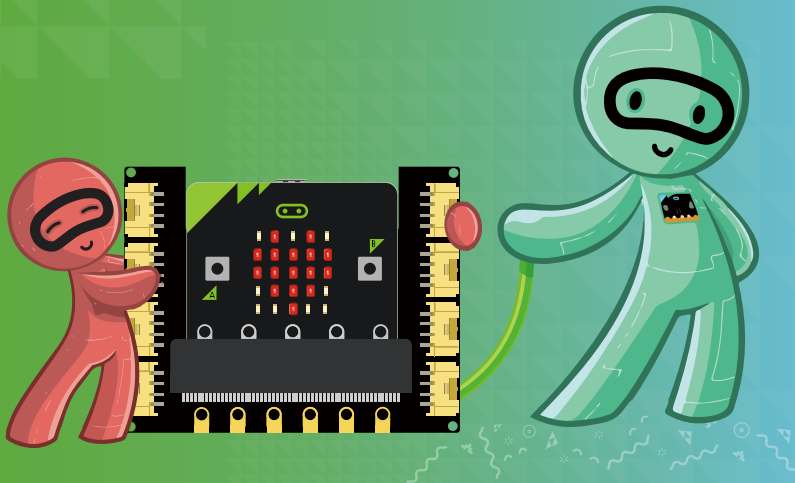


# Crowtail STEAM Edu Kit for Micro:bit

Project Quick Start Guide

*JUST FOR YOUR KIDS*



# Table of Contents

About this kit	1-4
Get to know micro:bit	5-12
Start Your Journey Here	13
1. Digital Eraser	13
2. Tilt Direction	14
3. Timekeeper	15
4. Dice Game	16
5. Telegraph	17
6. Pac-Man	18-20
7. Talented Singer	21
8. Polite Kid	22
9. Light Activated Alarm	23-24
10. Logic Block	25-26
11. Rainbow LED	27-29
12. Angry Girl	30-32
13. Unsocial boy	33-35
14. Remote Controller	36-39
15. Soil Moisture	40-41
16. Automatic Gate	42-44
17. Magnetic Field Detector	45-46
Keep Exploring	47



## About this kit

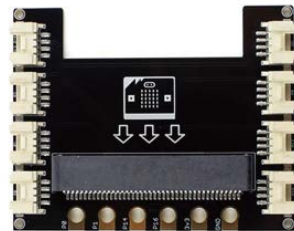
---

The **Crowtail STEAM Edu Kit for Micro:bit** brings numerous creative possibilities for you to dig out, bringing them from imagination to the real world. This kit includes **12 Crowtail modules** so you can learn microcontroller programming and build innovative objects or plan out unique projects. Compared with the traditional kit, this **Crowtail STEAM Edu Kit** is especially suitable for children. All of this Crowtail kit has a standard 4 pin connectors, so you can realize your creative ideas easily; just plug and play.

# Introduction to modules

## 1. Crowtail base shield for micro:bit

The Crowtail base shield is the core part of this kit, it acts as a bridge between Micro:bit board and Crowtail modules. This shield integrates **6 Crowtail connectors** and **2 Crowtail UART/IIC connectors** on board to allow you to plug Crowtail modules into the base shield to communicate with micro:bit. Apart from that, the shield also reserves some pinouts such as P0, P1 for you to use a crocodile clip to do some simple projects.



## 2. Crowtail touch sensor

The Crowtail touch sensor uses the touch IC TTP223-B to detect human touch by sensing the change of capacitance. It still works as a button, you can use it to replace the traditional push button.



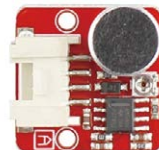
## 4. Crowtail ultrasonic sensor

The Crowtail ultrasonic sensor is a non-contact distance measurement module, it can be used to build projects that output different effects such as lights or sound based on changes in distance.



## 3. Crowtail sound sensor

The Crowtail sound sensor consists of a microphone, a LMV358 amplifier and a band pass filter to detect noises nearby and convert the sound waves to electrical signals. You can use it to visualize the sound level or as a sound trigger to control the circuit.



## 5. Crowtail buzzer

The Crowtail buzzer consists of a device that vibrates very quickly when electricity passes through it, the vibration generates a buzzing sound that moves through the air and into our ears. You can use the buzzer to make ringtones or play music.



## 6. Crowtail LED

This is a simple LED, just input power to light it up.



## 8. Crowtail vibration motor

The Crowtail vibration motor consists of a permanent magnet coreless DC motor, it vibrates when the input logic is HIGH, like the vibration of a phone.



## 10. Crowtail 9G servo

The Crowtail 9G servo comes with a Crowtail standard connector. Compared to a traditional servo, a major benefit is that you can plug into base shield without jumper wires.



## 7. Crowtail RGB LED

The Crowtail RGB LED string integrates 4 pcs of WS2812 on the board, so all LEDs can be controlled by only one signal pin. The RGB LED can form all colors, and you can control each LED separately to show different colors.



## 9. Crowtail moisture sensor

The Crowtail moisture sensor uses two probes to pass a current through the soil, then reads the resistance to determine the moisture level. It can be used in an automatic plant watering device to monitor if plants need watering.



## 11. Crowtail magnetic switch

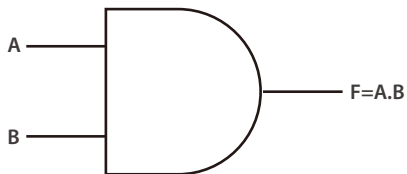
The Crowtail magnetic switch is based on encapsulated dry reed switch MKA14103, which is a double-ended type and may be actuated by a ruthenium contact, a permanent magnet or a combination of both. The magnetic switch could be a wonderful choice if you want to turn on and off a circuit based on magnetic proximity.



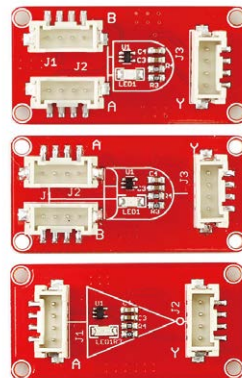
## 12. Logic block (AND/ OR/ NOT)

The Crowtail-Logic Block allows you to understand the logic gate by inserting different logic gates one by one and observing the progress of the input to the output!

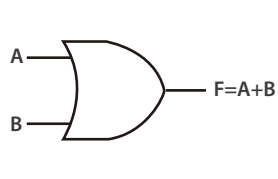
The following illustration and table show the circuit symbol and logic combination for an AND gate. If 0 is called "false" and 1 is called "true", the output is "true" when both inputs are "true". Otherwise, the output is "false".



Input		Output
A	B	$F=A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

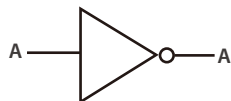


The output is "true" if either or both of the inputs are "true". If both inputs are "false", then the output is "false".



Input		Output
A	B	$F=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

The NOT gate only has one input, it reverses the logic state.

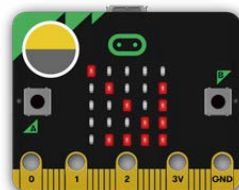
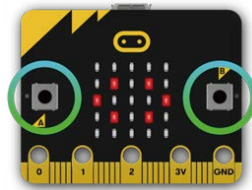
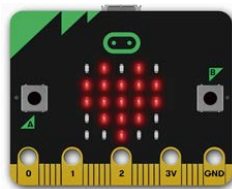


Input	Output
A	$A'$
0	1
1	0

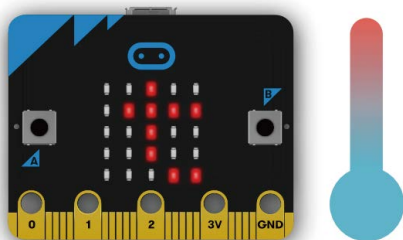
# Get to know micro:bit

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 form a very small section of a screen, and display 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.

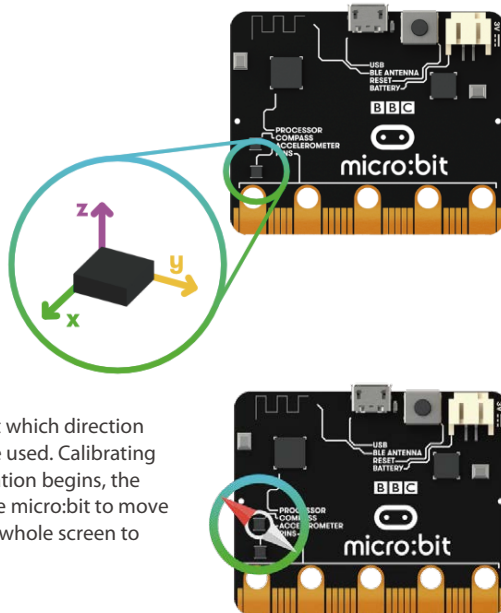


The **temperature sensor** transforms the temperature in the ambient environment into digital signals that can be read by the micro:bit for calculating the current temperature of the device.



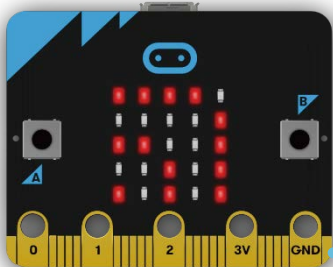
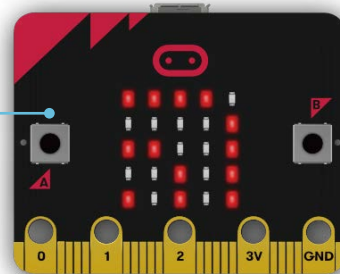
The **compass** detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used. Calibrating the compass ensures the compass results are accurate. When the calibration begins, the micro:bit will scroll the instruction "Tilt to fill screen". You should tilt the micro:bit to move the dot into the center of the screen around until you have filled up the whole screen to calibrate the compass.

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



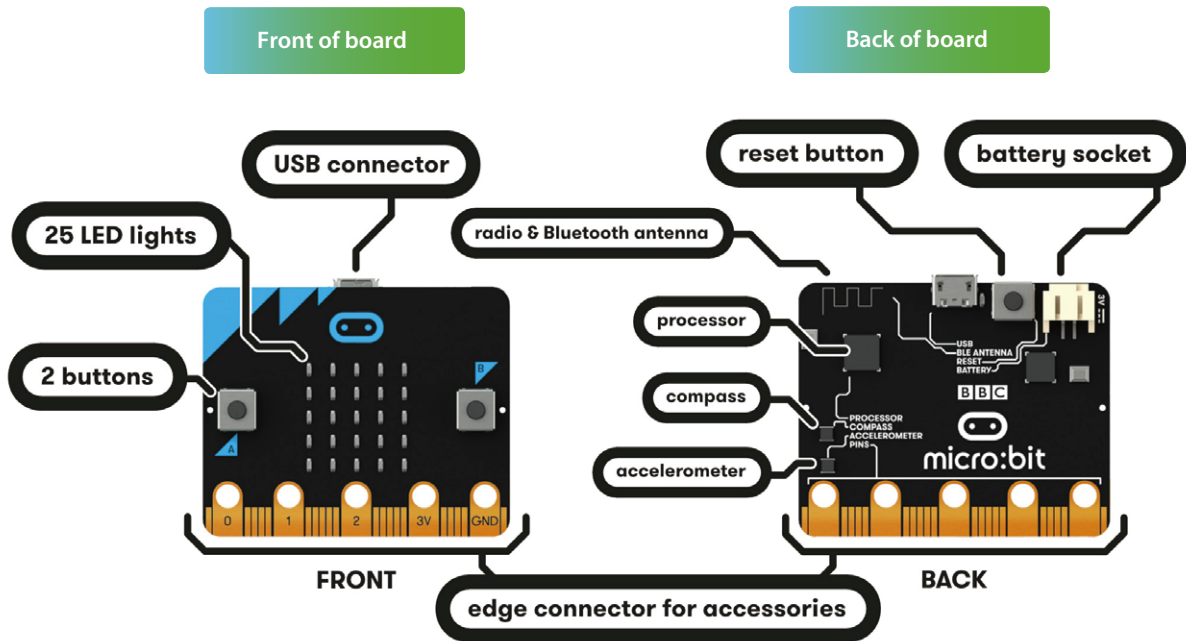


The **radio** feature allows you to communicate wirelessly between micro:bits.



The **Bluetooth Low Energy antenna** allows the micro:bit to send and receive Bluetooth signals, you can wirelessly communicate between micro:bit and phones, PCs, or 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 is shown below.



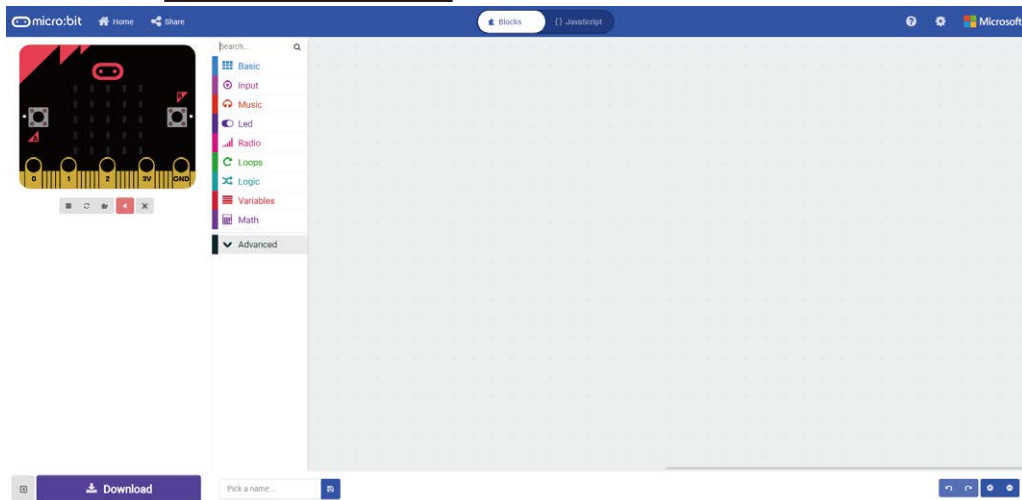
# Quick start

## 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" .

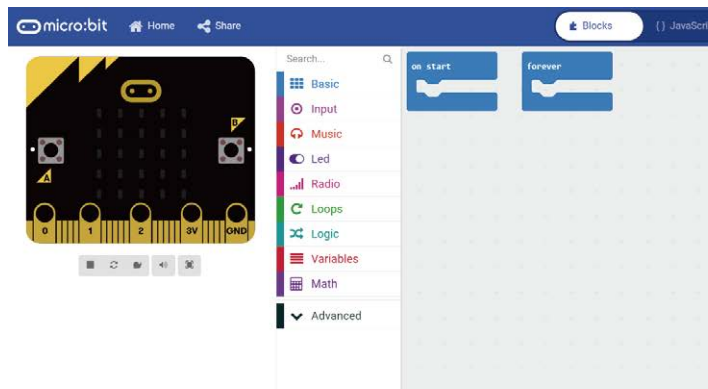
## Step 2. Program it

Using the official online editor <https://makecode.microbit.org/>, write your micro:bit code.

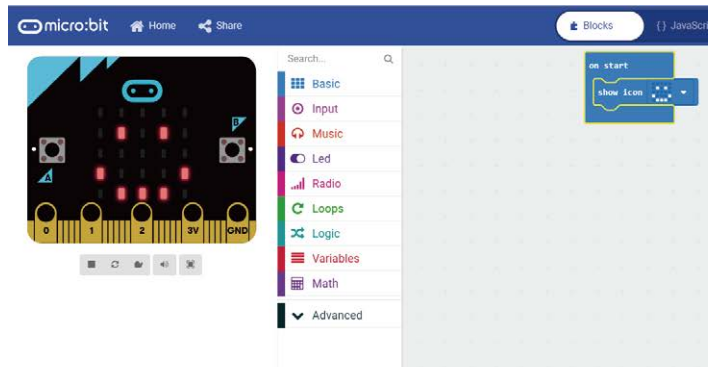


The left part is a simulator, it will simulate the work of micro:bit after finishing the program. The middle part is our command area, there are various program blocks we need. The rightmost area is the script area, you can drag and drop instructions into this area to form your program. Drag and drop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

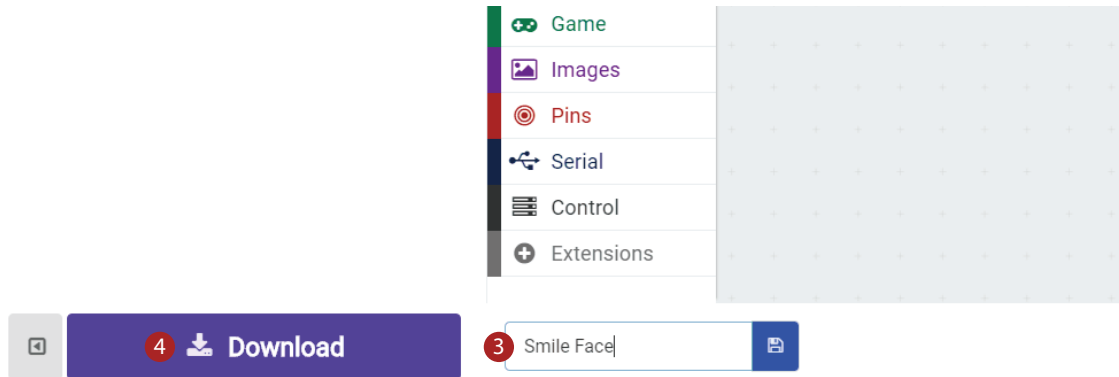
**Step 1.** Select “*on start*” block in the command area, and drag it out to place it in the script area.



**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 your 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 smiley face*.

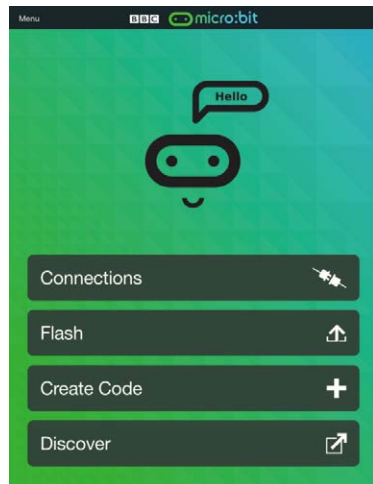
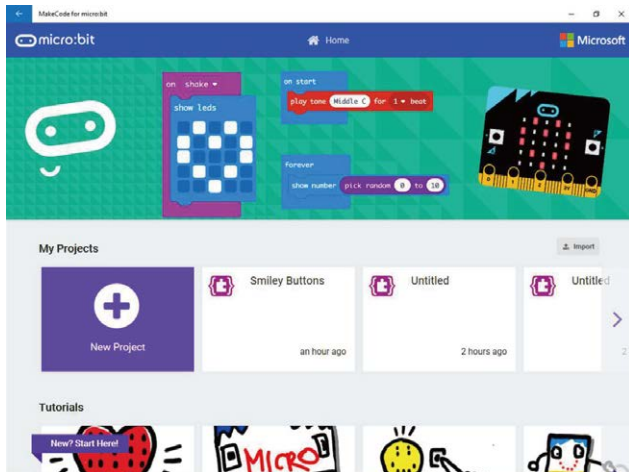
*Note:*

*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 are 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](https://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.

# Start Your Journey Here

## 1.Digital Eraser

Make a digital eraser to delete the content shown on the LED screen of micro:bit. Press the button A. The input contents will show on the LED screen, press the button B, all contents will disappear. Remember that you can input the content you like by changing "show icon" , "show number" and "show string" blocks.

### Required Parts:

- Micro:bit x 1

### Programming with Makecode

#### Steps

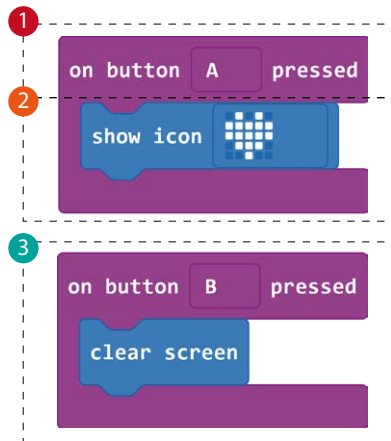
- 1 Place an '**on button A pressed**' block to run code when button A is pressed.
- 2 Place a '**show icon**' block inside on button A pressed to display a heart on the screen. Press the button A in the simulator to see the heart.
- 3 Add '**on button B pressed**' and '**clear screen**' blocks to clear the heart when we press the button B.
4. **Download** the program to micro:bit, start your journey to digital mastery.



Mission: **Master two buttons**

Difficulty Level: ★

Try to control the LED screen by pressing button A and button B at the same time.



## 2.Tilt Direction

Make a tilt direction to know which direction you are tilting the micro:bit.

### Required Parts:

- Micro:bit x 1

### Programming with Makecode

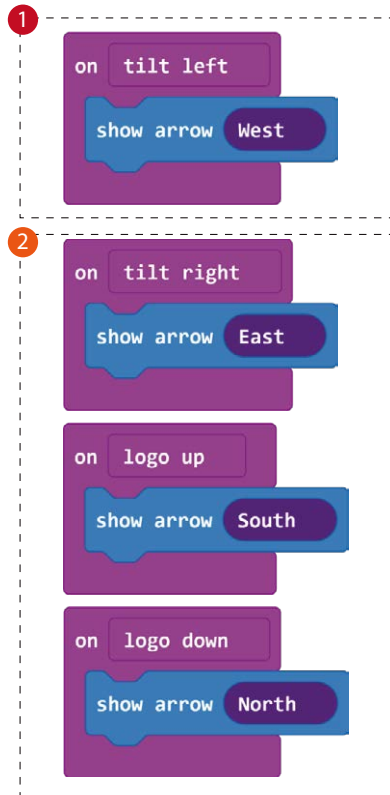
#### Steps

- 1 Start an *event handler*, this handler works when you tilt and lift the micro:bit.
- 2 Start event handlers like step 1, this handlers tasks for you.
3. **Download** the program, tilt your micro:bit to see the direction.

#### Mission: Measure the angle

Difficulty Level: ★★

The accelerometer senses when the micro:bit is moved, try to show the moved angle on LED screen.





### 3.Timekeeper

Make a stopwatch timer, let micro:bit count from 1 to 60 and count one number per second.

#### Required Parts:

- Micro:bit x 1

#### Programming with Makecode

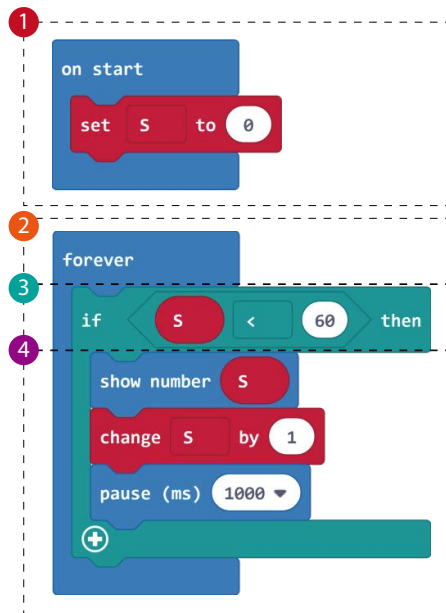
##### Steps

- 1 At first, *define a variable as "S"* . Then place the set s to 0 block inside the on start block to initialization variable.
- 2 Put the *'if logic' block* into the *'forever' block*.
- 3 Set the logic judgment index to *"S<60"* .
- 4 If S is less than 60, then it will change 1 by one second, like on a clock.
5. **Download** the program to your micro:bit, see how many words you can write in 60 seconds!

Mission: **Make a watch**

Difficulty Level: ★★★

A day is 24 hours, an hour is 60 minutes, one minute is 60 seconds. After knowing the system of the clock, let's make a watch to show the current time.



## 4.Dice Game

Create a digital dice on micro:bit, and play games with your friends.

### Required Parts:

- Micro:bit x 1

### Programming with Makecode

#### Steps

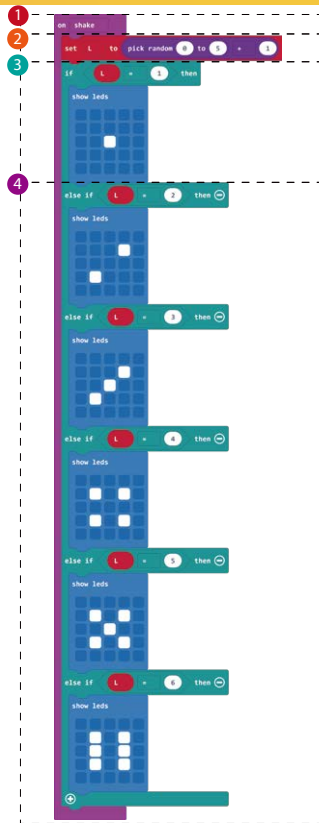
- 1 Add an '**on shake**' block to run code when you shake the micro:bit.
- 2 Add an **L variable** and place the set L to block in the shake event. Then Add a '**pick random**' block to pick a random number from 1 to 4 and store it in the variable named L.
- 3 Place an '**if**' block under the pick random and check whether L is equal to 1. Add a '**show ledLEDs**' block to show the number 1.
- 4 Click the **+** button to add an else section, add block inside the else to show number 2.3.4.5.6
5. Click on the **SHAKE button** in the simulator, you will see the numbers on the screen if you try enough times. Then download the program to your micro:bit, and play games with your friends!



#### Mission: Compete in spelling

Difficulty Level: ★★★

Change the number to 26 letters, the letter will come out randomly after shaking it. Taking it as the first letter, let's see who write the most words in one minute.



## 5.Telegraph

Use the radio to send messages to your friends, just like a telegraph can send messages.

### Required Parts:

- Micro:bit x 2

### Programming with Makecode

#### • *Sending Hello*

- 1 On start, *add the radio set group block* to set the group ID for radio communication.
- 2 *Add blocks* to *send a message* when button A is pressed.
3. Download the program to your micro:bit that will be used for sending.

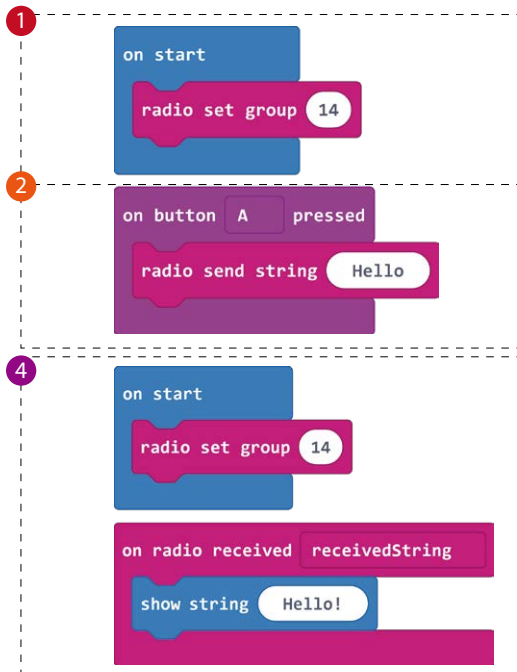
#### • *Receiving Hello*

- 4 Add an '*on receivedString*' block that will run code whenever a new message comes in. The receivedString variable contains the string that we sent.
5. **Download** the program to another micro:bit that will be used for receiving, then try to send "Hello" between your micro:bits.

#### Mission: *Express your love*

Difficulty Level: ★★★★★

Give your Mom a piece of micro:bit, and send a ❤️ to show how much you love her.



## 6.Pac-Man

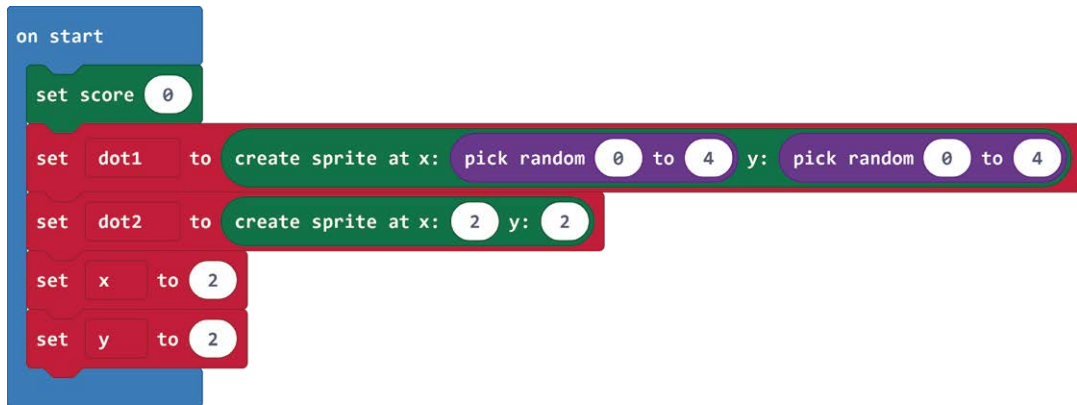
Let's play the Pac-Man game on micro:bit. After eating the beans, your score will increase by one. Remember that once you move out of the range of the LED screen, the game will be over, and the screen will show your score.

### Required Parts:

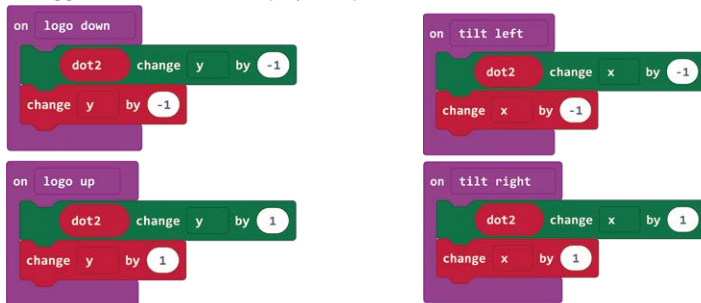
- Micro:bit x 1

### Programming with Makecode

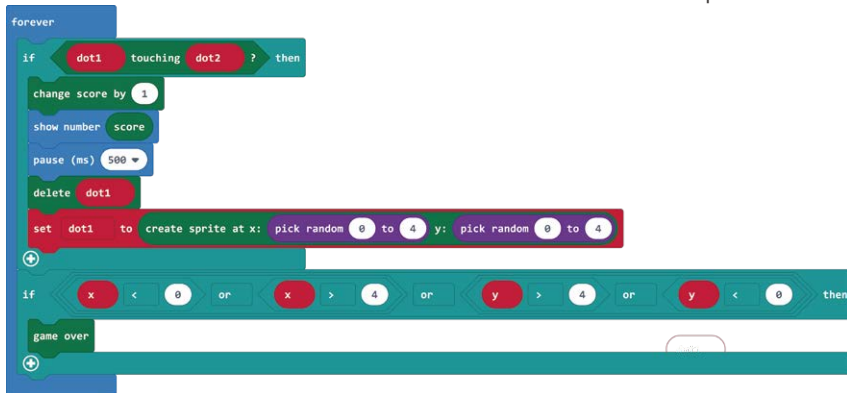
**Step 1.** Firstly, **set the score**, create the variable **dot1** to store the position of the bean; create variable **dot2** to store the position of the player; create the variable **x, y** to record the player's position. The bean appears at random locations on the screen, and the player's **initial position is (2,2)**.



**Step 2.** Use the 'event trigger' block to move the player's position.



**Step 3.** Add an 'if logic' block inside the forever block to check if the player has eaten the bean. If you eat it, the score will increase by 1 and show the score on the screen. Then it will delete the bean and add a new bean at a random position.



**Step 4.** Throughout the whole process, the player's position will be monitored all the time. The game will be over if the position is *out of the range of the screen*.

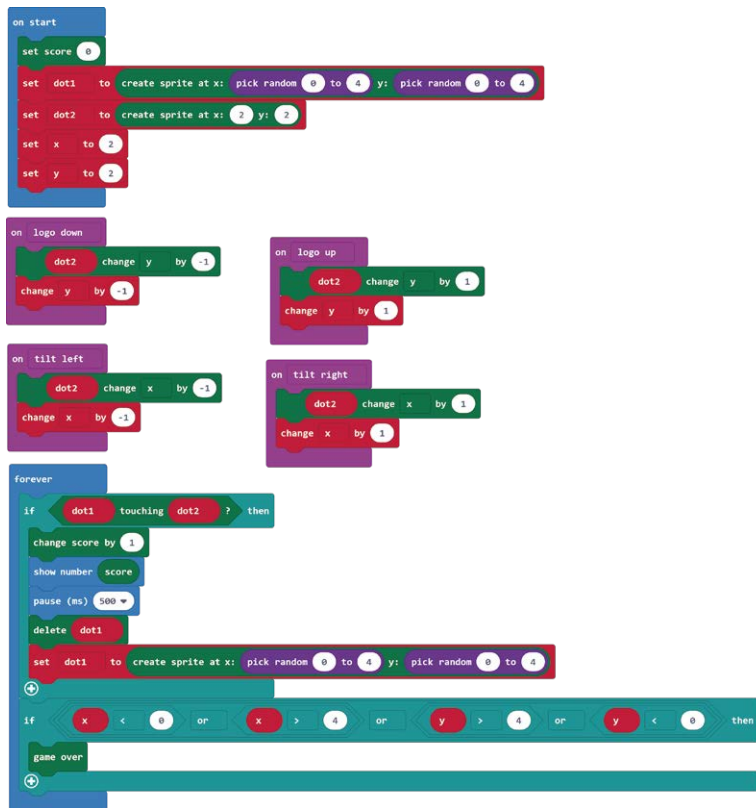
**Step 5. Download** the program, enjoy the game!



### Mission: Catch the bean

Difficulty Level: ★★★★★

Try to make a bean catcher game, the principle is the same as the Pac-Man. Remember to place the bean catcher at the bottom of the screen, the beans fall vertically from the top. The game will be over if you don't catch the bean.



## 7.Talented Singer

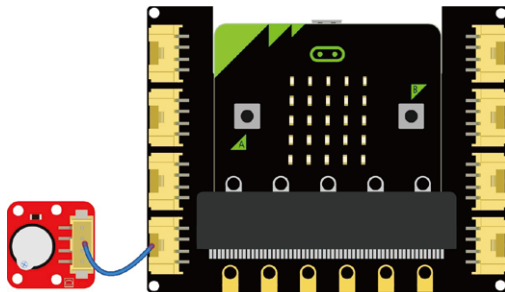
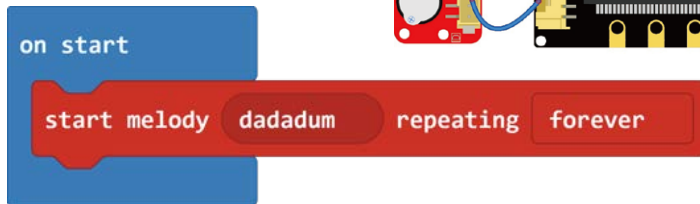
Let's play music on micro:bit, there are about 20 songs in the music library, use the buzzer to bring out the music.

### Required Parts:

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail buzzer x 1

### Programming with Makecode

Step 1. Add a 'start melody' block inside the on start block.



Step 2. **Download** the program to the micro:bit, then enjoy the music.



### Mission: Write a rhythm

Difficulty Level: ★

Use your imagination to create a unique song, you can change the play tone, beat, tempo reset time, etc, on the "Music" block.

## 8. Polite Kid

When you say “Hi” to micro:bit, it will respond with “Hello” .

### Required Parts:

- Micro:bit x 1
- Crowtail sound sensor x 1
- Crowtail base shield for micro:bit x 1

### Programming with Makecode

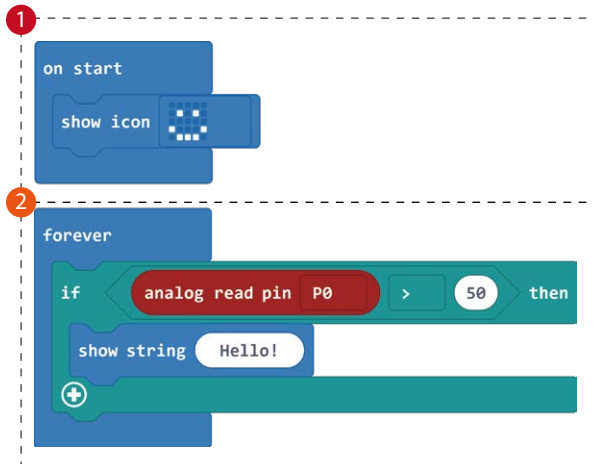
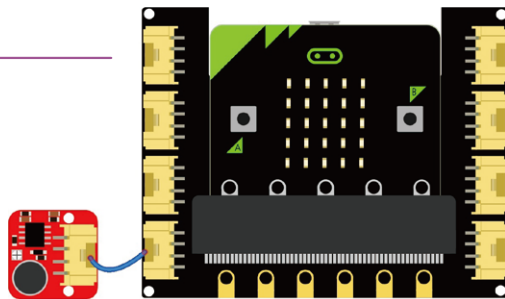
#### Steps

- 1 **Show a smile face** on the screen before you talk to your micro:bit.
- 2 Add an **'if block'** to determine if the value of the sound is greater than 50, and if so, display 'hello' on the screen
3. **Download** the program, then say “Hi” to your micro:bit!

Mission: **Detect voice level**

Difficulty Level: ★★

Use the sound sensor to measure your voice, the louder it is, the more LEDs will light up on your micro:bit.



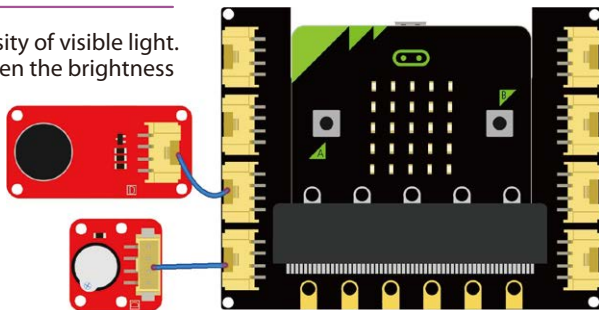


## 9.Light Activated Alarm

The micro:bit has a light sensor, it can measure the intensity of visible light. Let' s make a micro:bit alarm, it will beep and vibrate when the brightness is greater than the set value.

### Required Parts:

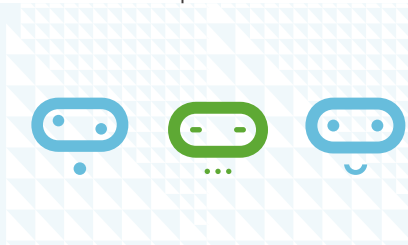
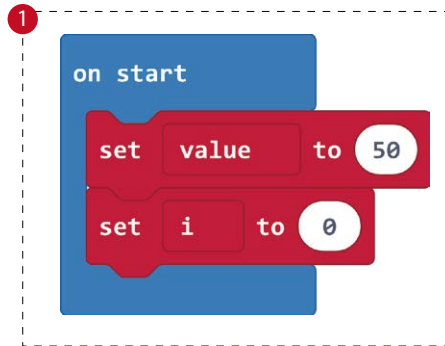
- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail buzzer x 1
- Crowtail vibration motor



### Programming with Makecode

#### Steps

1 On start, **create a variable "value"** to store the light value, and set the boundary of light level to 50. **Create a variable "i"** to store the number of times the button A has been pressed.



2 Create a variable "read" to store the current light level. Then add an *if logic block* to determine when the alarm clock rings.

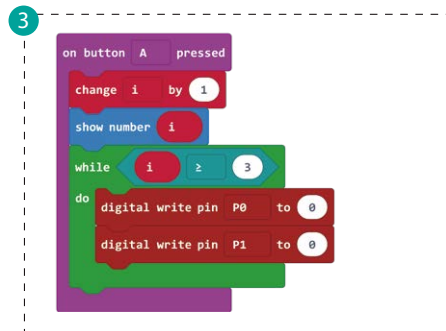
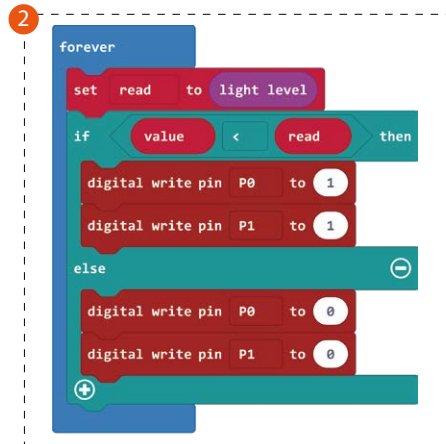
3 The clock will always ring unless you *press the button A* for a specified number of times.

4. **Download** the program to your micro:bit, put the micro:bit in a bright place to see what will happen.



Mission: **Build an alarm clock** Difficulty Level: ★★★★★

Make an alarm clock to wake you up at the time you set.



## 10.Logic Block

This is a fantastic project. Here the input is the same, but the output is different.

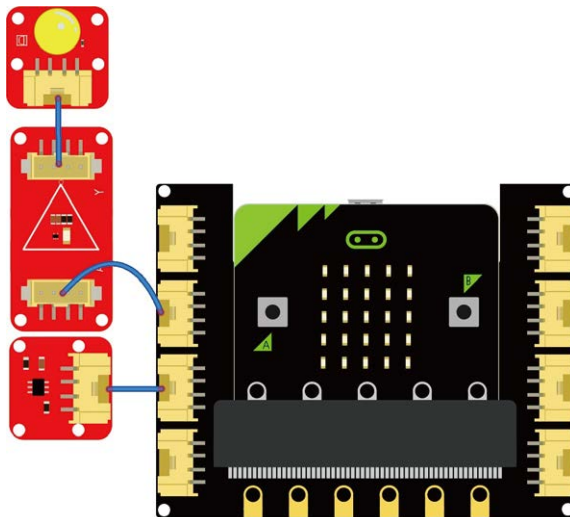
### Required Parts:

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail touch sensor x 1
- Crowtail logic OR block x 1
- Crowtail logic NOT block x 1
- Crowtail LED x 1

### Programming with Makecode

#### Steps

- 1 Add the **'if logic' block** to decide input digital voltage.



2. **Download** the program to your micro:bit, then connect the logic OR block and the NOT block to your micro:bit separately. *Press the touch sensor* to see what will happen. The program and circuit are suitable for NOT and OR gates. Although the OR gate has two inputs, we just use one input to control the output. Try to press the touch sensor, you will see different phenomena!

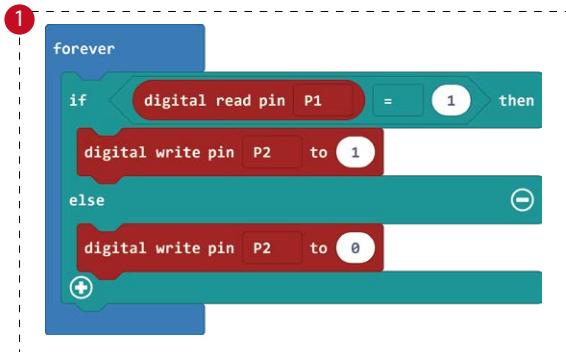
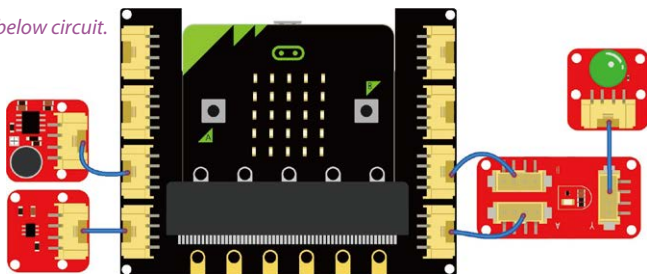
Mission: **Wake him up**

Difficulty Level: ★★★

The boy falls into sleep so deep in the class, you need to shake him and call his name to wake him up. How do you simulate this situation with your micro:bit?

After learning the logic block, we know that the AND gate has two inputs. Connect the touch sensor and sound sensor to the input, only if you press the touch sensor and talk to the sound sensor at the same time will you “wake him up” .

*You can build your circuit following the below circuit.*

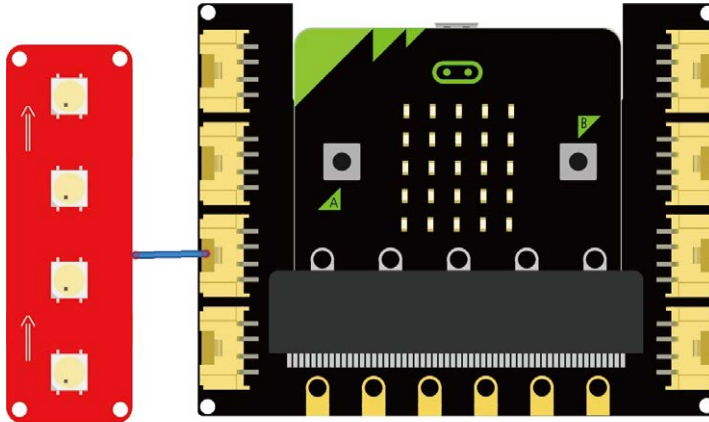


## 11.Rainbow LED

Use the micro:bit to drive the Crowtail RGB LED to make a rainbow LED.

### ***Required Parts:***

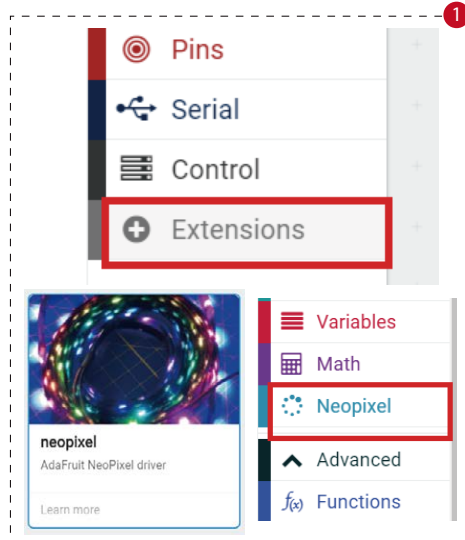
- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail buzzer x 1
- Crowtail RGB LED x 1



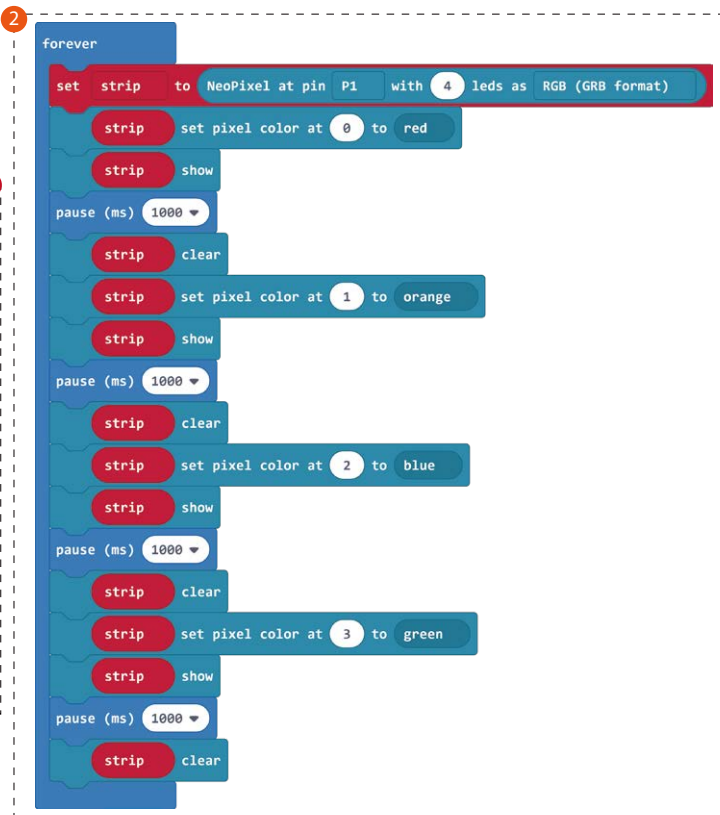
## Programming with Makecode

### Steps

- 1 Click on the **Extensions** to add the “Neopixel” library to the command area.



- 2 In the forever loop, **determine the quantity and color** of RGB LED that we will light.



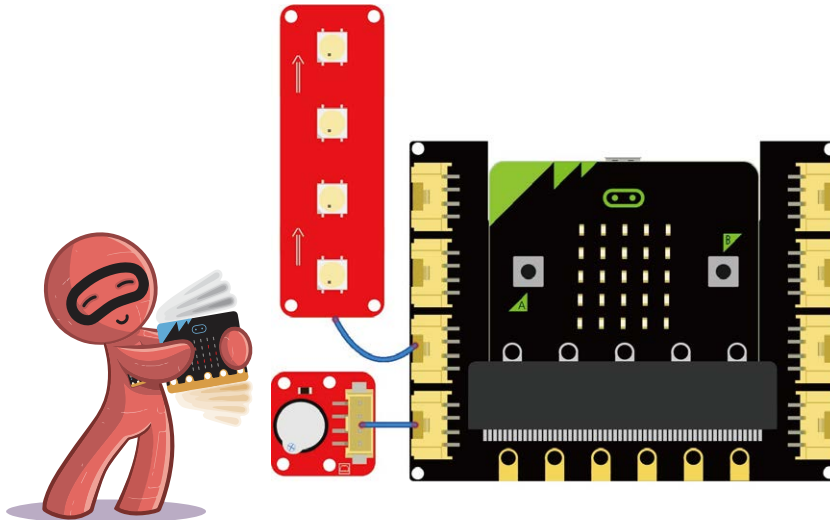
## Mission: Celebrate a birthday

Difficulty Level: ★★★

Use the RGB LED and buzzer to celebrate a birthday for your friends. Light up the RGB LED and play a "happy birthday" song at the same time.



*You can build your circuit following the below circuit.*

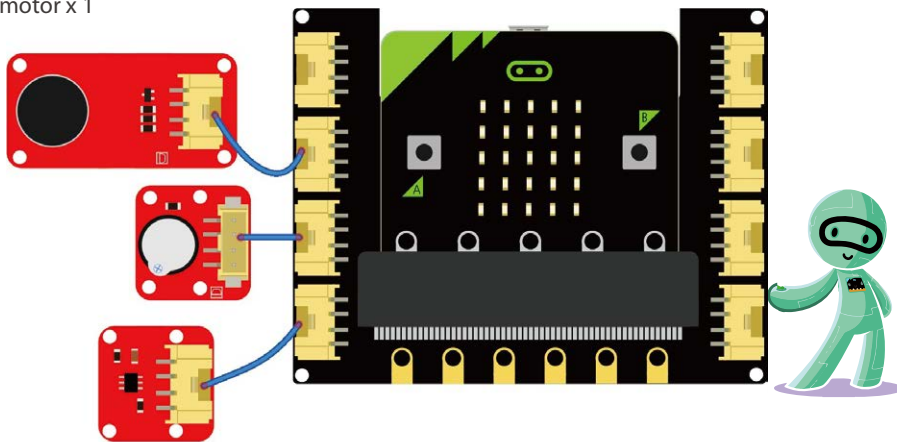


## 12. Angry Girl

Use micro:bit, touch sensor, buzzer and vibration motor to simulate an angry girl. When you touch the touch sensor, it will shout and jump to show its anger.

### **Required Parts:**

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail touch sensor x 1
- Crowtail buzzer x 1
- Crowtail vibration motor x 1

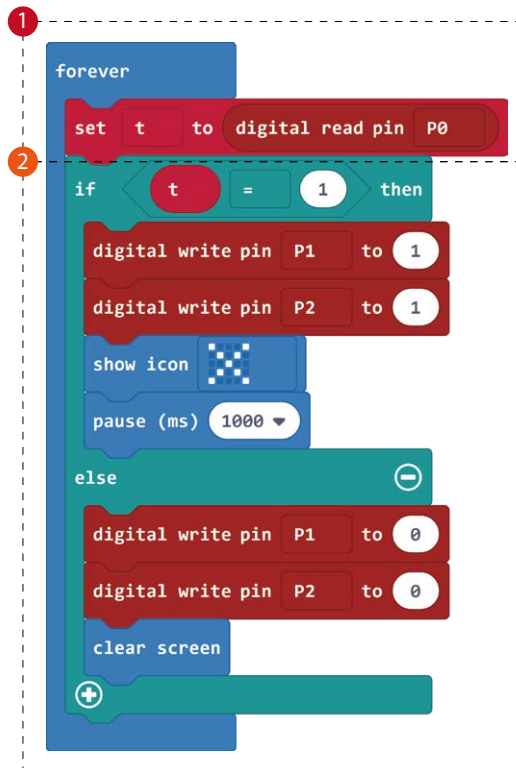
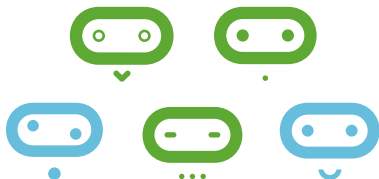




## Programming with Makecode

### Steps

- 1 **Create a variable** to store the value detected by touch sensor.
- 2 Add an **if logic block** to determine when the micro:bit would shout.
3. **Download** the program to micro:bit, press the touch sensor to see what will happen.



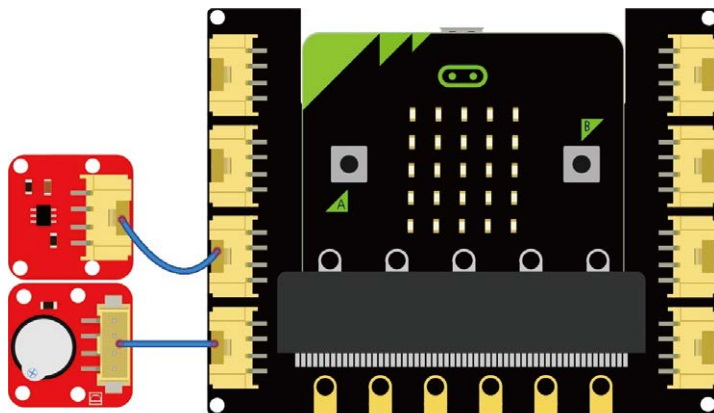


### Mission: **Let her calm down**

Difficulty Level: ★★★

Try to calm the angry girl down after you press the touch sensor, the micro:bit will play light music and the LED will change to be a breathing lamp to calm her down.

*You can build your circuit following the below circuit.*



## 13.Unsocial boy

Use the micro:bit to simulate an unsocial boy. He doesn't like others getting too close to him. Connect the LED to your micro:bit, and the micro:bit will light LEDs in different colors to show his feelings according to different distances.

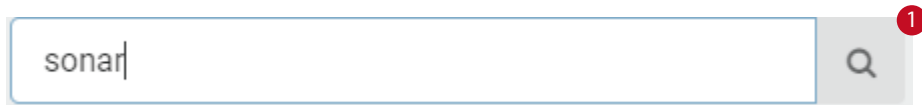
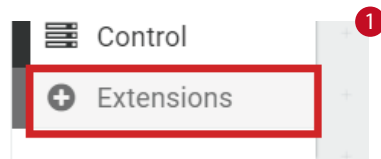
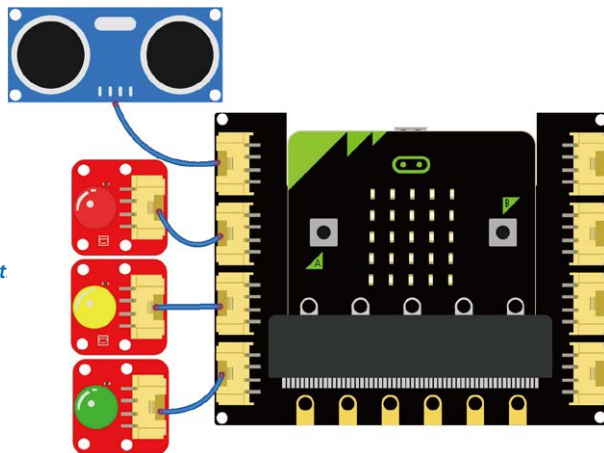
### Required Parts:

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail LED x 3 (red, yellow, green)
- Crowtail ultrasonic sensor x 1

### Programming with Makecode

#### Steps

- 1 Firstly, [click on the Extensions](#), then search "sonar" , [add the sonar package](#) to your Makecode.





sonar

A Microsoft MakeCode package to



Math



Sonar



Advanced

2 Create a **variable** to store the distance that ultrasonic sensor detects, set the unit to cm.

3 Set the program so that when the distance that ultrasonic sensor detects is **more than 100cm**, the **green LED will light up**, which means it is a comfortable distance.

4 When the distance is **between 50cm and 100cm**, the **yellow LED will light up**, that means the distance is slightly uncomfortable.

5 When the distance is **below 50 cm**, the **red LED will light up**. The boy feels very uncomfortable that others have got too close to him.



```
2 forever
  ping trig P13
  set distance to echo P15
  unit cm

3 if distance > 100 then
  show icon [happy face]
  digital write pin P0 to 1
  digital write pin P1 to 0
  digital write pin P2 to 0

4 else if distance > 50 and distance < 100 then
  show icon [neutral face]
  digital write pin P0 to 0
  digital write pin P1 to 1
  digital write pin P2 to 0

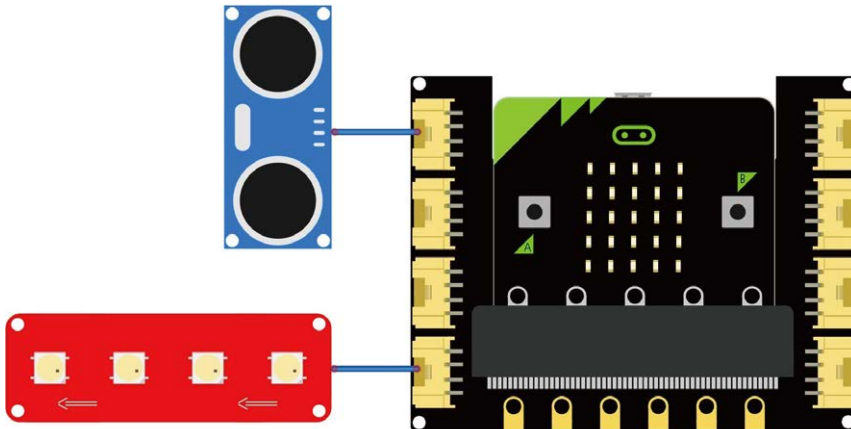
5 else if distance < 50 then
  show icon [sad face]
  digital write pin P0 to 0
  digital write pin P1 to 0
  digital write pin P2 to 1
```

Mission: **Make friends with him**

Difficulty Level: ★★★

Try to use the ultrasonic sensor and RGB LED to make a more social boy. When someone approaches him, the RGB LED will flash and the LED screen will show a smiling face. The boy feels so happy to make friends with you.

*You can build your circuit following the below circuit.*

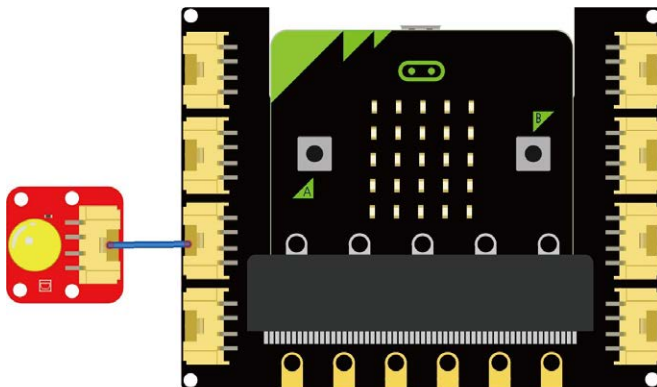
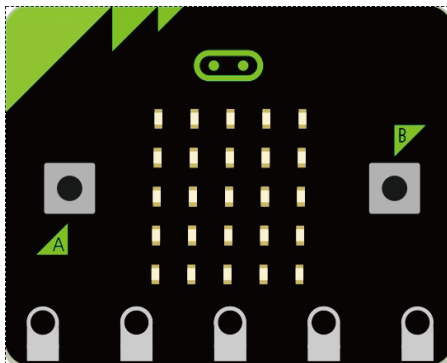


## 14.Remote Controller

Make the micro:bit become a remote control and use the buttons on your remote control to turn the LEDs on and off.

### *Required Parts:*

- Micro:bit x 2
- Crowtail base shield for micro:bit x 1
- Crowtail LED x 1



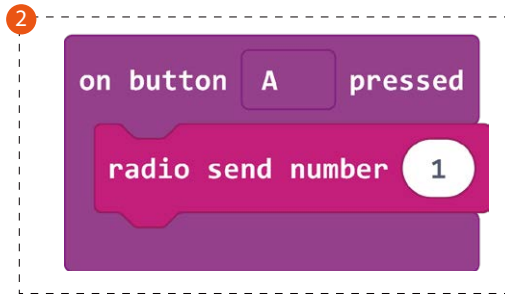
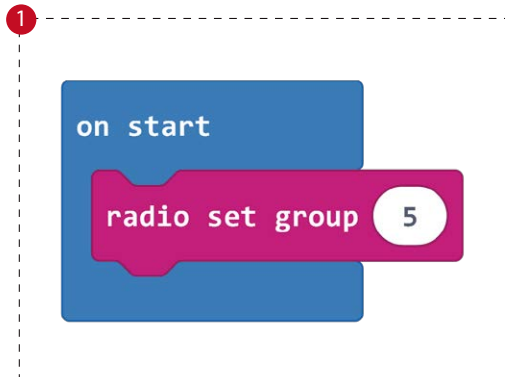
## Programming with Makecode

- Sending the order

1 On start, add the '*radio set group*' block to set the group ID for radio communication.

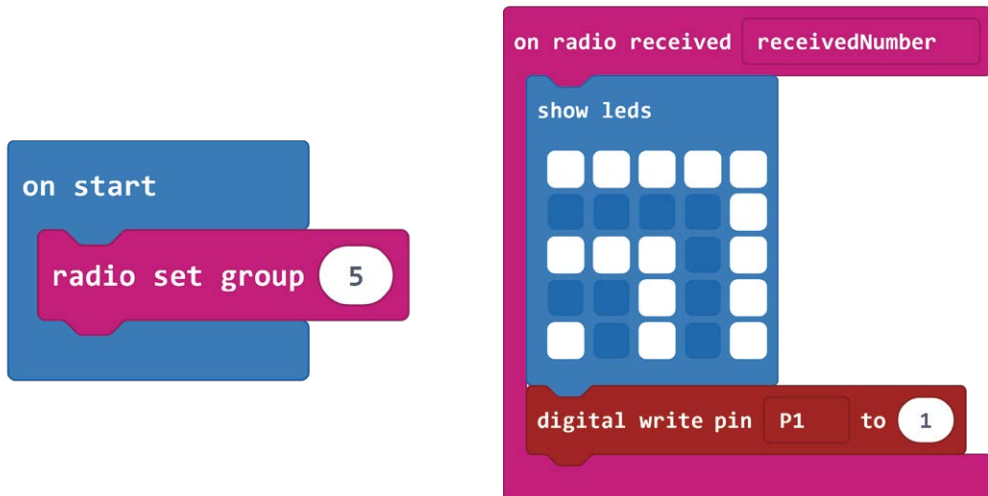
2 Add blocks to *send numbers* when *button A is pressed*.

3. **Download** the program to the micro:bit that will be used as a controller.



- Receiving the order

4. Add an '*on receivedNumber*' block that will run code whenever a new order comes in, the receivedNumber variable contains the number that we sent.



5. **Download** the program to your other micro:bit that will be used as the receiver.

6. *Press the button A* on the controller to see what will happen.



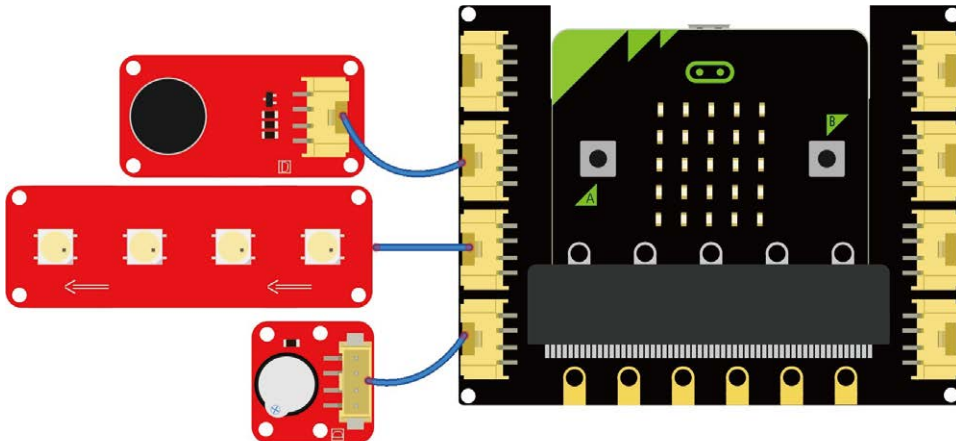


## Mission: Let's rock

Difficulty Level: ★ ★ ★ ★

Try to use this remote controller to control the RGB LED, buzzer and vibration motor at the same time. How about we press button A and button B to make them all work at the same time?

*You can build your circuit following the below circuit.*

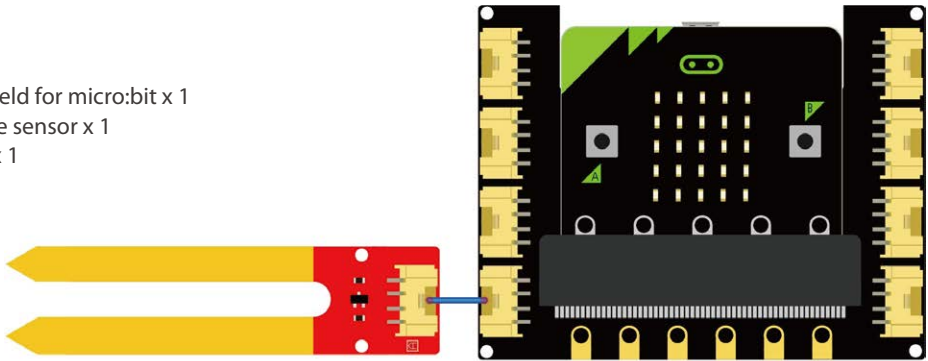


## 15. Soil Moisture

Use the moisture sensor to detect the moisture of soil and show the value on the LED screen.

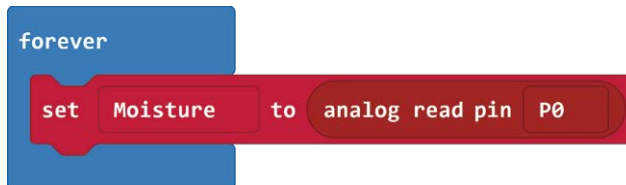
### Required Parts:

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail moisture sensor x 1
- Crowtail buzzer x 1



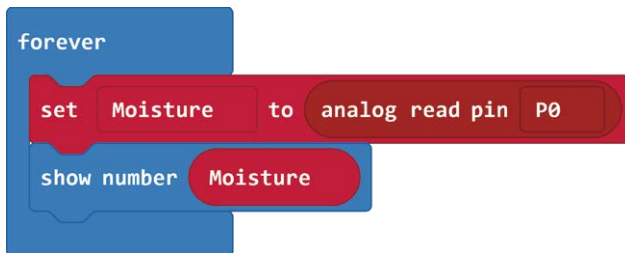
### Programming with Makecode

Step 1. Firstly, *create a variable "Moisture"* to store the moisture of soil.



**Step 2.** *Display the detected soil moisture value* on the LED screen.

**Step 3. Download** the program to your micro:bit, and plug the moisture sensor into the soil to read the soil moisture. You will view the analog value of soil on the LED screen.

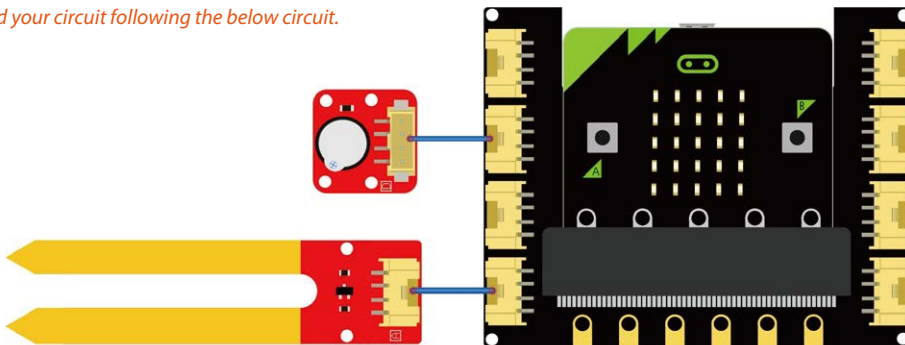


### Mission: Plant guard

Difficulty Level: ★★★

Try to use the micro:bit, moisture sensor and buzzer to make a guard for your plant. When the soil moisture is below the set value, the buzzer will warn you to water your plant.

*You can build your circuit following the below circuit.*



## 16. Automatic Gate

Automatic gates are widely used in public places, such as shopping malls, hotels, banks, etc. In this project we will use the ultrasonic sensor and servo to simulate the function of automatic gates.

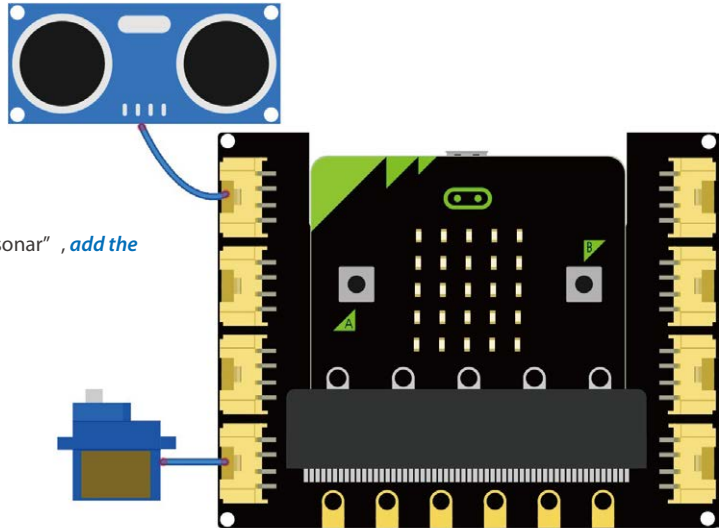
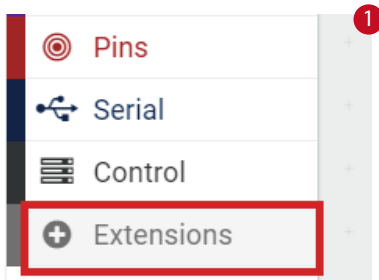
### Required Parts:

- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail ultrasonic sensor x 1
- Crowtail 9G servo x 1

### Programming with Makecode

#### Steps

- 1 Firstly, [click on the Extensions](#), then search “sonar” , [add the sonar package](#) to your Makecode.



sonar



Math



Sonar



Advanced

2 Create a variable to store the distance that the ultrasonic sensor detects, set the unit to cm.

3 Set the program, when the **distance** between the object and ultrasonic sensor is **less than the set value**, the servo will rotate 90 degrees to simulate an **opening door**.

4. **Download** the program to your micro:bit, then *put your hand near the ultrasonic sensor* to see what will happen.



2

forever

set distance to

ping trig P13  
echo P15  
unit cm

3

if distance < 10 then

servo write pin P0 to 0

pause (ms) 1000

else

servo write pin P0 to 90

pause (ms) 1000

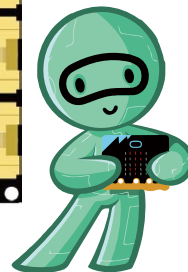
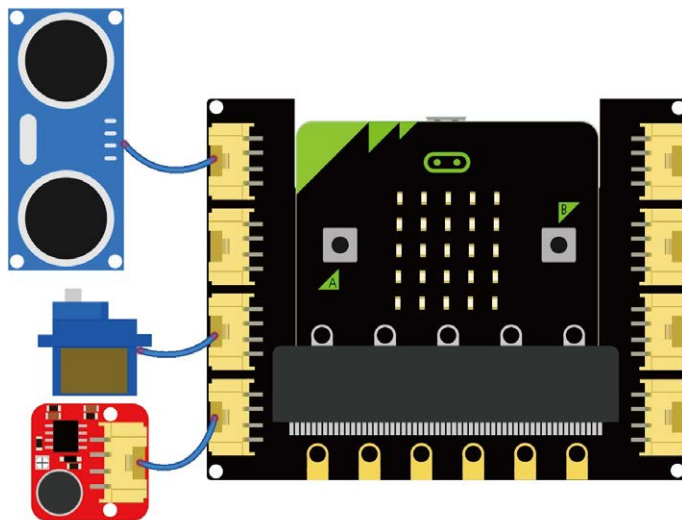
## Mission: Open Sesame!

Difficulty Level: ★★★

Use your sound sensor, ultrasonic sensor, and servo to make a door where only if you stand in front of the door and say a magical phrase such as “Open Sesame” will it open for you.



*You can build your circuit following the below circuit.*

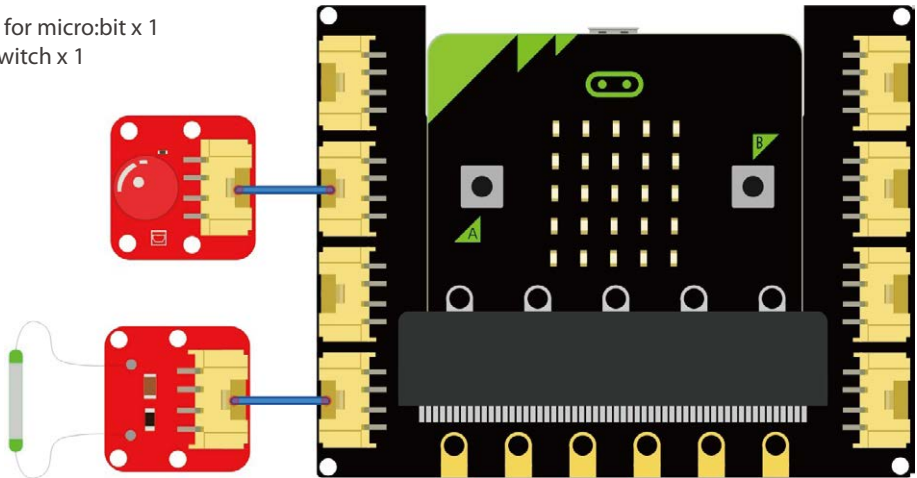


## 17. Magnetic Field Detector

Use the magnetic switch to make a magnetic field detector. When it detects a magnetic field, the LED will light up to remind you that there is a magnetic field nearby.

### *Required Parts:*

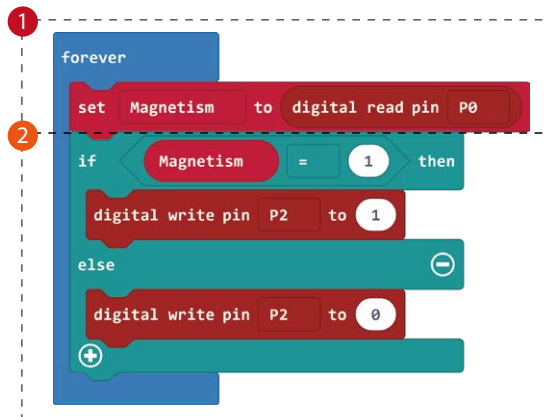
- Micro:bit x 1
- Crowtail base shield for micro:bit x 1
- Crowtail magnetic switch x 1
- Crowtail LED x 1
- Magnet x 1



## Programming with Makecode

### Steps

- 1 Create a variable "Magnetism" to store the state of the magnetic switch.
- 2 When the *magnetic switch detects a magnet*, the LED will light up.
3. **Download** the program to your micro:bit, then use the magnet to create a magnet field. Put the magnet near the magnetic switch to see what will happen.

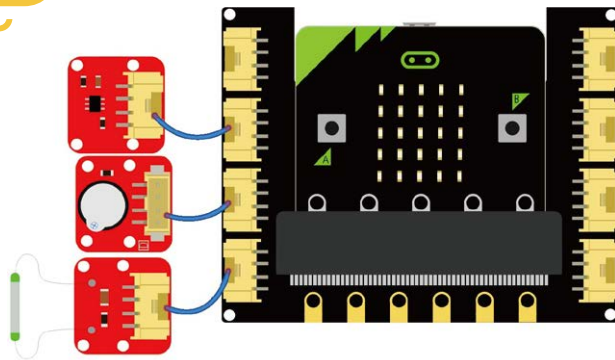


### Mission: No entry

Difficulty Level: ★ ★ ★

Everyone needs privacy, are you annoyed that others will enter your room without your permission? Let's make a door magnetic switch alarm with a magnetic switch and a buzzer. The buzzer will beep to alert you when someone enters your room.

You can build your circuit following the right circuit.





# Keep Exploring

We also provide **MicroPython lessons** for this kit, it can be downloaded on our official website. You have just finished your invention journey with this guide, but there are unlimited possibilities as to what you can do with our Crowtail modules. Try to build your own circuit and make your own projects. Share your creations with your friends. You may get fresh new ideas from them.



Cultivating Creativity through Projects, Passion, Peers and Play.

- Mitchel Resnick



MAKE YOUR MAKING EASIER



+86 0755-23204330



[www.elecrow.com](http://www.elecrow.com)



[info@elecrow.com](mailto:info@elecrow.com)



8<sup>th</sup> Floor, F-building, Fusen industry  
park, Gushu Town, Bao'an District,  
Shenzhen, China.

