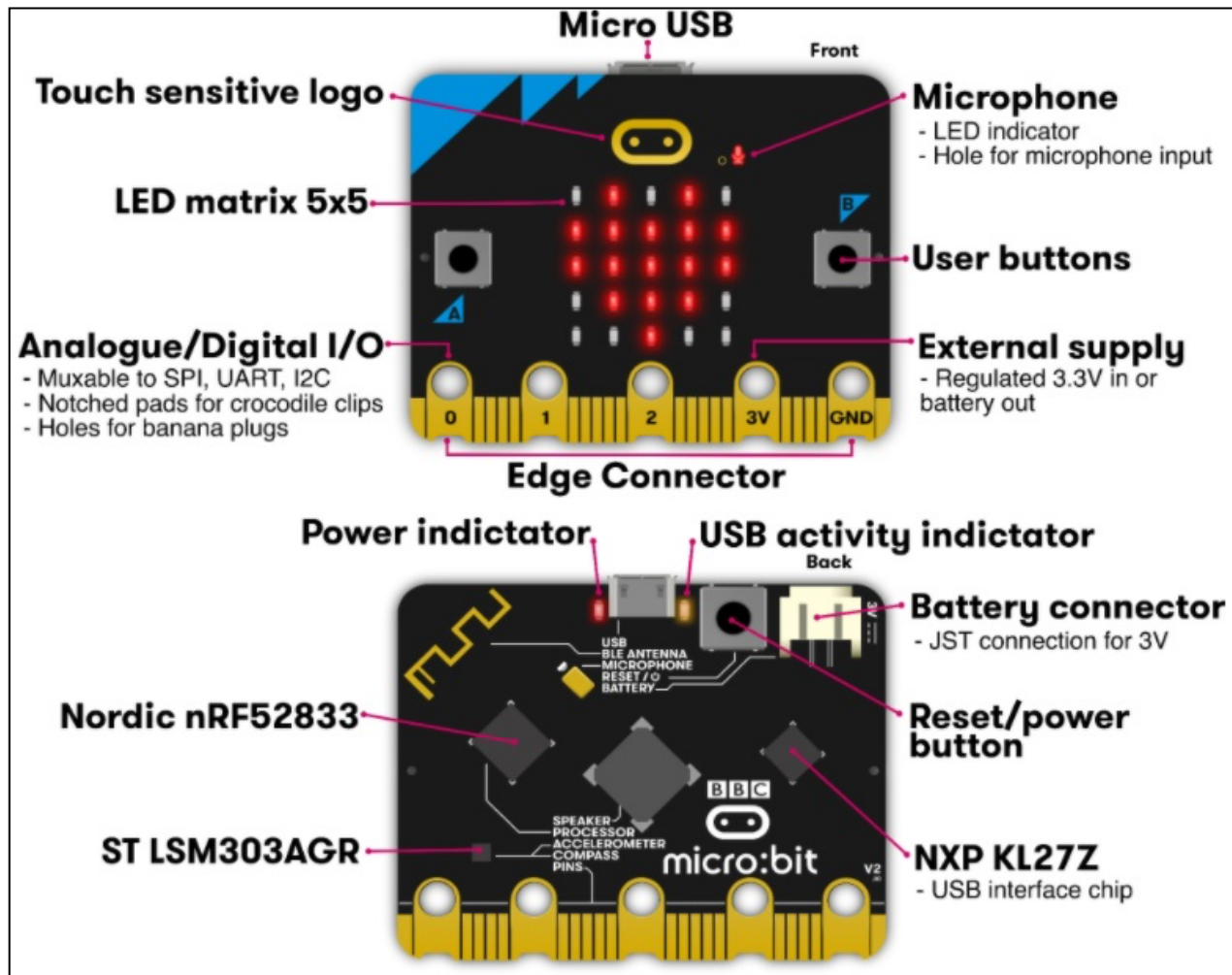
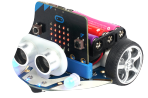


GO GO ROBOT


BBC MicroBit



Bluetooth Pairing




To enter **Pairing Mode** on the micro:bit you'll need to


- hold down the **A**, **B** and **reset** buttons simultaneously
- **release the reset button**. The LED screen will fill and then show the Bluetooth logo , followed by a pairing pattern or number
- Copy the pattern into the app or follow the instructions on your mobile device to pair.

Blocks language

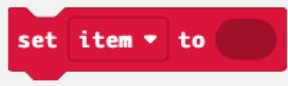
Blocks snap into each other to define the program that your micro:bit will run. Blocks can be event (buttons, shake, ...) or need to be snapped into an event to run. The **on-start** event runs first.




Loops
Loops and repetition.



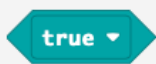
Logic
Logic operators and constants.



Variables
Variables.

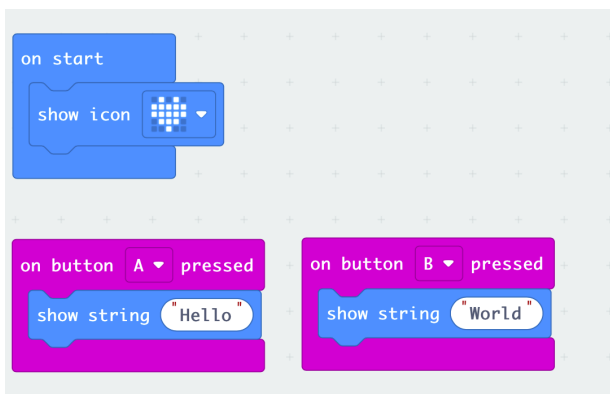


if
Conditional statement.



Boolean
True or false values.

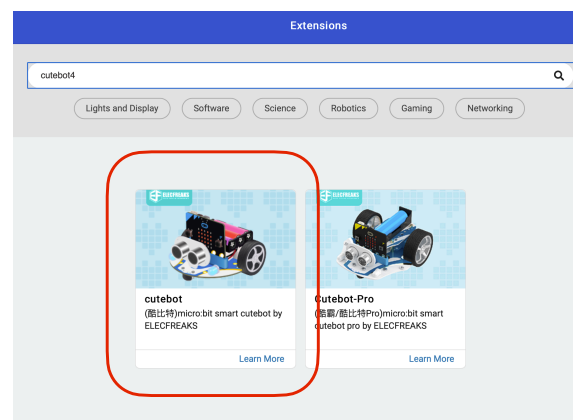
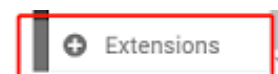
Our First Program



Best Practices

- Put a unique "Show icon" for each program in the on start of the program you are writing so you know it has downloaded correctly (explore other options for displaying on the LEDs as well).

Adding Cutebot Extension



Dance in Figure Eight

- Drag “go straight at full speed” and “set left wheel speed, right wheel speed” bricks into the Forever brick in turns.
- Divide the “figure-of eight” track into six parts: move forward for 200ms at the beginning, set the speed of the left wheel is faster than the right and set to move for 1000ms after, then go straight for 200ms. Right now you have completed the half part of “figure-of-eight”.
- Complete the second half part of “figure-of-eight” in a similar way.

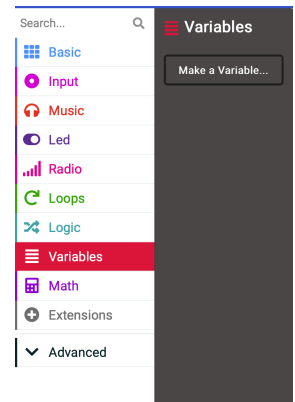
```

forever
  go straight at full speed
  pause (ms) 200
  set left wheel speed 100 right wheel speed 40
  pause (ms) 1000
  go straight at full speed
  pause (ms) 200
  go straight at full speed
  pause (ms) 200
  set left wheel speed 40 right wheel speed 100
  pause (ms) 1000
  go straight at full speed
  pause (ms) 200

```

Object Avoidance

- Set a **Sonar** variable to save the detected **Cm** value in the **Forever** brick.
- If the detected value is between **2** and **20** which means there is obstacle being detected in the front 20cm far, set the left wheel speed to **0** and right to **-50**, make a right turn at a random time to complete an obstacle avoidance.
- If not, move forward at its full speed.



```

on start
  Go straight at full speed

forever
  set sonar to HC-SR04 Sonar unit cm
  if <sonar < 20 and sonar > 2> then
    Set left wheel speed 0 % right wheel speed -50 %
    pause (ms) pick random 100 to 200
  else
    Go straight at full speed

```

Line Tracking

- Drag three **if** bricks into the **Forever** brick.
- If the status of line-tracking sensors is **○ ●**, that indicates the left probe doesn't detect the black line while the right probe detects the black line.
- Set the left wheel speed to **50** and right to **25**, make a right turn by the different speed of the two wheels and go back to the black line.
- If the status of line-tracking sensors is **● ○** Make a left turn to go back to the black line.
- When the status is **● ●** that means the **Cutebot** runs along with the black line at the speed of **50**.

```

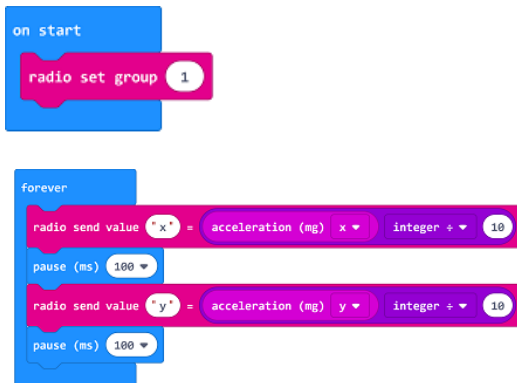
forever
  if Tracking state is ○ ● then
    Set left wheel speed 50 % right wheel speed 0 %
  if Tracking state is ● ○ then
    Set left wheel speed 0 % right wheel speed 50 %
  if Tracking state is ● ● then
    Set left wheel speed 50 % right wheel speed 50 %

```

Remote Control

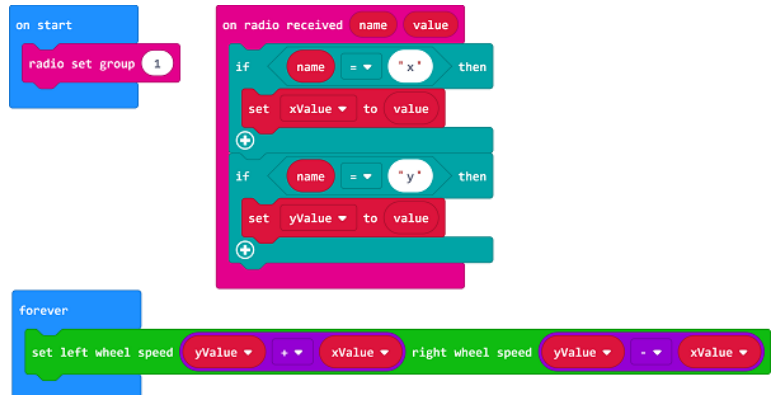
Sending

- Set “radio set group” to 1 in the **On start** brick.
- Set **x** whose value is given by “acceleration (mg) x” exactly divides 10 to the radio value in **forever** brick.
- Set **y** whose value is given by “acceleration (mg) y” exactly divides 10 to the radio value in **forever** brick.
- The scope of the acceleration value is 0~1024, which can be regarded roughly as the speed value in 0~100 after dividing 10.



Receiving

- Set the “radio set group” to 1 in the **On start** brick. Items must be the same with the remote control for the correct match.
- Drag two “if” bricks into the **on radio received** brick and judge if the radio received value **name** is **x** or **y**.
- If the radio received value **name** is **x**, it is the data for **X** and then save the **value** in the variable **xValue**.
- If the radio received value **name** is **y**, it is the data for **y** and then save the **value** in the variable **yValue**.
- In **forever** brick, set the left wheel speed to **yValue+xValue** and right wheel speed to **yValue-xValue**.



Sending and receiving code can be in the same program.

Challenges

More information

Visit ProfC's Github for this document and a link to more resources for Micro-bit and Open Source hardware.

