

1. About the micro:bit dance robot

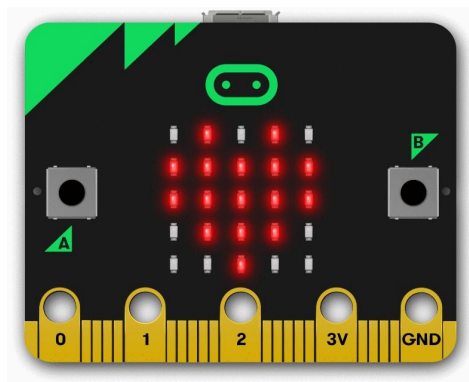
This is a cute programmable robot, you can control it with micro:bit. The body is assembled, so just plug the micro:bit and then play with it, which is great for you to choose as your entry-level robot. The robot comes with various sensors, such as ultrasonic sensor, touch sensor, sound sensor, buzzer, RGB LED and etc for you to learn about electronics, robot and programming at the same time. There are two ways to program the robot. You can use Microsoft Makecode, an easy way to learn programming. Or, you can use Micropython , which gives you python access to control the robot.



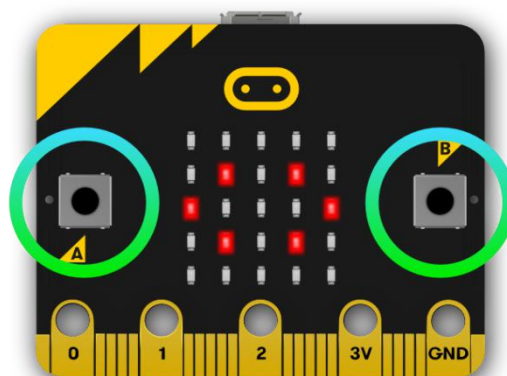
2. About micro:bit

The BBC micro:bit is a pocket-sized micro-computer that can be used for all sorts of cool creations, from robots to musical instruments, the possibilities are endless. It can be coded from any web browser in Blocks, Javascript, Python, Scratch, and etc.

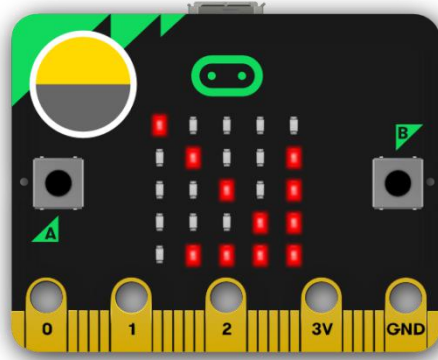
- The micro:bit has 25 individually-programmable LEDs, these 5x5 red LEDs forms a very small section of a screen, it displays information like words, numbers, and pictures through a combination of ON and OFF LEDs.



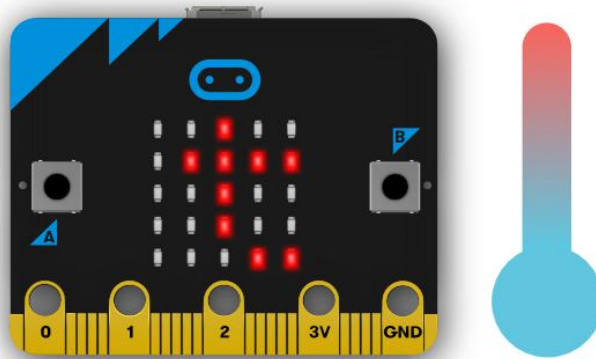
- There are two buttons on the front of the micro:bit - button A and button B, you can detect when these buttons are pressed, allowing you to trigger code on the device. You can detect events such as a single press, a double press, and a long press by programming.



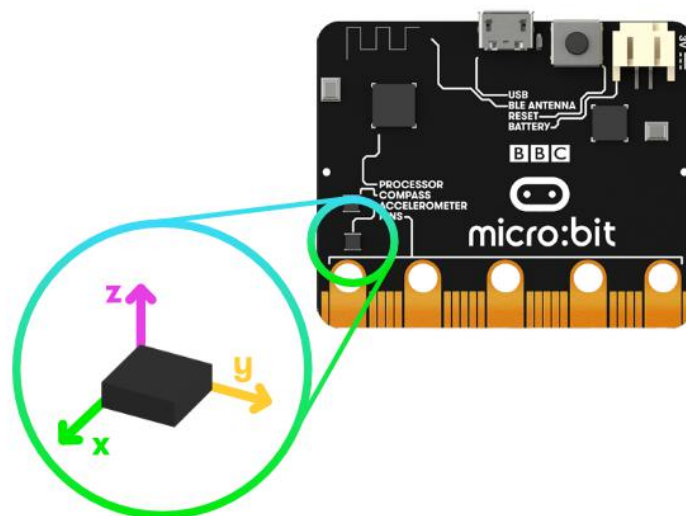
- The LED screen works as a basic light sensor by reversing the LEDs of the screen to become an input, which allows you to detect the ambient light.



- The temperature sensor transforms the temperature in the ambient environment into digital signals that can be read by the micro:bit for calculating the current temperature of the device.

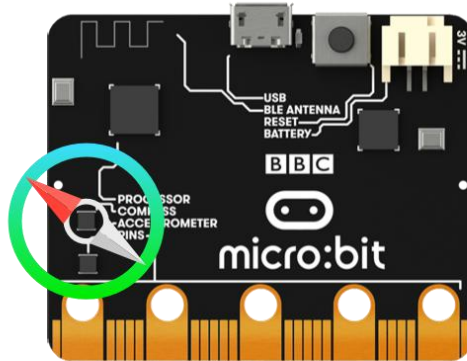


- The accelerometer measures the acceleration of your micro:bit, it can track motions like tilt, free fall and shake.

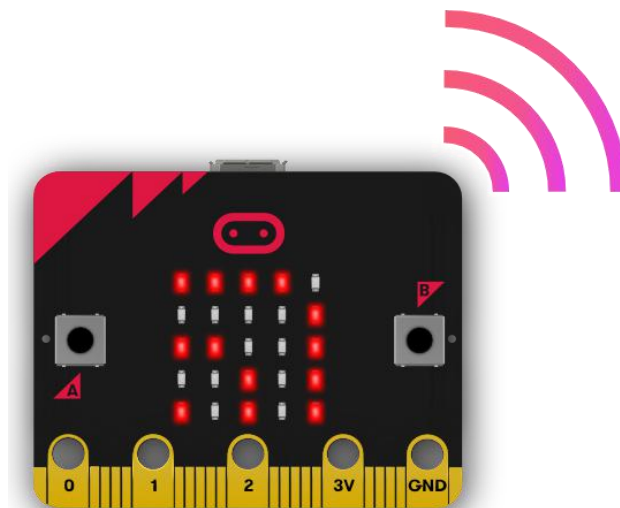


- The compass detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used. Calibrating the compass ensures the compass results are accurate. When the calibration

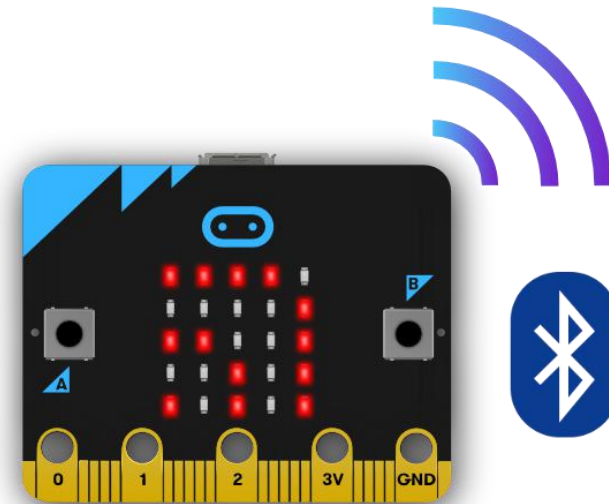
begins, the micro:bit will scroll the instruction “Tilt to fill screen”. You should tilt the micro:bit to move the dot in the centre of the screen around until you have filled up the whole screen to calibrate the compass.



- The radio feature allow you to communicate wirelessly between micro:bits.



- The Bluetooth Low Energy antenna allows micro:bit to send and receive Bluetooth signals, you can wirelessly communicate the micro:bit with phones, PCs and tablets. Before using the Bluetooth antenna you need to pair your micro:bit with another device.



The layout of these components on the micro:bit shows as below.

The BBC micro:bit

A pocket-sized, codeable ARM based computer designed for computer education. With a built-in compass, motion detection and Bluetooth technology. With an ambition to inspire digital creativity to develop a new generation of tech' pioneers in collaboration with the element14 Community.

PICK YOUR PROJECT

Fitness Tracker

Requires the use of:

- 8 Accelerometer
- 10 LED Array

Gaming

Requires the use of:

- 5 Button A
- 9 Button B
- 10 LED Array

Metal Detector

Requires the use of:

- 4 Magnetometer
- 10 LED Array

Door Bell / Alarm

Requires the use of:

- 3 Bluetooth
- 8 Accelerometer
- 10 LED Array

Robotics

Requires the use of:

- 5 Button A
- 6 Battery
- 9 Button B
- 3 Bluetooth
- 8 Accelerometer
- 7 Edge Connector

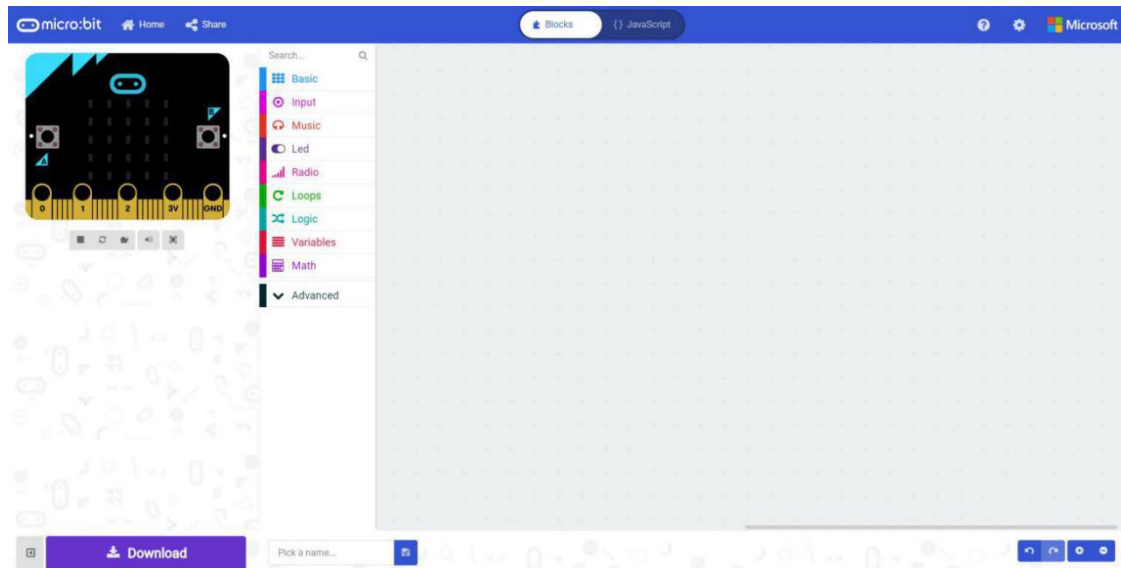
Quick start

Step 1: Connect it

Connect the micro:bit to your computer via a USB cable, then your micro:bit will show up on your computer as a drive called "MICROBIT".

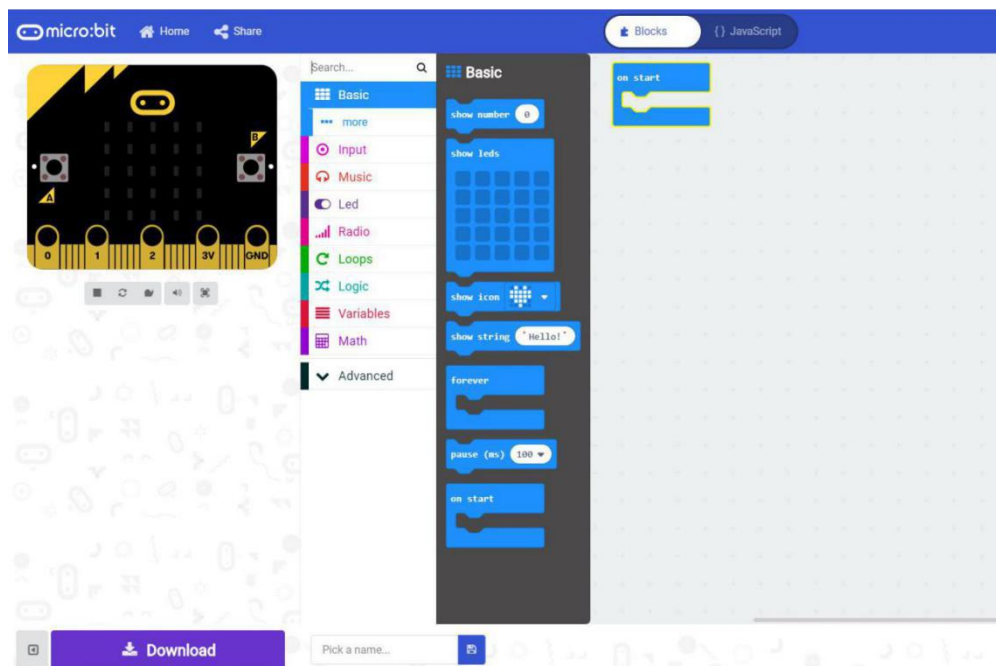
Step 2: Program it

Using [the official online editor](#), write your micro:bit code.

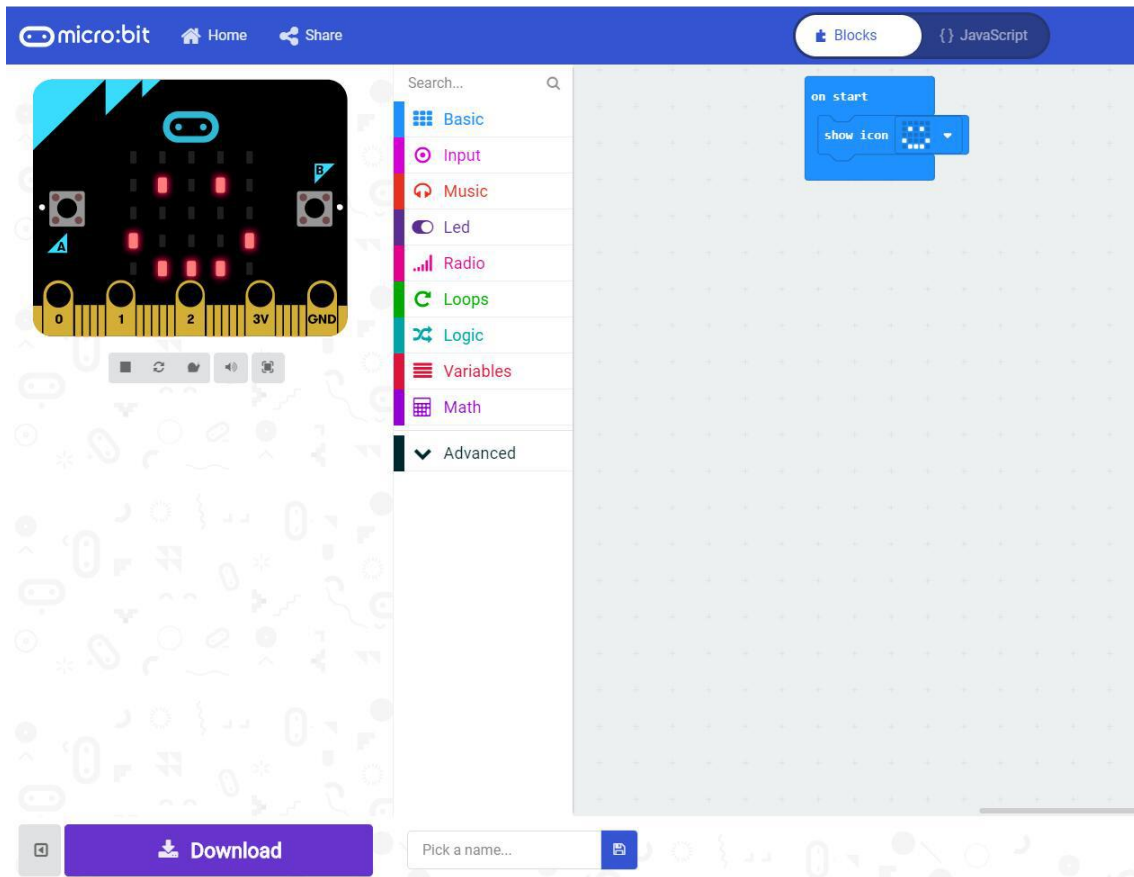


The left part is a simulator, it will simulate the work of micro:bit after finishing the program. The middle part is our command area, there are various program blocks we need. The rightmost area is the script area, you can drag and drop instructions into this area to form your program. Drag and drop some blocks and try your program on the Simulator in the Makecode Editor, the image below that shows how to program a smile face.

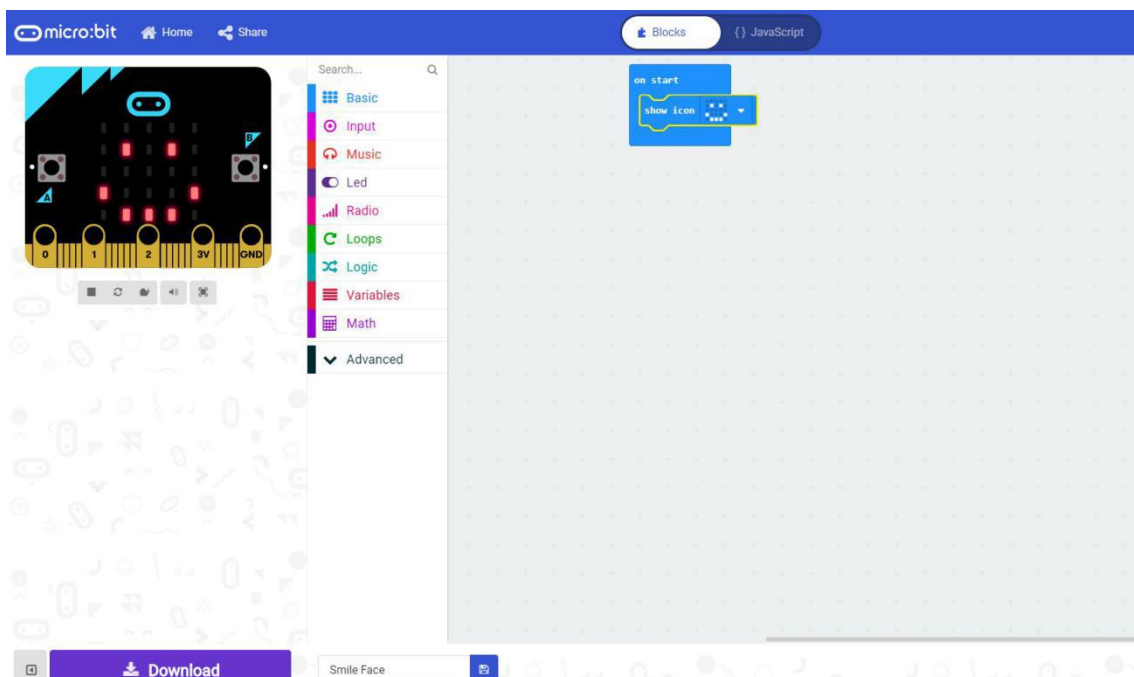
Step 1. Select “on start” block in the command area, and drag it out to place it in the script area.



Step 2. Drag out the “show icon” block to place it inside the “on start” block.



Step 3. Pick a name for your program, we name this project as “Smile Face”.



Step 4. Download the code

Click the Download button in the editor, this will download a “hex” file that micro:bit can read. Once the hex file has downloaded, copy it to your micro:bit like copying a file to a

USB drive.

Step 5. Play

The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is programmed, the code will run automatically once it's finished. Then look at your micro:bit, the screen will show a smile face.

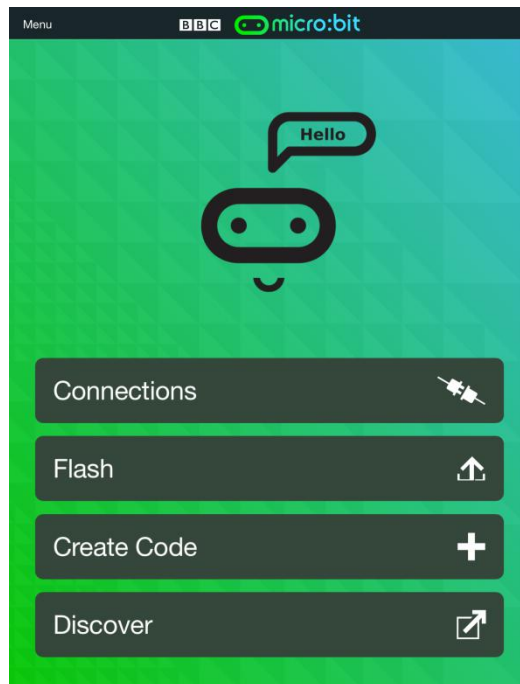
Note:

The micro:bit drive will automatically eject and come back each time you program it, but your hex file will be gone. It can only receive hex files and won't store anything else.

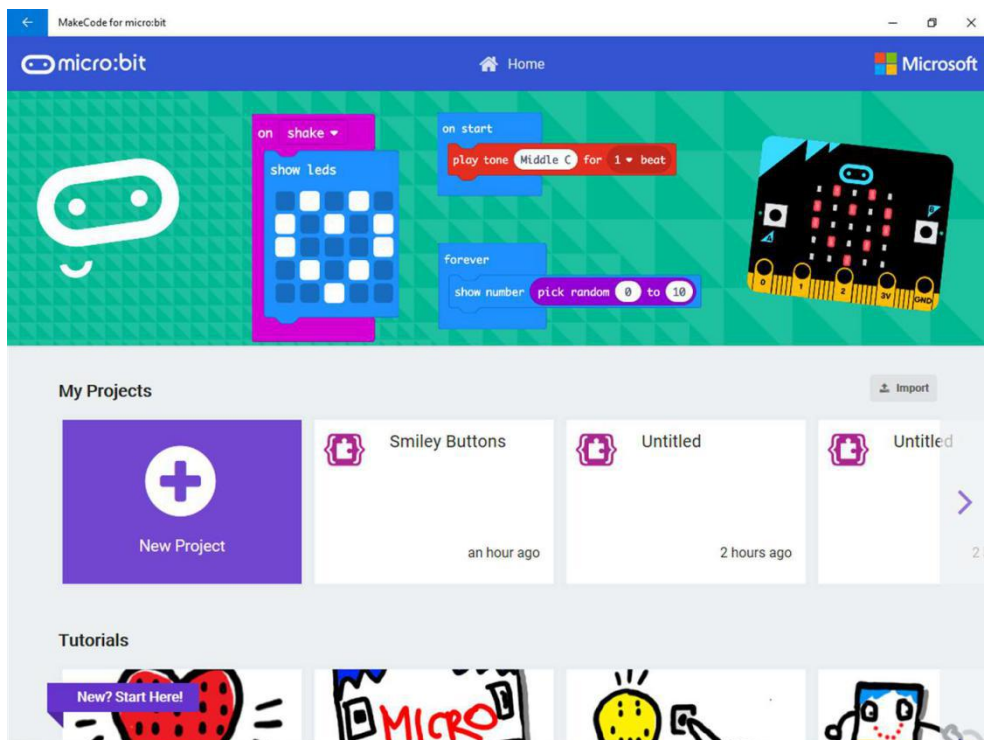
APP for Micro:bit

There is no wire needed if you use Bluetooth to send the code to your micro:bit from your mobile device. What you need to do is make sure your micro:bit is powered up and within easy reach of the phone or tablet running the App. There are two types of APPs available for micro:bit which based on Microsoft Makecode languages and Javascript, respectively micro:bit companion app and Windows 10 APP.

The micro:bit app is the essential companion to the BBC micro:bit, available for iOS and Android. Download the official micro:bit app for Android or iOS from the app store, the app contains handy instructions on how to pair your micro:bit with the app.



The Windows 10 APP has a few extra features over the editor at makecode.microbit.org that will work on micro:bits with up-to-date interface firmware: automatically program your micro:bit over USB, without needing to drag-and-drop the file onto the micro:bit drive; directly read serial data from your micro:bit for data logging and other fun experiments.

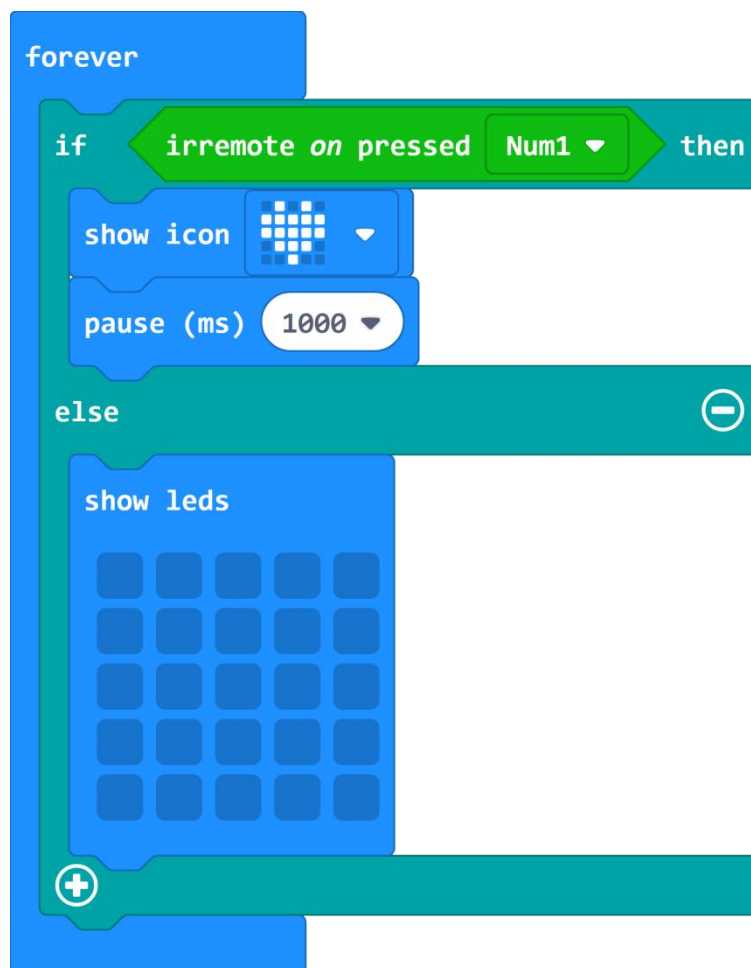


3. Start your journey here

Lesson 1. Show Your Love

Have you ever met a situation that you want to show your love to someone but you are too shy to say out? How about using this lovely robot to show your love remotely? Press the number 1 on the remote controller, then you will see a heart on the screen. How interesting, let's do it now!

Program the robot as follow



Tips

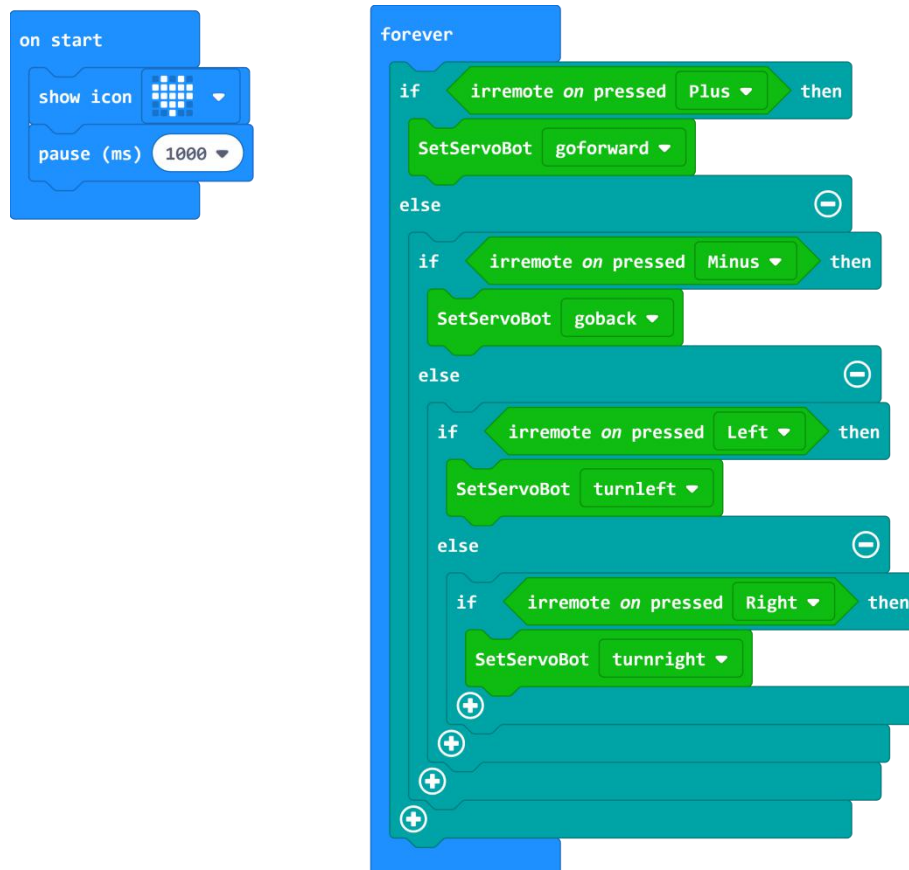
1. You can choose the number on the remote controller you like as a control number.
2. You can also show words, message you like on the screen.
3. First of all, please add the library of the robot to the MakeCode editor.

<https://github.com/Ceoi-Fung/LittleBlueMan>

Lesson 2. Obedient Robot

Let's make a cute and obedient robot, which will be very interesting. The robot can be controlled by the magical number, when you press the plus key, it will go forward. Press the minus key, it will go backward. It will turn left when you press the left key, turn right when you press the right key.

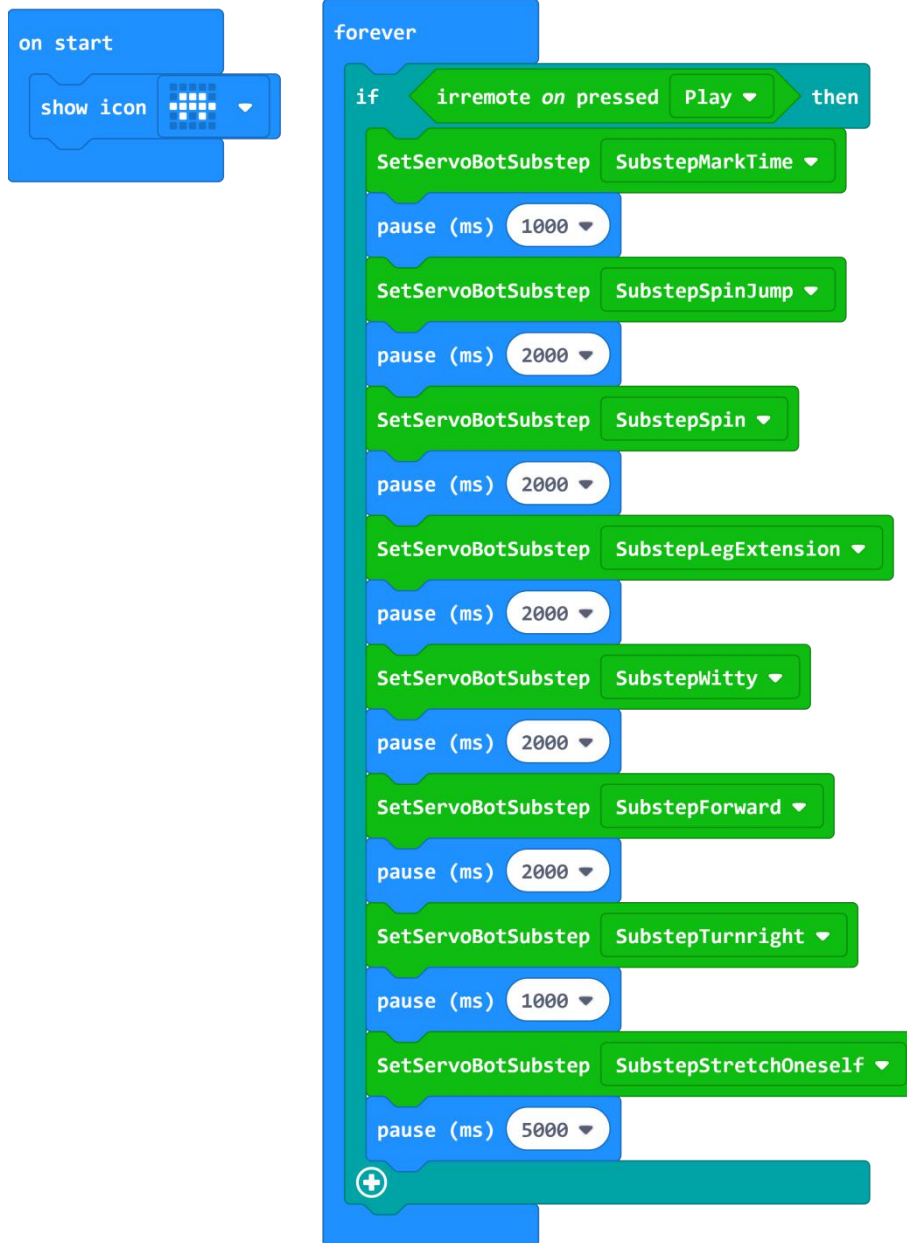
Program the robot as follow



Lesson 3. Dancing Robot

Our robot is a talented dancer, press the play key on the remote controller, it will show you how it dances.

Program the robot as follow



Tips

The robot can dance at least 9 kinds of dances, you can make it dance the way you like by programming.

Lesson 4. Dancing with Light

Our robot contains various sensors, there are two LEDs on the shoulder of the robot. How about let the robot dance with the bright light? Press the play key on the remote controller, the LED will be on and the robot will dance at the same time.

Program the robot as follow

```
on start
  show icon [LED Icon]
  pause (ms) 1000

forever
  if irremote on pressed Play then
    SetServoBotSubstep SubstepStretchOneself
    digital write pin P12 to 1
    pause (ms) 3500
    digital write pin P12 to 0
  else
    digital write pin P12 to 0
```

Lesson 5. Dancing with Music

The robot contains a buzzer, so you can let the robot dance with music. Press the play key on the remote controller, then enjoy the dance and music.

Program the robot as follow

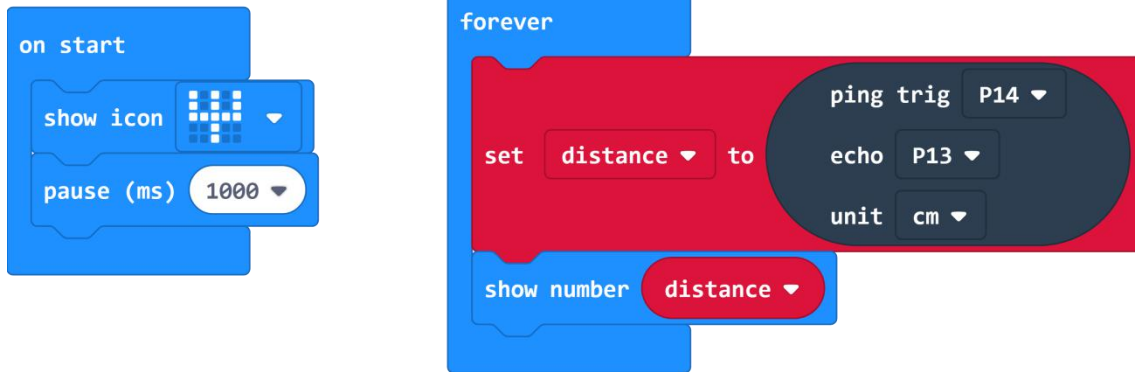
```
on start
  show icon [robot icon]
  pause (ms) 1000

forever
  if irremote on pressed Play then
    start melody dadadam repeating once
    SetServoBotSubstep SubstepWitty
    pause (ms) 1000
  if irremote on pressed Play then
    start melody entertainer repeating once
    SetServoBotSubstep SubstepLegExtension
```


Lesson 6. Ranging Robot

There is an ultrasonic sensor on the head of the robot, how about using the robot to measure the distance?

Program the robot as follow



Lesson 7. Following Robot

Sometimes, the robot needs a safe distance, if you are too close to it, it will step back until reaches its safety distance. When you are too far away from the robot, it will move close to you to remain a safe distance all the time.

Program the robot as follow

The image shows two blocks of code in a Scratch-like environment. The first block is an 'on start' block containing a 'show icon' block with a grid icon and a 'pause (ms)' block set to 1000. The second block is a 'forever' loop containing a 'ping trig' block with 'P14' as the trigger, 'P13' as the echo, and 'cm' as the unit. This is followed by an 'if' block where 'distance' is compared to '15'. If true, it runs 'SetServoBotSubstep' with 'SubstepForward'. If false, it runs another 'if' block where 'distance' is compared to '5'. If true, it runs 'SetServoBotSubstep' with 'stop'. If false, it runs a third 'if' block where 'distance' is compared to '0'. If true, it runs 'SetServoBotSubstep' with 'SubstepBack'. There are also three empty 'if' blocks at the bottom of the loop, each with a plus sign to indicate they can be added.

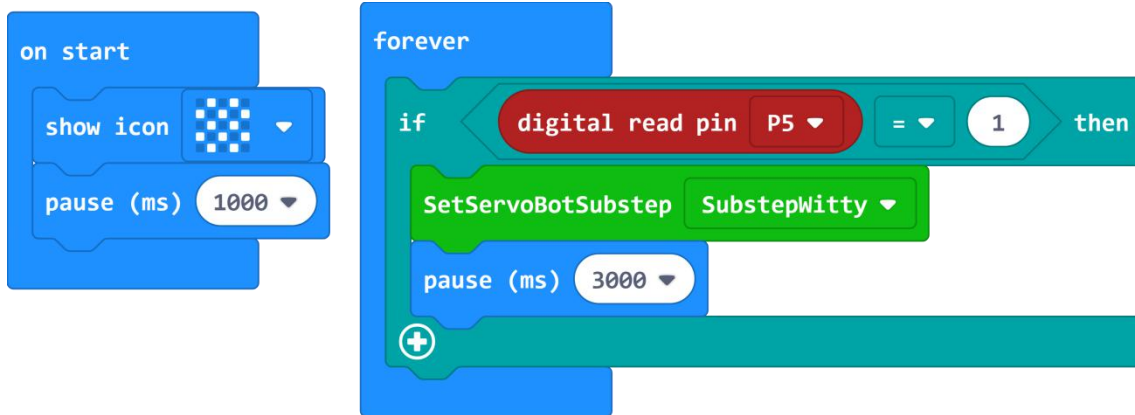
```
on start
  show icon [grid icon]
  pause (ms) 1000

forever
  ping trig P14
  echo P13
  unit cm
  if distance > 15 then
    SetServoBotSubstep SubstepForward
  else
    if distance > 5 then
      SetServoBotSubstep stop
    else
      if distance > 0 then
        SetServoBotSubstep SubstepBack
      if
      if
      if
```

Lesson 8. Naughty Robot

Touch the shoulder of the robot, it will dance naughty.

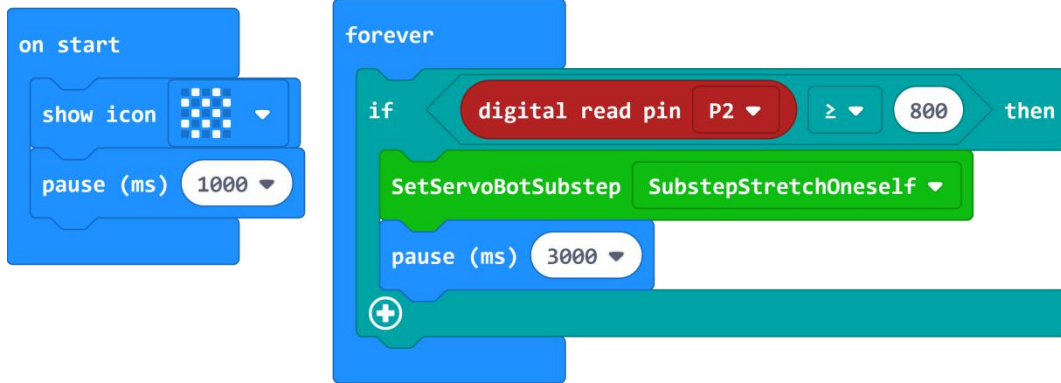
Program the robot as follow



Lesson 9. Listening Robot

There is a sound sensor on the robot, it can detect and react to the sound in the environment. Speaking to the robot, then see what will happen.

Program the robot as follow



Lesson 10. Don't Shake

The robot doesn't like shaking, if you shake it, it will show an unhappy expression and make a sound to express its unhappiness.

Program the robot as follow



Lesson 11. Obstacle Avoidance Robot

The robot is smart, it can recognize and avoid the obstacle. There are two IR reflective sensor on the two sides of the robot, it will turn right when the IR reflective sensor on the right side detects the obstacle. The robot will turn left when the IR reflective sensor on the left side detects the obstacle.

Program the robot as follow

The image shows a sequence of code blocks for programming a robot. It starts with an 'on start' block containing a 'show icon' block with a grid icon. This is followed by a 'forever' loop containing two conditional blocks. The first conditional block checks if 'digital read pin P8' is equal to '0'. If true, it sets 'SetServoBotSubstep' to 'SubstepTurnleft'. If false, it sets 'SetServoBotSubstep' to 'stop'. The second conditional block checks if 'digital read pin P15' is equal to '0'. If true, it sets 'SetServoBotSubstep' to 'SubstepTurnright'. If false, it sets 'SetServoBotSubstep' to 'stop'. The loop concludes with a 'pause (ms) 100' block.

```
on start
  show icon [grid icon]
  forever
    if (digital read pin P8 = 0) then
      SetServoBotSubstep SubstepTurnleft
    else
      SetServoBotSubstep stop
    if (digital read pin P15 = 0) then
      SetServoBotSubstep SubstepTurnright
    else
      SetServoBotSubstep stop
    pause (ms) 100
```