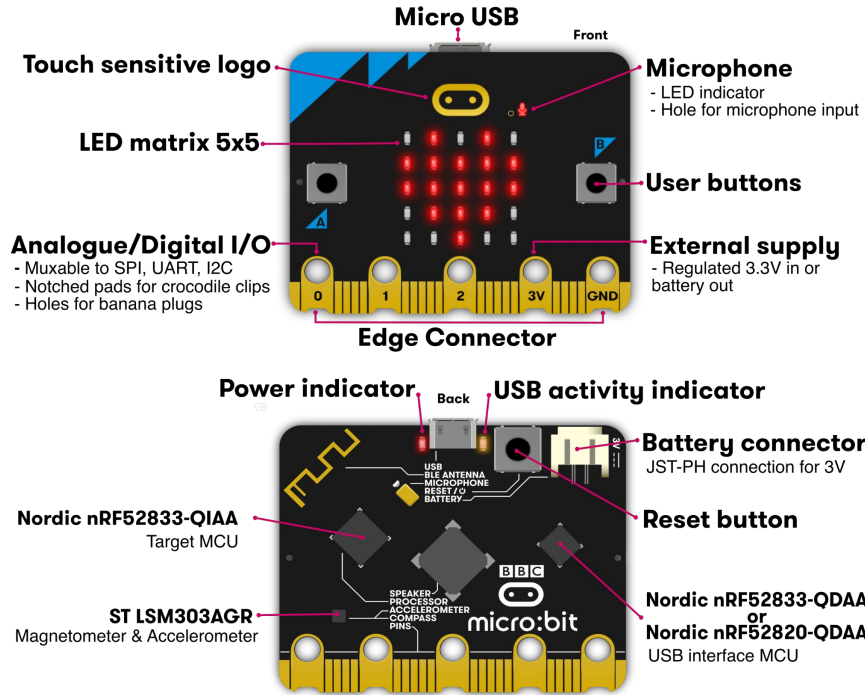
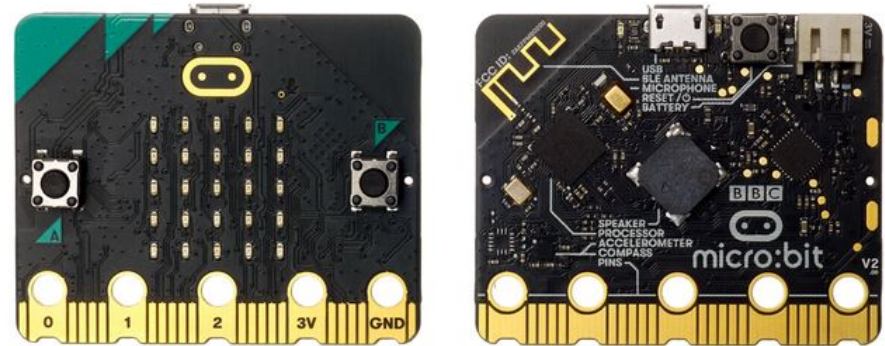
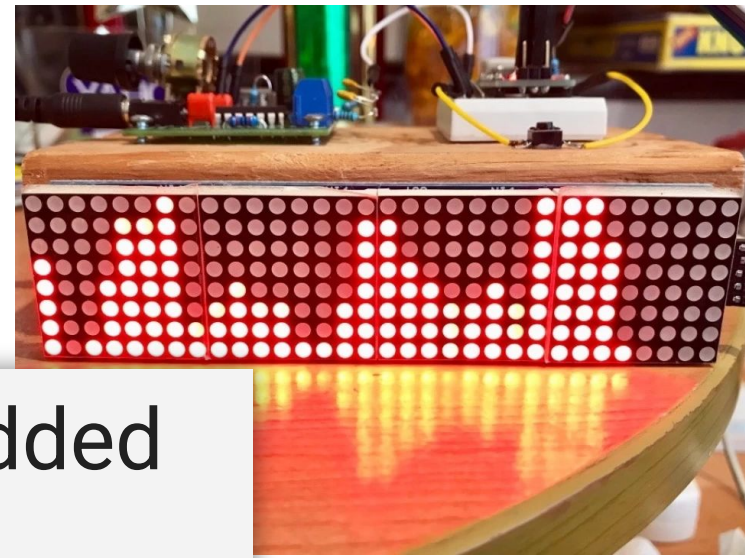
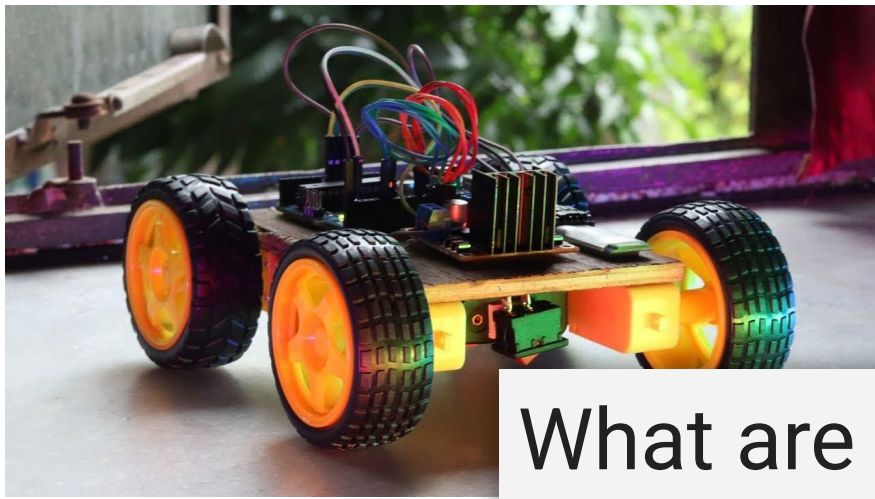




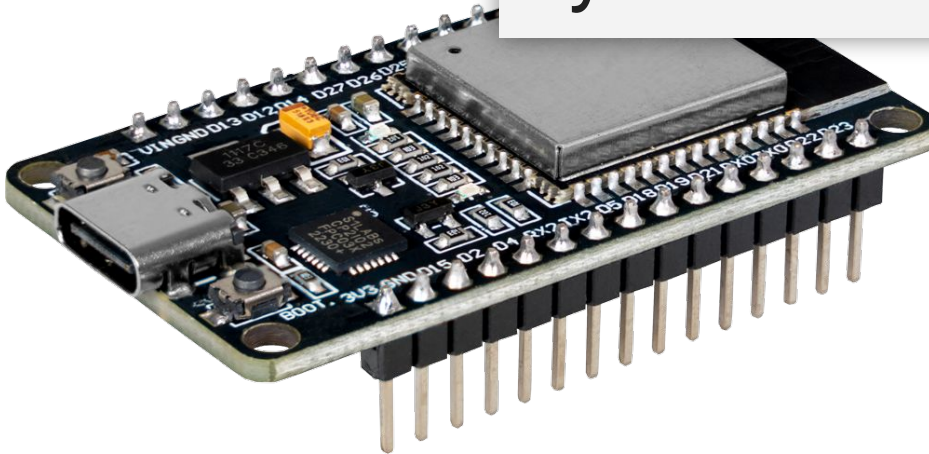
Microbit!



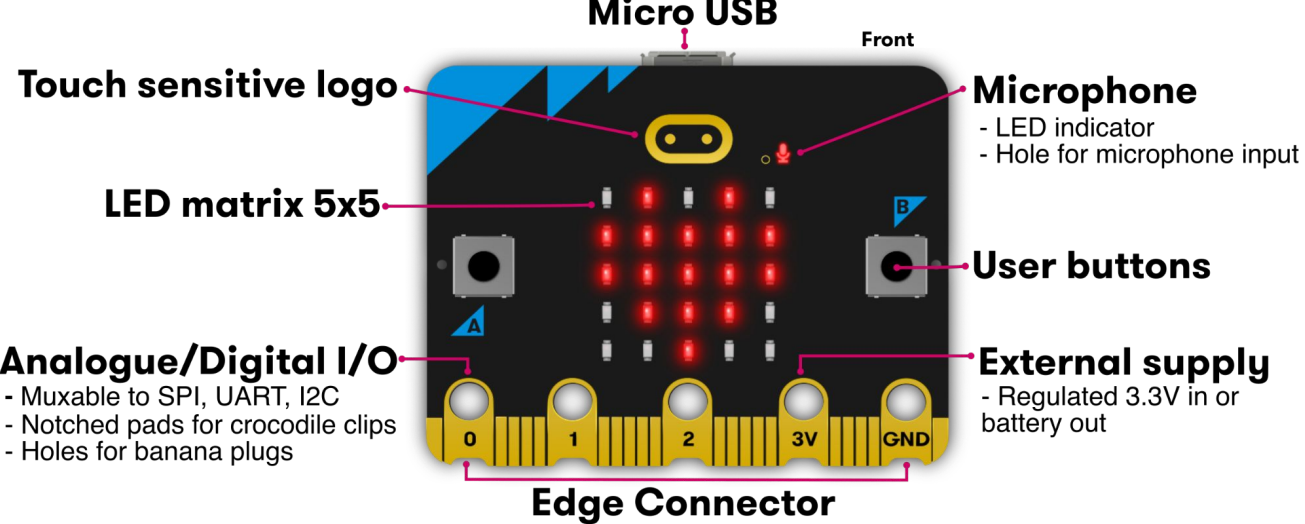
I am TAing a lab finishing at 6pm - Could exec please hand out microbits to people before I arrive?



What are Embedded Systems?



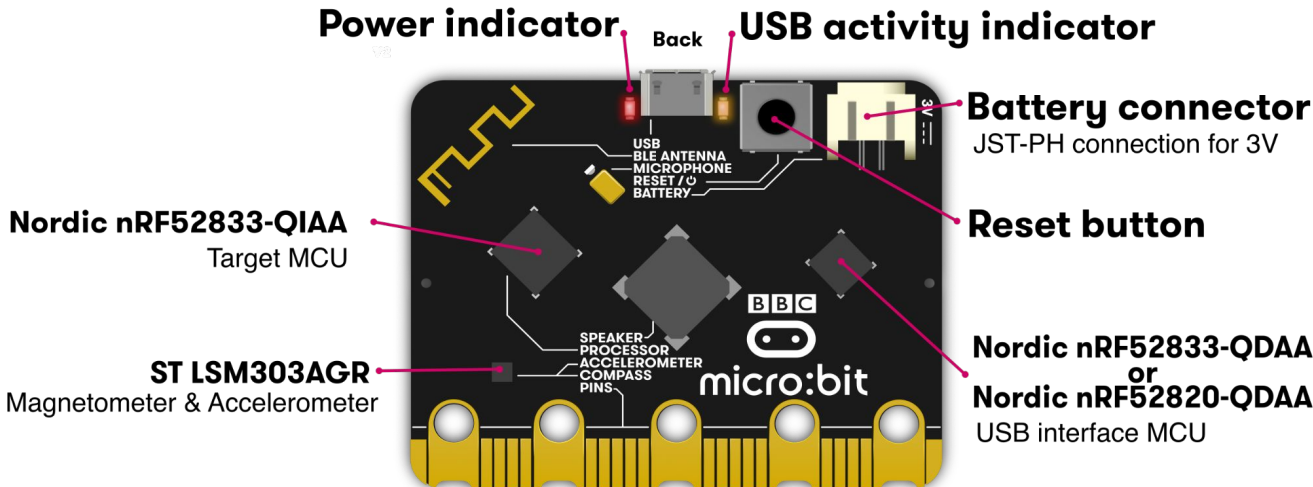
Many Products Depend on Embedded Systems

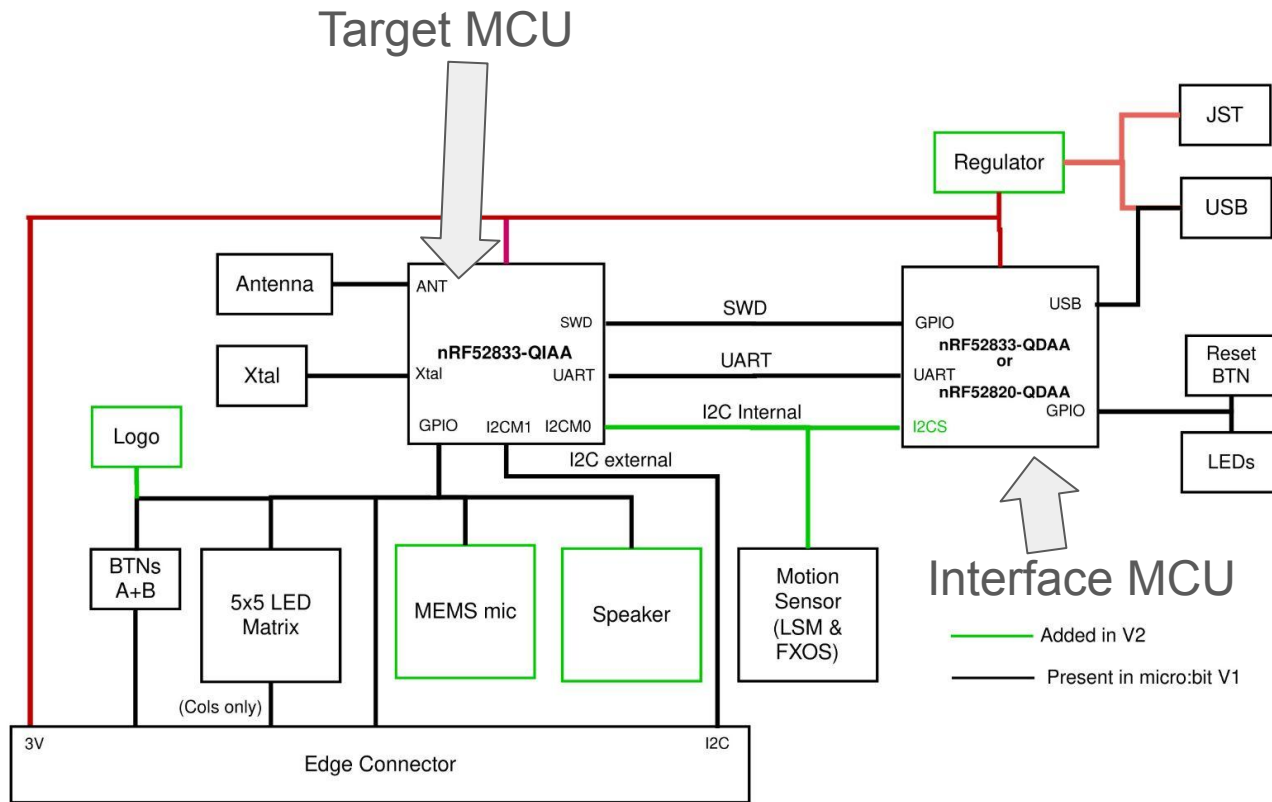


These are some of the different inputs and outputs on the microbit!

Sensors you could use:

- Accelerometer
- Magnetometer
- Light sensor
- Compass
- Microphone
- Button(s)
- Temperature





This is how the hardware of the microbit is laid out. There is a main processor (Target MCU) that runs application code and controls most IO. There is also an interface MCU that controls USB interface, and flashing the microbit.

(this is a bit simplified - Y1s will learn more later this year!)

Task menu

[Intro - Task 1 Write your name \(makecode\)](#)

[Easy - Task 2 Fizzbuzz \(makecode\)](#)

[Easy - Task 3 Compass \(makecode\)](#)

[Intermediate - Task 4 Reaction time game \(makecode\)](#)

[Intermediate - Task 5 Fizzbuzz \(C++ CODAL\)](#)

[Difficult - Task 6 Reaction time game \(C++ CODAL\)](#)

[Expert - Task 6 BLE Keyboard \(C++ CODAL\)](#)



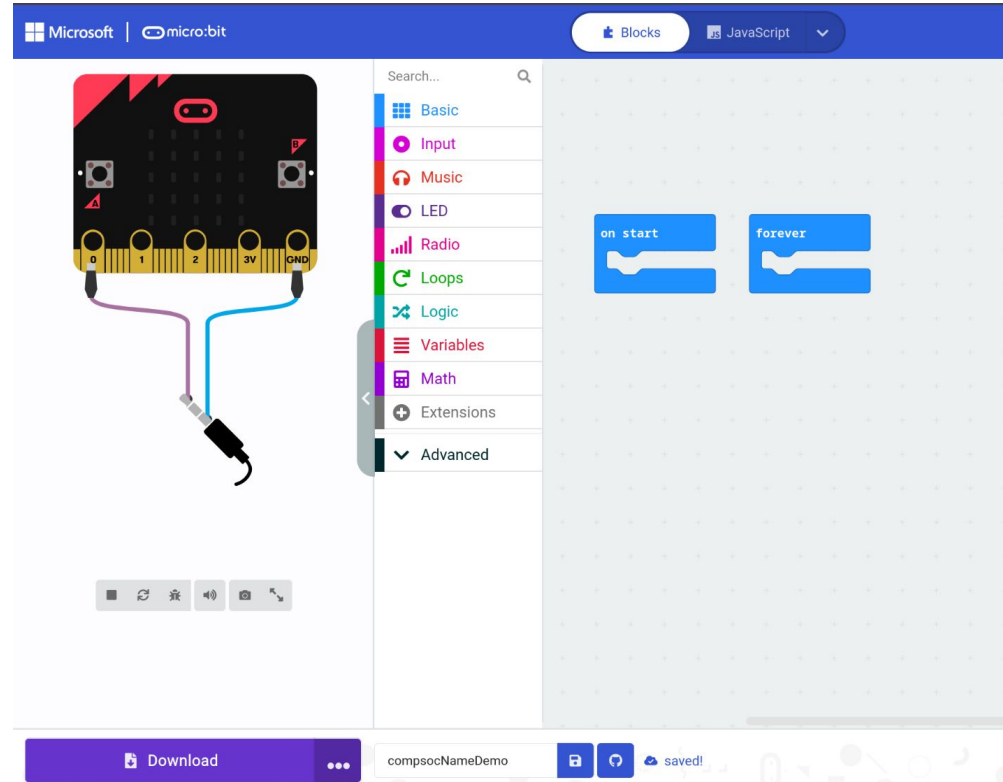
Task 1: MakeCode

Write your name!

Go to <https://makecode.microbit.org> and create a new project

The code editor is displayed! You can drag and drop blocks (similar to scratch). The on start loop runs code once when the microbit starts.

The forever loop runs code in an infinite loop (eg: main program loop)

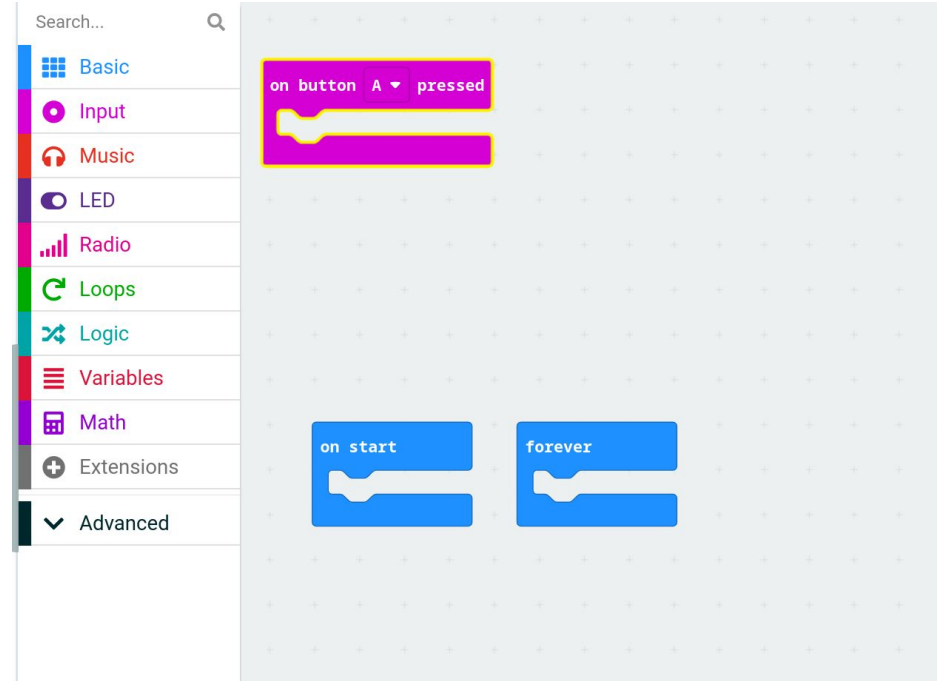


Secret answers!

Task 1: MakeCode

Write your name!

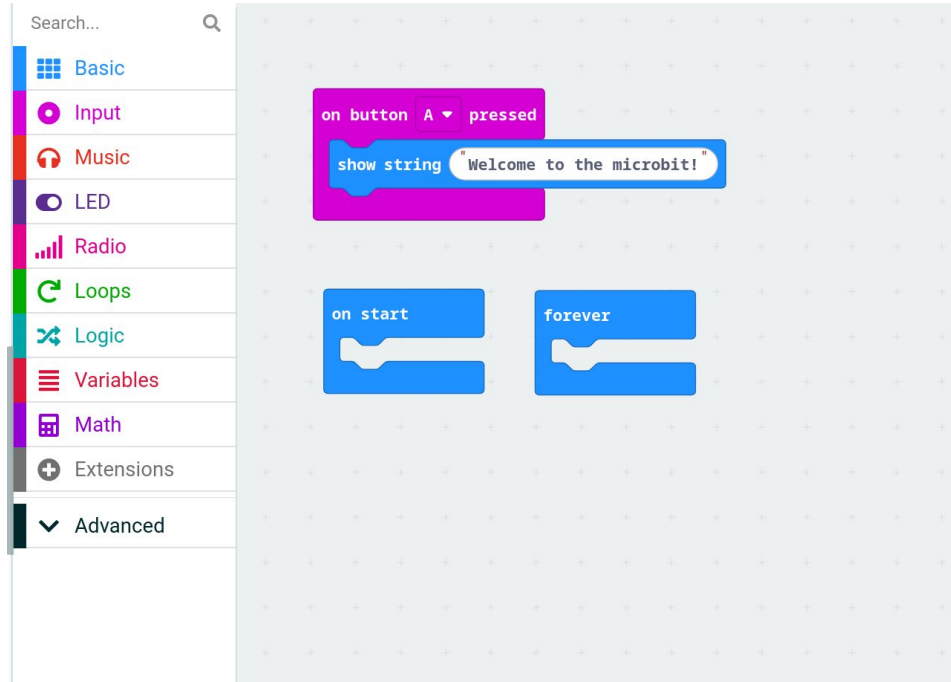
Drag over an “on button a pressed” block from the input tab.



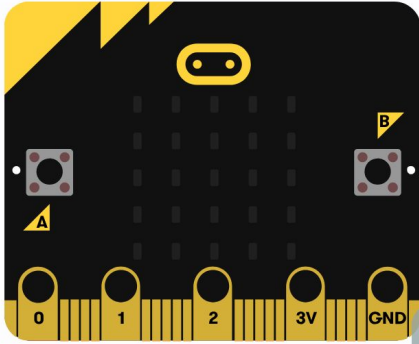
Task 1: MakeCode

Write your name!

Use a show string block from the basic tab to display your name!



Task 1: MakeCode Write your name!



Download it!

To upload the microbit plug it into your computer.

Click the download button. It will either download a hex file, that you can copy over to the MICROBIT usb device.

Alternatively when clicked it will connect via WebUSB and upload the program itself (for this use chrome)

Makecode task 2 Fizzbuzz

For task 2 we will be making a simple program based on the game “Fizz buzz”

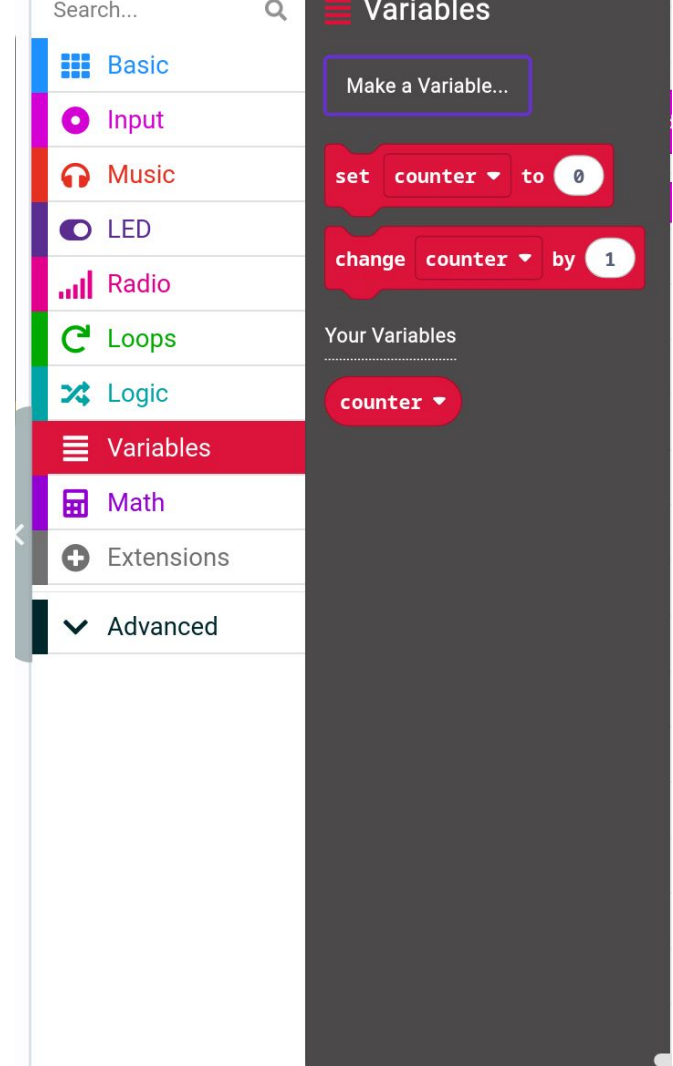
https://en.wikipedia.org/wiki/Fizz_buzz

Players take turns to count incrementally, replacing any number divisible by three with the word "fizz", and any number divisible by five with the word "buzz", and any number divisible by both three and five with the word "fizzbuzz".

1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, Fizzbuzz, 16, 17, Fizz, 19, Buzz, Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29, Fizzbuzz, 31, 32, Fizz, 34, Buzz, Fizz, 37, ...

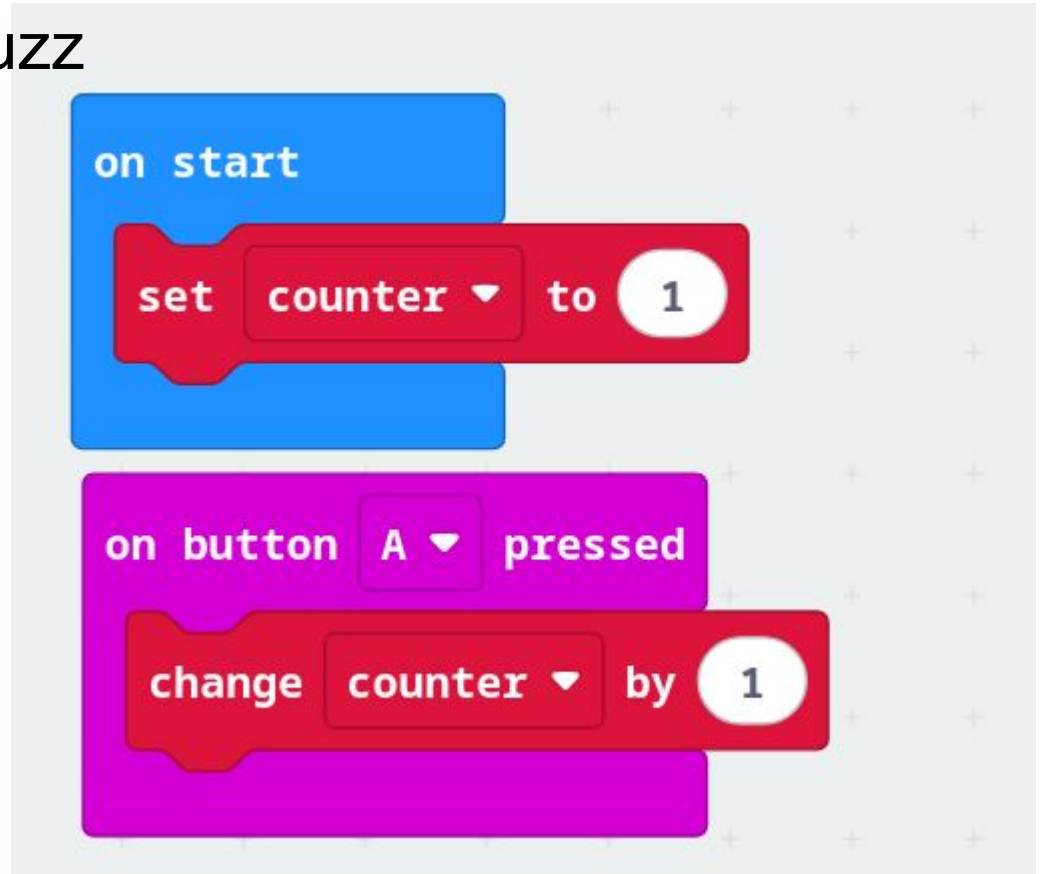
Makecode task 2 Fizzbuzz

Make a variable and call it whatever you like



Makecode task 2 Fizzbuzz

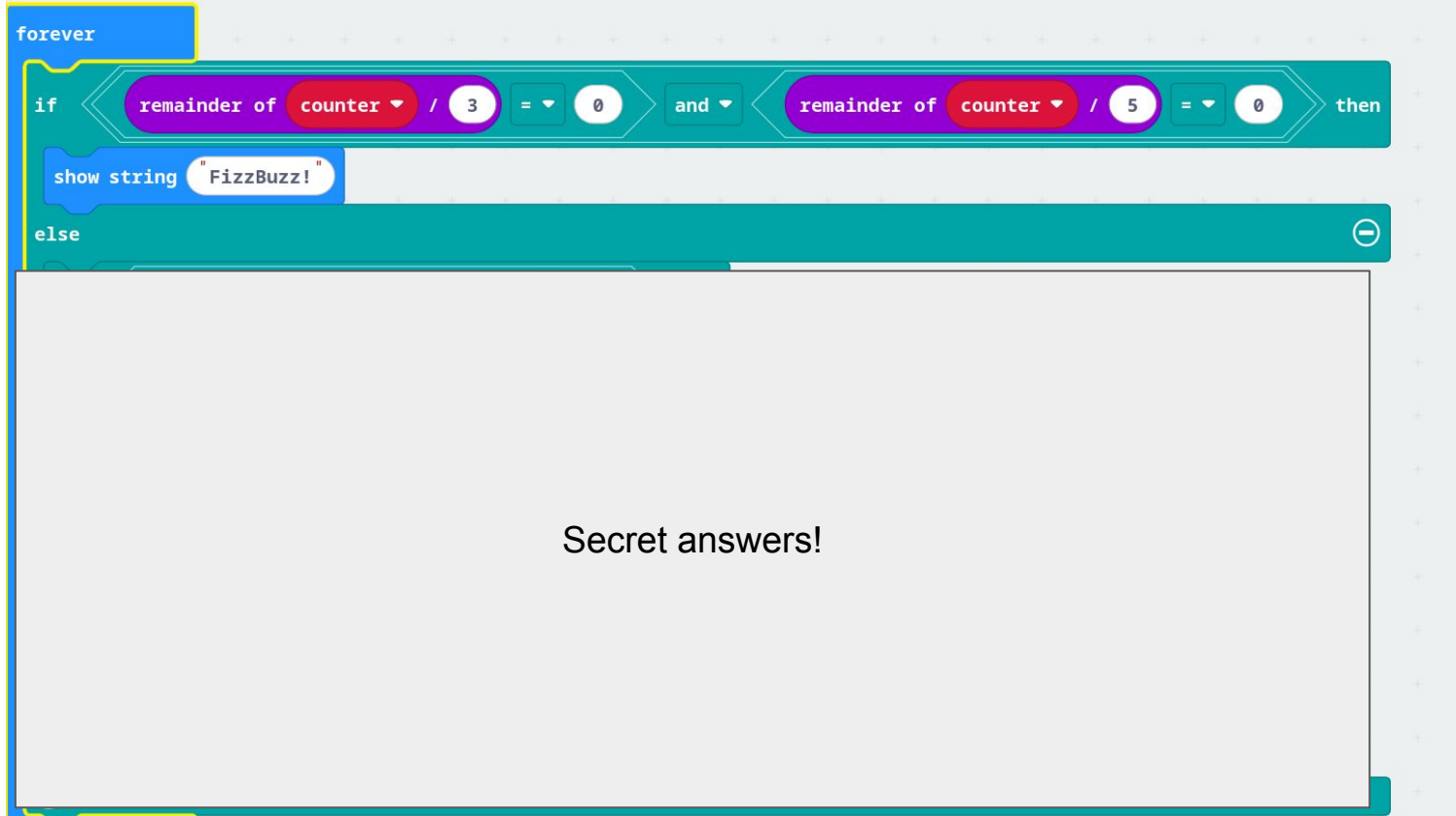
On start, you want to set your counter variable to a known value. Each time A button is pressed, increment the counter by 1.



Makecode task 2 Fizzbuzz

Use an if else statement, and blocks from the logic and math tabs to create the main program logic of fizzbuzz in the forever loop.

We've given you the first one, try to complete the rest of the conditionals on your own!



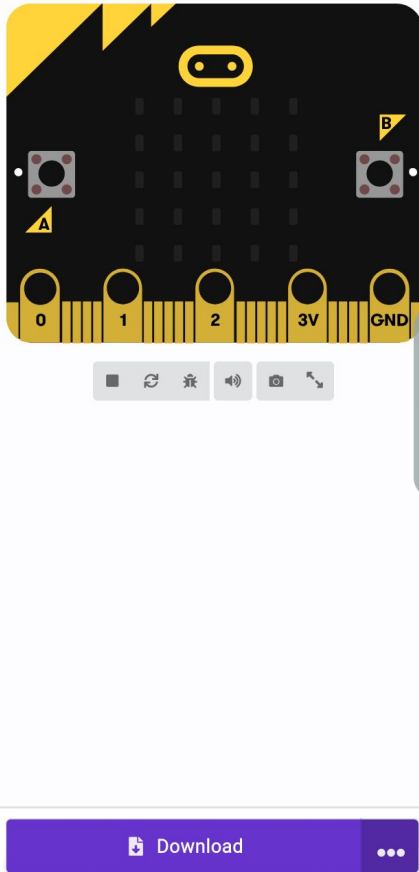
Makecode task 2 Fizzbuzz

Download it!

To upload the microbit plug it into your computer.

Click the download button. It will either download a hex file, that you can copy over to the MICROBIT usb device.

Alternatively when clicked it will connect via WebUSB and upload the program itself (for this use chrome)

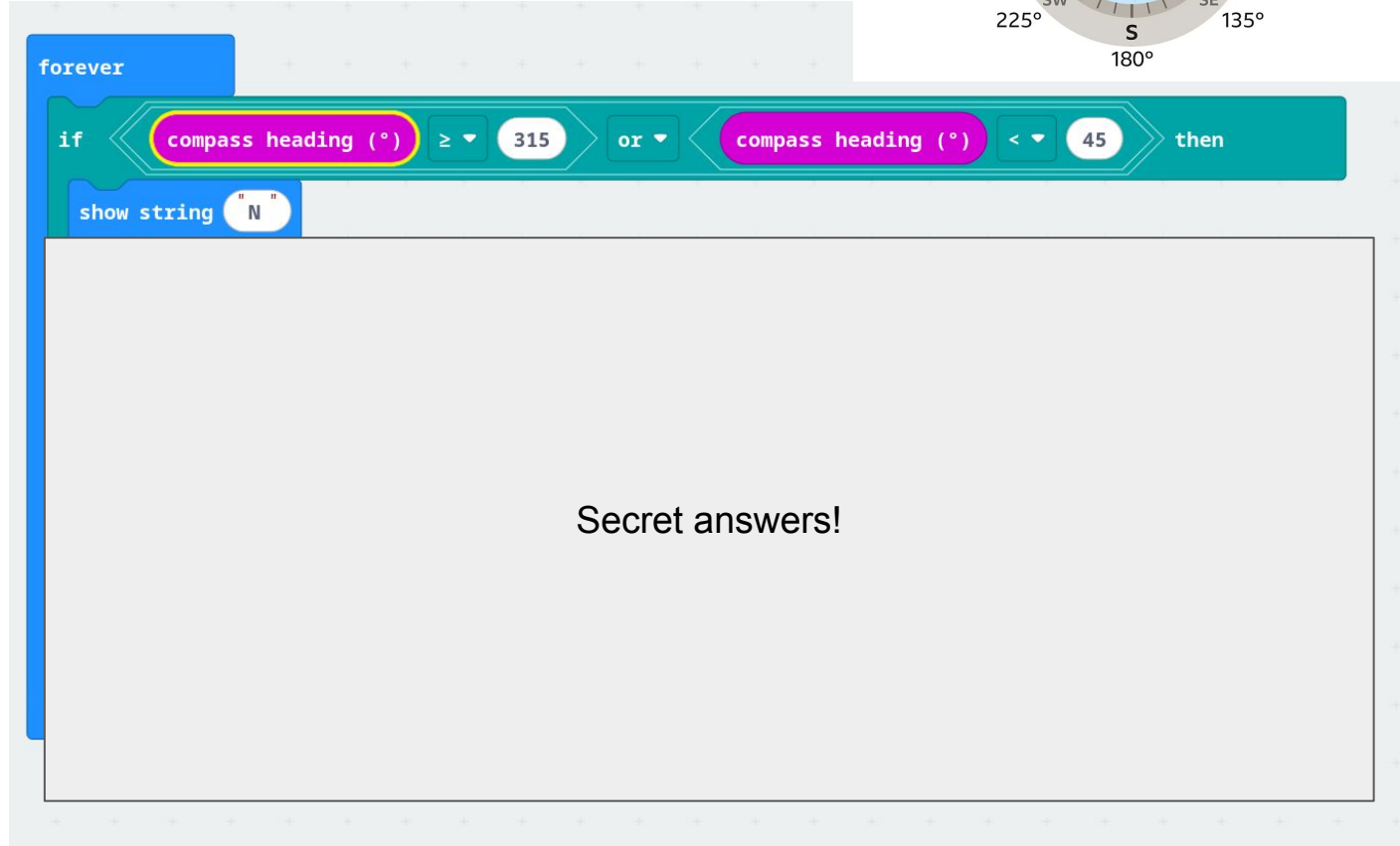
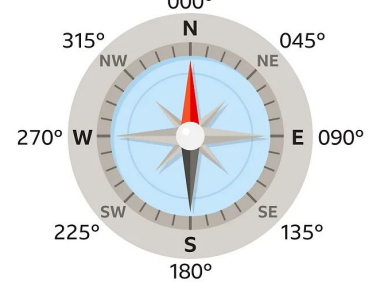


Secret answers!

Makecode task 3 - Compass

Use the compass heading block from the input tab, alongside logic blocks.

Create the rest of the if/else statements following the first example - you can see example compass headings in the top right image



Secret answers!

Makecode task 3 - Compass

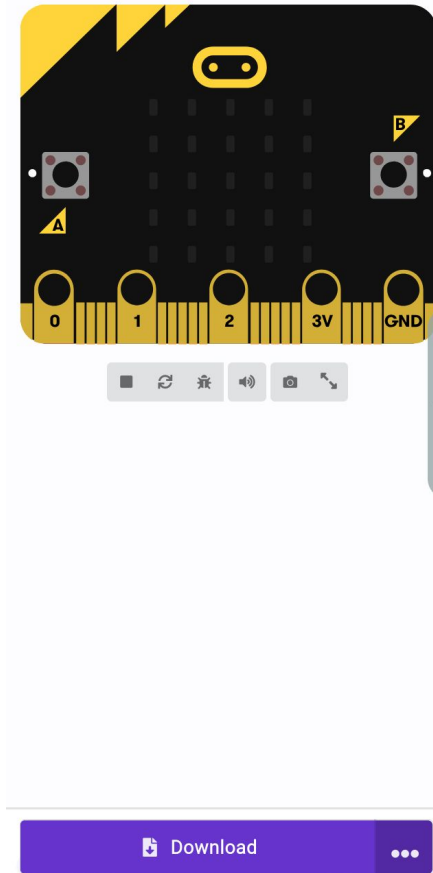
Download it!

For the compass task you will need to tilt the microbit to fill in all the lights before the program will run - follow the instructions on the microbit.

To upload the microbit plug it into your computer.

Click the download button. It will either download a hex file, that you can copy over to the MICROBIT usb device.

Alternatively when clicked it will connect via WebUSB and upload the program itself (for this use chrome)



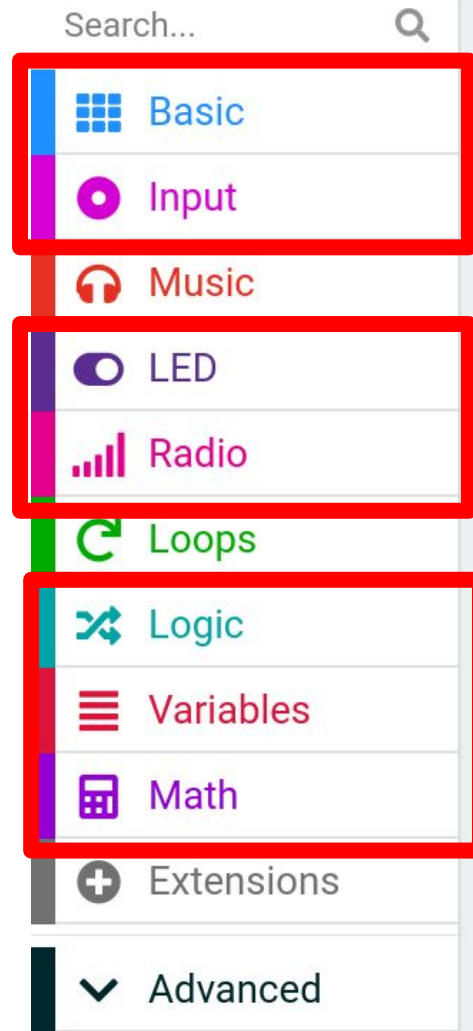
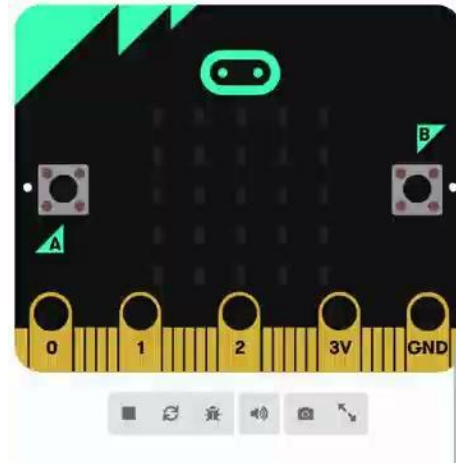
Secret answers!

Makecode task 4 - Reaction time game!

Using the blocks we've been using, design a reaction time game!

I've circled the tabs we used some of the blocks from when we made our version but feel free to get creative.

Video of how the game should work ->



CODAL task 5 - Fizzbuzz

Clone this repo <https://github.com/lancaster-university/microbit-v2-samples>

Navigate into the cloned repo folder and run `python3 build.py` (lab PCs have dependencies installed already)

This will generate a MICROBIT.hex file in the repo, from the main.cpp file in the sources directory. Drag the hex file over to your microbit usb and then run it. The microbit demo file will run!

Docs for the microbit library <https://lancaster-university.github.io/microbit-docs/ubit/>

CODAL task 5 - Fizzbuzz

First we have to check if the button is pressed.

We can use `uBit.buttonA.isPressed()` inside an if statement, but this is not ideal as it may not register properly if the button is held or if the screen is currently printing.

To solve this we use event listeners.

```
uBit.messageBus.listen(MICROBIT_ID_BUTTON_A,  
MICROBIT_BUTTON_EVT_CLICK, onButtonA);
```

Event of interest

Event handler

ID of component we
want to listen to

CODAL task 5 - Fizzbuzz

Can you complete the program? Fill in the if statement, and add the remaining if statements within the event handler.

There's a template to get you started at

<https://github.com/LUCompSoc/microbitExamples2026> (try not to look at the answers!)

Once you're done, edit main.cpp in the repo and build the hex file like you did before!

CODAL task 6 - Reaction time game!

Using what you've learnt when writing fizzbuzz, make a simple reaction time game - ie user presses button to start, waits a random amount of time, screen lights up, and the time from the screen lighting up to button press is measured. Reference the microbit documentation, below are some starting points.

<https://lancaster-university.github.io/microbit-docs/ubit/display/#ubitdisplay>

<https://lancaster-university.github.io/microbit-docs/ubit/button/>

<https://lancaster-university.github.io/microbit-docs/ubit/#sleep>

<https://lancaster-university.github.io/microbit-docs/ubit/#systemtime>

Bonus 7 - CODAL BLE Keyboard task (for year 2+)

This task is the most tricky!

<https://github.com/bsiever/microbit-pxt-blehid>

Referencing this repo, can you figure out a way to send strings using the microbit as a BLE keyboard? You can use different actions ie: shake, button presses, noise level, temperature, light to trigger different phrases to be sent.

This is a tough one!