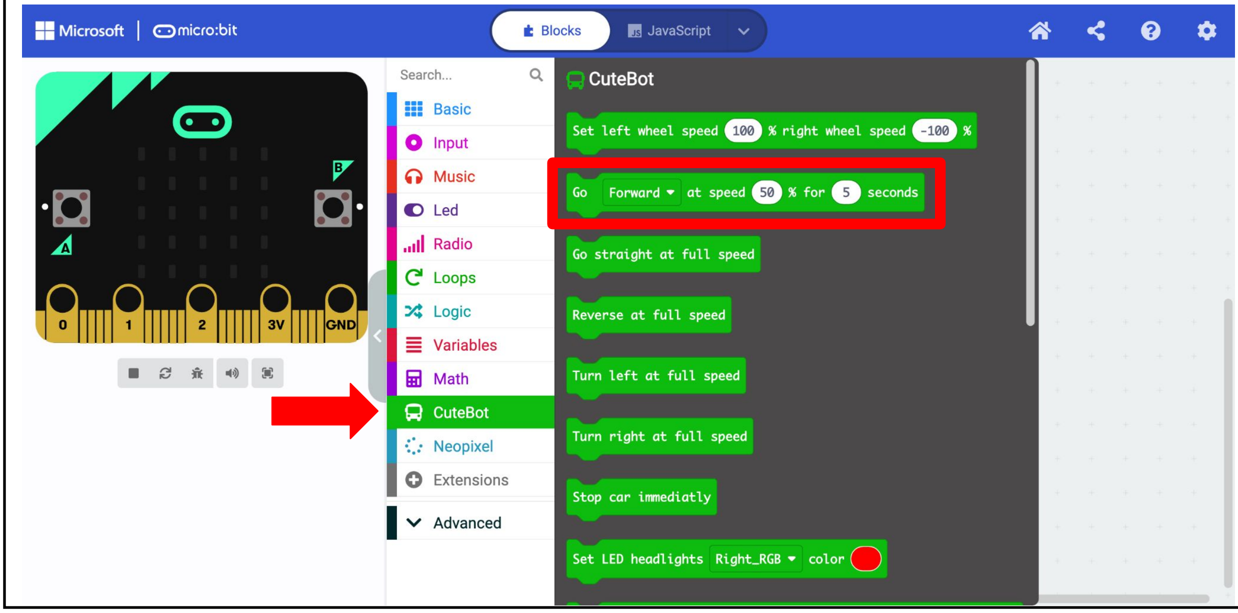


Double-click the text “Hello!” and change to “IoT4Ag!”



```
on start
  show string "IoT4Ag!"
```

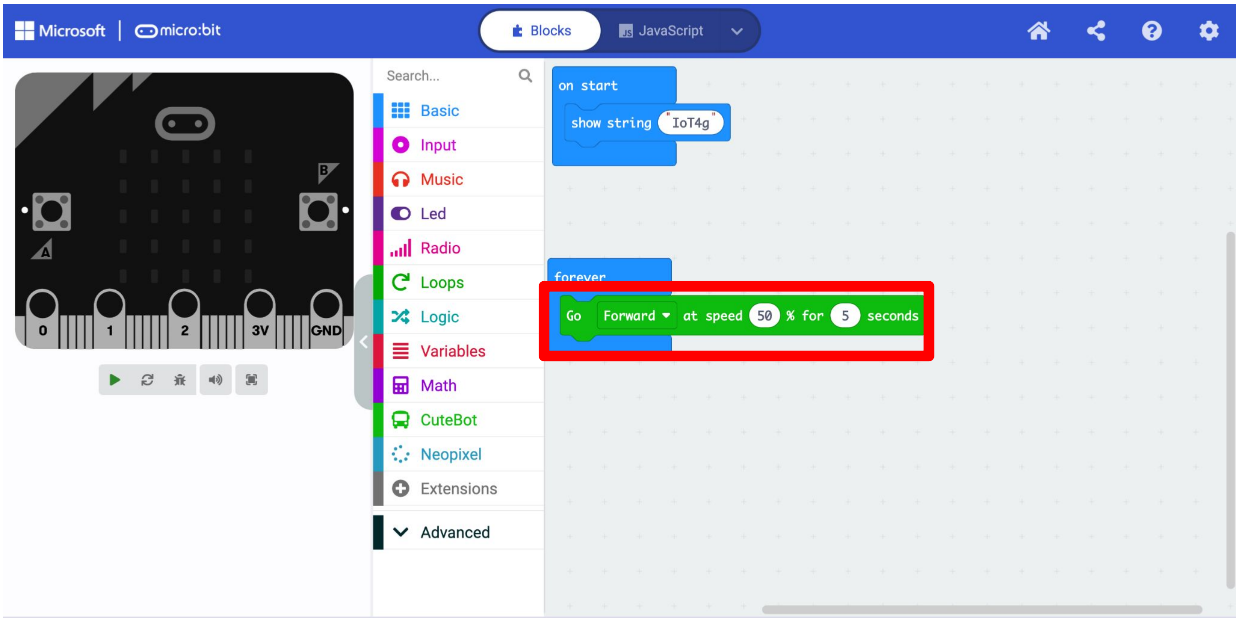
Next, go to the CuteBot menu from the code library and drag out the “go forward at speed 50 for 5 seconds” block.



CuteBot

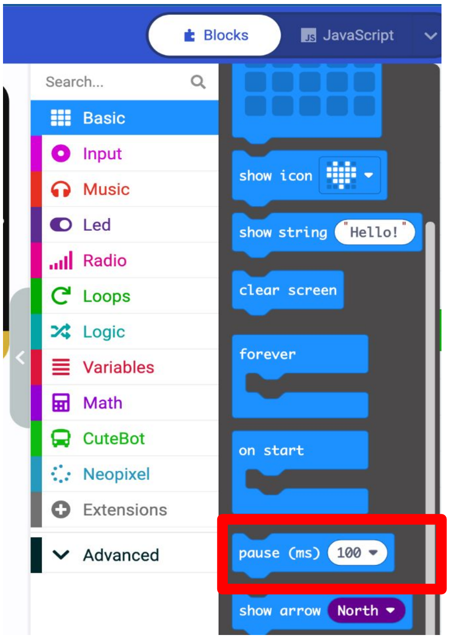
```
Set left wheel speed 100 % right wheel speed -100 %
Go Forward ▾ at speed 50 % for 5 seconds
Go straight at full speed
Reverse at full speed
Turn left at full speed
Turn right at full speed
Stop car immediatly
Set LED headlights Right_RGB ▾ color
```

Drag it into the forever block.

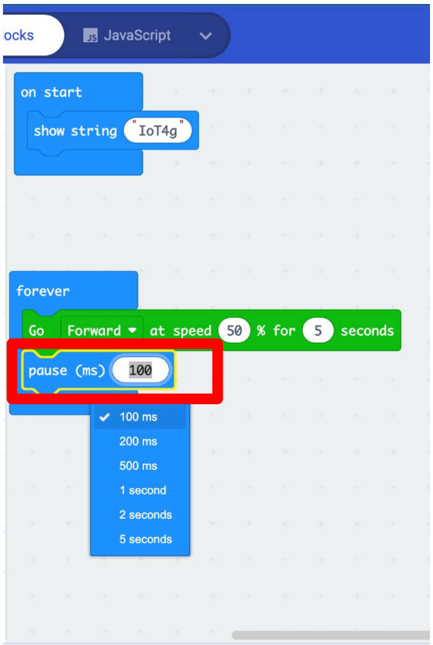


```
on start
    show string "IoT4g"

forever
    Go Forward at speed 50 % for 5 seconds
```

Then go to the Basic menu and drag out a “pause (ms)” block.

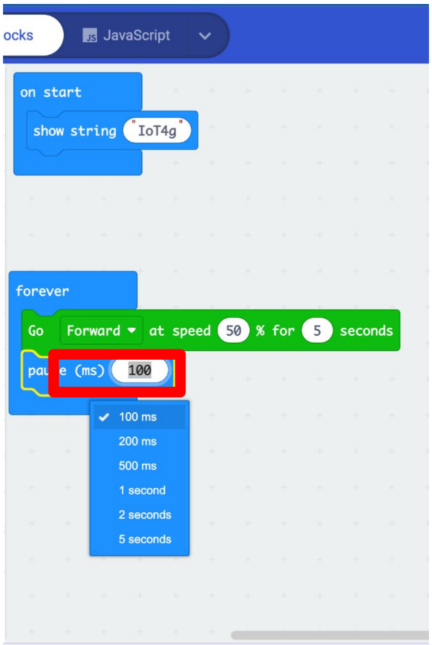


Insert it inside the forever block after the “go forward...” block.



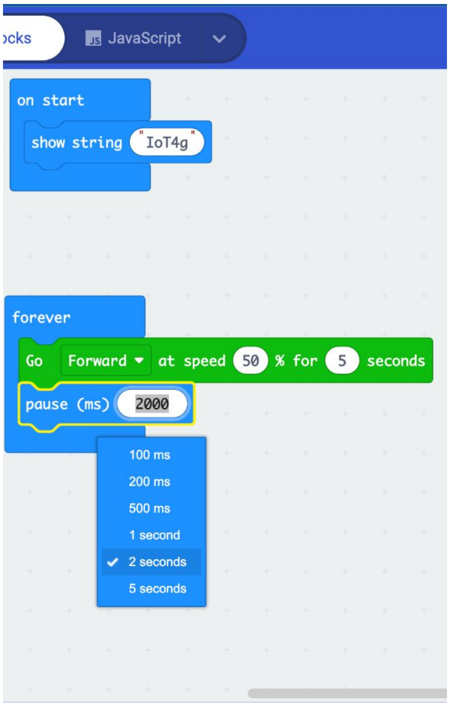
```
on start
    show string "IoT4g"

forever
    Go Forward ▾ at speed 50 % for 5 seconds
    pause (ms) 100
```

Once it is in the forever block, click the “100” value and change it to 2 seconds.



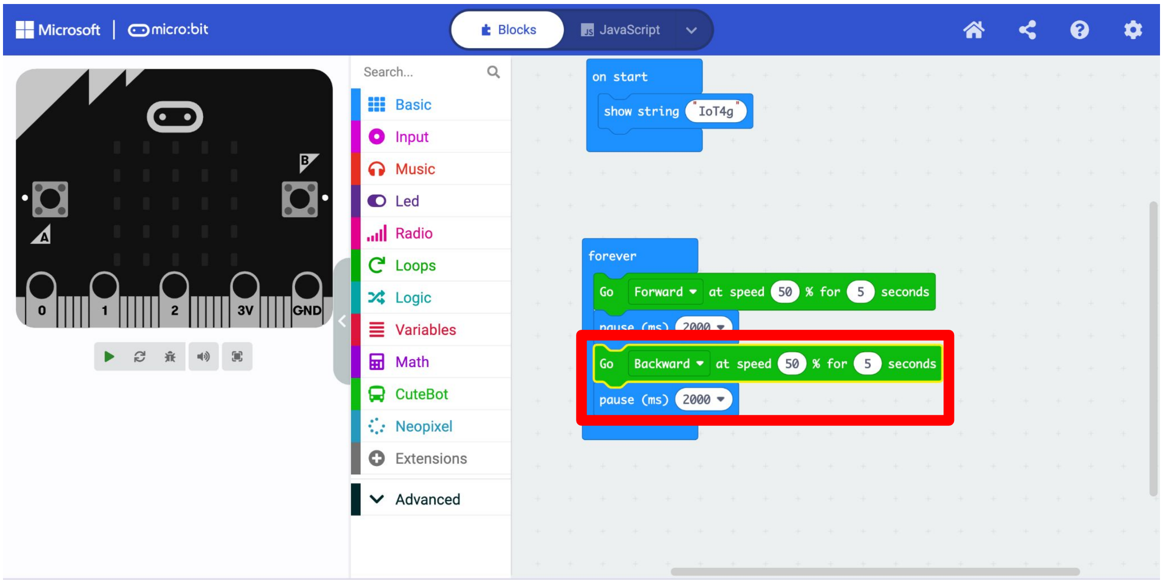
```
on start
    show string "IoT4g"

forever
    Go Forward ▾ at speed 50 % for 5 seconds
    pause (ms) 100
```

It will show up as “2000” since it is units of milliseconds (ms).



```
on start
    show string "IoT4g"

forever
    Go Forward ▾ at speed 50 % for 5 seconds
    pause (ms) 2000
```

Select another “go forward at speed 50 for 5 seconds” block and drag it into the forever block under the pause block. Now, change the “forward” to “backward” in this new block. Add another pause block for 2 seconds after the “go backward...” block.

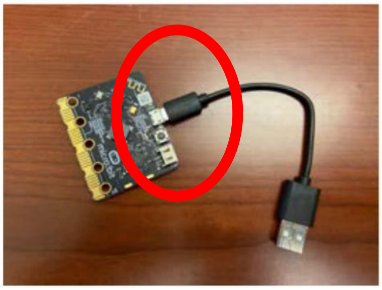


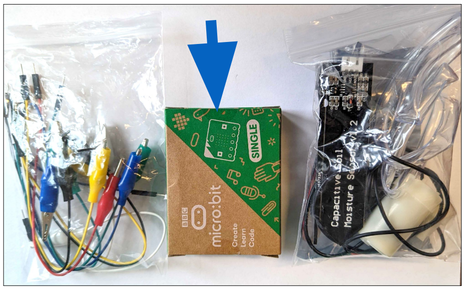
```
on start
    show string "IoT4g"

forever
    Go Forward ▾ at speed 50 % for 5 seconds
    pause (ms) 2000 ▾
    Go Backward ▾ at speed 50 % for 5 seconds
    pause (ms) 2000 ▾
```

The program is now complete.
It is time to plug the USB cable into the micro:bit and connect it to the computer.

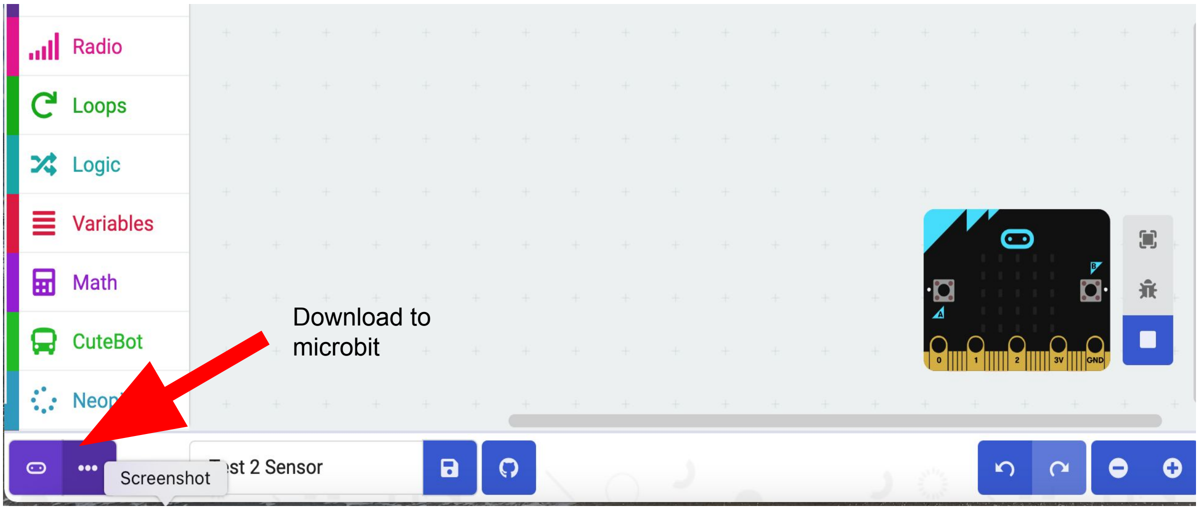
Be careful to be gentle when making this connection.

Note: when you plug in the micro:bit to the computer for the first time it will run it's default greeting program that will buzz and flash its LEDs and say "hello".



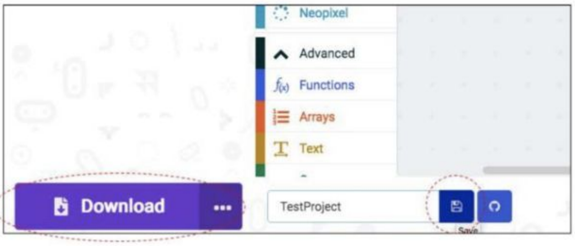


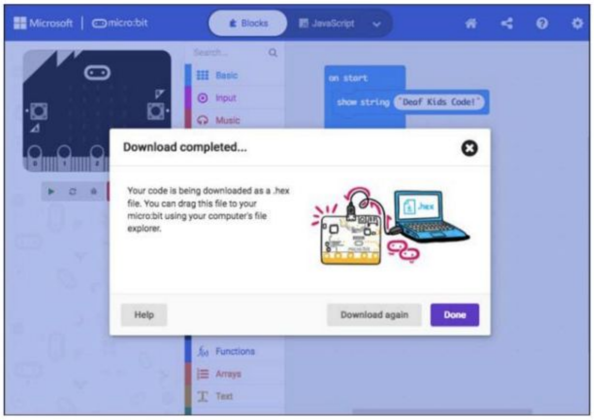
Save your "message" code to the micro.bit (The short way)



Download to microbit

Once plugged into the USB port on the computer, click the “save” button next to the program name or the “Download” button on the bottom left. You will see a dialog box pop up that says your program is being downloaded as a .hex file. You can drag this file to your micro:bit using your computer’s file explorer. Once you do this and the program loads, the LEDs on the micro:bit should display “IoT4Ag!” scrolling by while still plugged into the computer.

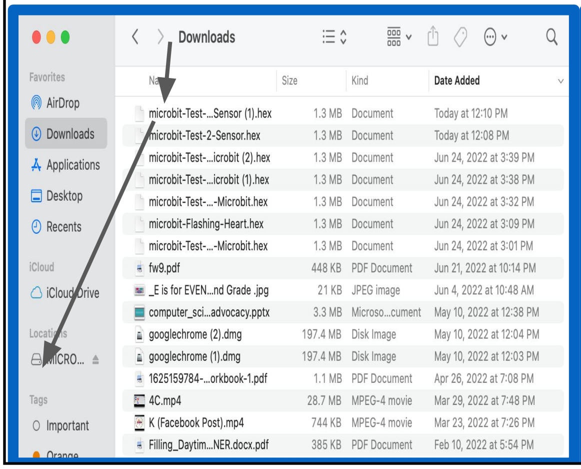




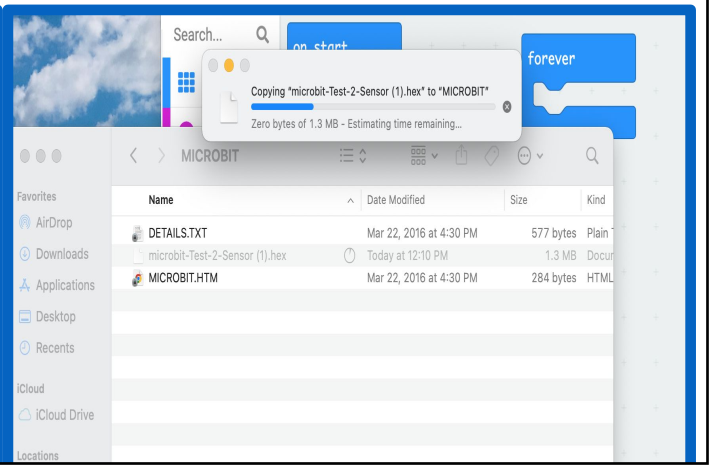
Download completed...

Your code is being downloaded as a .hex file. You can drag this file to your micro:bit using your computer's file explorer.

(The long way): Go to “File” select “Downloads”
In the Finder, drag the latest download to the Micro.bit Processor attached to your computer.



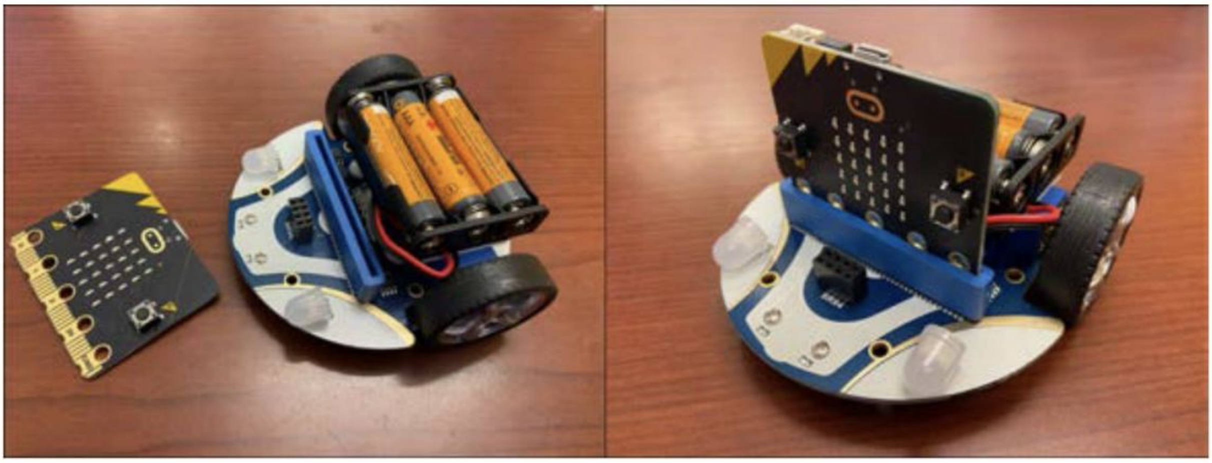
Na	Size	Kind	Date Added
microbit-Test-...Sensor (1).hex	1.3 MB	Document	Today at 12:10 PM
microbit-Test-2-Sensor.hex	1.3 MB	Document	Today at 12:08 PM
microbit-Test-...icrobit (2).hex	1.3 MB	Document	Jun 24, 2022 at 3:39 PM
microbit-Test-...icrobit (1).hex	1.3 MB	Document	Jun 24, 2022 at 3:38 PM
microbit-Test-...-Microbit.hex	1.3 MB	Document	Jun 24, 2022 at 3:32 PM
microbit-Flashing-Heart.hex	1.3 MB	Document	Jun 24, 2022 at 3:09 PM
microbit-Test-...-Microbit.hex	1.3 MB	Document	Jun 24, 2022 at 3:01 PM
fw9.pdf	448 KB	PDF Document	Jun 21, 2022 at 10:14 PM
_E is for EVEN...nd Grade .jpg	21 KB	JPEG image	Jun 4, 2022 at 10:48 AM
computer_sci...advocacy.pptx	3.3 MB	Microso...cument	May 10, 2022 at 12:38 PM
googlechrome (2).dmg	197.4 MB	Disk Image	May 10, 2022 at 12:04 PM
googlechrome (1).dmg	197.4 MB	Disk Image	May 10, 2022 at 12:03 PM
1625159784-...orkbook-1.pdf	1.1 MB	PDF Document	Apr 26, 2022 at 7:08 PM
4C.mp4	28.7 MB	MPEG-4 movie	Mar 29, 2022 at 7:48 PM
K (Facebook Post).mp4	744 KB	MPEG-4 movie	Mar 23, 2022 at 7:26 PM
Filling_Daytim...NER.docx.pdf	385 KB	PDF Document	Feb 10, 2022 at 5:54 PM



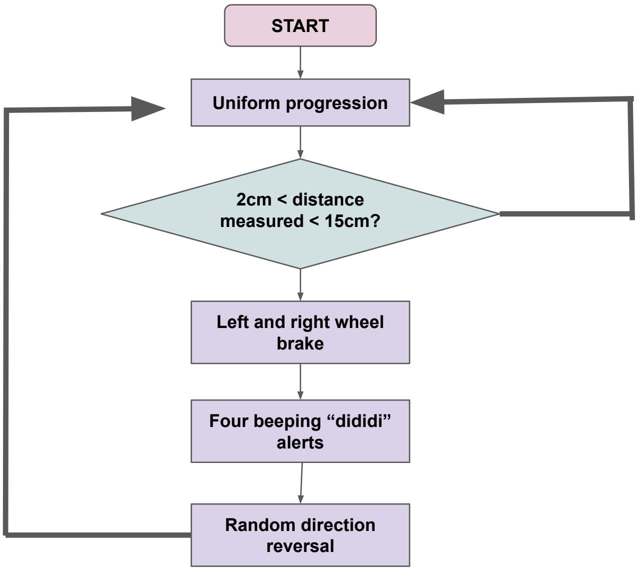
Copying “microbit-Test-2-Sensor (1).hex” to “MICROBIT”
Zero bytes of 1.3 MB - Estimating time remaining...

Name	Date Modified	Size	Kind
DETAILS.TXT	Mar 22, 2016 at 4:30 PM	577 bytes	Plain T
microbit-Test-2-Sensor (1).hex	Today at 12:10 PM	1.3 MB	Docur
MICROBIT.HTM	Mar 22, 2016 at 4:30 PM	284 bytes	HTML

Unplug the micro:bit from the computer and insert it into the CuteBot with the LED side facing out. Make sure to push it down all the way in its connector. Flip the power switch to the ON position and watch the CuteBot go!



Automatic Collision Avoidance



START

Uniform progression

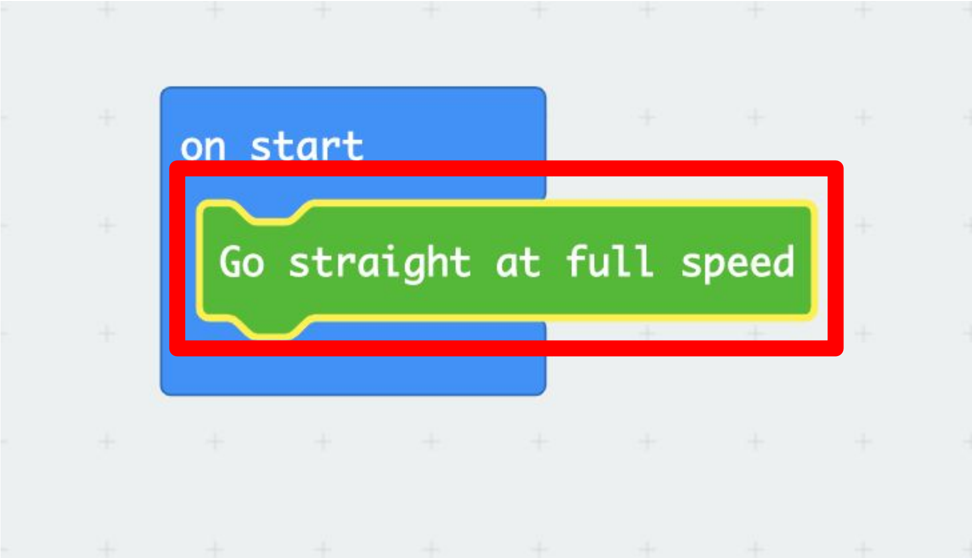
2cm < distance measured < 15cm?

Left and right wheel brake

Four beeping “dididi” alerts

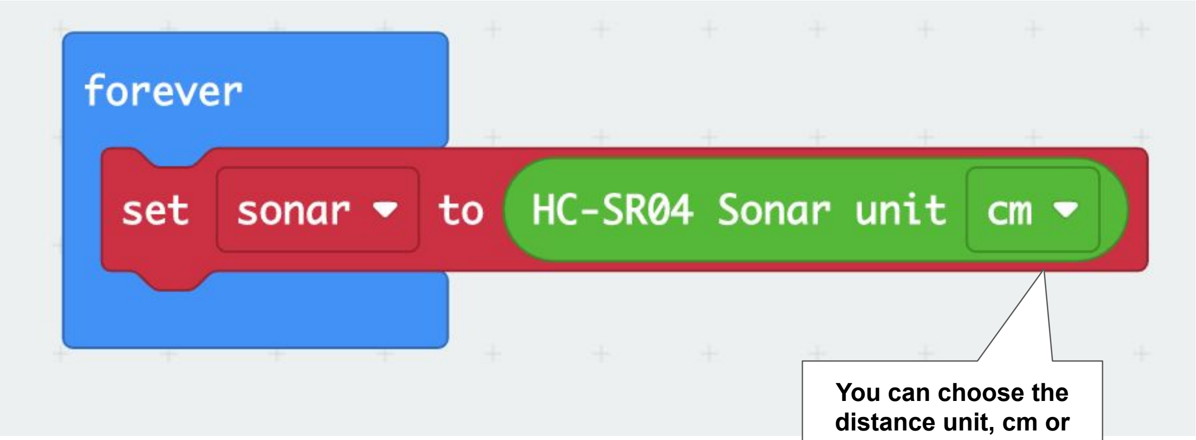
Random direction reversal

Step 1: Move forward at full speed by selecting the block from the CuteBot menu and adding it to “on start”.



on start
Go straight at full speed

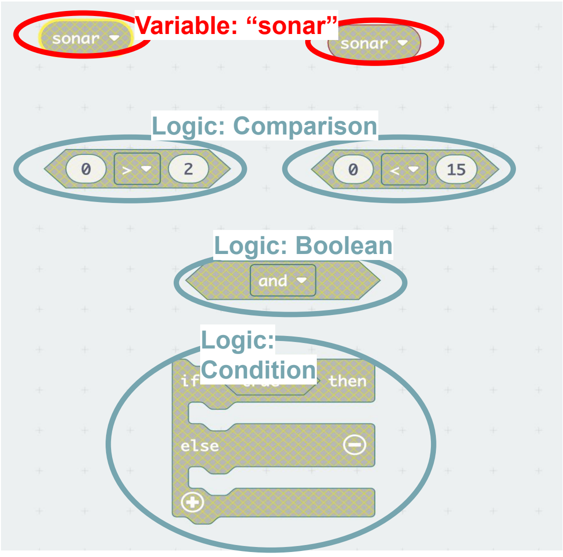
Step 2: Make a “sonar” variable and set it to the detected “HC-SR04 Sonar unit” in cm within “forever”.

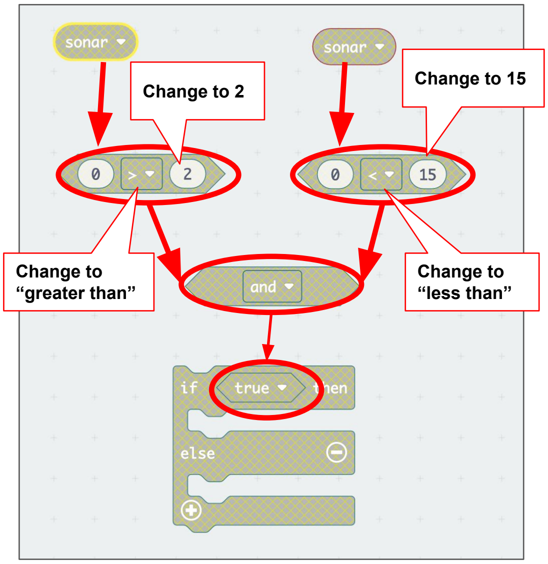


forever
set sonar ▾ to HC-SR04 Sonar unit cm ▾

You can choose the distance unit, cm or inches

Step 3: Check if the distance measured by the ultrasonic sensor (now saved in our “**sonar**” variable) is between 2cm and 15cm by using various functions identified below.





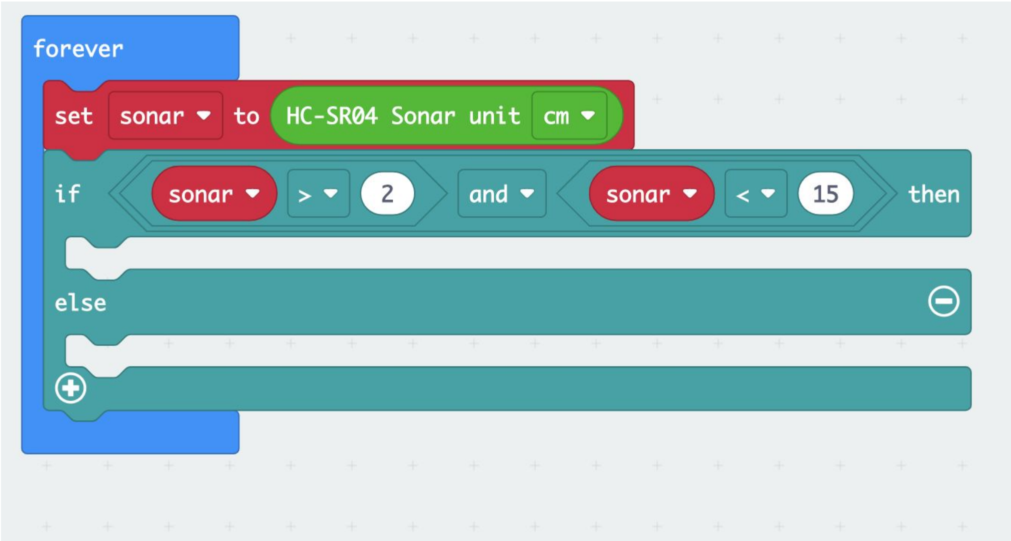
Change to 2

Change to 15

Change to “greater than”

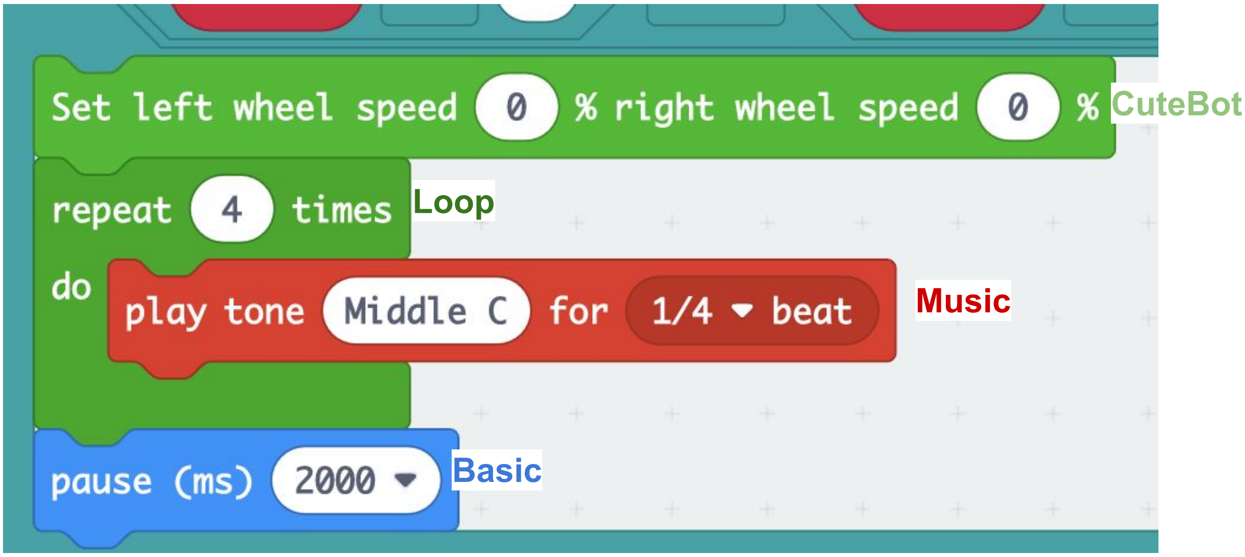
Change to “less than”

Your code should look like this.



```
forever
    set sonar to HC-SR04 Sonar unit cm
    if sonar > 2 and sonar < 15 then

    else

```

Step 4: If the **“sonar”** value is between 2cm and 15cm, **set the speed of both wheels to 0 to stop the cutebot from moving,** then it **alarms four times**. Then **pause for 2 seconds**. Use the image below to create the code. Use the labels to help locate the code in the code library.



```
Set left wheel speed 0 % right wheel speed 0 %
repeat 4 times
do
    play tone Middle C for 1/4 beat
pause (ms) 2000
```

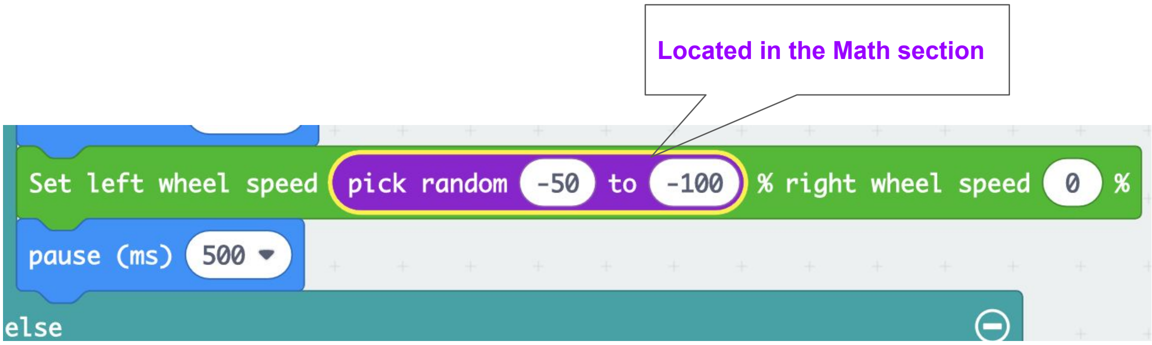
CuteBot

Loop

Music

Basic

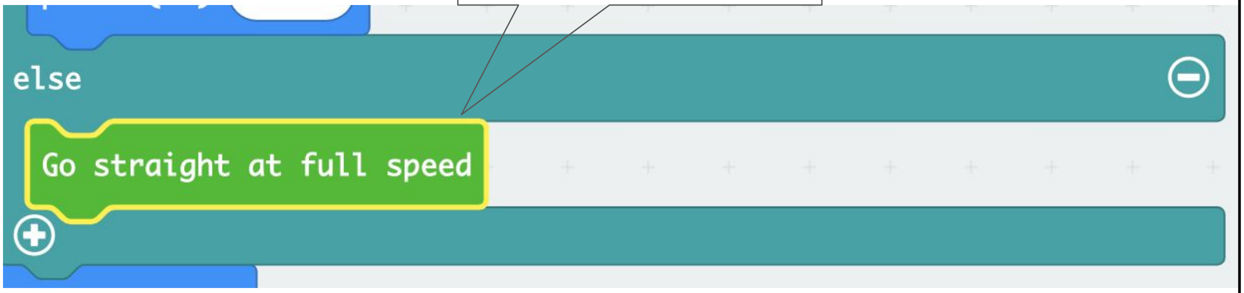
Step 5: **Set the left wheel speed** to a **random** value between -50 to -100. The CuteBot backs up and turns its direction. Then **pause** 500ms.



Located in the Math section

```
Set left wheel speed pick random -50 to -100 % right wheel speed 0 %
pause (ms) 500
else
```

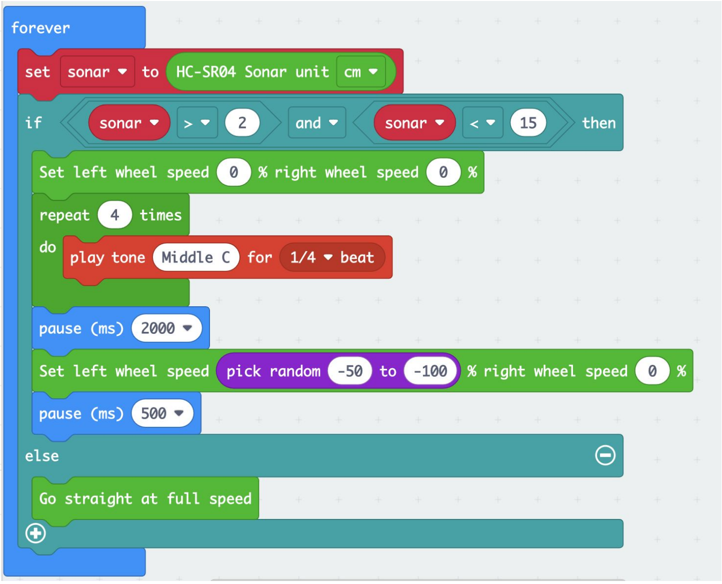
Step 6: Move the CuteBot ahead at **full speed** if the distance measured in the fourth step is not between 2 to 15 cm away from the CuteBot.



```
else
  Go straight at full speed
```

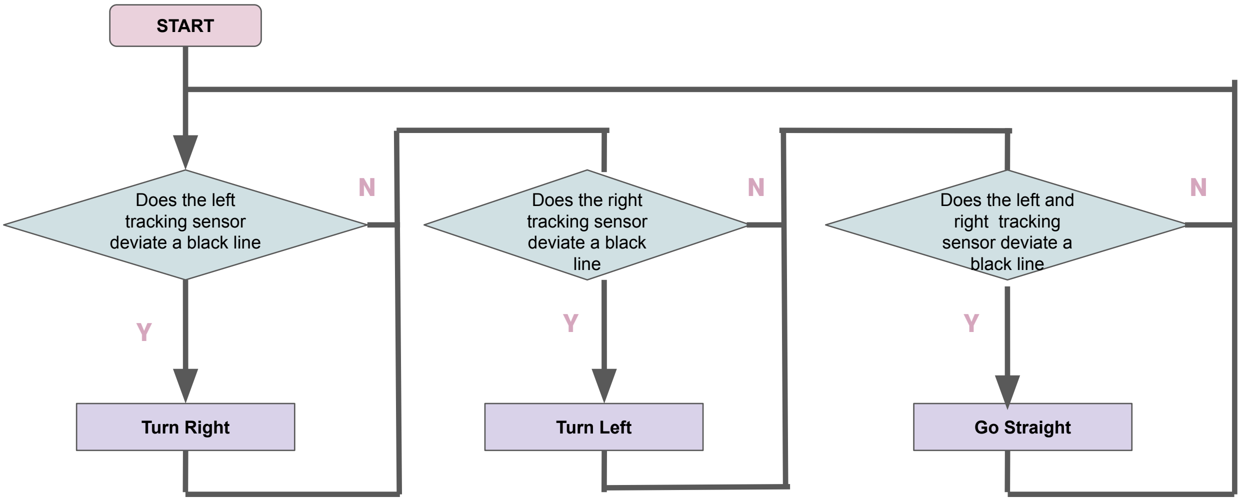
Located in the CuteBot section

Your code should look like this.

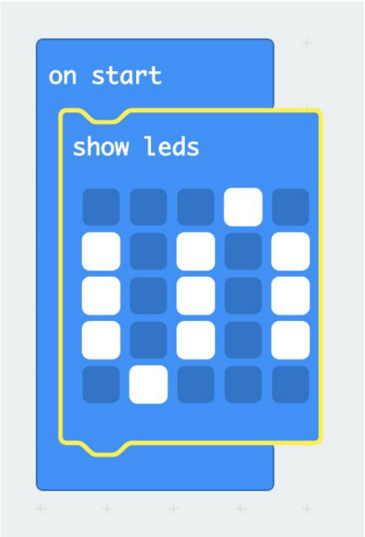


```
forever
  set sonar to HC-SR04 Sonar unit cm
  if sonar > 2 and sonar < 15 then
    Set left wheel speed 0 % right wheel speed 0 %
    repeat 4 times
    do
      play tone Middle C for 1/4 beat
    pause (ms) 2000
    Set left wheel speed pick random -50 to -100 % right wheel speed 0 %
    pause (ms) 500
  else
    Go straight at full speed
```

Line-tracking

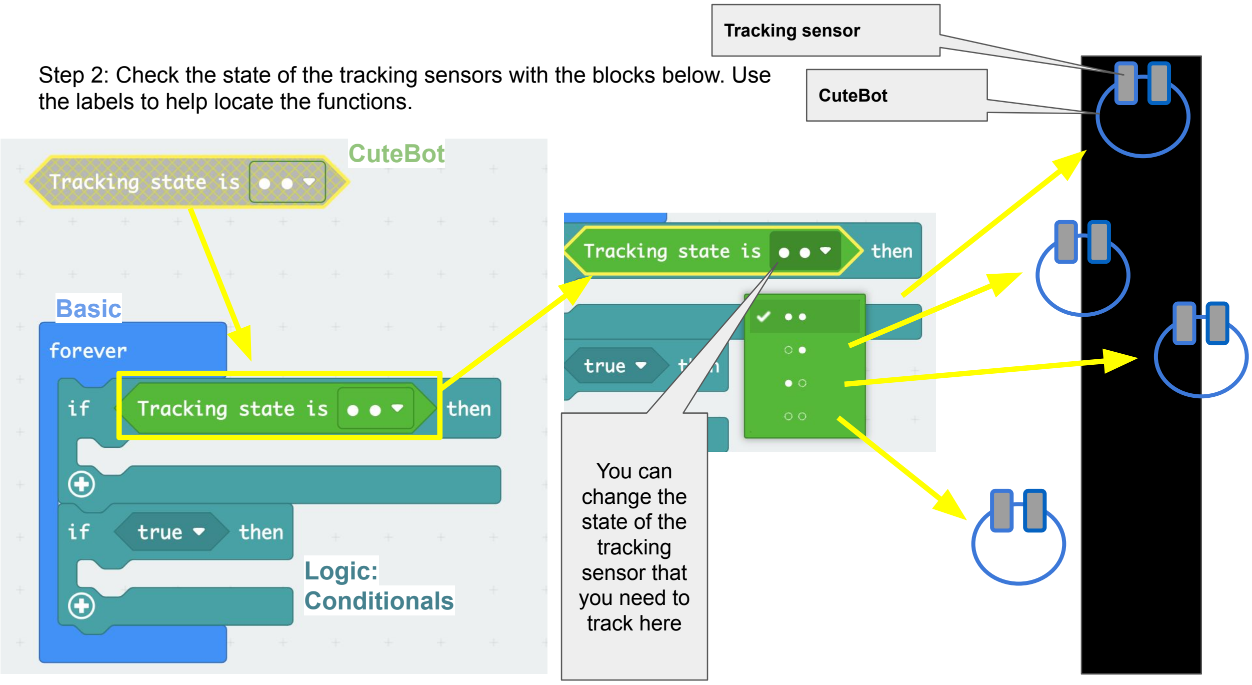

START
Does the left tracking sensor deviate a black line
N
Y
Turn Right
Does the right tracking sensor deviate a black line
N
Y
Turn Left
Does the left and right tracking sensor deviate a black line
N
Y
Go Straight

Step 1: Display a “S” shape on start using the **Basic “show leds”** function.



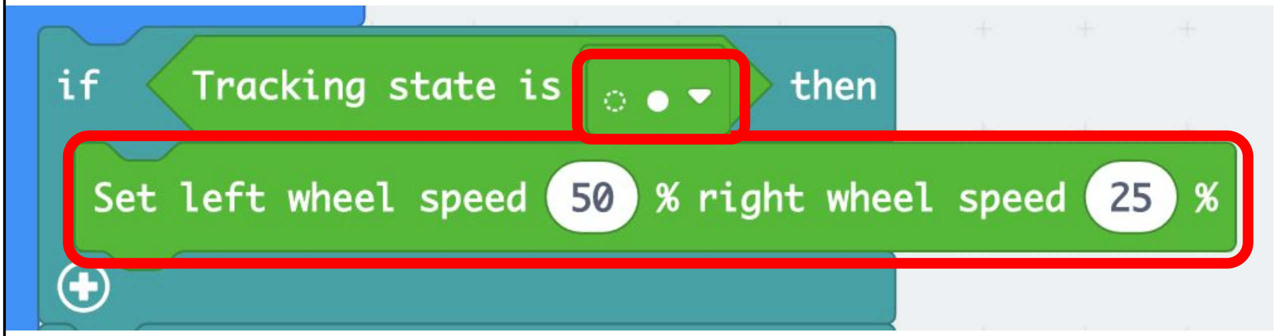
```
on start
  show leds
```

Step 2: Check the state of the tracking sensors with the blocks below. Use the labels to help locate the functions.



You can change the state of the tracking sensor that you need to track here

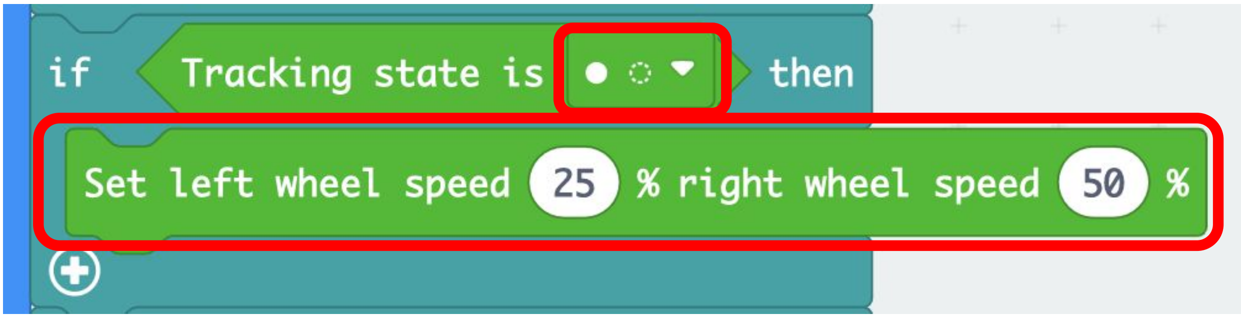
Step 3: When the **tracking sensor on the left side** detects no black line, **set the speed of the left wheel to be faster than that of the right wheel** to correct its movement.



if Tracking state is ○ ● then

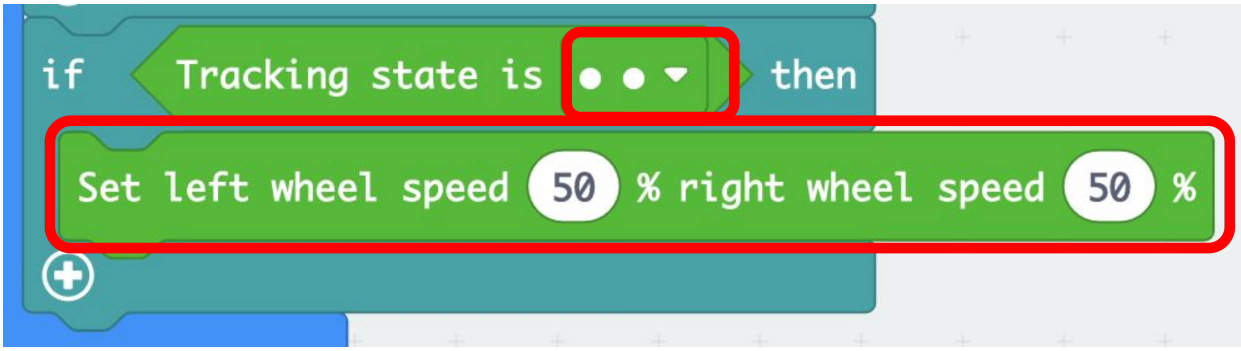
Set left wheel speed 50 % right wheel speed 25 %

Step 4: When the **tracking sensor on the right detects** no black line, the **speed of the right wheel has to be adjusted slower than that of the left** in the same way as you did in the third step.



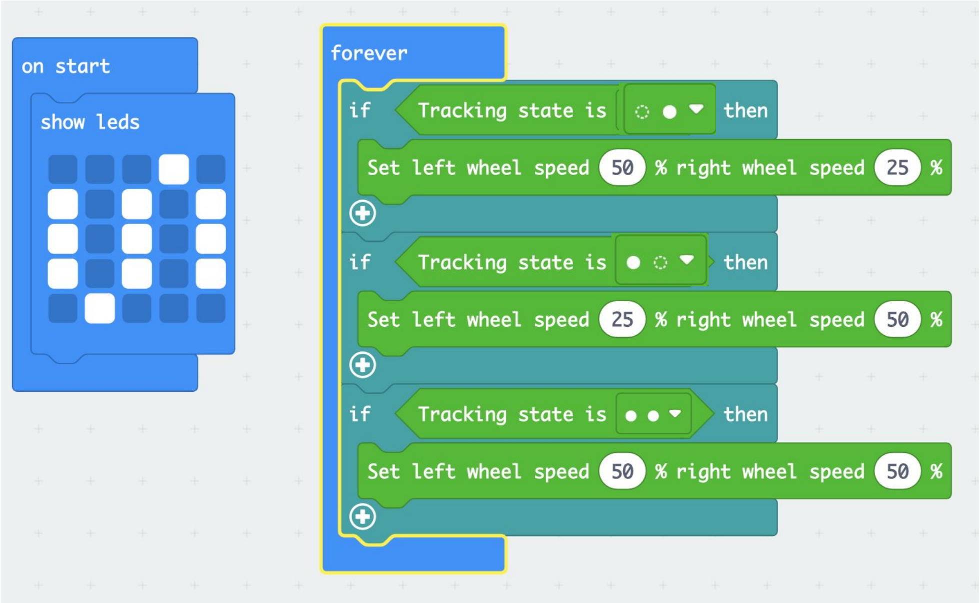
```
if Tracking state is ● ◌ then
    Set left wheel speed 25 % right wheel speed 50 %
```

Step 5: When both **tracking sensors detect black lines**, **move forward at an even speed.**



```
if Tracking state is ● ● then
    Set left wheel speed 50 % right wheel speed 50 %
```

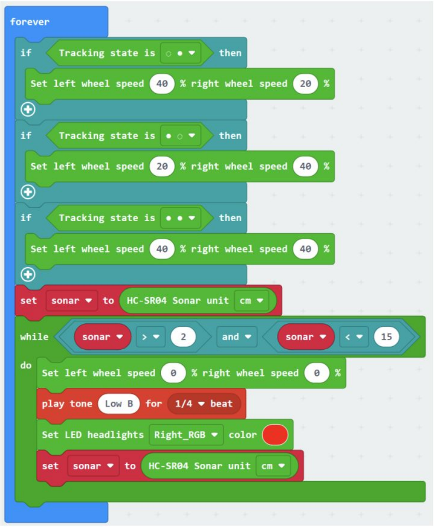
Your code should look like this.



```
on start
    show leds
        . . . # .
        # . # . #
        # . # . #
        # . # . #
        . # . . .

forever
    if Tracking state is ◌ ● then
        Set left wheel speed 50 % right wheel speed 25 %
    if Tracking state is ● ◌ then
        Set left wheel speed 25 % right wheel speed 50 %
    if Tracking state is ● ● then
        Set left wheel speed 50 % right wheel speed 50 %
```

Activity: Combine line following with collision avoidance

Combine the previous two activities to make the CuteBot follow the black line, and come to a stop and turn on a buzzer and an LED when you place an obstacle in its path. The CuteBot should resume moving along a line when you remove the obstacle.



```
forever
    if Tracking state is ◌ ● then
        Set left wheel speed 40 % right wheel speed 20 %
    if Tracking state is ● ◌ then
        Set left wheel speed 20 % right wheel speed 40 %
    if Tracking state is ● ● then
        Set left wheel speed 40 % right wheel speed 40 %
    set sonar to HC-SR04 Sonar unit cm
    while sonar > 2 and sonar < 15
    do
        Set left wheel speed 0 % right wheel speed 0 %
        play tone Low B for 1/4 beat
        Set LED headlights Right_RGB color 🟥
        set sonar to HC-SR04 Sonar unit cm
```

Questions!

1. What happens when you increase or decrease the speed of the robot?
2. Try and create a separate function that performs obstacle detection, and call the function in the main (forever) loop. (Tip: Search for function blocks in the search bar for easy access)

3.4. Open ended challenge: Combine sensor interfacing with line following and automatic collision avoidance

In this activity, you will make the CuteBot stop at every obstacle (cup of soil) on the path, and measure temperature, humidity and soil moisture at that obstacle.

Manually place the soil moisture sensor in the cup, make the bot display a Red or Green LED to indicate moisture level, remove the soil moisture sensor, and then continue moving to another obstacle – repeat.