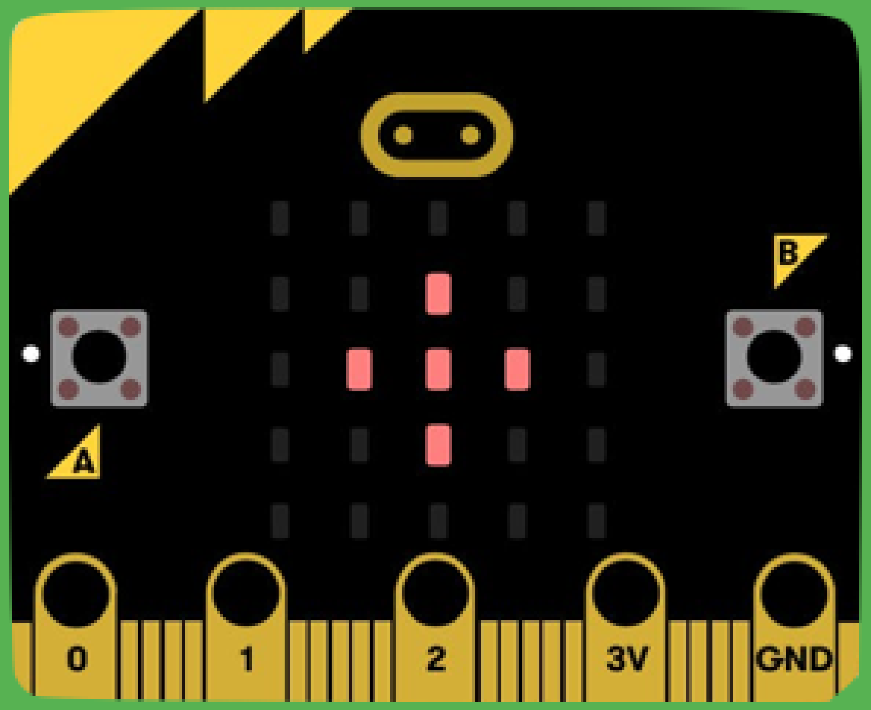
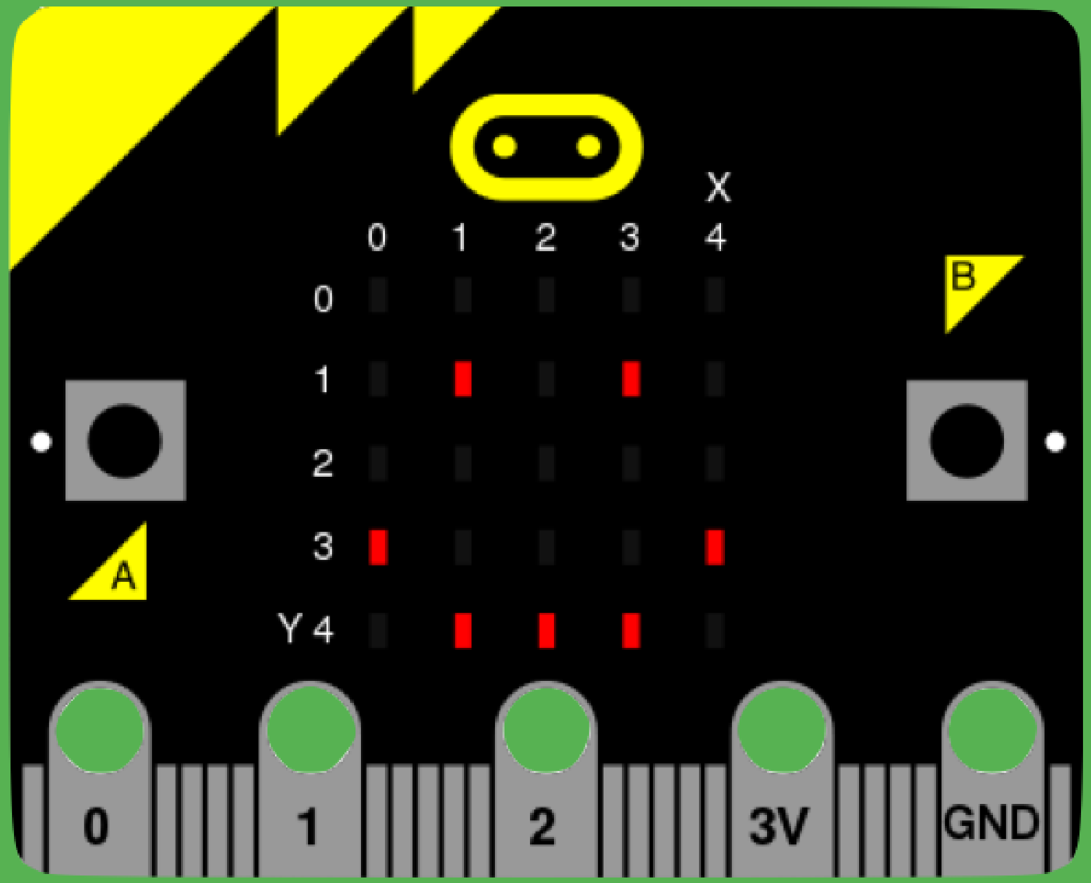


Micro:Bit 101

Explore with Micro:Bit - Tutorial for Micro:Bit V1

@CoraEdTech



What is Micro:Bit?

The micro:bit is a tiny computer that makes coding tangible and promotes digital creativity.

You can program instructions to tell it what to do by using the web-based editor MakeCode, and download codes as .hex files - a version of your codes that your micro:bit can read.

Micro:Bit V1 Overview

Getting To Know Micro:Bit V1

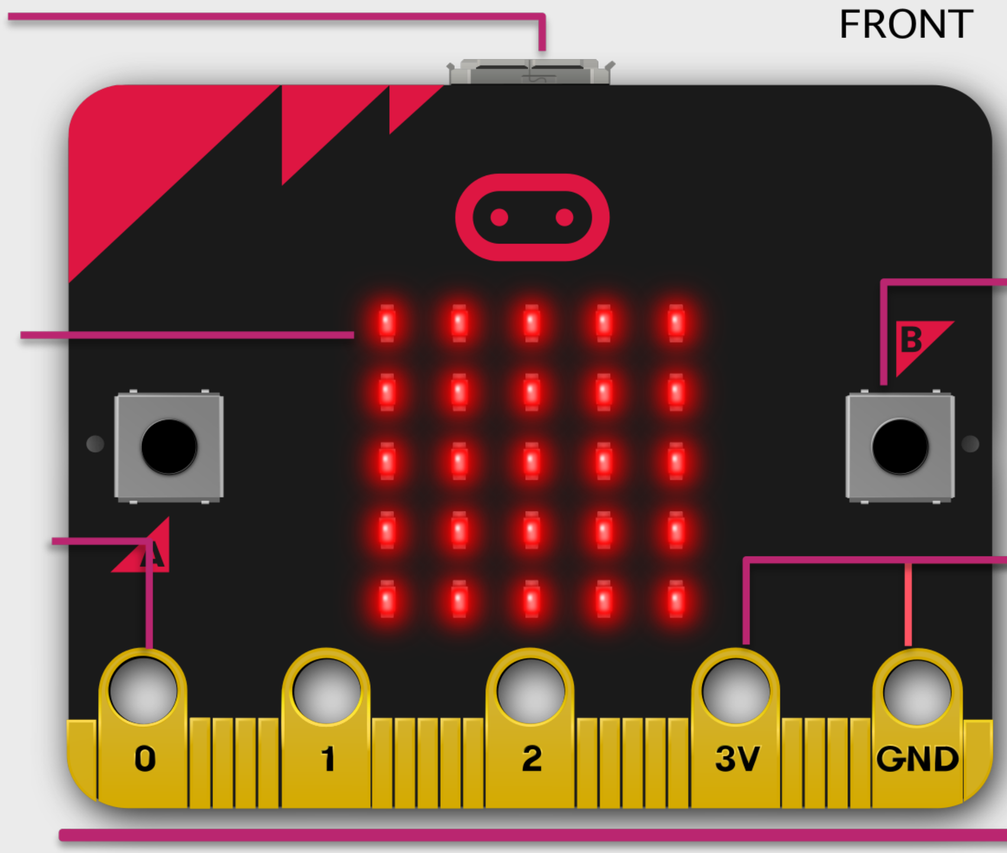
Micro USB
Drag-and-drop programming
MSC, UART, CMSIS-DAP, webUSB

5x5 LED Matrix

Digital/analog IO
Muxable to SPI, UART, I2C

Pads for crocodile clips

Holes for banana plugs



User buttons

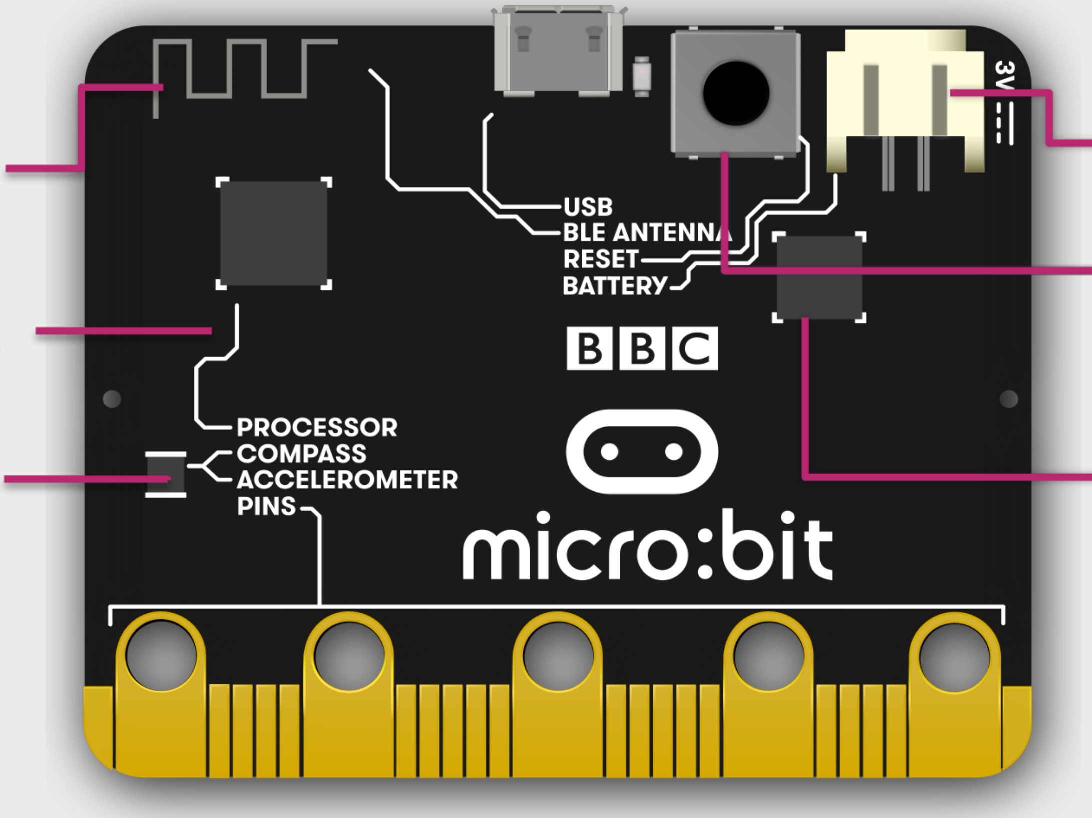
External supply
Regulated 3.3V in or battery out

Edge Connector

2.4GHZ Antenna
Bluetooth low energy
Broadcast radio

Nordic nRF51822

Motion Sensor
ST LSM303AGR



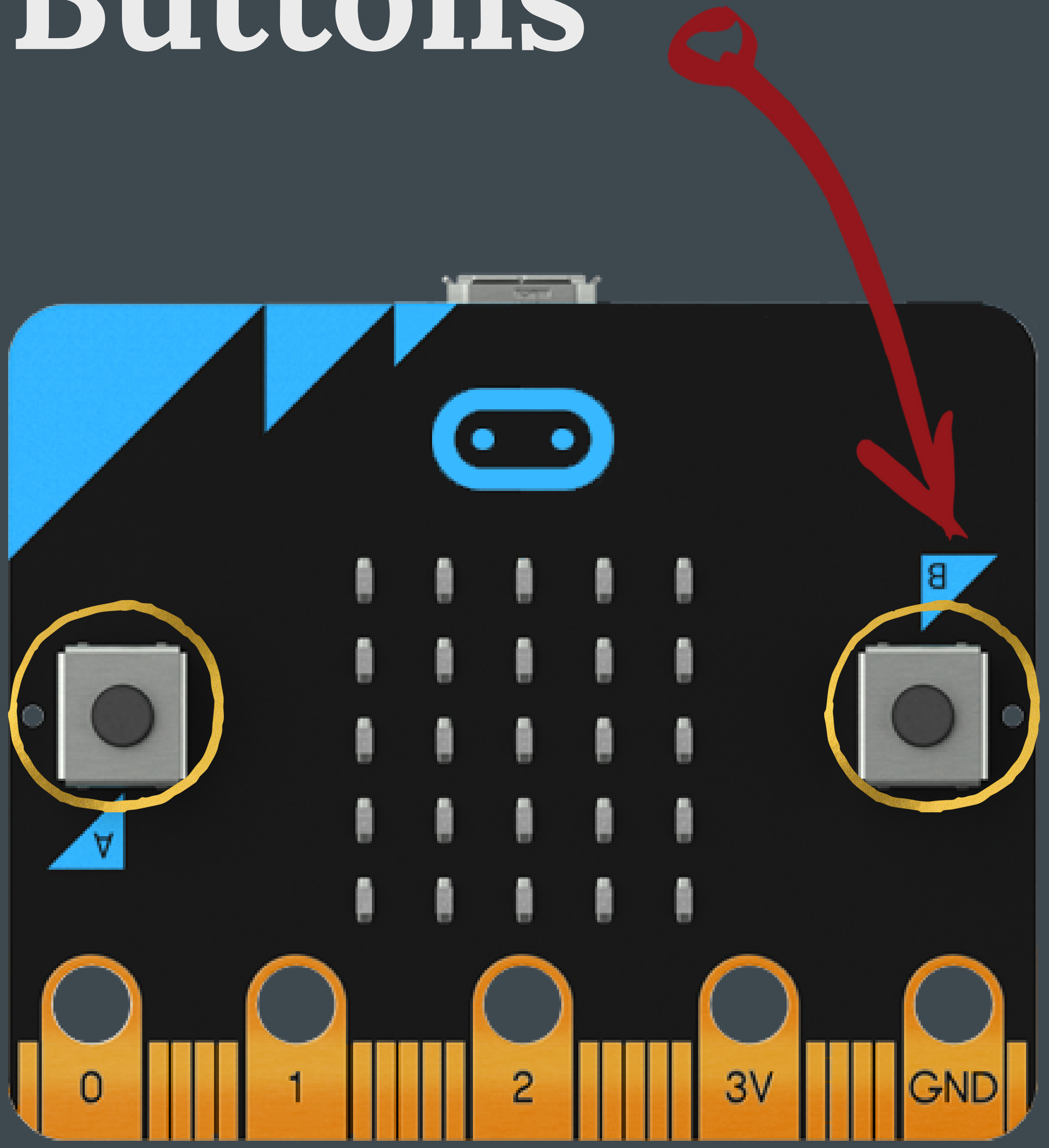
Battery connector
JST connection for 3V

Reset Button

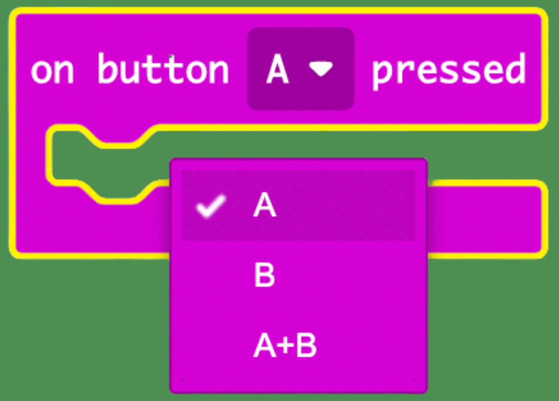
NXP KL26Z
USB Interface chip

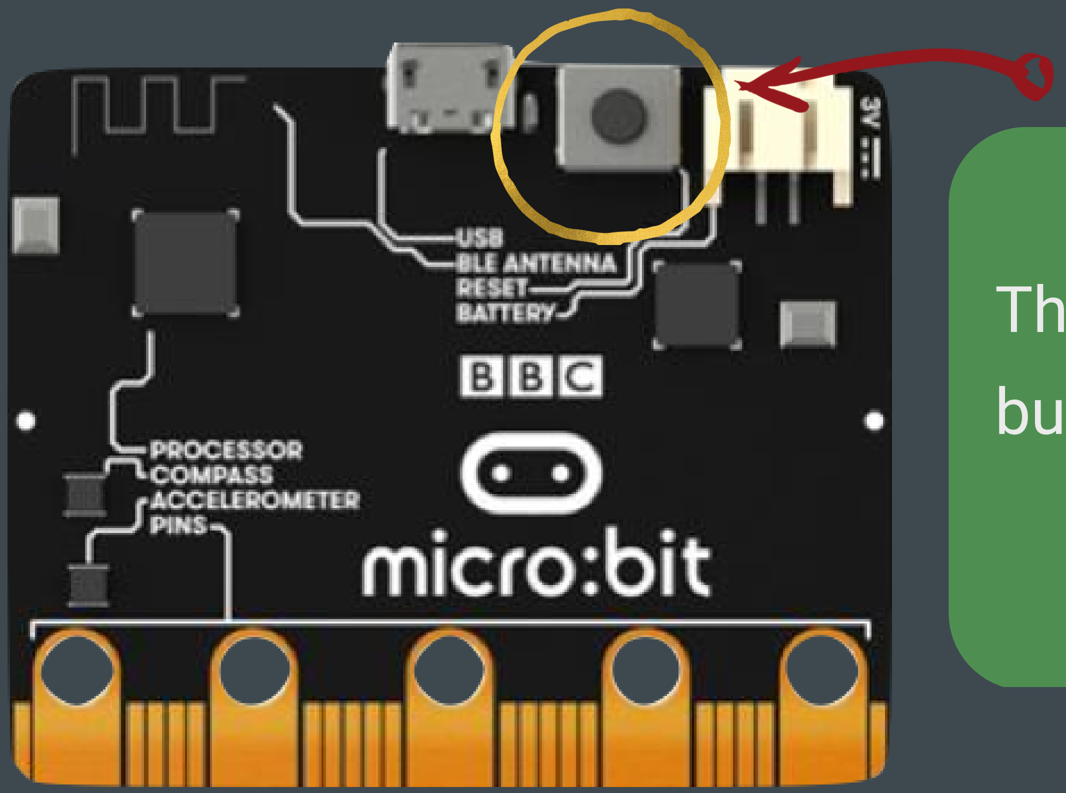
@CoraEdTech

Buttons



Front buttons A and B are sensors on Micro:Bit. Like other triggers, your code can start with pressing buttons.

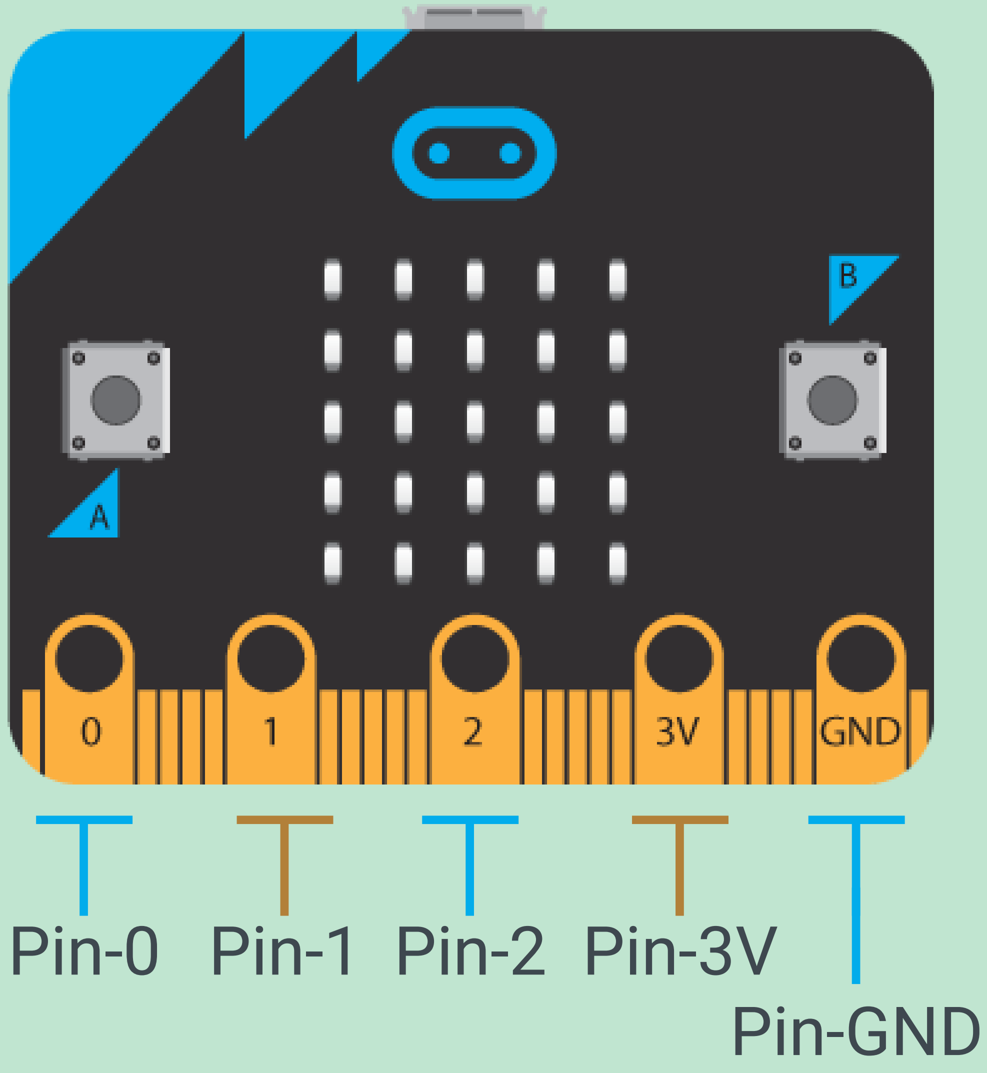

```
on button A ▾ pressed
```



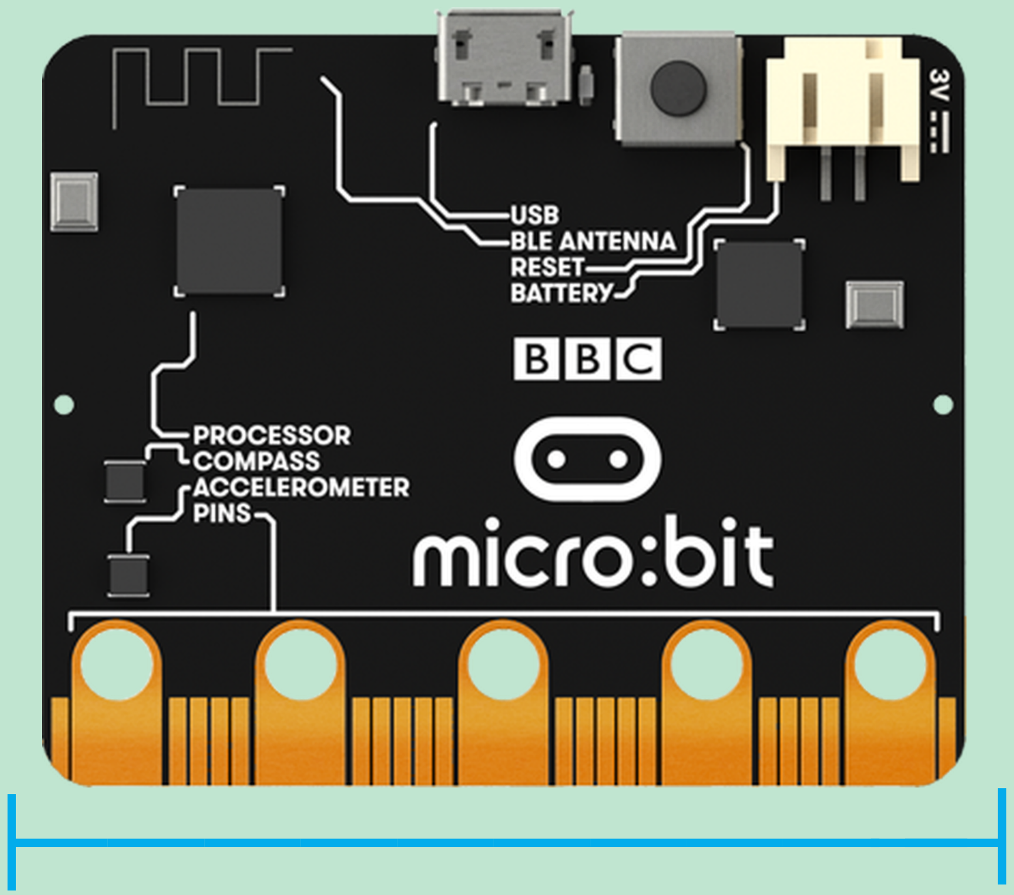
The micro:bit has a reset/power button located on the back.

Getting To Know Micro:Bit V1

Pins



On the bottom edge of your BBC micro:bit there are 25 gold strips, called pins. These pins allow you to really get creative. You can create circuits, connect external things like buzzers and motors and make your own fun projects.



Edge connector for accessories

@CoraEdTech

Getting To Know Micro:Bit V1

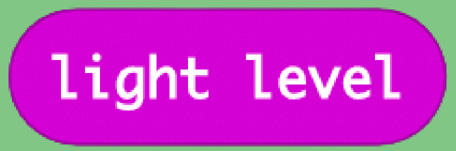
LEDs

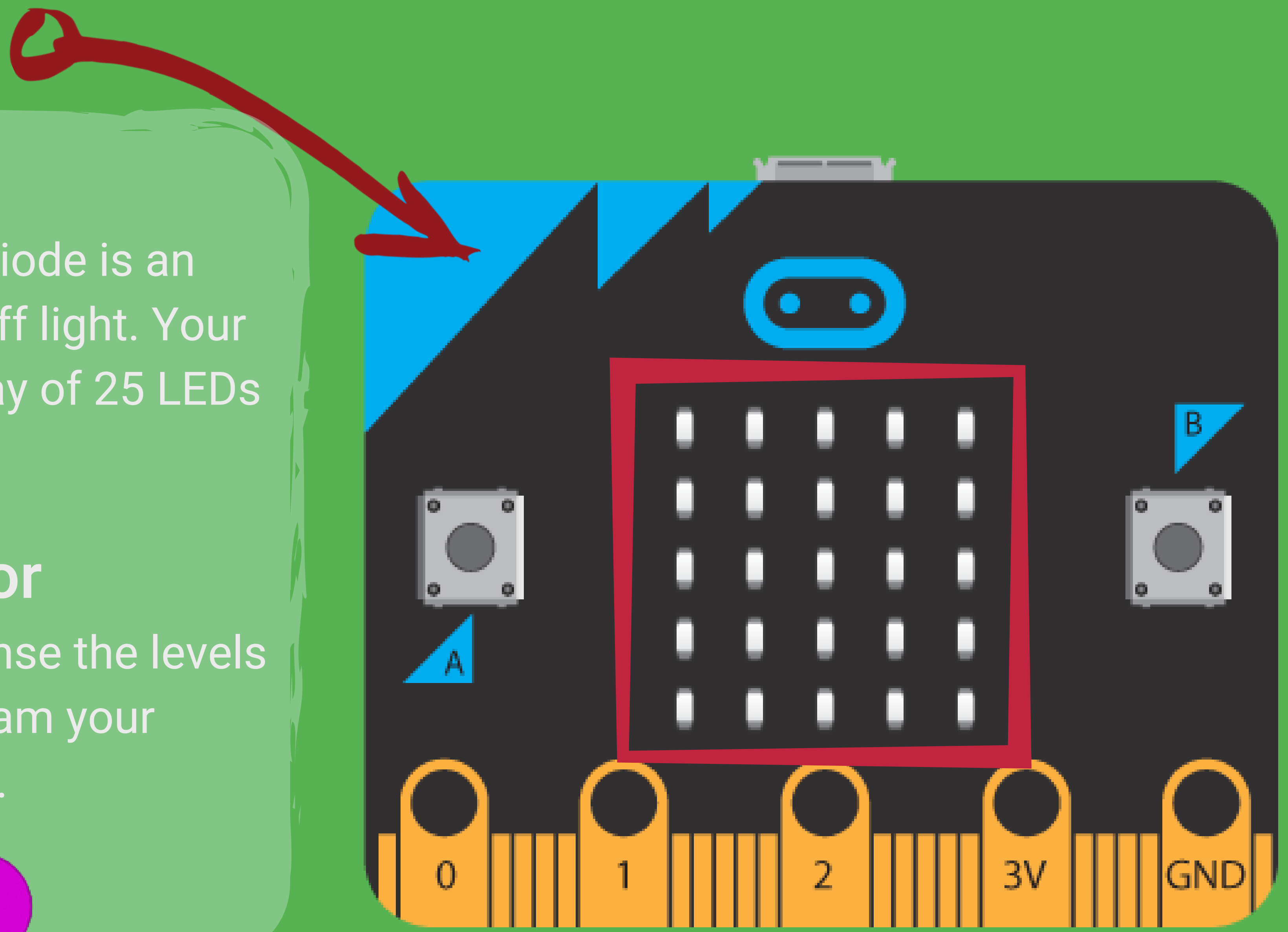
LEDs

An LED, or light-emitting diode is an output device that gives off light. Your BBC micro:bit has a display of 25 LEDs for you to program.

Light sensor

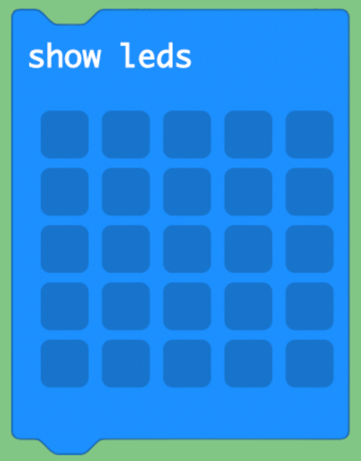
Micro:Bit uses LEDs to sense the levels of light and lets you program your micro:bit as a light sensor.


light level



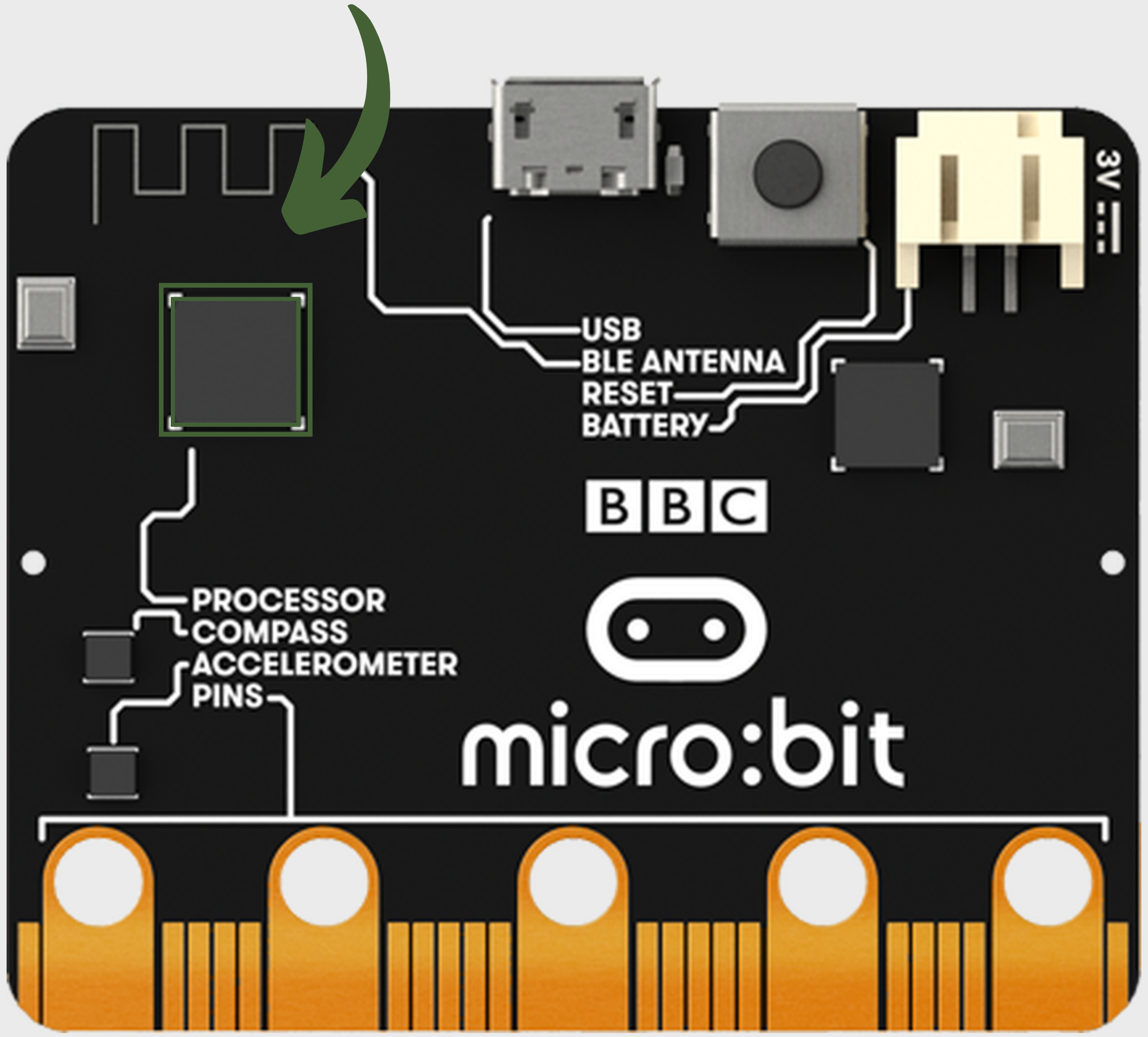
Output

LEDs on Micro:Bit V1 is the output. You can use them to express information within 5*5 LED grids.


show leds

@CoraEdTech

Processor



What is it?

A processor is sometimes called the **‘brains’ of a computer** and each Micro:Bit has a micro processor inside it. It’s an essential part of your micro:bit as it runs the programs you write.

Temperature sensor

temperature (°C)

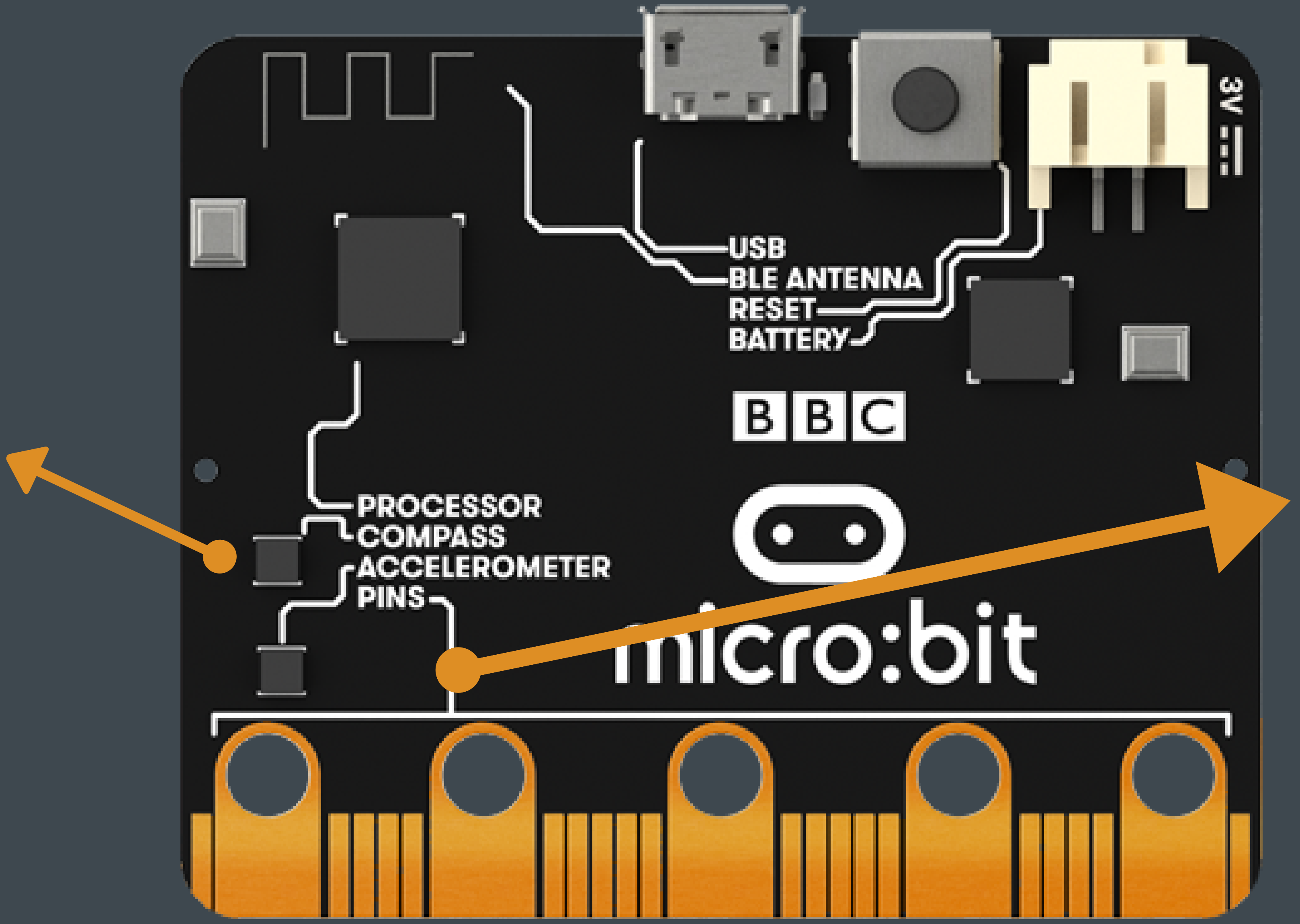
A temperature sensor is an input device that measures temperature. A Micro:Bit has **a temperature sensor inside the processor** which can give you an approximation of the air temperature.

Accelerometer & Compass

Compass

A digital compass is an input sensor that detects magnetic fields.
Each Micro:bit has an inbuilt compass that can detect the direction in which it is facing.

`compass heading (°)`



Accelerometer

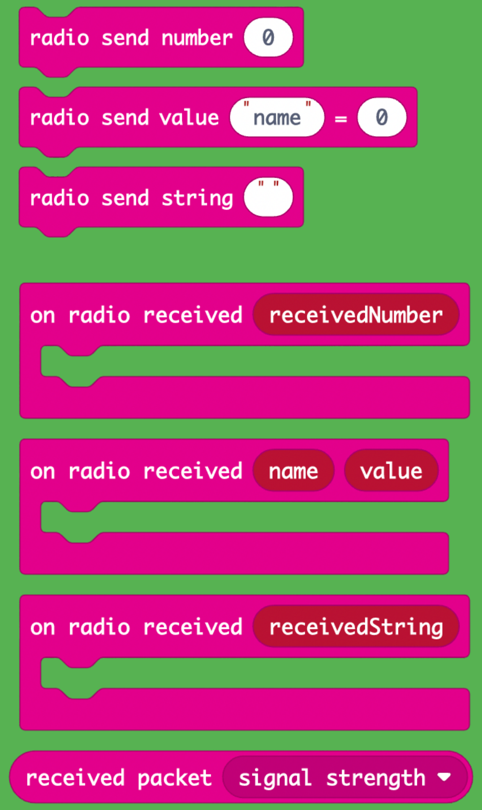
An accelerometer is a motion sensor that measures movement.
The accelerometer in each Micro:bit detects when you tilt it left to right, backwards and forwards and up and down.

`acceleration (mg) x ▾`

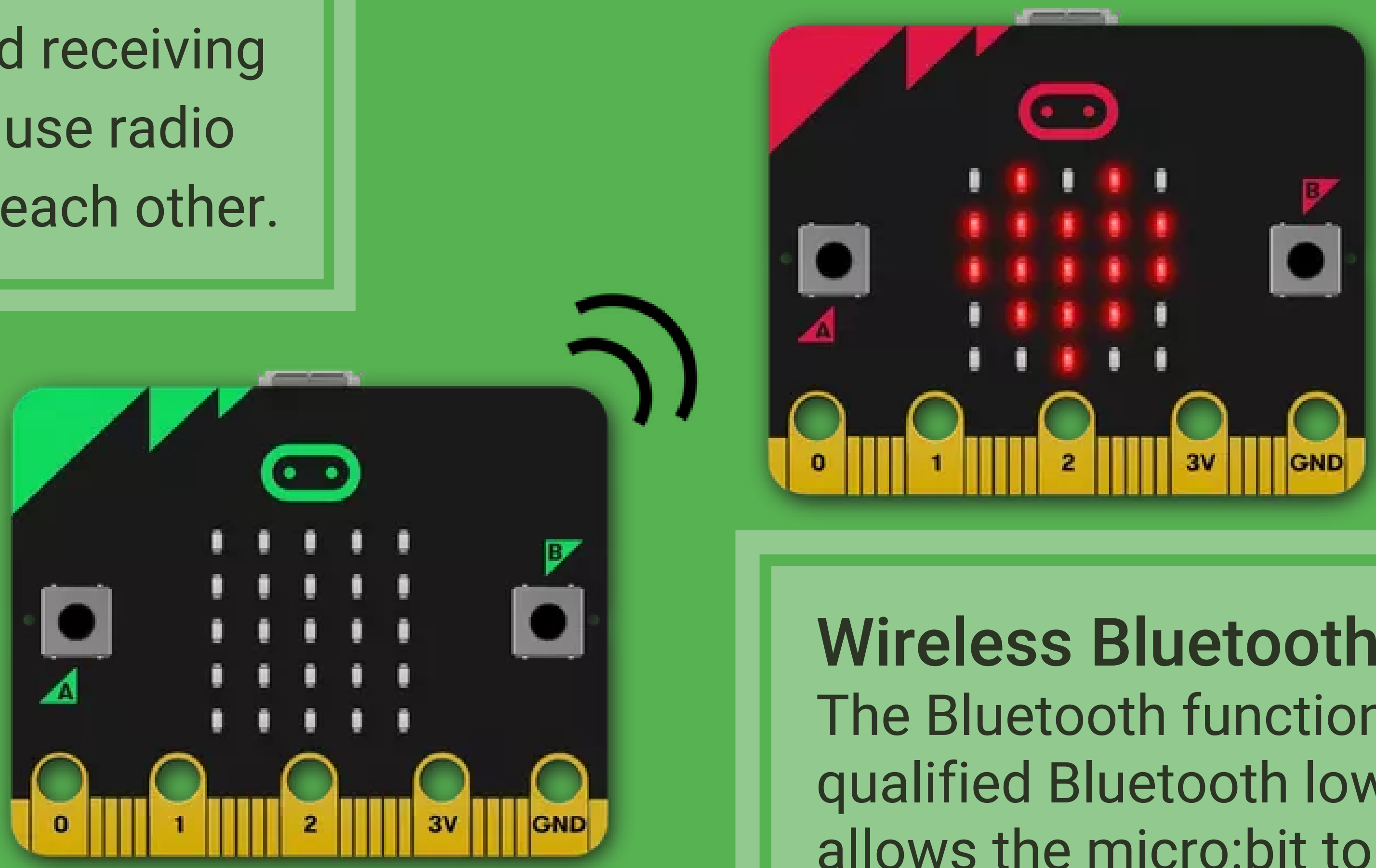
Communications

Radio

Radio is a way of sending and receiving messages and micro:bit can use radio waves to communicate with each other.

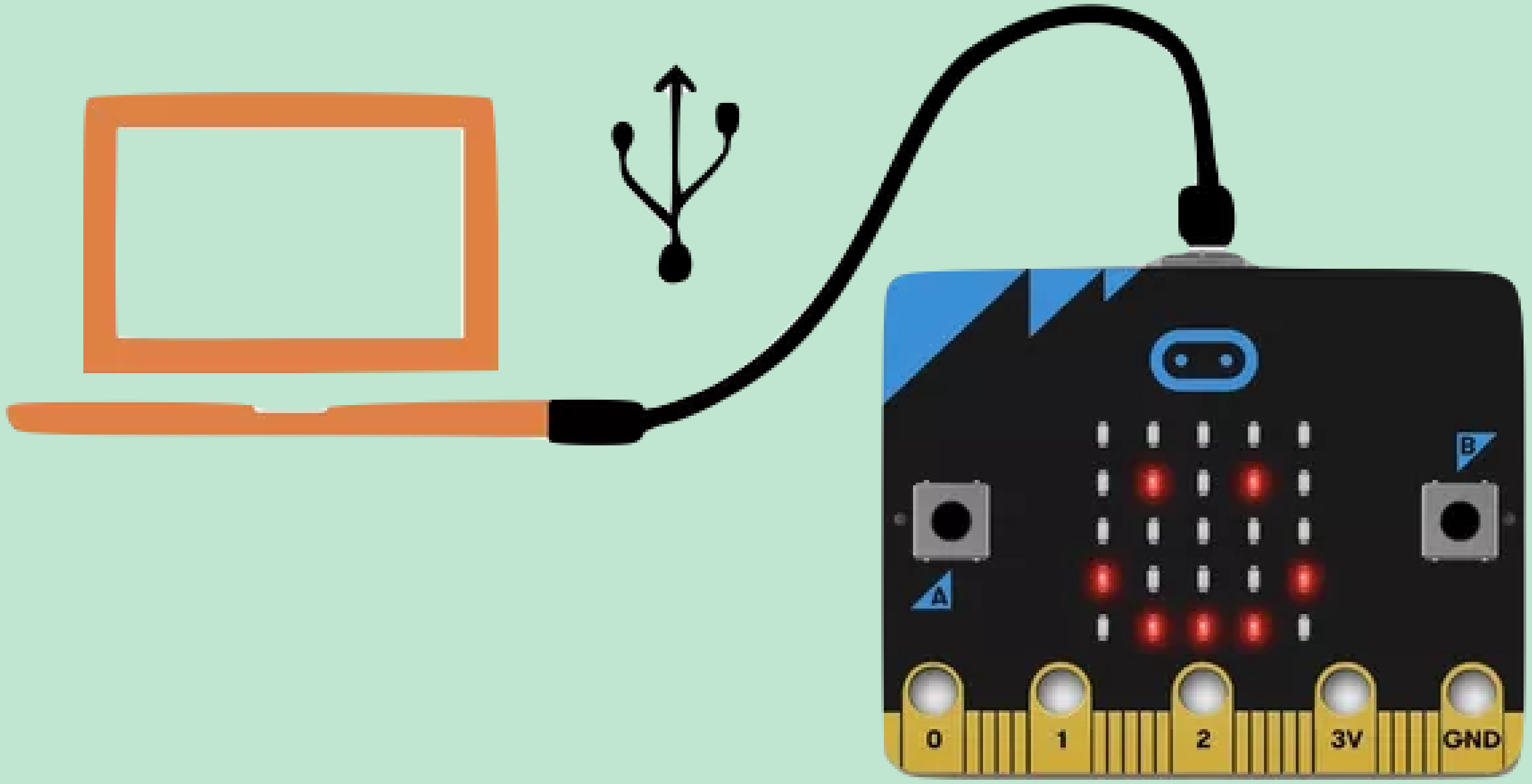


```
radio send number 0
radio send value "name" = 0
radio send string ""

on radio received receivedNumber

on radio received name value

on radio received receivedString

received packet signal strength ▾
```



Wireless Bluetooth

The Bluetooth function provides a fully qualified Bluetooth low-energy stack. This allows the micro:bit to communicate with a wide range of Bluetooth devices, including smartphones and tablets.



USB interface

USBs, or Universal Serial Buses, are are used to connect, communicate and power computers and digital devices.

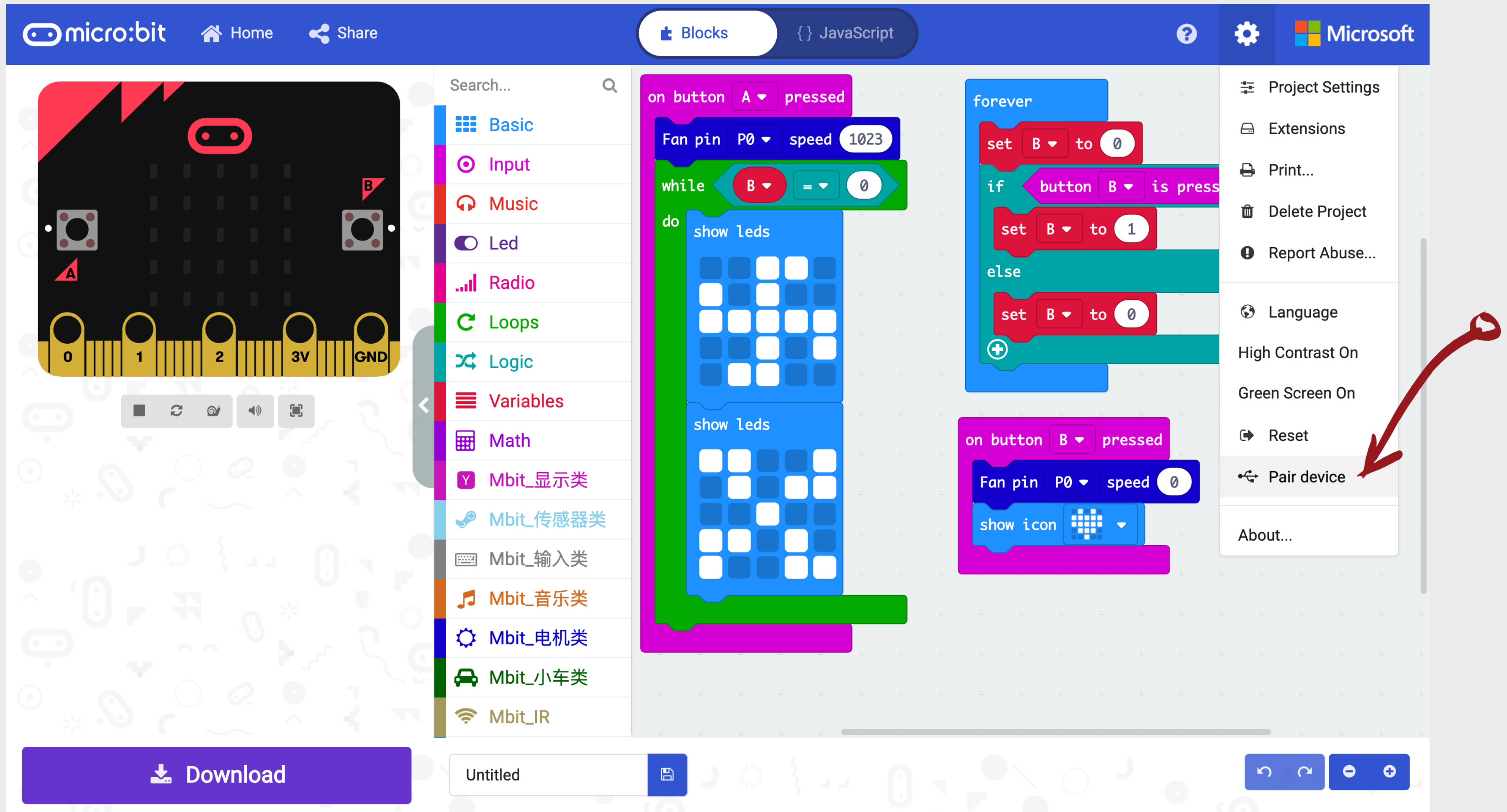
The Micro:bit has a USB interface to allow you to connect your computer to your micro:bit using a micro USB cable and power your micro:bit.

Connection

Getting To Know Micro:Bit V1

Connect Micro:Bit with MakeCode

Use USB Interface



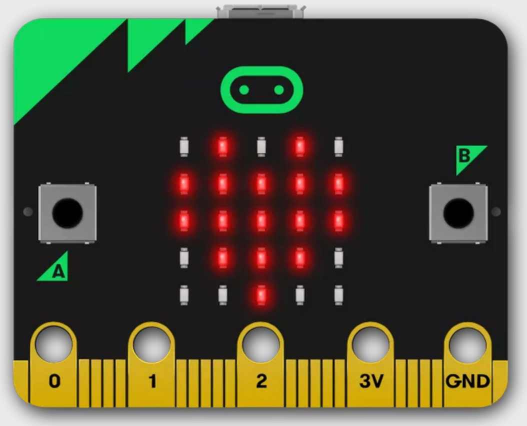
```
on button A pressed
  Fan pin P0 speed 1023
  while B = 0
  do
    show leds
    show leds

forever
  set B to 0
  if button B is press
    set B to 1
  else
    set B to 0

on button B pressed
  Fan pin P0 speed 0
  show icon
```

- From the MakeCode editor settings menu, choose **Pair Device**. You will see a pop-up in the browser that lists any micro:bits you have connected with a USB cable.
- Select the device named **BBC micro:bit CMSIS-DAP** and then **Connect**. You will see a message telling you that a connection has been made.
- Try downloading your program, you should find that it is now flashed directly to the micro:bit each time you press **DOWNLOAD**.

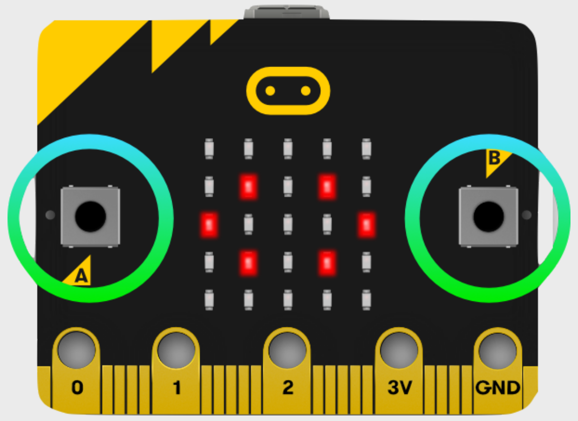
@CoraEdTech

Sensor Practice

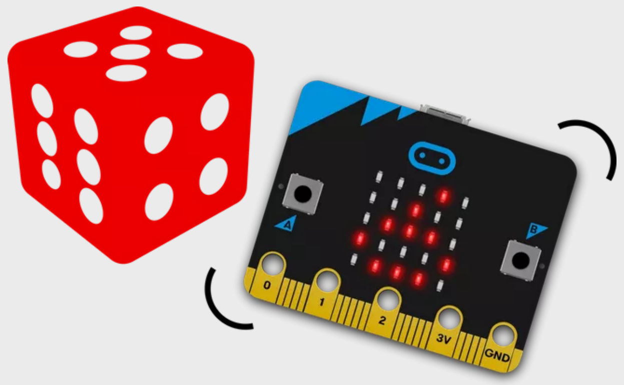
Learn from hands-on activities



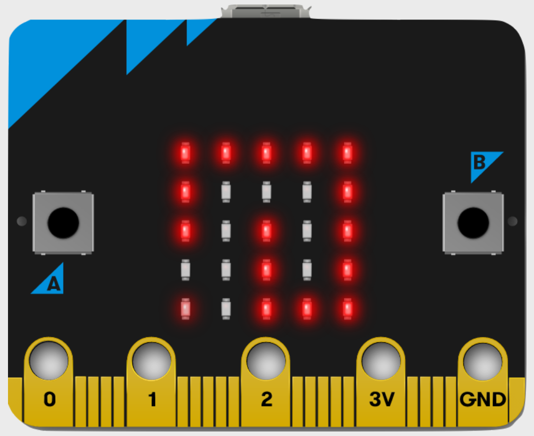
Beating Heart



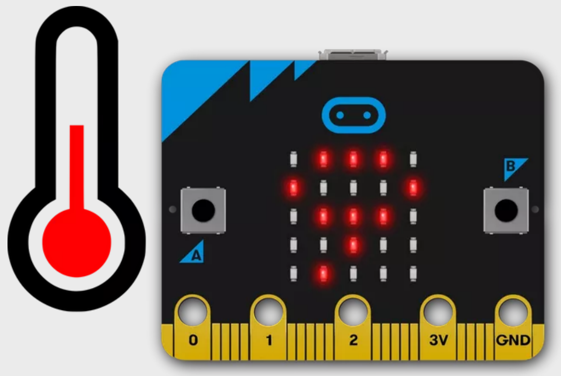
Press It!



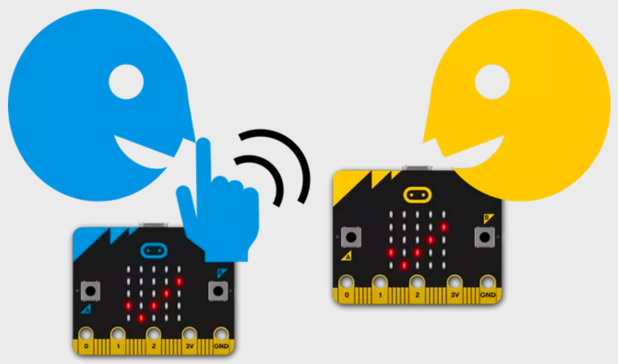
Roll the Dice



Floating Sand



Thermometer



Talk to Me

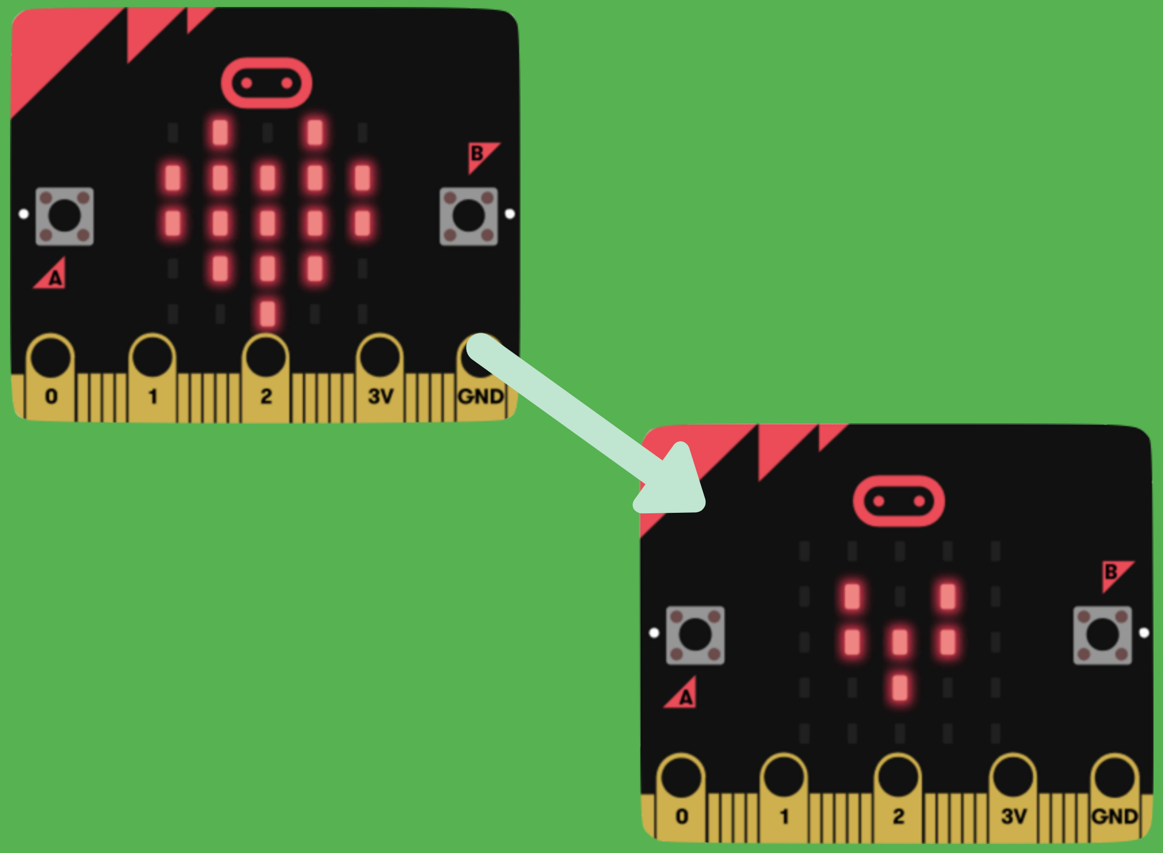
@CoraEdTech

Beating Heart

Use two LED images to make a beating heart.

Requirements:

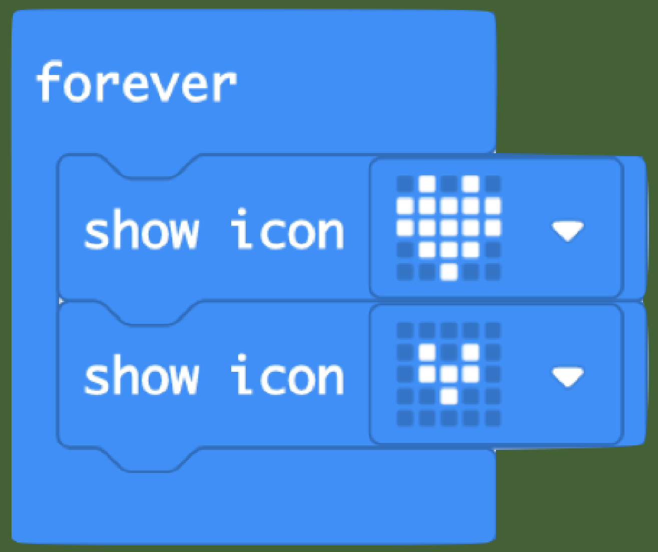
Programme your Micro:Bit to make LEDs show big heart and turn to the small heart - make it look like a beating heart.



Solutions:

Use "show icon"/"show LEDs" to program different icons on LEDs.

You can use "forever" block to make a loop.

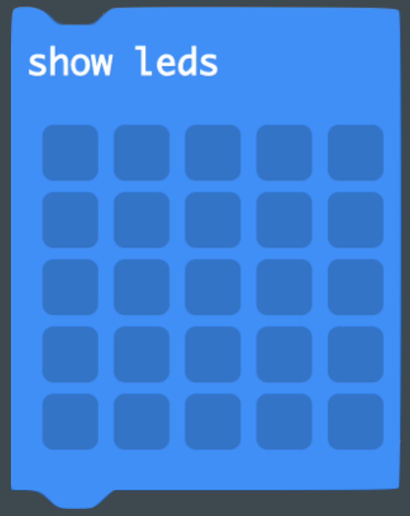


```
forever
  show icon
  show icon
```

Challenge:

Can you try other different icons? Maybe animals, squares or circles?

Or you can create your own icons by lightening LEDs.



```
show leds
```

@CoraEdTech

Press It!

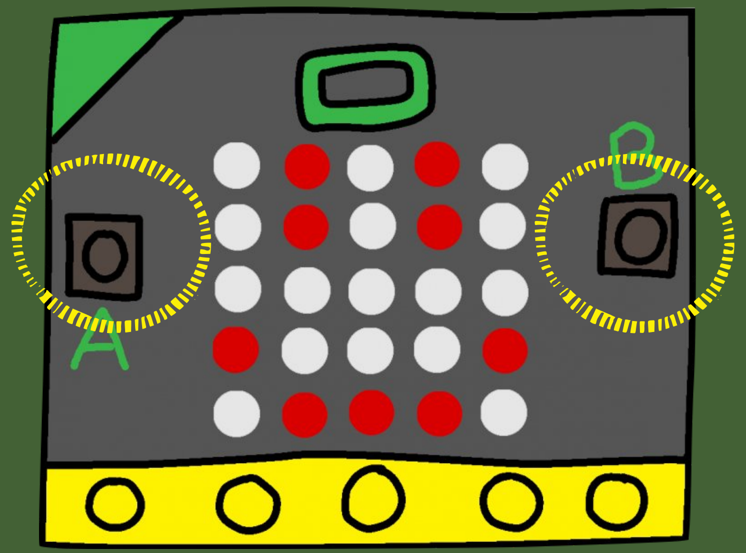
A game that using buttons as triggers.

Requirements:

Two players press buttons A/B at the same time.

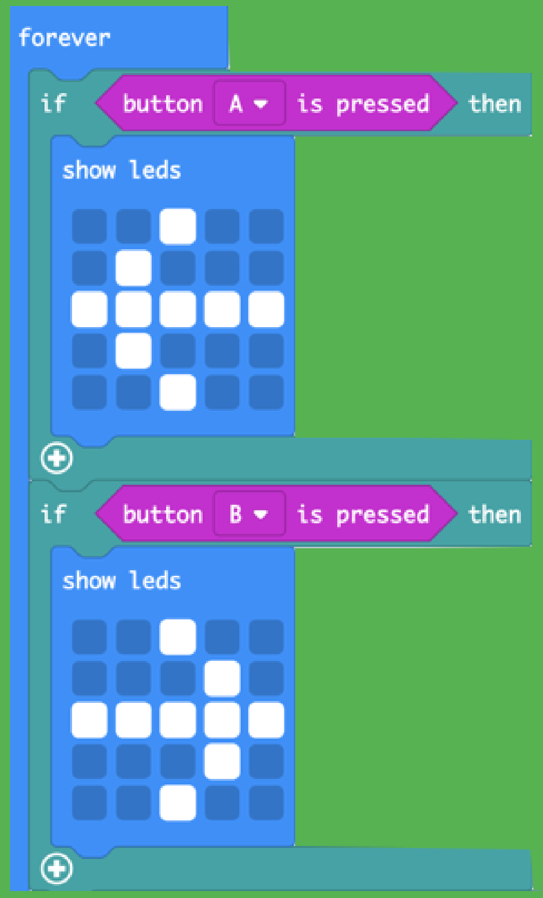
If button A is pressed first, LEDs will show an arrow point to A or vice verse.

If two players press at the same time, LEDs will show a smily face.

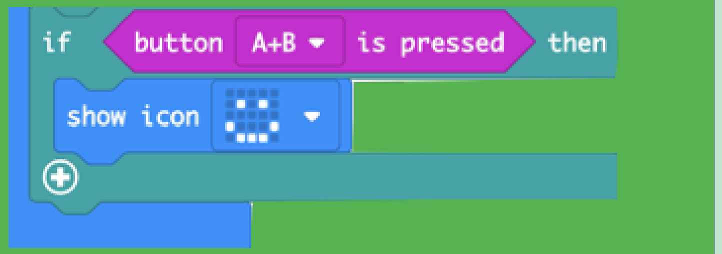


Solutions:

Use conditional block "if... then..." and input block "when button is pressed" to insert conditions in your programming.



```
forever
  if button A ▾ is pressed then
    show leds
  ⊕
  if button B ▾ is pressed then
    show leds
  ⊕
```



```
if button A+B ▾ is pressed then
  show icon ▾
⊕
```

Challenge:

Can you try to use buttons to make different animations? For instance, you can press A to make a beating heart; press B to make a shrink square.

```
if true ▾ then
⊕
```

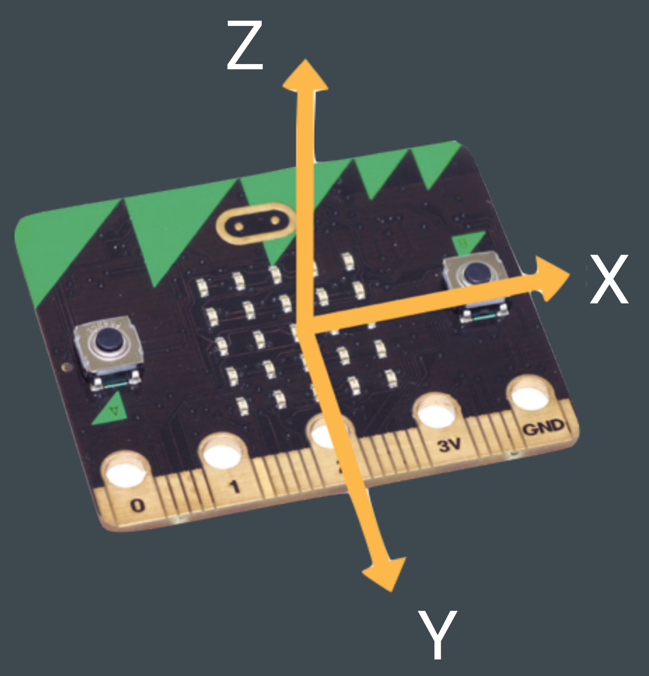
@CoraEdTech

Floating Sand

Use accelerometer to measure the movement of Micro:Bit.

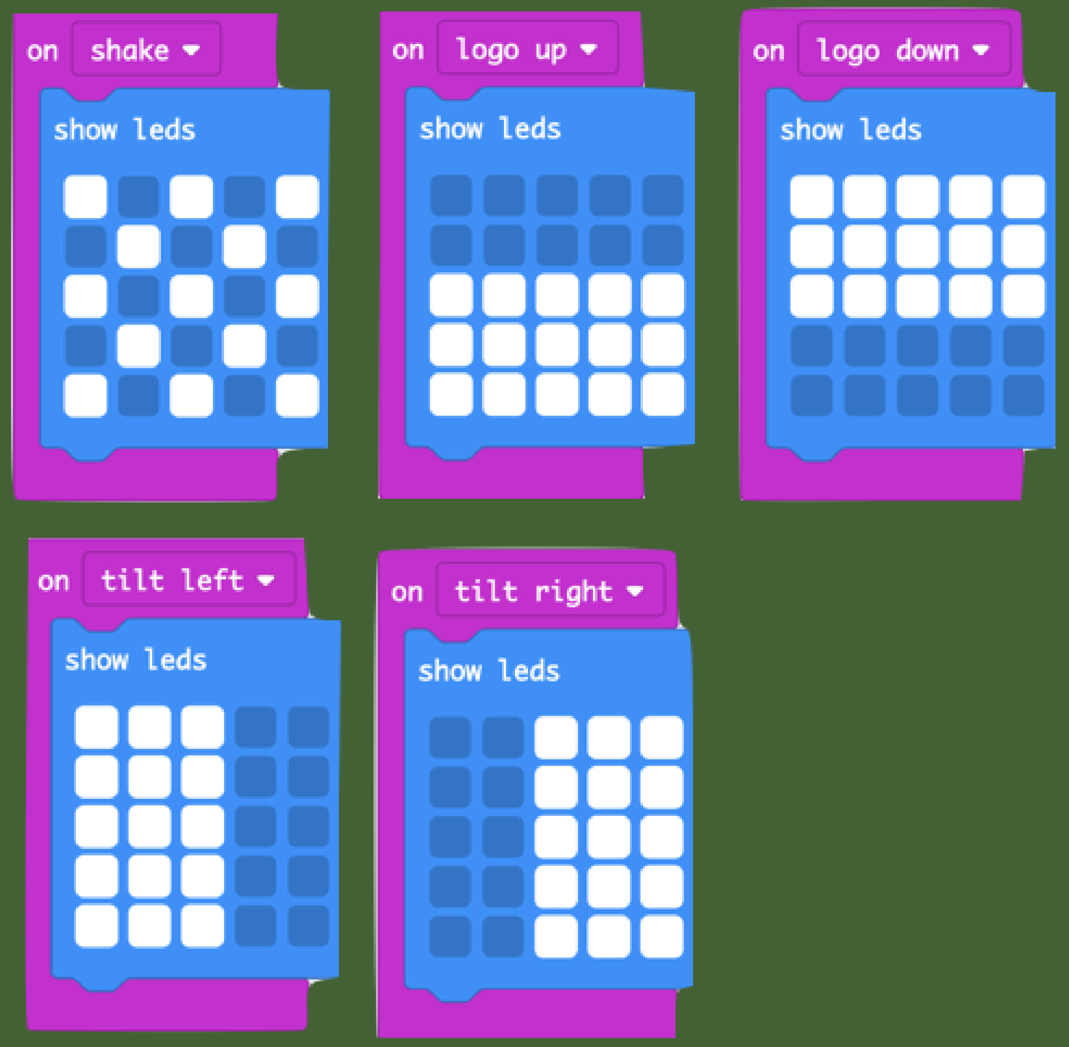
Requirements:

Shake Micro:Bit - LEDs lighten separately;
Tilt to the left - Left side of LEDs lighten;
Tilt to the right - Right side of LEDs lighten;
Tilt forwards - the upper side of LEDs lighten;
Tilt backwards - bottom of LEDs lighten.



Solutions:

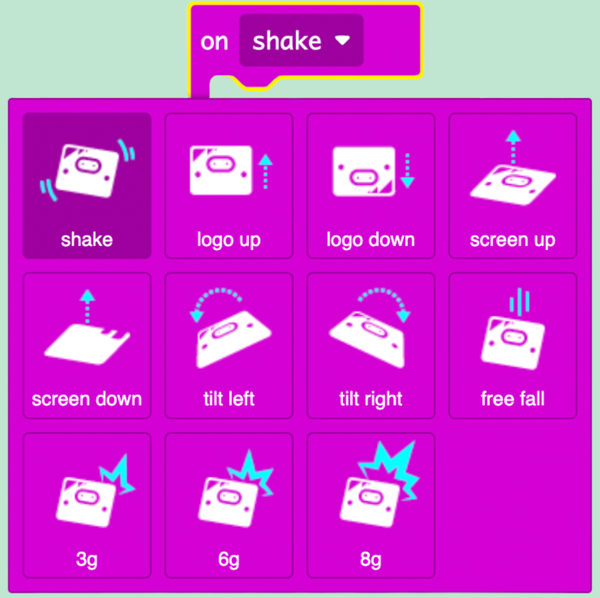
Input blocks could help Micro:Bit sensor movement precisely.



```
on shake
  show leds
    # . # . #
    . # . # .
    # . # . #
    . # . # .
    # . # . #

on logo up
  show leds
    . . . . .
    . . . . .
    # # # # #
    # # # # #
    # # # # #

on logo down
  show leds
    # # # # #
    # # # # #
    # # # # #
    . . . . .
    . . . . .

on tilt left
  show leds
    # # # . .
    # # # . .
    # # # . .
    # # # . .
    # # # . .

on tilt right
  show leds
    . . # # #
    . . # # #
    . . # # #
    . . # # #
    . . # # #
```

Challenge:

Add more gestures to see what are differences among all triggering gestures.



@CoraEdTech

Roll Dice

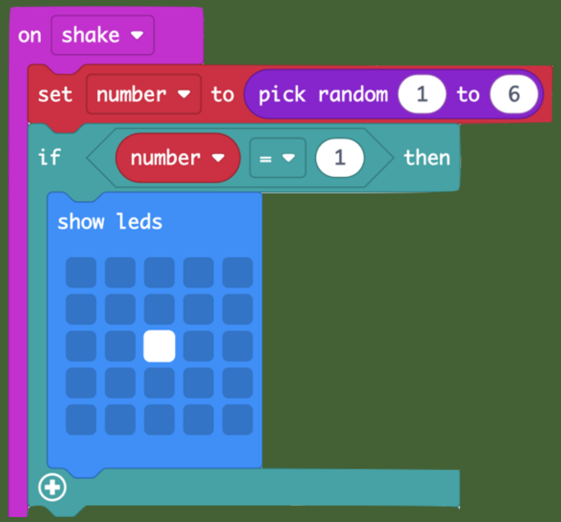
Shake your Micro:Bit to show a random LED picture.

Requirements:

Shake your Micro:Bit - on the LEDs, it shows a random LED picture. It can be one bright LED one, or two, or three... Exact the same like dice we usually play!

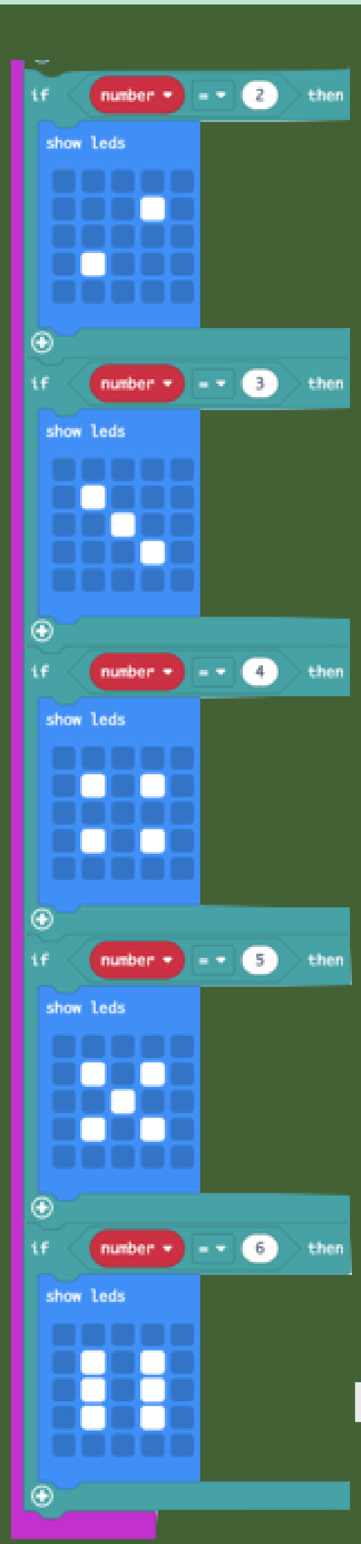
Solutions:

Use "on shake" block to trigger the Micro:Bit.
Set a variable called "number" and pick a random number between 1 to 6 (in math block list).
Set different LED pictures when the number comes as different results.



```
on shake
  set number to pick random 1 to 6
  if number = 1 then
    show leds
      . . . . .
      . . . . .
      . . # . .
      . . . . .
      . . . . .
```

Picture 1



```
  if number = 2 then
    show leds
      . . . . .
      . . . # .
      . . . . .
      . # . . .
      . . . . .
  if number = 3 then
    show leds
      . . . . .
      . # . . .
      . . # . .
      . . . # .
      . . . . .
  if number = 4 then
    show leds
      . . . . .
      . # . # .
      . . . . .
      . # . # .
      . . . . .
  if number = 5 then
    show leds
      . . . . .
      . # . # .
      . . # . .
      . # . # .
      . . . . .
  if number = 6 then
    show leds
      . . . . .
      . # . # .
      . # . # .
      . # . # .
      . . . . .
```

Picture 2

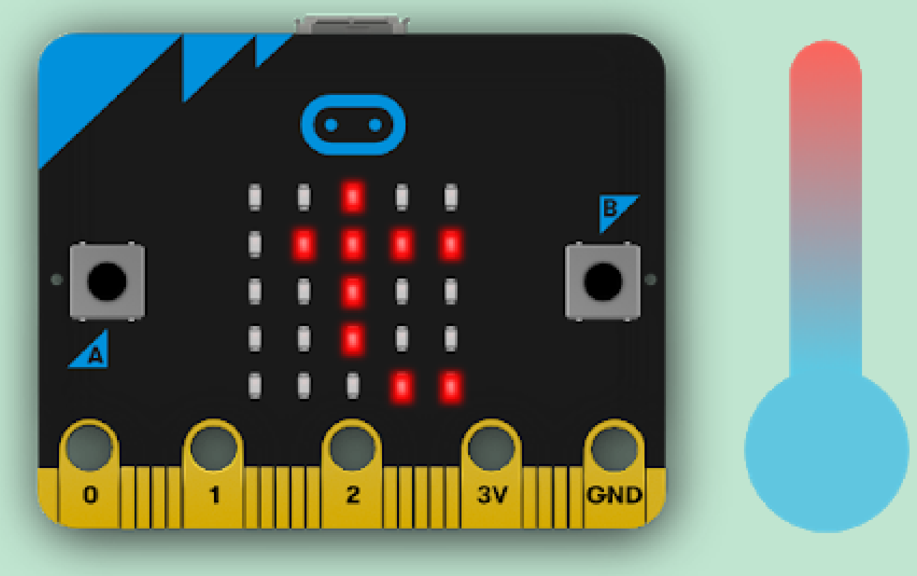
@CoraEdTech

Thermometer

Use temperature sensor to sense the temperature and make it shown on LEDs.

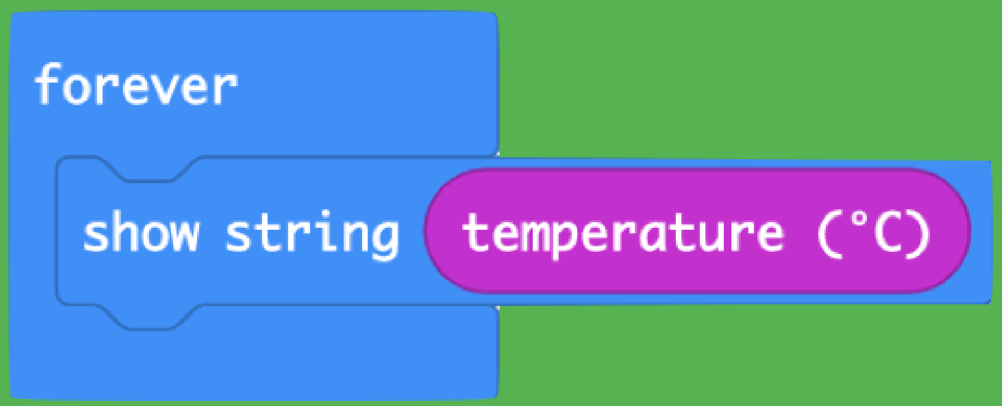
Requirements:

Use temperature sensor to sense the room temperature. The digit will show on LEDs.



Solutions:

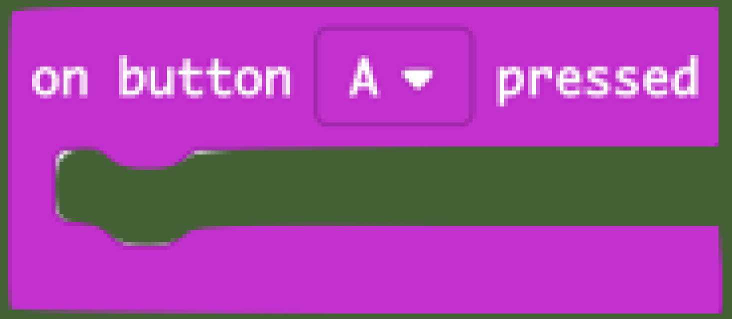
Use input block "temperature" to make your Micro:Bit sense the temperature.
"Show string" block can put the digit to LEDs.



```
forever
  show string  temperature (°C)
```

Challenge:

Can you use buttons to make the Micro:Bit switch the number of temperature to other pictures; and switch it back?



```
on button A ▾ pressed
```

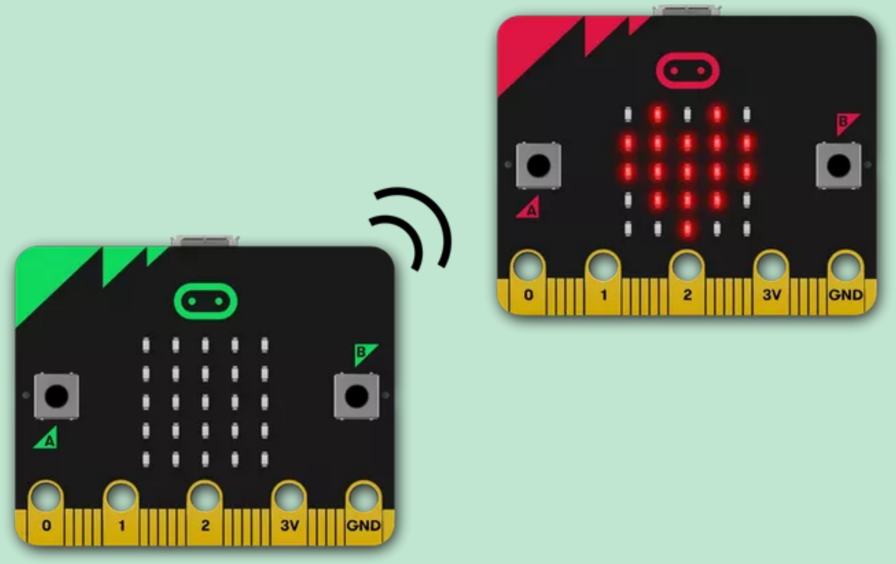
Talk To Me

Use radio to communicate with each other - work in pairs.

Requirements:

When button A on Micro:Bit A is pressed, Micro:Bit B will receive the signal and show "A" on LEDs;

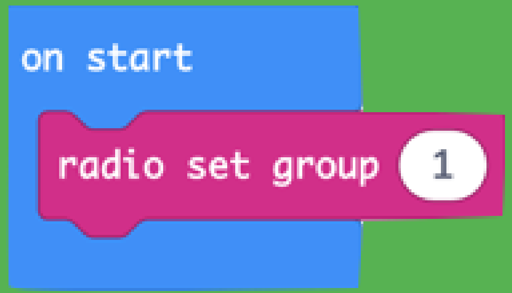
When button B is pressed, Micro:Bit B will receive another signal and show "B" on its LEDs.



Solutions:

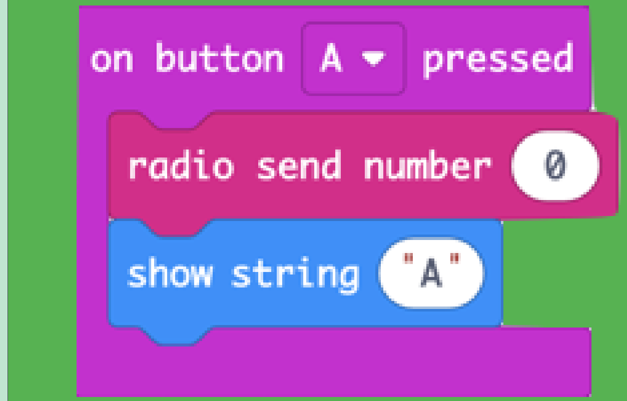
Use "radio set group" to make a program have the group ID you tell it for sending and receiving with radio.

Micro:Bit A:

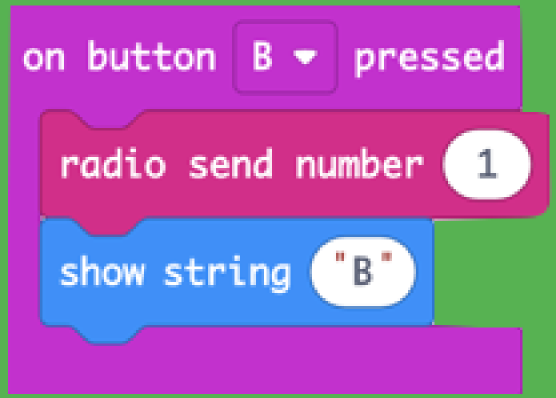


```
on start
    radio set group 1
```

The radio group number should be the same for both Micro:Bits.



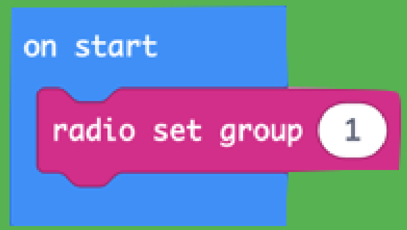
```
on button A ▾ pressed
    radio send number 0
    show string "A"
```



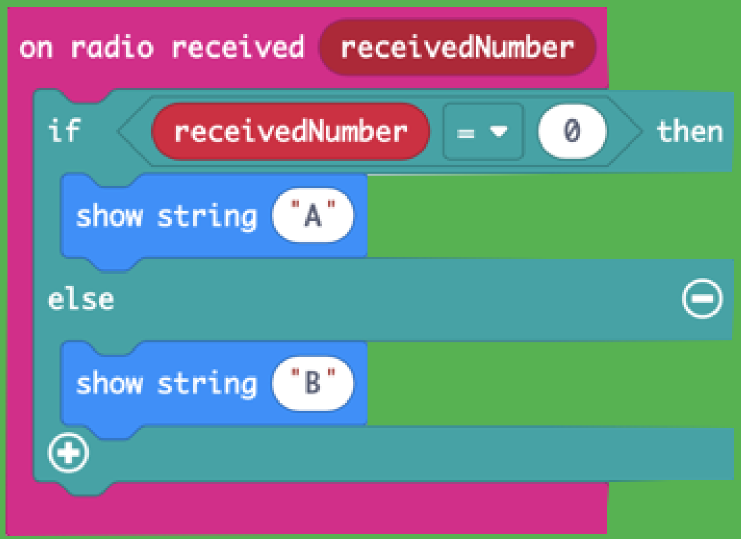
```
on button B ▾ pressed
    radio send number 1
    show string "B"
```

When it receives signal 0, Micro:Bit B will show string A; otherwise, it will show B.

Micro:Bit B:



```
on start
    radio set group 1
```

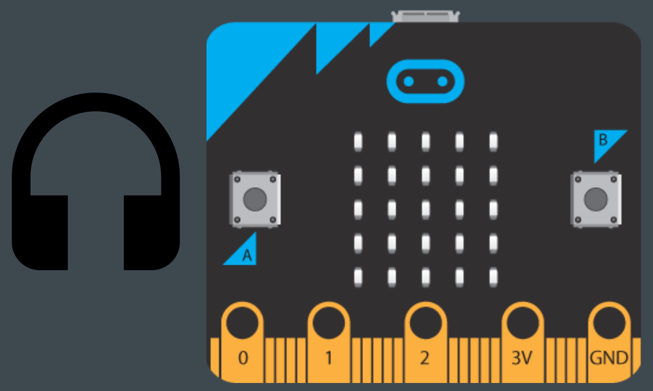


```
on radio received receivedNumber
    if receivedNumber = ▾ 0 then
        show string "A"
    else ⊖
        show string "B"
    ⊕
```

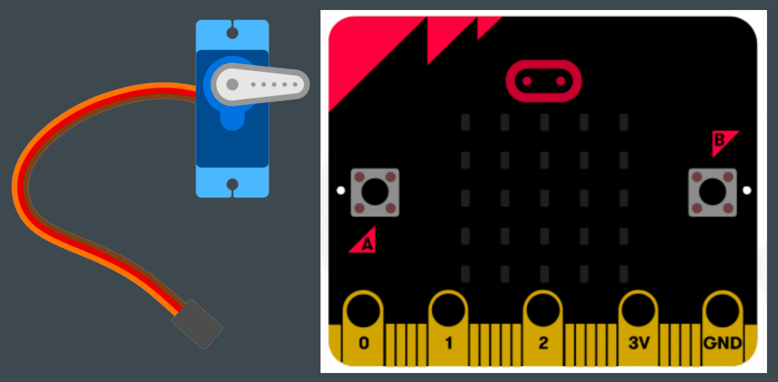
@CoraEdTech

Using Add-on Components with Micro:Bit

We'll dive in Micro:Bit by using add-on components.



Listen Music



Make It Move

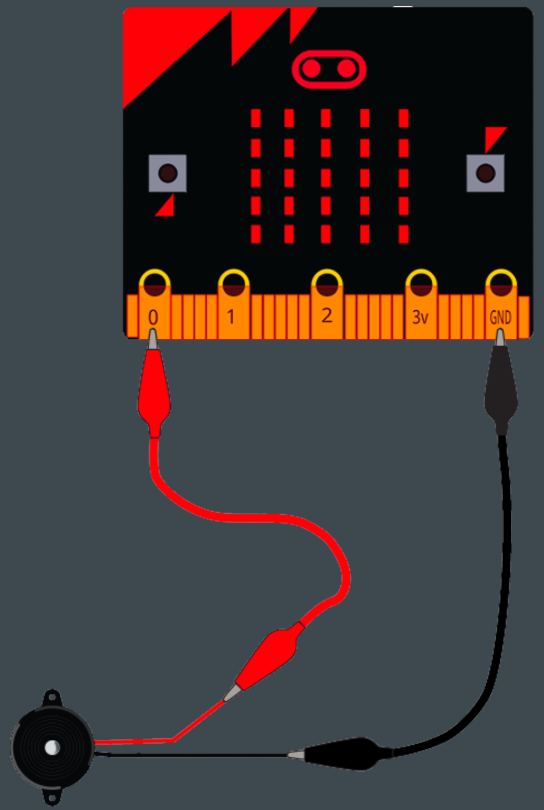
@CoraEdTech

Music Time!

Micro:Bit V1 doesn't have speakers - but we can attach a buzzer to make it sing.

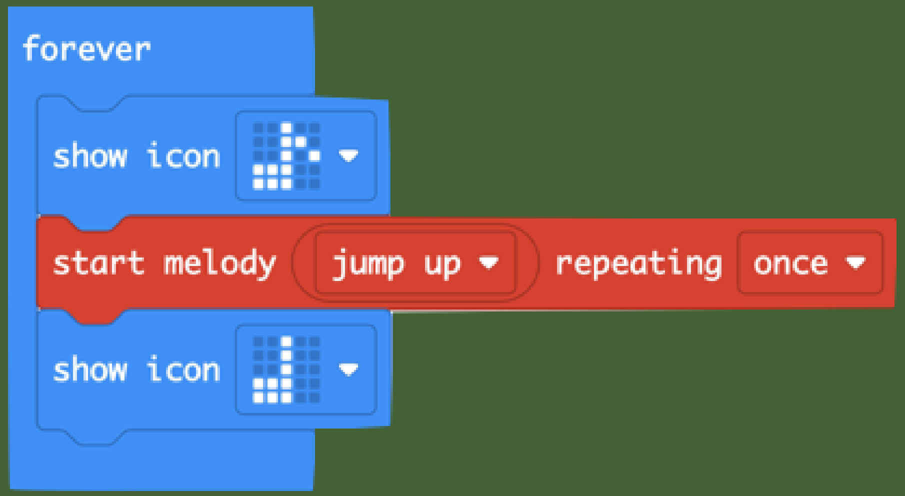
Requirements:

- This project requires two alligator clips and a headset.
- The GND connects to G plug of headset; The left/right sound area connects to signal pin.



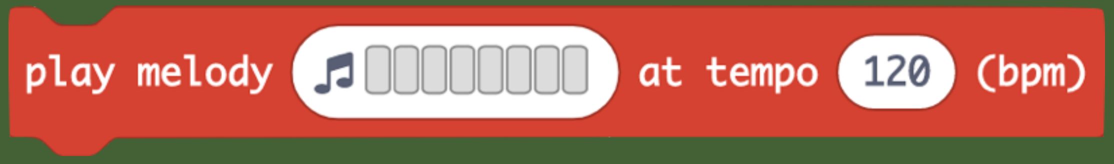
Solutions:

Use "show icon" blocks to show that your music is playing.



```
forever
    show icon
    start melody jump up repeating once
    show icon
```

You can also compose music by using the following block:



```
play melody ♬ at tempo 120 (bpm)
```

Challenges:

- Use buttons to control music playing.
- Show players the name of the music.
- Compose your own music.

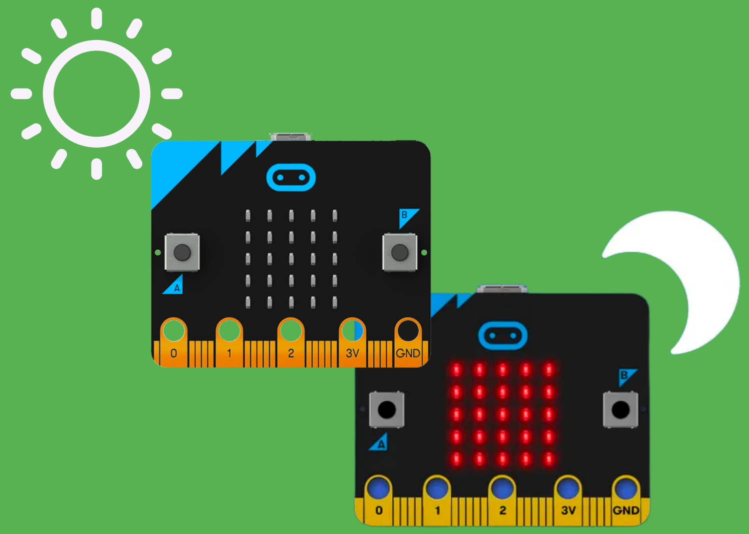
@CoraEdTech

Make It Move

Attach servo to Micro:Bit to make the movement as an output.

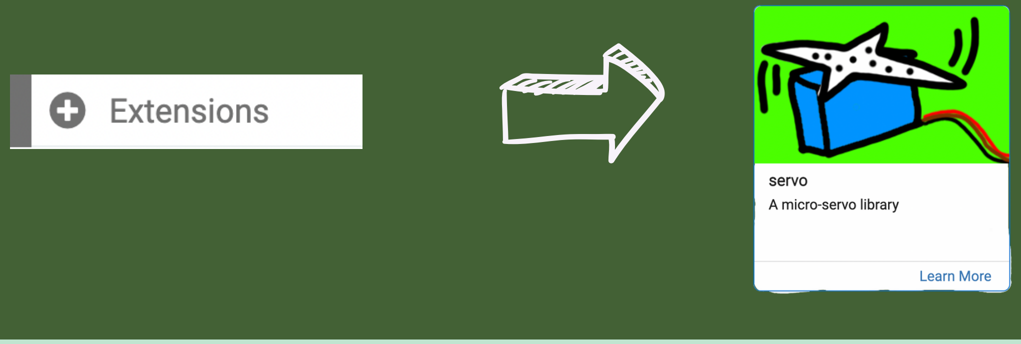
Requirements:

Make the servo move when the environment is bright. Cover your Micro:Bit to mimic the nighttime, so the servo will 'go to bed' and stop moving.

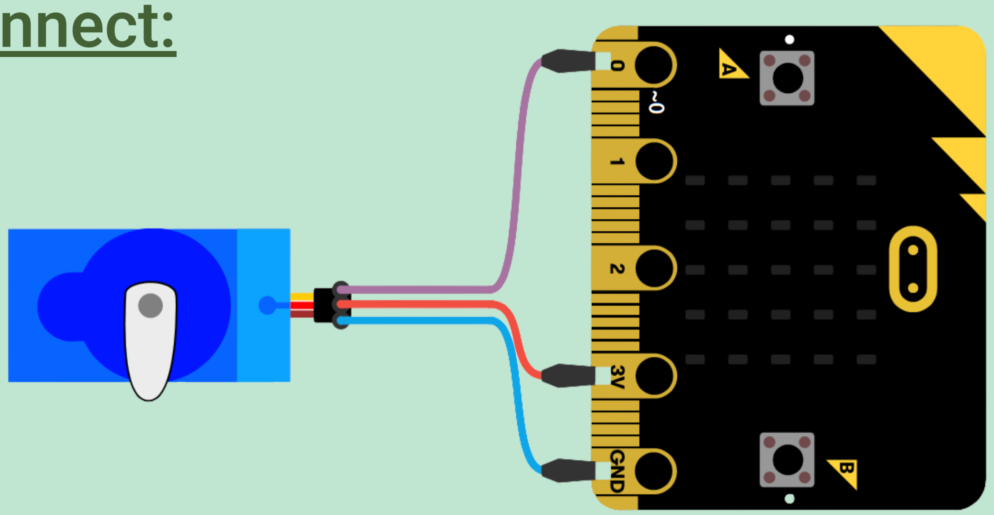


Add Extension:

The servo blocks are not listed on the default categories. You need to go to Extensions and add the servo blocks.

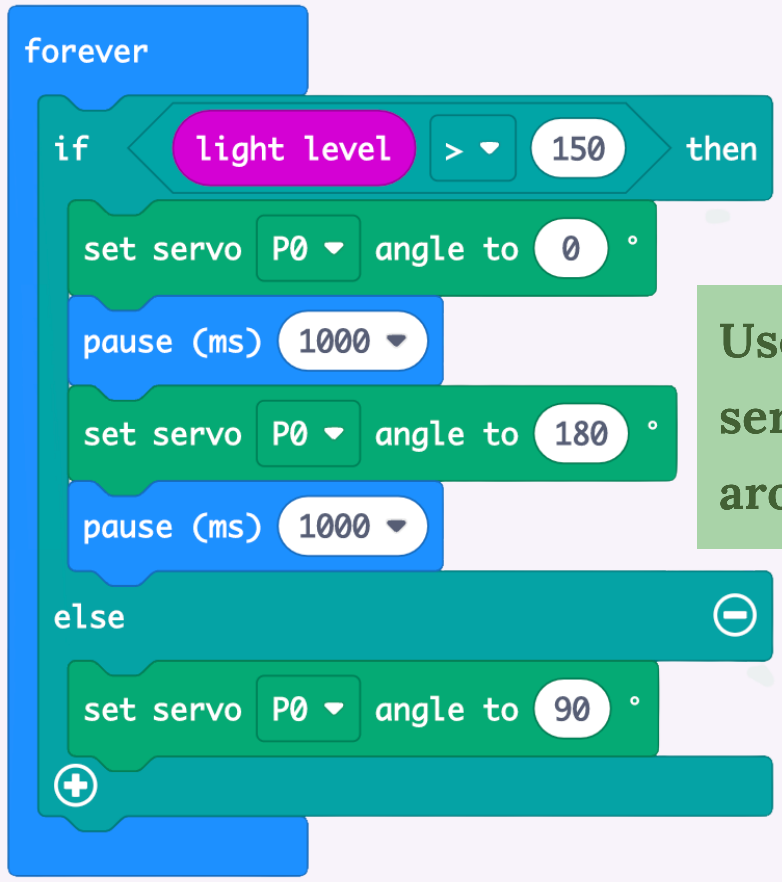


Connect:



Solutions:

In this solution, we use the conditional block to set the brightness condition to trigger the servo.



```
forever
  if light level > 150 then
    set servo P0 angle to 0 °
    pause (ms) 1000
    set servo P0 angle to 180 °
    pause (ms) 1000
  else
    set servo P0 angle to 90 °
```

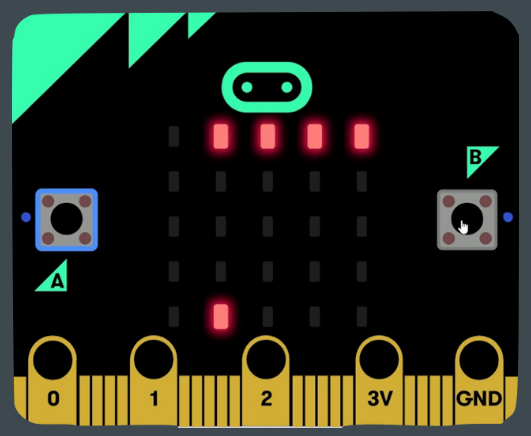
Use pause blocks to give servo time to move around.

@CoraEdTech

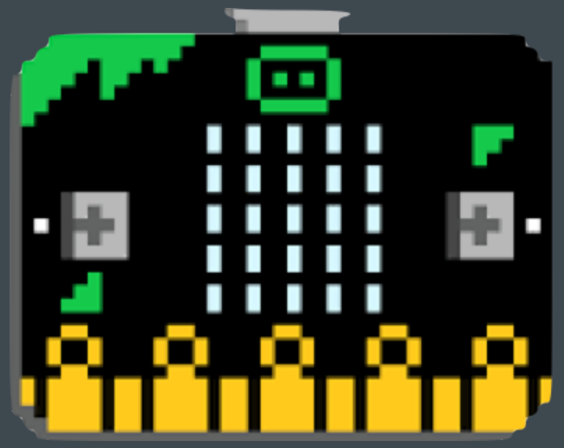
Extra Activities:

Turn Micro:Bit into Gameboy

Turn your Micro:Bit into gameboy - Let's code on it to make interesting games on this little chip!



Dodgeball



Shooting Game

@CoraEdTech

Space Invaders

Program on Micro:Bit to create the fun Space Invaders game!

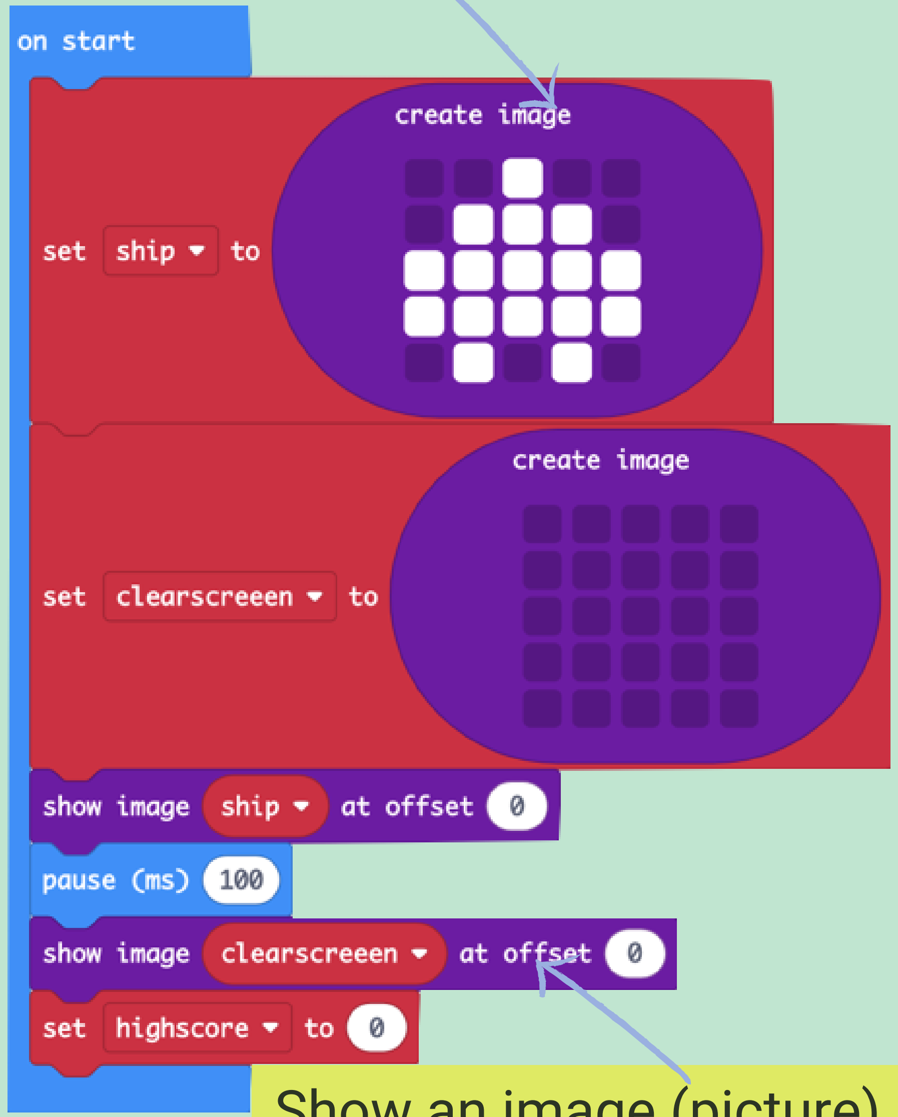
Requirements:

When you see the spaceship appears on the screen, press the button A to catch it. You have 5 opportunities to capture it. Let's see how brave you are at protecting your planet!

Variables:

- clearscreeen
- delay
- highscore
- onscreen
- score
- ship

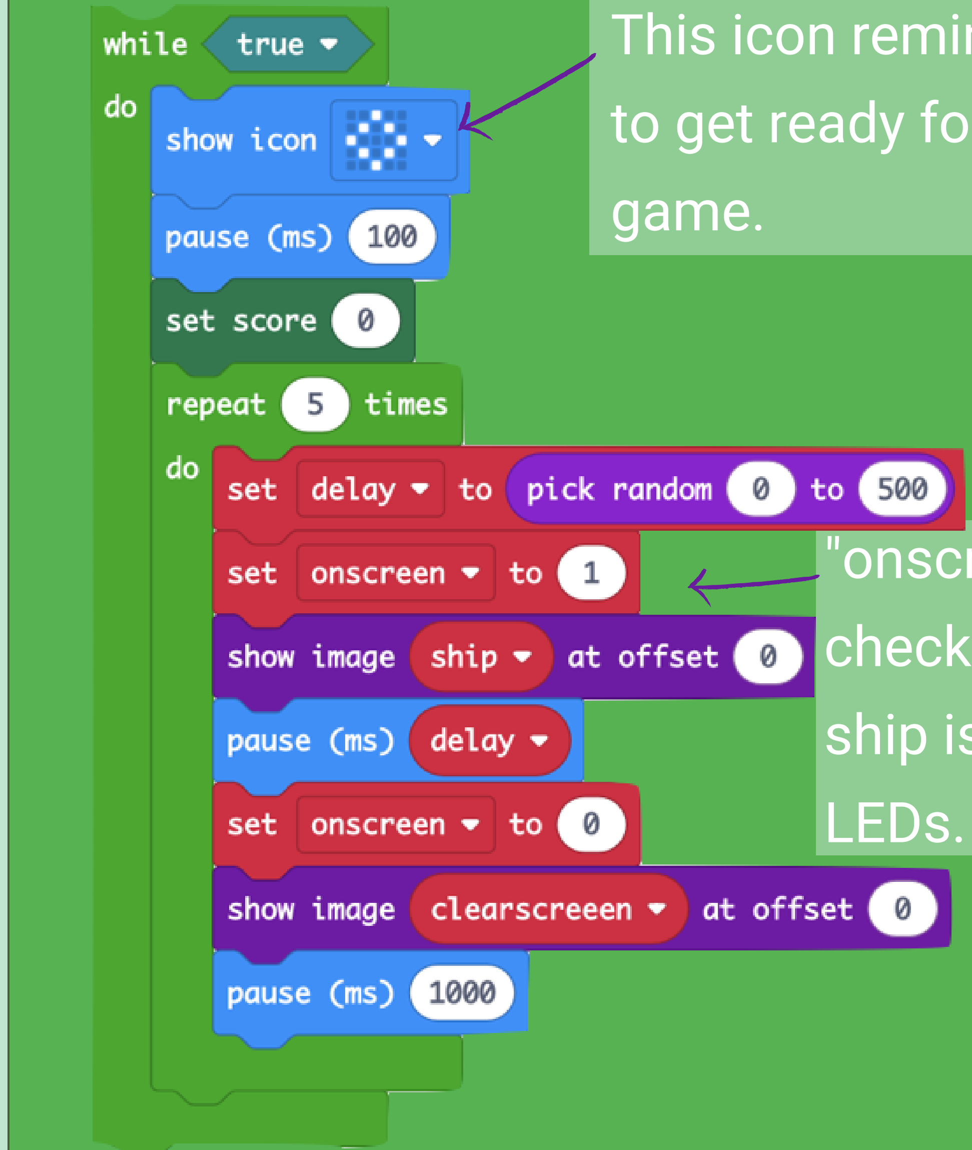
Setting a variable with an image is because we will use it for further coding.



```
on start
    set ship to create image
    set clearscreeen to create image
    show image ship at offset 0
    pause (ms) 100
    show image clearscreeen at offset 0
    set highscore to 0
```

Show an image (picture) on the LED screen.

Step 1:
Set up the game in the beginning, including show the game image.



```
while true
do
    show icon
    pause (ms) 100
    set score 0
    repeat 5 times
    do
        set delay to pick random 0 to 500
        set onscreen to 1
        show image ship at offset 0
        pause (ms) delay
        set onscreen to 0
        show image clearscreeen at offset 0
        pause (ms) 1000
```

This icon reminds players to get ready for the game.

"onscreen" is to check if your ship is shown on LEDs.

Step 2:
Set your spaceship to show on LEDs randomly within 0 to 500 milliseconds. Make it to repeat 5 times.

@CoraEdTech

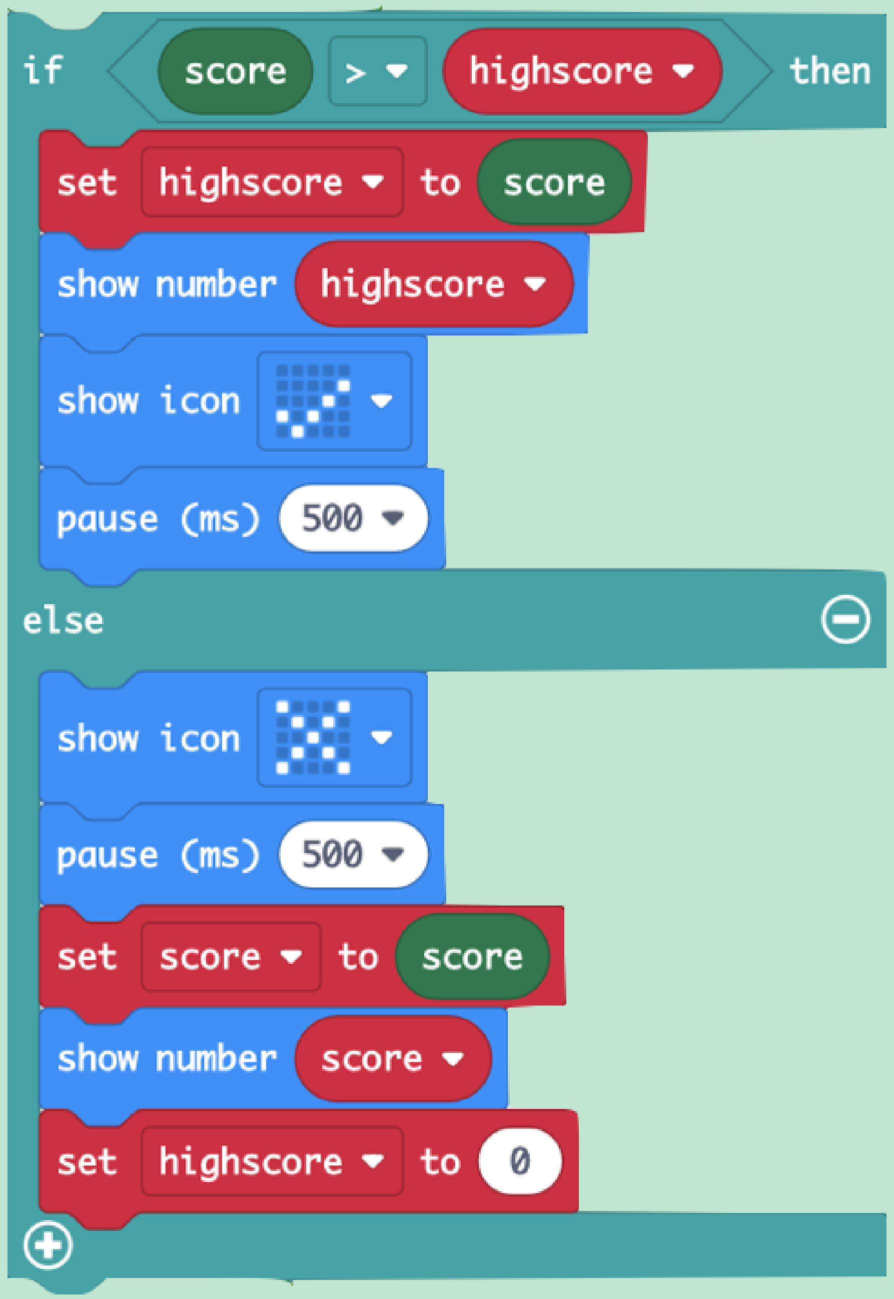
Space Invaders

Program on Micro:Bit to create the fun Space Invaders game!

Step 3:

Each round you play will have a score - if the score is higher than previous rounds, you set yourself a higher goal.

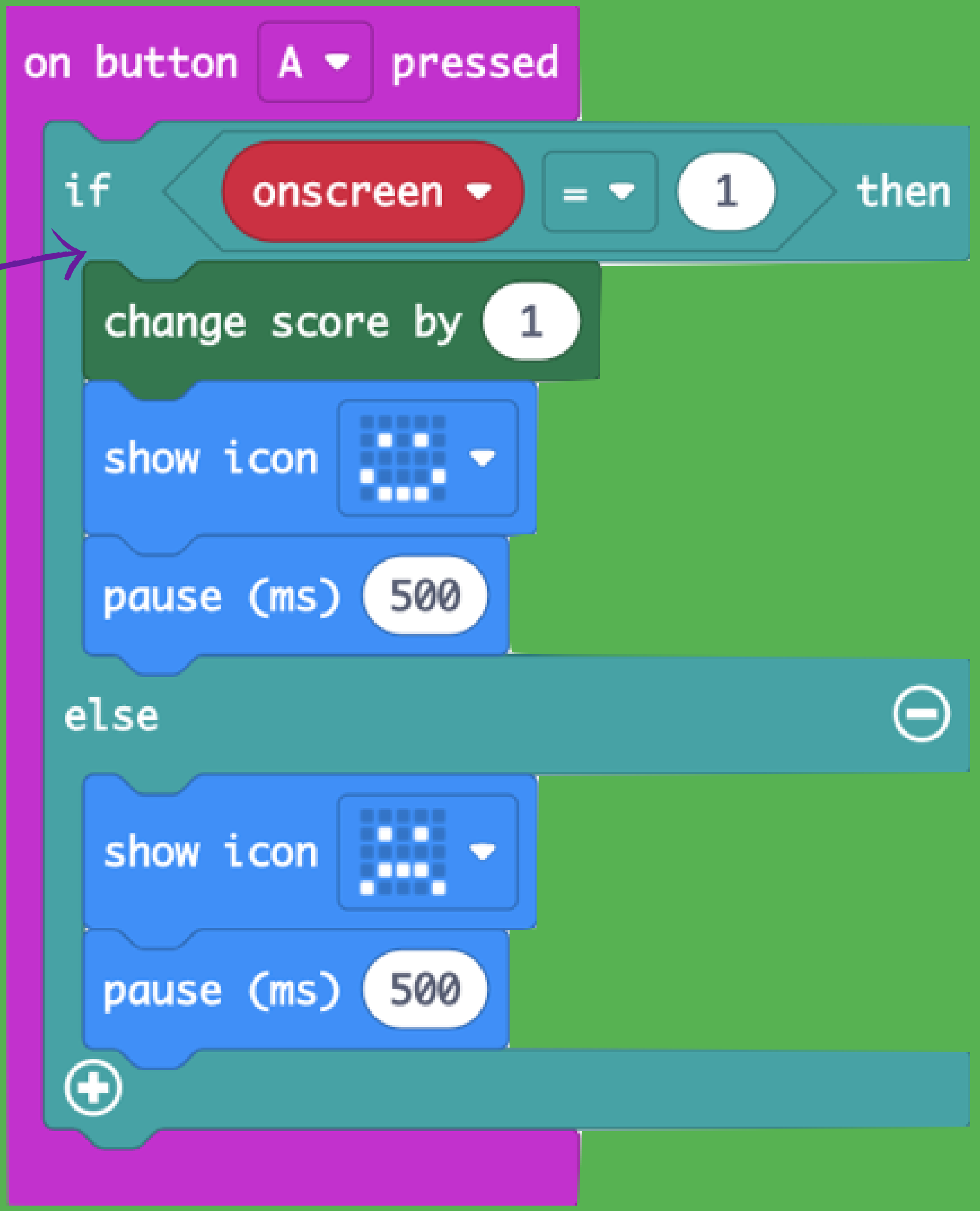
If you couldn't reach to your previous high score, you have to restart it again.



```
if score > highscore then
    set highscore to score
    show number highscore
    show icon
    pause (ms) 500
else
    show icon
    pause (ms) 500
    set score to score
    show number score
    set highscore to 0
```

Once you lose the game, the highscore will be set as 0; which means the game will start over again.

Onscreen is the block to sense your captures.



```
on button A pressed
    if onscreen = 1 then
        change score by 1
        show icon
        pause (ms) 500
    else
        show icon
        pause (ms) 500
```

Step 4:

When button A is pressed, your Micro:Bit will detect if you capture the space ship. If so, you'll score one; otherwise, you'll get a sad face.

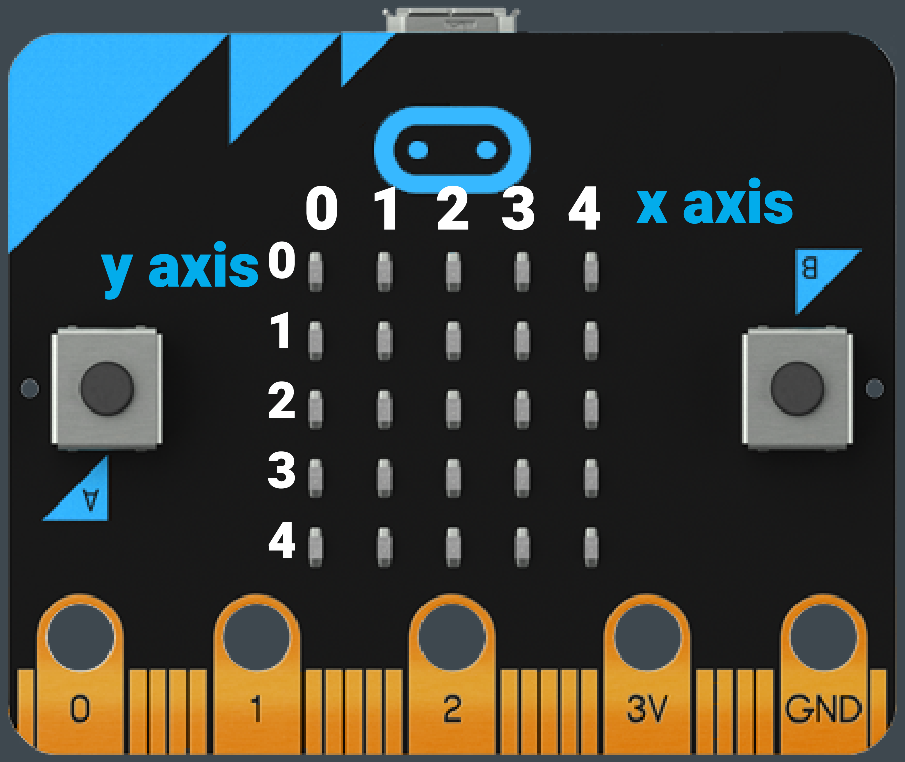
@CoraEdTech

Shooting Game

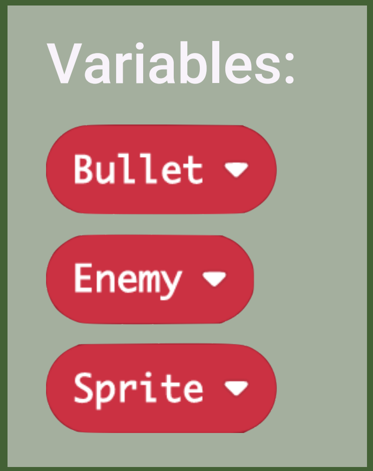
Use bullets to shoot your goals - let's see how many scores you can get!

Requirements:

Press buttons A, B to control the movement of your sprite. Press button A+B to shoot. The game last for 10 seconds. You'll see your final score once the game finishes.



x&y axis



Variables:

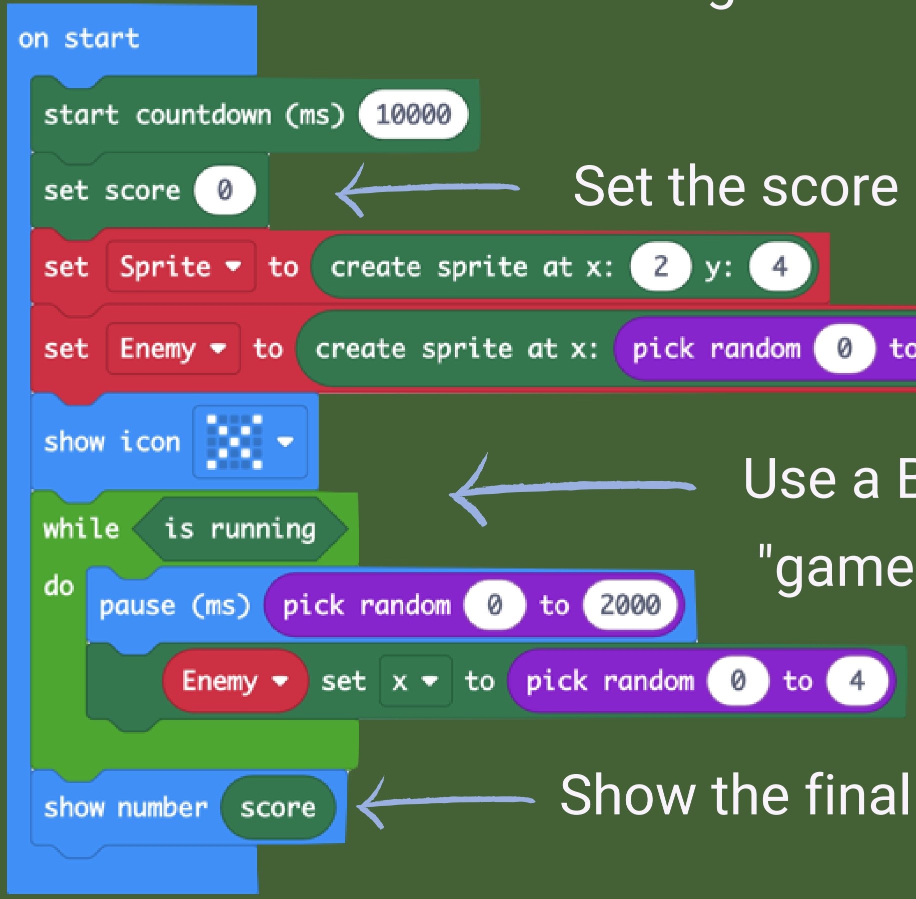
```
Bullet ▾
Enemy ▾
Sprite ▾
```

Step 1:

This is the setup of this game. The game lasts for 10 seconds. In the beginning, we put the sprite in the middle-bottom of LEDs.

Enemies always show up at the top of LEDs.

When the game starts, enemy shows up randomly.



```
on start
  start countdown (ms) 10000
  set score 0
  set Sprite ▾ to create sprite at x: 2 y: 4
  set Enemy ▾ to create sprite at x: pick random 0 to 4 y: 0
  show icon ▾
  while is running
  do
    pause (ms) pick random 0 to 2000
    Enemy ▾ set x ▾ to pick random 0 to 4
  show number score
```

Set the score as 0 in the beginning.

Use a Boolean condition as long as "game is running".

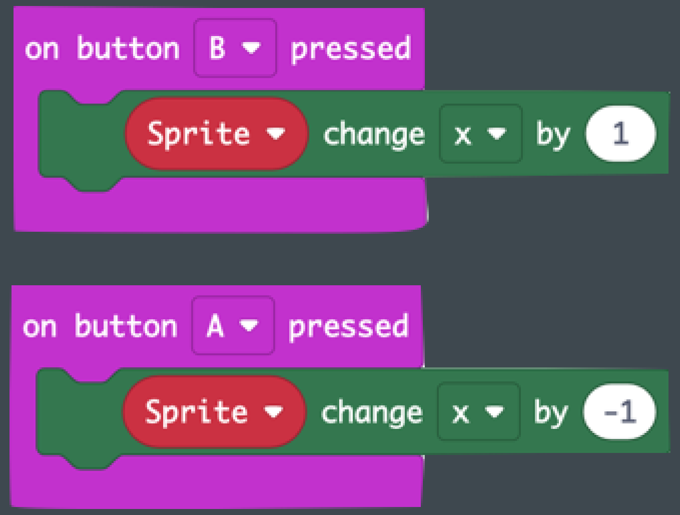
Show the final score once the game finishes.

@CoraEdTech

Shooting Game

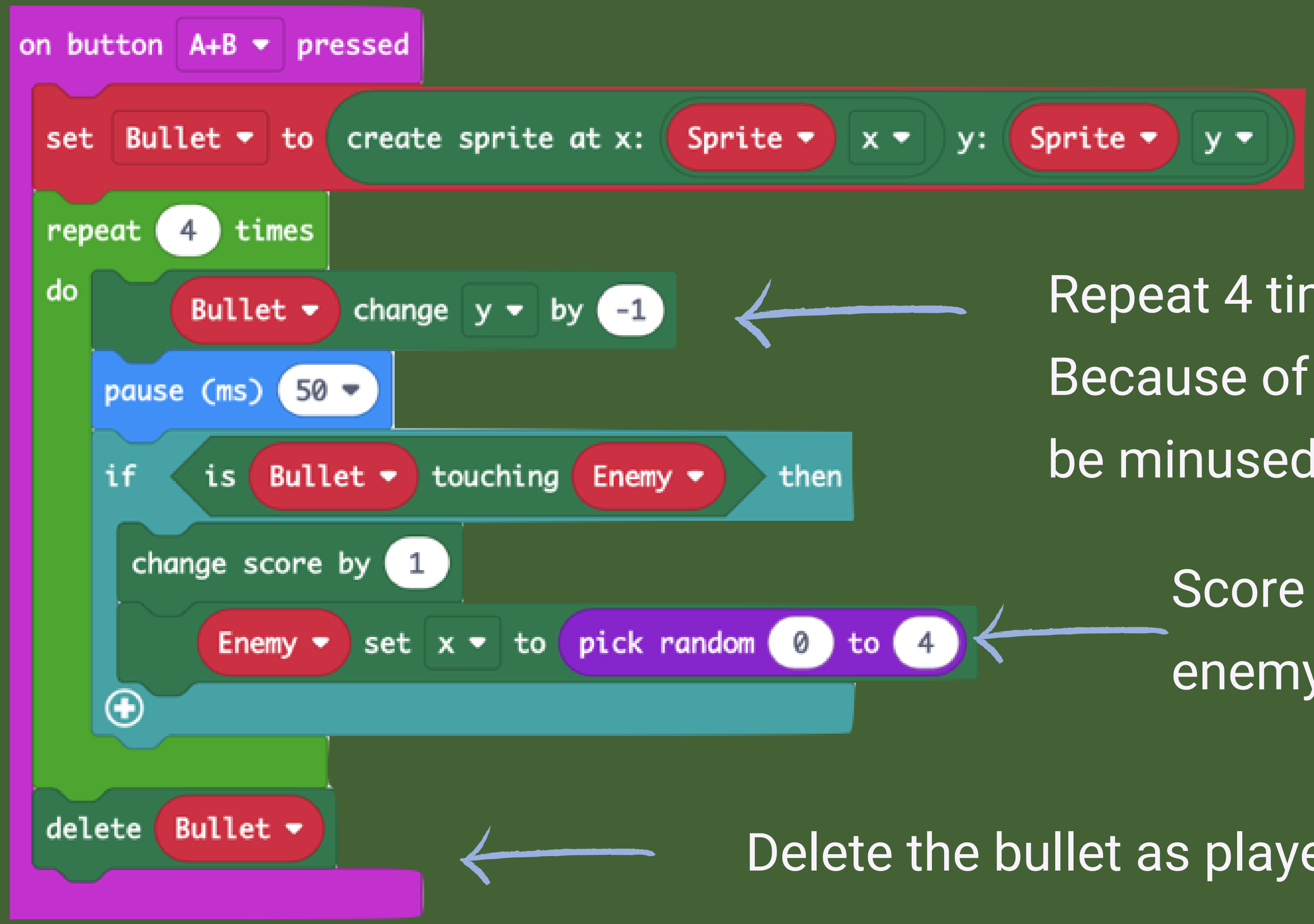
Use bullets to shoot your goals - let's see how many scores you can get!

Step 2:
Use buttons A and B to control the movement of sprite (it can be moved side to side).



```
on button B ▾ pressed
    Sprite ▾ change x ▾ by 1

on button A ▾ pressed
    Sprite ▾ change x ▾ by -1
```

Step 3:
When button A+B is pressed, the sprite will shoot bullets. As the y axis has 5 grids, it needs to move up 4 steps to shoot enemies. If the bullet touches enemy, players will score one.



```
on button A+B ▾ pressed
    set Bullet ▾ to create sprite at x: Sprite ▾ x ▾ y: Sprite ▾ y ▾
    repeat 4 times
    do
        Bullet ▾ change y ▾ by -1
        pause (ms) 50 ▾
        if is Bullet ▾ touching Enemy ▾ then
            change score by 1
            Enemy ▾ set x ▾ to pick random 0 to 4
        ⊕
    delete Bullet ▾
```

The bullet always appears as the same x axis like the sprite.

Repeat 4 times until the bullet reaches to the top of LEDs. Because of the y axis setting on Micro:Bit, the y needs to be minused when bullet moves up.

Score one if the bullet touches the enemy, also the enemy will move to another place randomly.

Delete the bullet as players can shoot a new one.

@CoraEdTech