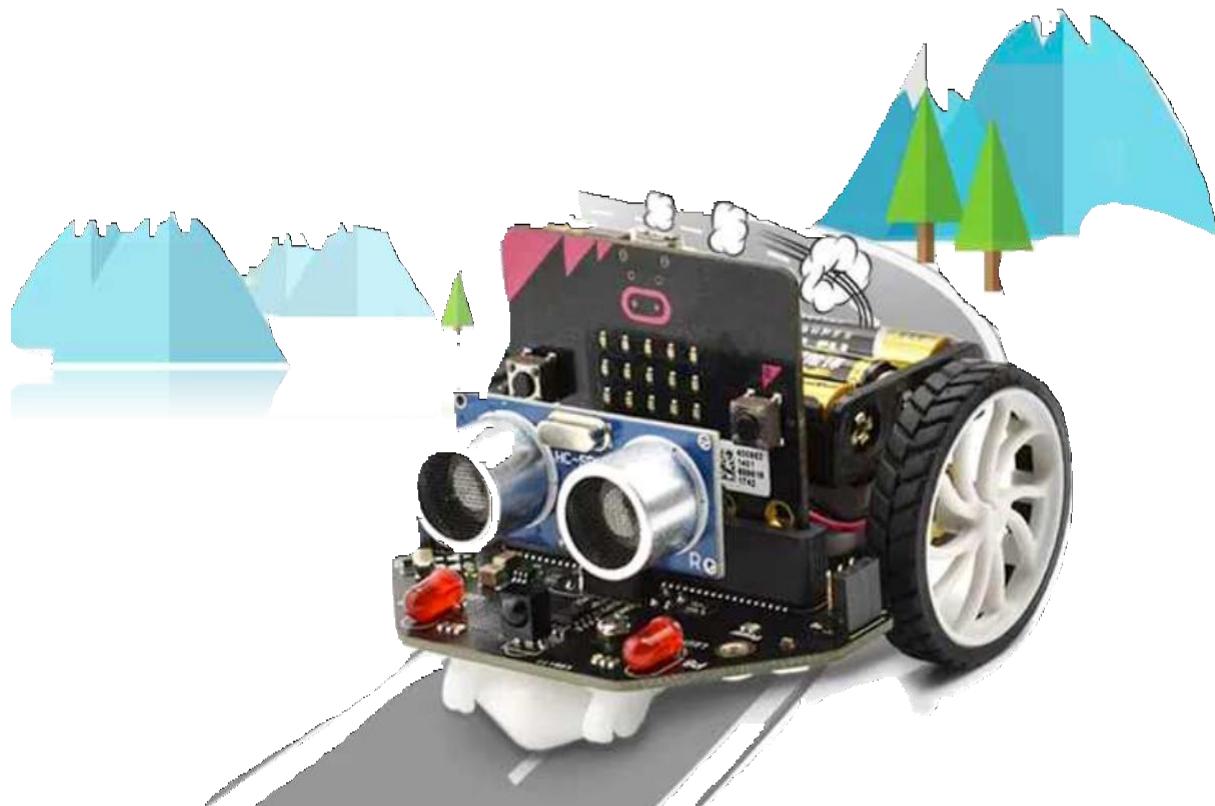


## STEM Education Series

# Maqueen & micro:bit Graphical Programming

Chen Baojie Sha Liangwen



TSINGHUA UNIVERSITY PRESS

Beijing

## Introduction

The programming courses in this book are specially designed for all robotic hobbyists and STEM educators. Most people may think that robotic learning is a high-cost investment, actually, it is not the case when compared with what achievements we may make in robotic learning process. A high-quality and funny teaching equipment can arouse students' interests in learning robot. By designing and controlling robots, they can get known about robotic engineering, programming and machine learning and so on. Maqueen is a mini robot based on micro:bit with various output modes, integrating sound, temperature, and light sensor. Students can program with Maqueen to realize all kinds of playing methods. Furthermore, the large quantity of interacting cases could let them enjoy the happiness of programming learning. Besides that, it can also develop students' logic thinking and creativity. The book is suitable for the middle and elementary students, also can be used as an introduced textbook for STEM beginners' training education.

There is an anti-fake label of TSINGHUA UNIVERSITY PRESS on every book's cover, books without the label are not allowed to be sold.

The book is published by TSINGHUA UNIVERSITY PRESS. All rights reserved.

Copyright infringement complaining Tel: **010-62782989 13701121933**

Editor in charge: Zhang Chi

Author: Chen Baojie Sha Liangwen

Translator: Frances Zhou Nana Wang

## Preface

Why should kids learn to program? What's the significance and value of learning programming? We may ask these questions often as parents, but for kids, instead of considering why and what, they just thought that programming robot is fun and interesting, and they are so pleased to show their awesome robots work to others. Learning robotics and coding can be a kind of gift this era brought to them.

Programming is an exciting and satisfying progress, since in programming, we have to analyse and ponder all kinds of possibilities and then to come up with an acceptable method to solve the problem. In this process, it can greatly improve kids' creativity, logical thinking and problem-solving ability, furthermore, coding is able to instill persistence, build frustration tolerance and enhance concentration.

Teaching robotics in school is getting more and more prevalent throughout the country. "like English, programming will become a personal ability rather than programmer's professional skill in the future, and it will integrate into every aspects and details of our social life." says Chen Yue, a professor at Computer College of Zhejiang University. Programming robots helps to develop children' imagination and creativity so that they can easily adapt the society of AI in the future.

Maqueen is a mini robot car intensively designed for STEM education, small but powerful. It is developed on the basis of the popular micro:bit board. Children can code and create by the board, low cost and easy to learn, which is precisely what the original intention of STEM education is. Accelerometer, temperature sensors and other commonly used sensors are all integrated on micro:bit board, and it is also equipped with Bluetooth and wireless communication. There are ultrasonic sensor, motor and line-tracking sensor on the car body. Everyone can control the robot car easily by programming.

We use Mind+ programming software in this book. Mind+ is a Scratch 3.0-based graphical Programming platform that supports all kinds of open source hardware such as Arduino, micro: bit. Drag and drop code blocks to make programs, or use high-level programming language like Python, C, C++ etc. to code. Easy to experience the joy of creating.

This book is composed of 20 teaching cases, and each case is designed on the basis of teaching practices. It aims to inspire all children to participate in programming and pave the path for them to further develop and create.

## Content

<b>Chapter 1. Customize Emoji.....</b>	<b>1</b>
1.1 Micro:bit.....	1
1.2 Mind+ Interface.....	3
1.3 Algorithms Analysis.....	7
1.4 Example Program.....	7
1.5 LED Matrix.....	10
1.6 Training Camp .....	12
<b>Chapter 2. Funny Electronic Organ .....</b>	<b>13</b>
2.1 Sound Module .....	13
2.2. Function Module .....	15
2.3 Micro:bit Pins .....	16
2.4 Training Camp .....	19
<b>Chapter 3. Catching Apple.....</b>	<b>20</b>
3.1 Algorithms Analysis.....	20
3.2 Programming .....	20
3.3 Programming Result.....	25
3.4 Training Camp .....	26
<b>Chapter 4. Cat and Mouse.....</b>	<b>27</b>
4.1 Accelerometer .....	27
4.2 Algorithms Analysis.....	28
4.3 Programming .....	28
4.4 Variable and Broadcast.....	34
4.5 Programming Result.....	36
4.6 Training Camp .....	37
<b>Chapter 5. Stress Buster .....</b>	<b>38</b>
5.1 Algorithms Analysis.....	38
5.2 Programming .....	38
5.3 Program Description .....	43
5.4 Programming Result.....	44

5.5 Training Camp .....	45
<b>Chapter 6. Chatbot .....</b>	<b>46</b>
6.1 Get to know list .....	46
6.2 Algorithms Analysis .....	51
6.3 Programming .....	51
6.4 Temperature Sensor .....	54
6.5 Programming Result .....	54
6.6 Training Camp .....	55
<b>Chapter 7. Gorgeous Firework .....</b>	<b>56</b>
7.1 Clone .....	56
7.2 Algorithms Analysis .....	61
7.3 Programming .....	62
7.4 Programming Result .....	64
7.5 Light Sensor .....	65
7.6 Training Camp .....	65
<b>Chapter 8. Sound Paintbrush .....</b>	<b>66</b>
8.1 Pen Module .....	66
8.2 Algorithms Analysis .....	70
8.3 Programming .....	71
8.4 Expansion and Buzzer .....	73
8.5 Programming Result .....	76
8.6 Training Camp .....	76
<b>Chapter 9. Security Alarm .....</b>	<b>77</b>
9.1 Algorithms Analysis .....	77
9.2 Sensor and Multithread .....	77
9.3 Output and Input Module .....	81
9.4 Programming Result .....	83
9.5 Training Camp .....	85
<b>Chapter 10. Bubble Machine .....</b>	<b>86</b>
10.1 Algorithms Analysis .....	86
10.2 Scratch Mode Case .....	87

10.3 Code Mode Case .....	89
10.4 Servo .....	92
10.5 Training Camp .....	93
<b>Chapter 11. Get to know micro: Maqueen.....</b>	<b>94</b>
11.1 What is a robot? .....	94
11.2 Robot Application.....	96
11.3 Hello, Maqueen!.....	97
11.4 Have a Try! .....	99
<b>Chapter 12. Maqueen, let's move!.....</b>	<b>100</b>
12.1 Programming .....	100
12.2 Maqueen, let's move!.....	105
12.3 Training Camp .....	107
<b>Chapter 13. Turn on RGB Lights .....</b>	<b>108</b>
13.1 Programming .....	108
13.2 Time for Music .....	109
13.3 Training Camp .....	109
<b>Chapter 14. Ultrasonic Obstacle Avoidance.....</b>	<b>110</b>
14.1 Principle of Ultrasound.....	110
14.2 Obstacle Avoidance Program.....	112
14.3 Training Camp .....	113
<b>Chapter 15. Light-chasing Maqueen .....</b>	<b>114</b>
15.1 Get Data .....	114
15.2 Light-chasing .....	115
15.3 What is Forever? .....	116
15.4 Training Camp .....	118
<b>Chapter 16. Line-tracking Maqueen.....</b>	<b>119</b>
16.1 Single Line-tracking Sensor .....	119
16.2 Double Line-tracking Sensor .....	120
16.3 Training Camp .....	123
<b>Chapter 17. Infrared Remote-control Robot .....</b>	<b>125</b>
17.1 Get Key Value .....	126

17.2 Remote-control Maqueen.....	127
17.3 Training Camp .....	128
<b>Chapter 18. Radio Remote-control Maqueen.....</b>	<b>129</b>
18.1 Gamepad.....	130
18.2 Control Maqueen with Gamepad .....	132
18.3 Training Camp .....	135
<b>Chapter 19. Research and Rescue Maqueen .....</b>	<b>136</b>
19.1 Thinking Draft .....	136
19.2 Connection Diagram .....	137
19.3 Programming .....	139
19.4 Training Camp .....	142
<b>Chapter 20. Maqueen and IoT.....</b>	<b>143</b>
20.1 OBLOQ Module .....	144
20.2 Functions .....	145
20.3 Test Result .....	149
20.4 Training Camp .....	150
<b>Bibliography .....</b>	<b>151</b>

## Chapter 1. Customize Emoji

We can hardly stop using emoji now when we chat online, how about design a unique emoji for ourselves to display the emotion we feel now!

In this chapter, we will get to know micro:bit, and try using its LED panel display function. Besides, we will also get familiar with mind+, and learn how to create a sprite and build a background and so on.

### 1.1 Micro:bit

Micro:bit is an open source hardware ARM-based embedded system designed by BBC for use in computer education, developed under the cooperation of Microsoft, Samsung, ABM, and so on. From robots to Lego, micro:bit can be used to build all sorts of fantastic projects-the possibilities are endless!

At present, micro:bit is promoted by the British non-profit organization Micro:bit Educational Foundation all over the world.

Visit the Website at <http://www.microbit.org>.

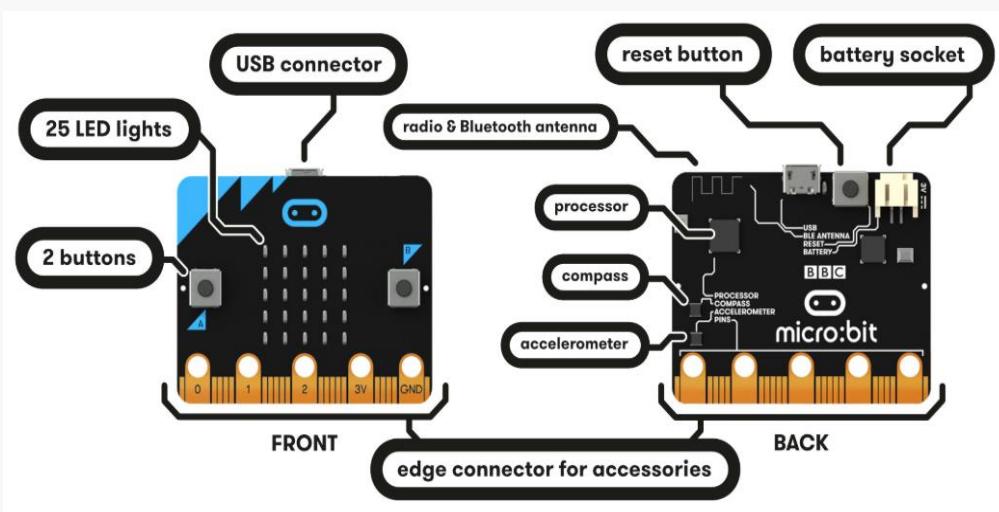


Figure 1-1

Micro:bit board, with a credit card size of 5cm×4cm, has 25 programming LEDs, two programmable buttons, light sensor, accelerometer, magnetometer, temperature sensor and Bluetooth module and so on.

Micro:bit integrated a state indicator (keep on when powered up and keep flashing when burning and communicating) and reset button (reboot the program quickly) near the micro USB.

### What can micro:bit do?

Micro:bit is equipped with built-in common sensors and its LED can display text, numbers and images. We can carry out many interesting experiments by graphical programming or coding. In addition, micro:bit can also be used together with crocodile clips and expansion board to interact with other electronics, for instance, to read sensor data, control servo and LED strip. It can totally fit all kinds of programming teaching and developing situations.

Micro:bit can be widely used in computer games, acousto-optic interactions, robotics, scientific experiments, wearable device and so on.

### Programming micro:bit

Micro:bit is compatible with a number of online code editors such as Blocks, javascript, Python, scratch etc. Without installing any software, users can directly program mciro:bit on the web page. While if you want use Bluetooth to send code from your mobile phone to the micro:bit, you need to download the micro:bit companion APP first. Visit micro:bit website at <http://micro:bit.org> to start coding.

There are many graphical programing software supporting micro:bit. However, most programming software only support some certain hardware and robots or even cannot work with any hardware at all. It would be perfect if there is a software that not only supports graphical programming, but also compatible with all sorts of hardware. Luckily, we get Mind+ here, and it can meet all the

requirements! It is suitable for primary and secondary school students as well as able to provide a great learning environment for makers who want to improve themselves by studying high-level programming language such as Arduino, python, C, C++ etc. Drag and combine code blocks to make programs, easy to find the joy of creating.

## 1.2 Mind+ Interface

Mind+ download address: <http://www.mindplus.cc/>

Mind+ interface includes: function menu bar, control, block, script, sprite and stage section.

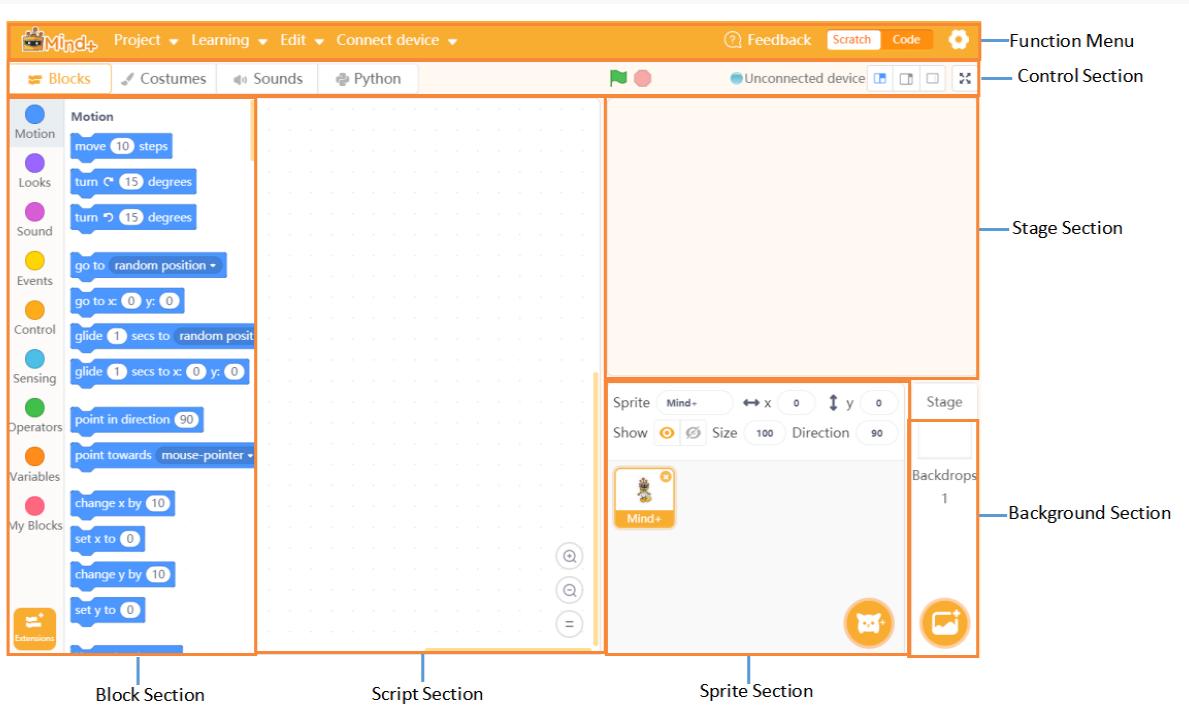


Figure 1.2 Mind+ Interface

### 1) Create a sprite

This is the backstage of the whole “stage”. Without the help of the menu bar, we can do nothing with this software. What functions are there behind the “stage”? Sprite is the main character in the play, and will show his performance on the

stage. Users can give instructions to a sprite by snapping blocks together in the script area. A robot Micko will be regarded as the main character by default when creating a new project. If you want to add new sprites to the project, just click the sprite library at the lower-right corner (next to the background library) to select the one you like. Then drag and combine the blocks in the script area to program and control the sprite. Each sprite has its own script, when you click the sprite in the sprite area, the related codes will be displayed on the script area.

There are so many ways to create a sprite, let's start to learn! First we will take a look about the sprite tool in Chart 1-1.

TOOL	Description
	Import a character from the computer
	Pick a sprite from sprite library
	Draw a new sprite
	Select a sprite from the sprite library

Chart 1-1

Open the sprite library, in which there are different character classifications. We can choose them according to our actual project.

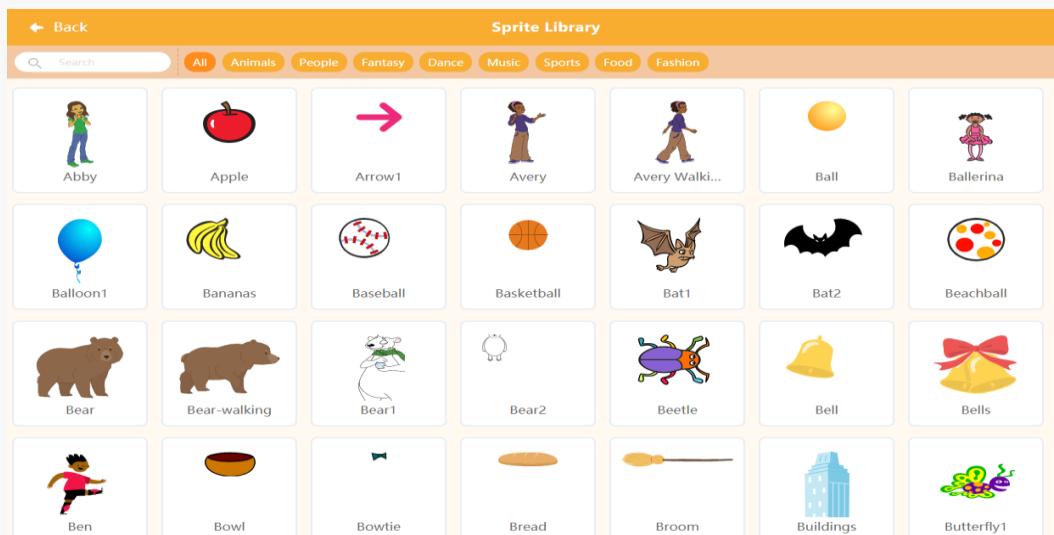


Figure 1-3

In order to make the sprite fit the whole story plot, you may need to adjust its size and costumes after importing it to stage, or even sometimes you have to draw a new sprite.

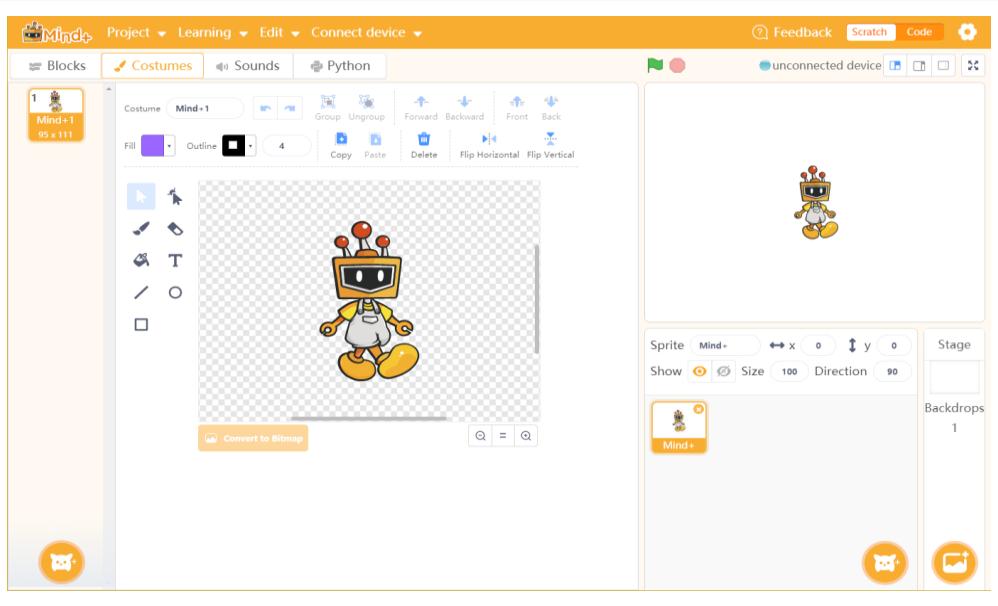


Figure 1-4

We can design different costumes for every sprite, for instance, when we add "Butterfly 1" to the stage, we can find there are two different costumes (some sprites only have one shape). This can basically make our sprite come to alive. Of course, we can add new sprite or switch the sprite's costumes.

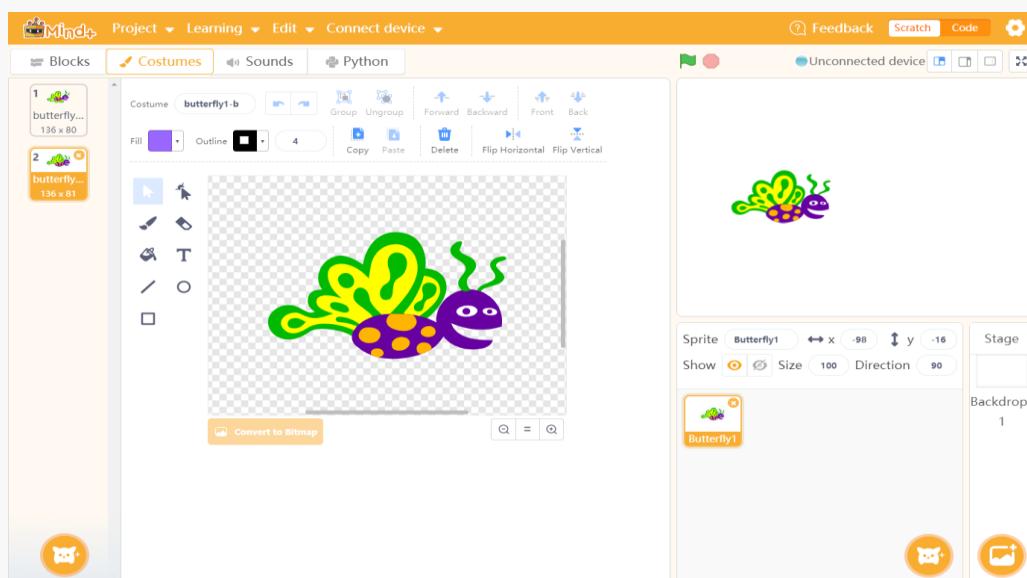


Figure 1-5

## 2) Build a background

Stage is the place where your sprites show their performance. Add background to the stage to provide a suitable living environment for the characters so as to make the entire story fun and lively. In the background area, we can choose an image we like from the background library, also we can upload a picture from the computer. Let's get to know the background library.

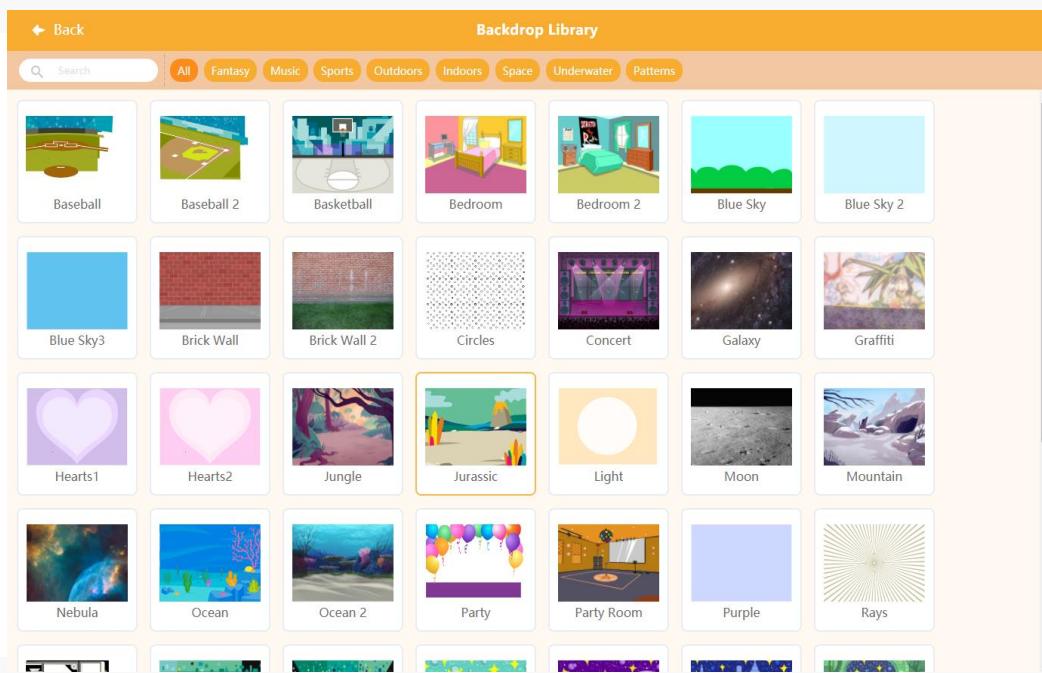


Figure 1-6

## 3) Coordinate system of stage

Each sprite has its own position on the stage. To let the whole play orderly, we have to make every sprite move along the exact routine in the coordinate system of the stage.



Figure 1-7

The center point of the stage is (0,0). The horizontal direction is X axis and vertical direction is Y axis. Take the center point as the original point, the right part of X axis, positive (+); the left part negative (-). The upper section of Y axis, positive (+); the lower section, negative (-). Get known about the coordinate system then we can well control the sprites to perform on the stage.

## 1.3 Algorithms Analysis

- Create two sprites: smiling face and crying face.
- Make the two sprites switch automatically every 3 seconds. Press the space key to trigger.
- Synchronize hardware. Realize to switch emojis on micro:bit board automatically.

## 1.4 Example Program

### 1) Build a new costume

Click the paint to draw new sprites, make a smiling face and a crying face.

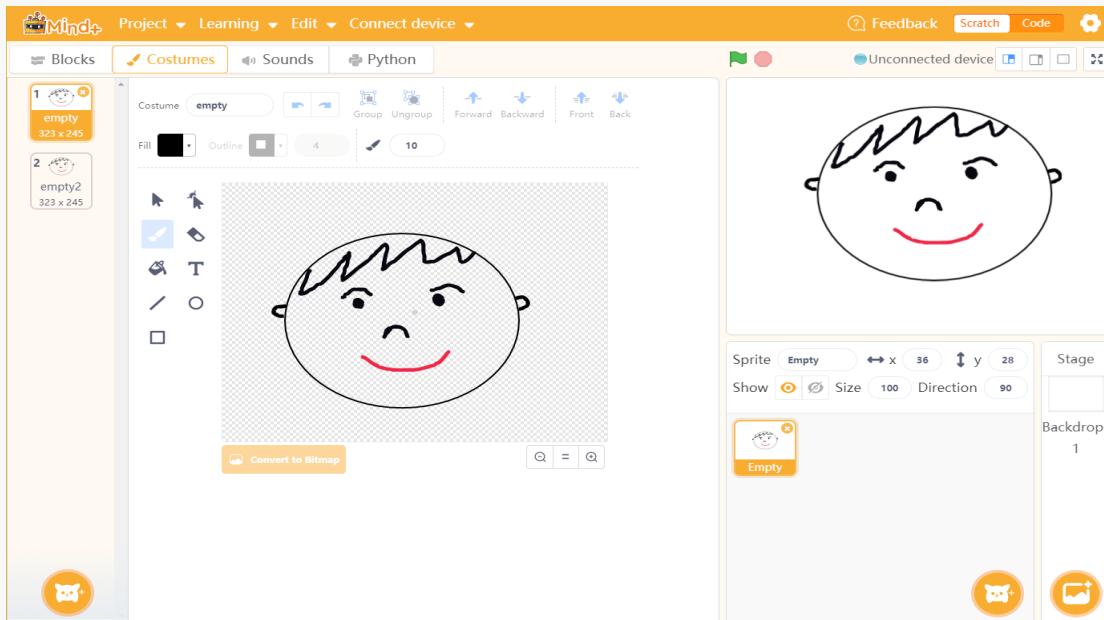


Figure 1-8

## 2) Sprite costume programming

1. Click and find suitable blocks to switch different style sprites.
2. When the space key is pressed, the program starts and it will switch to next sprite.

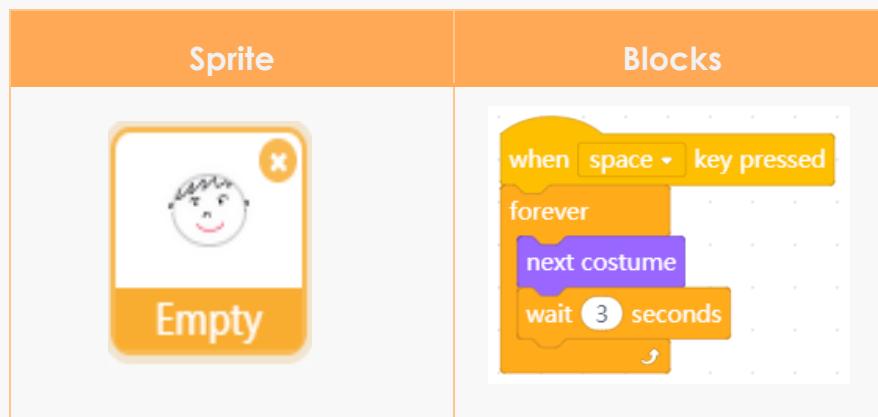


Chart 1-2

### Add Blocks:

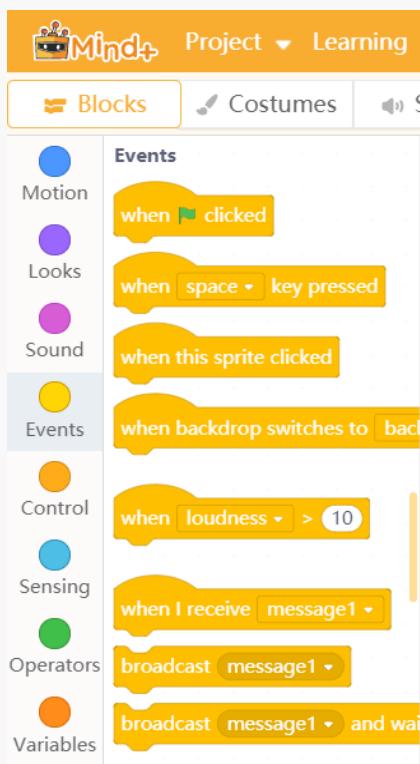
Drag blocks from the module area to script area, snap blocks together to control the sprite. The different color of the block represents different functions such as motion, look, sound etc.

## Delete blocks:

Drag block to the left to remove it, or right-click to delete block. Notice: when we want to delete block that has been nested in a group of blocks, we have to separate it from the others firstly.

## Get to know events:

The events module is usually used to detect the event operation of other modules. They are often used as the foundation of starting a program, so this module is relatively simple to learn.

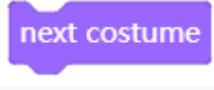
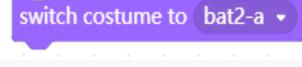


- ① When Green Flag clicked: click the green flag to enable the entire program;
- ② When (space key) pressed: press the space key to start the event, or replace space key with other keys to control the scripts underneath this block.
- ③ When this sprite clicked: scripts that wear the block will activate once its sprite is clicked.
- ④ When backdrop switches to backdrop (): scripts under this block will be triggered once the specified backdrop has been switched to on the stage.
- ⑤ When (loudness /timer)>(): it starts the scripts below it when a value (loudness or timer) is greater than another value (can be input).

⑥ When I receive (message 1): scripts that begin with this block will be invoked once the specified broadcast has been sent by a calling script.

⑦ Broadcast (): send a broadcast throughout the whole program. Any scripts in any sprites that are hatted with the When I Receive () block that is set to a specified broadcast will activate.

- ⑧ Broadcast () and wait: this block lets scripts send broadcasts, and have them wait until all scripts activated by the broadcast end.

Blocks	Description
	Switch the costume to the next in the costumes pane, if the current costume is the last in the list, the block will loop to the first.
	Change the sprite's costume to a specified one.
	Pause the script for the specified number of seconds
	Blocks held inside this block will be in a loop.
	Blocks held inside this block will loop a given amount of times before allowing the script to continue.

### Result:

Add a background for the stage. When the space key is pressed, the program starts and the two different emojis are switched.



Figure 1-10

## 1.5 LED Matrix

Switch different emojis on the micro:bit LED screen. What is it? LED stands for Light Emitting Diode. Micro:bit has 25 individually-programmable LEDs, allowing you to display text, numbers, and images.

We can use micro:bit LED screen to display our customized emojis.

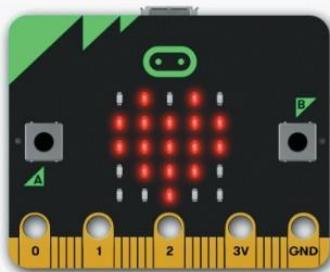


Figure 1-11

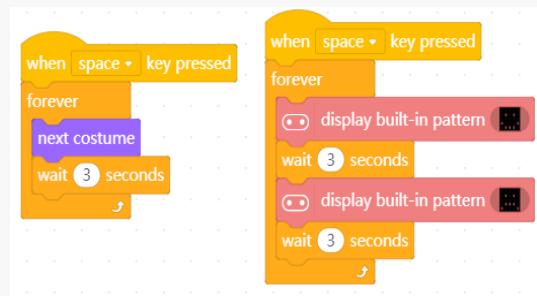


Figure 1-12

### Hardware Connection

Connect micro:bit to the computer with Micro USB, select the corresponding COM port in the Connect Device menu bar. Please note: if we selected the Scratch mode at the right-upper corner, then the program result can be displayed without downloading codes.

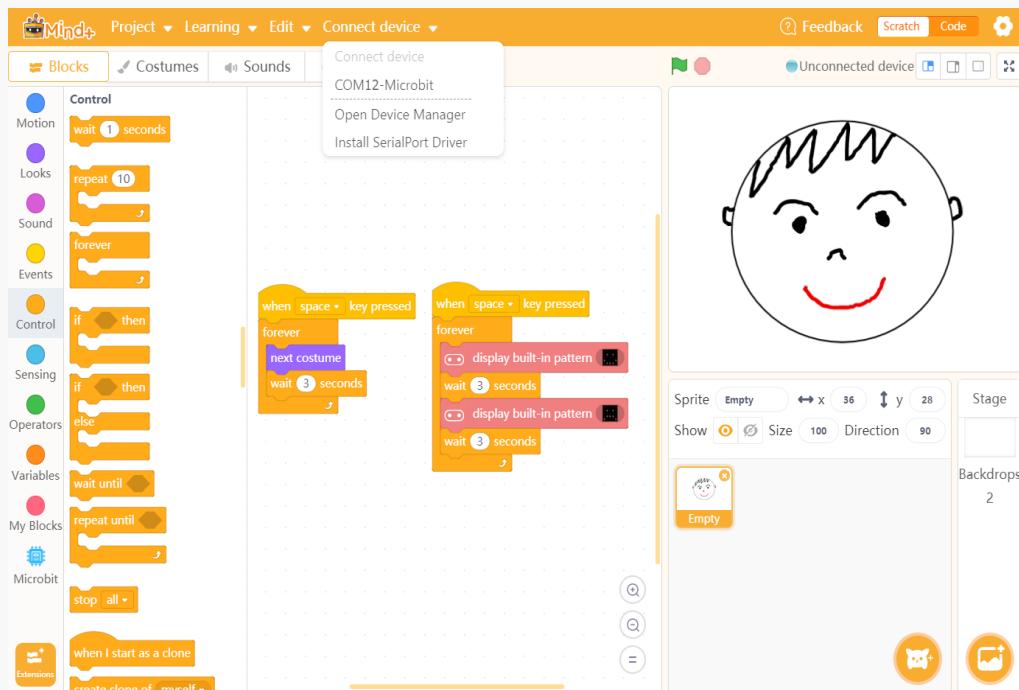


Figure 1-13

### Result:

Press the space key, let's see what the Mind+ and micro:bit LED screen will display.

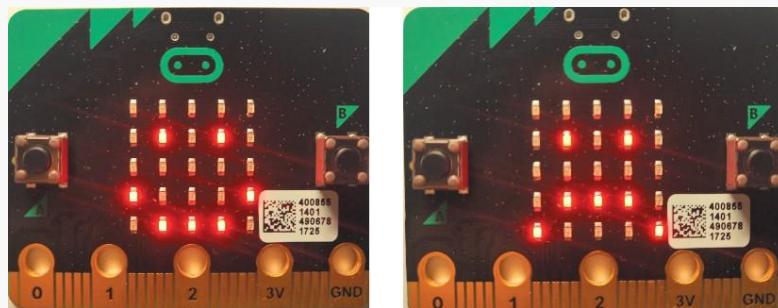


Figure 1-14

## 1.6 Training Camp

It may be bored if there are only pictures displayed on the screen, how about add some sounds for the emojis? Display the smiling face with laugh sound, and the crying face with cry sound. Have any idea how to do this? Come and try!

## Chapter 2. Funny Electronic Organ

“Twinkle twinkle little star, how I wonder what you are...”. Do you wanna try playing songs with keyboard? Micro:bit board can realize the functions of electronic organ, such as playing music, making musical accompaniment, adjusting playback speed, or even performing music. So many interesting functions, get started quickly if you like them! In this chapter, we are going to learn the “sound” module, micro:bit pins and button.

### 2.1 Sound Module

We will use sound module to write a program. The program needs to achieve the following functions: press down 1, 2, 3... on the keyboard, then the sound effect, Do, Re, Mi...will be displayed. Then we can create our own songs.

The eight numbers 1 to 8 on the keyboard respectively represent Do, Re, Mi, Fa, Sol, La, Ti, Do(high-pitch), and the corresponding English letters are C, D, E, F, G, A, B, C2. Shown as below.

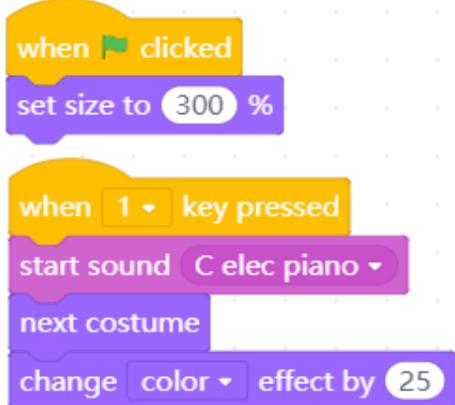
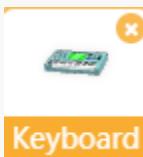
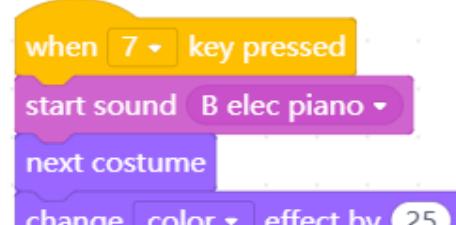
Note	Number	Letter	Scratch Script
Do	1	C	60
Re	2	D	62
Mi	3	E	64
Fa	4	F	65
Sol	5	G	67
La	6	A	69
Ti	7	B	71

Chart 2-1

### Program Analysis:

- Set the sprite “keyboard”, there are two different built-in shapes for keyboard: keyboard a and keyboard b.

- Add background for the stage.
- Press down the keys on the keyboard to start the programs, play the notes and switch the shape. The detailed programs are shown in chart 2-2:

Sprite	Code Blocks
	 
 Keyboard	 
	 
	 

Add background for the stage, run the programs and check the result.

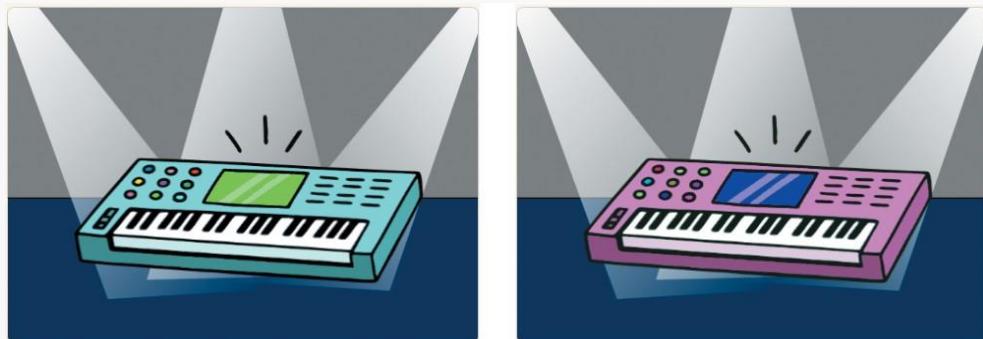


Figure 2-1

## 2.2. Function Module

Apparently, the sound module in Mind+ cannot meet all our needs. So we have to use the function module in “Extensions”. Open the “function” module in extensions, and select “music”. We can set the beat of the notes in this module.

A screenshot of the Mind+ software interface. At the top, there's a navigation bar with a back arrow, the text "Select Function", and a search bar. Below the navigation bar, there are tabs for Kit, Board, Sensor, Actuator, Communication, Display, and Function, with "Function" being the active tab. A "Search" button is also present. The main area is divided into two sections: "loaded:" and "not loaded:". The "loaded:" section contains one item: "Music" (represented by a piano icon), which is described as "Play instruments and drums.". The "not loaded:" section contains four items: "Pen" (represented by a pencil icon), "Video Sensing" (represented by a camera icon), "Google Translate" (represented by a Google logo with Chinese characters), and "IIC Scan" (represented by a yellow square icon). Each item has a brief description below it.

Figure 2-2

## 2.3 Micro:bit Pins

Now, let's take a look about the micro:bit pins. As figure 2-3 shows, there are 25 external connectors on the edge connector of micro:bit, which we refer to as "pins". Program motors, LEDs, or other electrical components with the pins, or connect extra sensors to control your code!

The first three, labelled 0, 1 and 2 are flexible and can be used for many different things-which means they are often called "general purpose input and output" (GPIO for short).

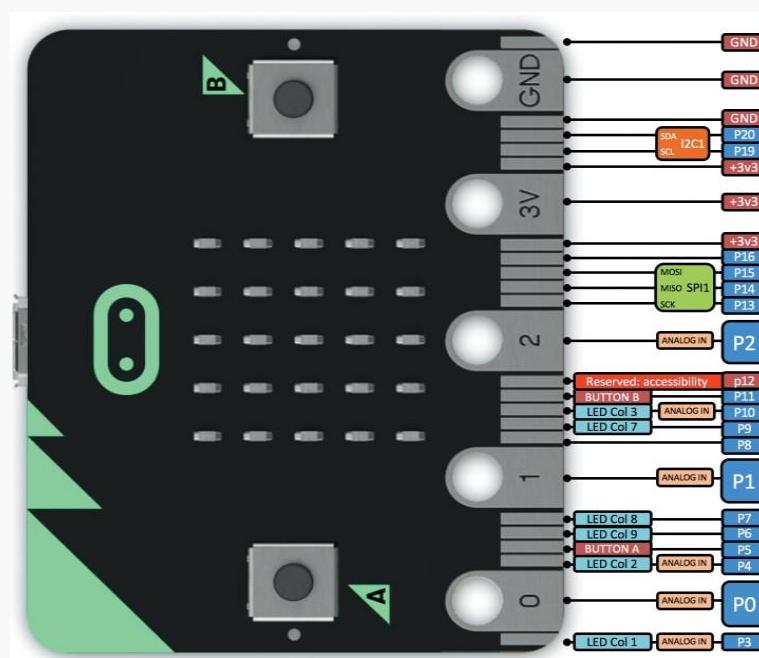


Figure 2-3

Connect micro:bit board and the computer with micro USB. Let pin P0 play the music "twinkle, twinkle little star", and P1 and P2 display different instruments. Press button A and button B to play faster or slower. As shown in Figure 2-4.

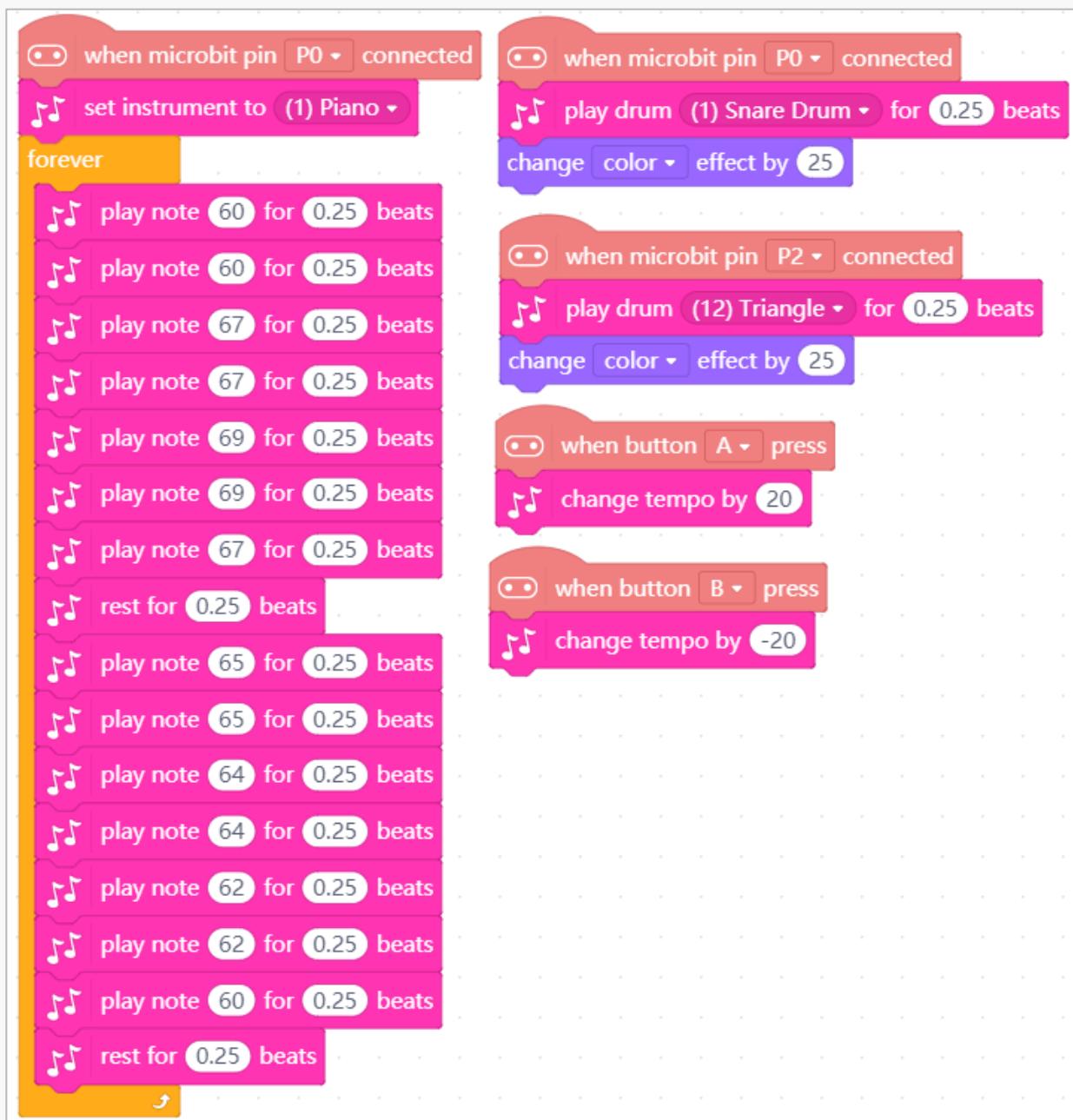
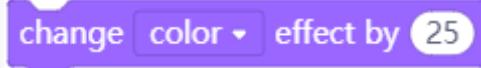
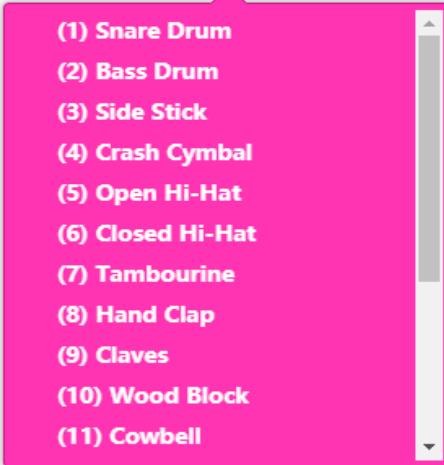


Figure 2-4

Code Blocks Descriptions:

Blocks	Description
	Click the green flag to enable the entire program
	Select the sprite "keyboard" in the sprite library, and then open the sound module, the corresponding "electronic organ" sound will

	appear. Switch the different English letters to play the electronic sounds Do, Re, Mi, Fa, Sol, La, Ti, Do(high-pitch).
	Change the sprite's color by increasing its color effect value
	When Pin P0 was connected, the program starts
	When button A was pressed, the program starts
	Loading the extended module. Select "Extensions"->"Function"->"Music". Note and beat of the sound were represented by numbers, which are matched with keys on the organ keyboard.
	Rest for () beats: to stop for several beats
	Control the playback speed: positive->faster; negative->slower
 	Display different drum sounds

**Result:**

Touch the micro:bit pin with your one hand, and touch the GND with another hand, by which to form a closed loop between the board and your body, then program starts. Touch pin P1 or P2 to add accompaniment for the performance, and press button A and B on the micro:bit to control the playback speed.

## 2.4 Training Camp

So far, we only tried the electronic organ, do you want to play other instruments? How to switch different instruments randomly, think over and give it a try!

## Chapter 3. Catching Apple

We are gonna make a catching apple game. Game players control the cat to move forth and back to catch the falling apples while avoiding the yellow lightning.

### 3.1 Algorithms Analysis

- Press the button A and B on the micro: bit to control the cat move left and right; game over if the lightning hits the cat;
- Duplicate the sprite “Apple”, and the apples appear and fall from the upper of the screen constantly. If the apple touched the cat, it disappears and reappears in the upper-side of the screen;
- Duplicate the sprite “Apple”, revise the waiting time after it falls;
- Add the sprite “Lightning”, the falling effect is same to apple, but when it hits the cat, game over.

### 3.2 Programming

**1. Create the main characters:** cat, apple and lightning.

**2. Program the cat:** the cat has two shapes. When the cat moves left or right, change its shape to the related action gesture so as to make it more like a cat walking.



Figure 3-1

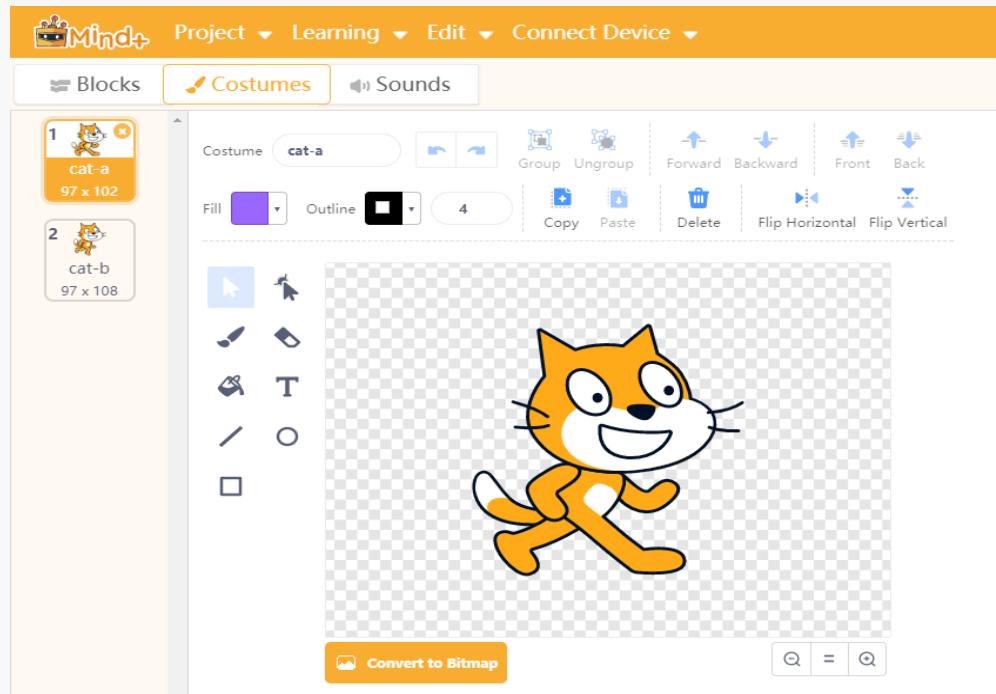


Figure 3-2

Press the button A and B on the micro:bit board to control the cat's movement. Change the value of x axis and the movement direction of the cat. The cat will catch the apple as it walks, if the lightning hits the cat, stop all and game over. Figure 3-3 is the entire program.

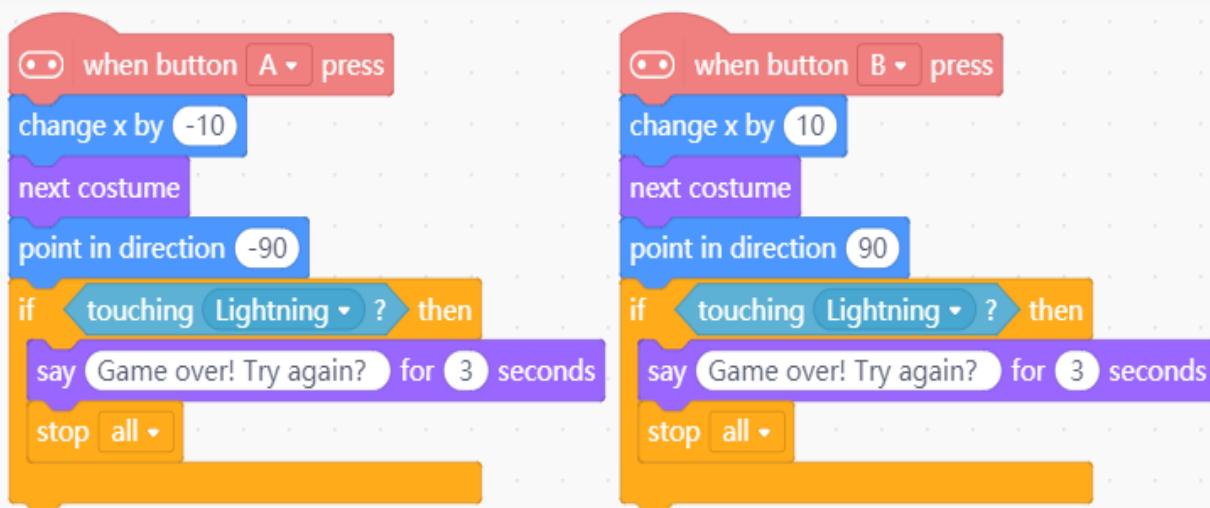


Figure 3-3

Code Block Description:

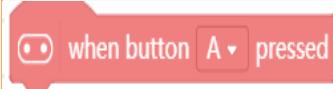
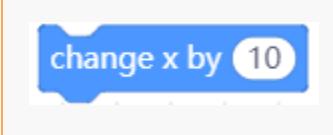
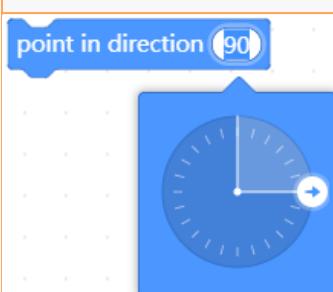
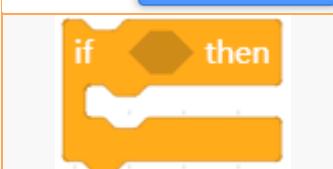
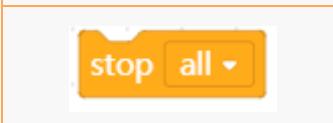
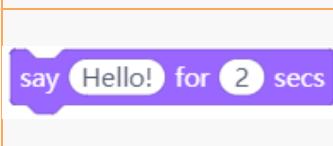
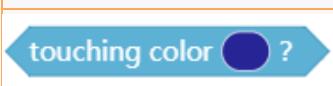
Code Block	Description
	Press button (A/B/A+B) to run the scripts underneath this block
	The horizontal direction is X axis: the right part of X axis, positive; the left part negative. The vertical direction is Y axis: the upper section of Y axis, positive; the lower section, negative.
	Change the value of the point in direction to make the sprite face toward different direction. 0 to 180 are on the right side and 0 to -180(up to down) are on the left side. 180 and -180 coincide with each other. 90 is horizontal right direction, while -90 is horizontal left.
	single branch condition block. If the condition is true, run the scripts inside it.
	Dual branches block. If the condition is true, run the scripts inside the "then" block, if not, run the scripts included in the "else" block.
	Stop all the programs
	The block displays a speech bubble with the specified text for the sprite that runs it, which appears on the screen for the specified amount of seconds.
	Check if the sprite is touching the mouse-pointer, edge or another sprite.
	Check if the sprite is touching a specified color.

Chart 3-1

### 3. Program the apple



Figure 3-4

**The position of the apple:** the value of y is unchanged, and **pick random 1 to 10** the value of x is random. Here we use the random function block. The block picks a pseudorandom number ranging from the first given number to the second, including both endpoints.

**The state of apple:** the value of y constantly minus 1 to realize the effect of falling.

**Interrupt detecting:** if the apple hits the cat, it will be hidden.

**Control the apple to appear and disappear:** the apple appears from the up-side and disappears when it arrives at the bottom.

**The state when the apple is caught:** when the apple hits the cat, it disappears which means that it was took by the cat.

**Falling state control:** since in the stage area, the value of Y ranges from 180 to -180, then the program needs run 360 times. The value of Y minus 1 every time.

#### 4. Program other apples

We can directly duplicate the sprite apple and its codes. However, when there are 4 apples to be appearing on the stage, what will happen? Since only the value of x is random, that is to say, these apples will fall from high at a same time. To make the game look much more realistic, we need to set different initial falling time for the apple, how to do that? There are many acceptable ways. Here we decide to realize the effect by adding more "wait" blocks before "display". And we can adopt random functions for the wait time. Just like this:

**wait pick random 1 to 5 seconds**

The rest apples' program is shown in figure 3-5.

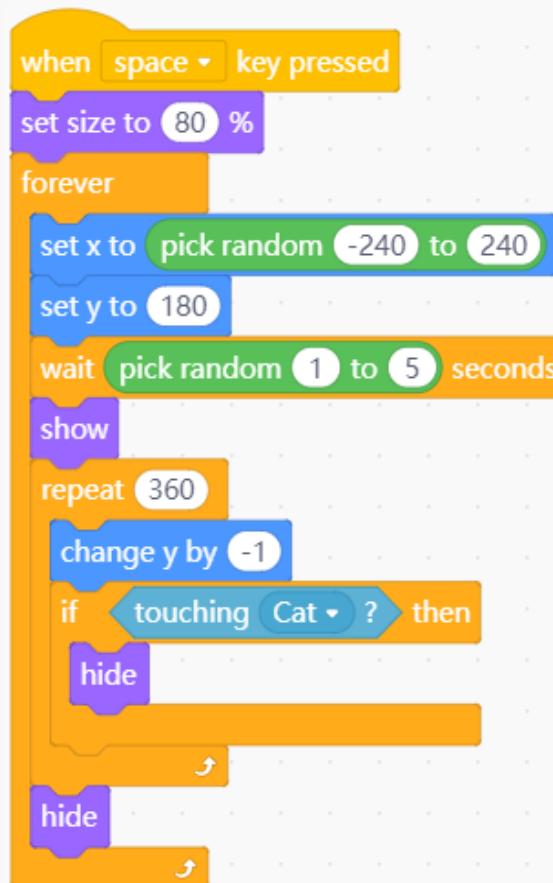


Figure 3-5

Duplicate the sprite and code orderly, as shown figure 3-6.

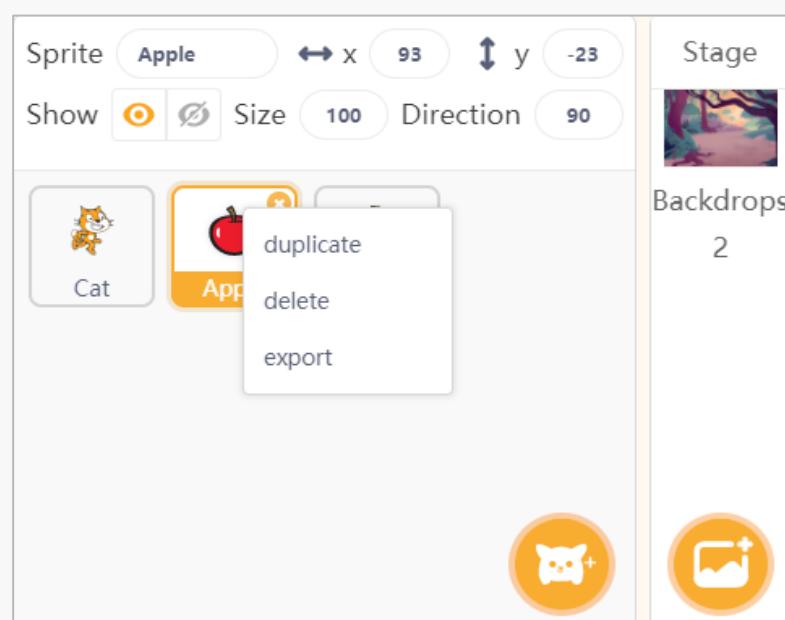


Figure 3-6

## 5. Program the lighting

What's the difference of the codes between the lightning and apple? The program of the lightning is shown below.

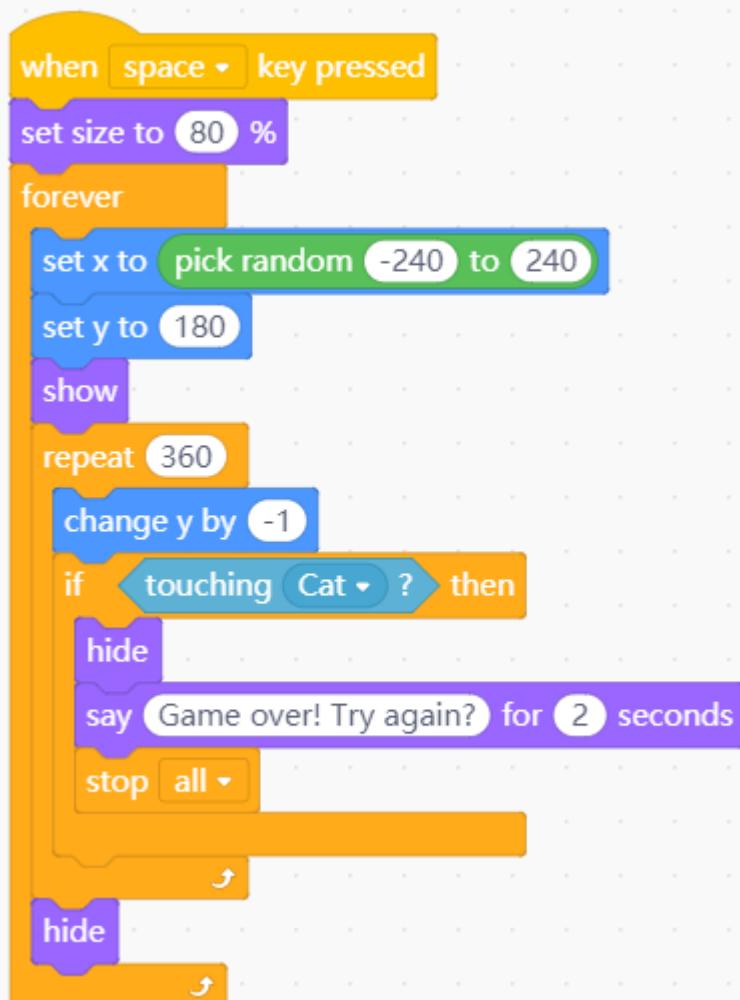


Figure 3-7

## 3.3 Programming Result

We can control the program through the buttons on the micro:bit now. Shown as below. Run the program, and check the result. Think what improvement we can make about this tutorial.

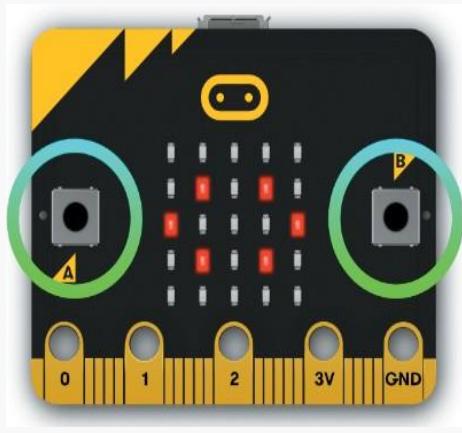


Figure 3-8



Figure 3-9

### 3.4 Training Camp

What if we make a two-player mode for the game as shown in figure 3-9. One player uses the left and right key to control the sprite's movement, another controls it by the buttons (A and B) on the micro:bit board, and then make a competition. Well, if so, what revisions should be made based on this program?



Figure 3-10

## Chapter 4. Cat and Mouse

It is not fun to play games without scoring! Well now, we will create a game and add score system. *Cat and mouse* is such a classical game! Today, we are gonna make a game like this, but there is something special about our main character cat, he does not need to run, but only stay where he is, turn around, and constantly fire hidden weapons to hit the scampering mouse.

Players get 1 point if the mouse gets hit once. The one who get the most scores within the given time will be the winner.

In this tutorial, we will also learn variable and the micro:bit accelerometer.

### 4.1 Accelerometer

Accelerometer, also called gravity sensor, is widely used in our daily life, for instance, gradienter, passometer in the smart phone. The accelerometer on a micro:bit measures acceleration as its name suggests or detects the movement along three axes. Get the data of these three axes, then the micro:bit board's gesture such as moving, shaking, tilting or free falling can be detected.

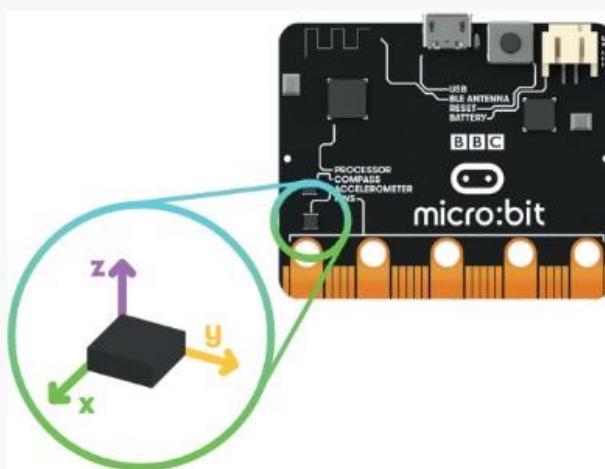


Figure 4-1

## 4.2 Algorithms Analysis

- There are three sprites in the game: cat, mouse, and ball. The ball is the weapon the cat uses to hit the mouse. The mouse has two shapes, we can choose one as the state when it gets hit.
- Tile the micro:bit board to control the cat's movement;
- The mouse moves randomly in the stage, if it touches the edge of the stage, it will point to the opposite direction from which it was coming;
- The ball is the cat's hidden weapon. It is placed under the cat. When button b is pressed, the ball moves forward until it touches the edge. If the ball hits the mouse, the player gets one point, at the same time, the mouse changes its shape and the screaming sound is played.

## 4.3 Programming

1. Add the sprites mouse, cat and ball, and set the size to 50%, 70% and 40% respectively. Then add a background for the stage. Shown as figure 4-2.

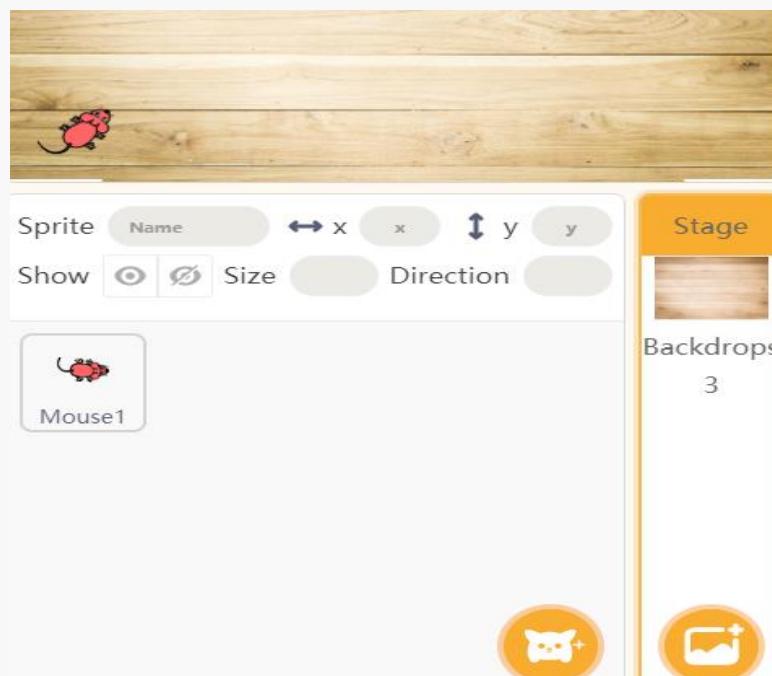


Figure 4-2

## 2. Shape design

Click the sprite mouse, we can find there are two shapes, mouse a and mouse b. Here we set the mouse b as the state when it gets hit. To make it obviously, we can fill the mouse's body and head with red color.

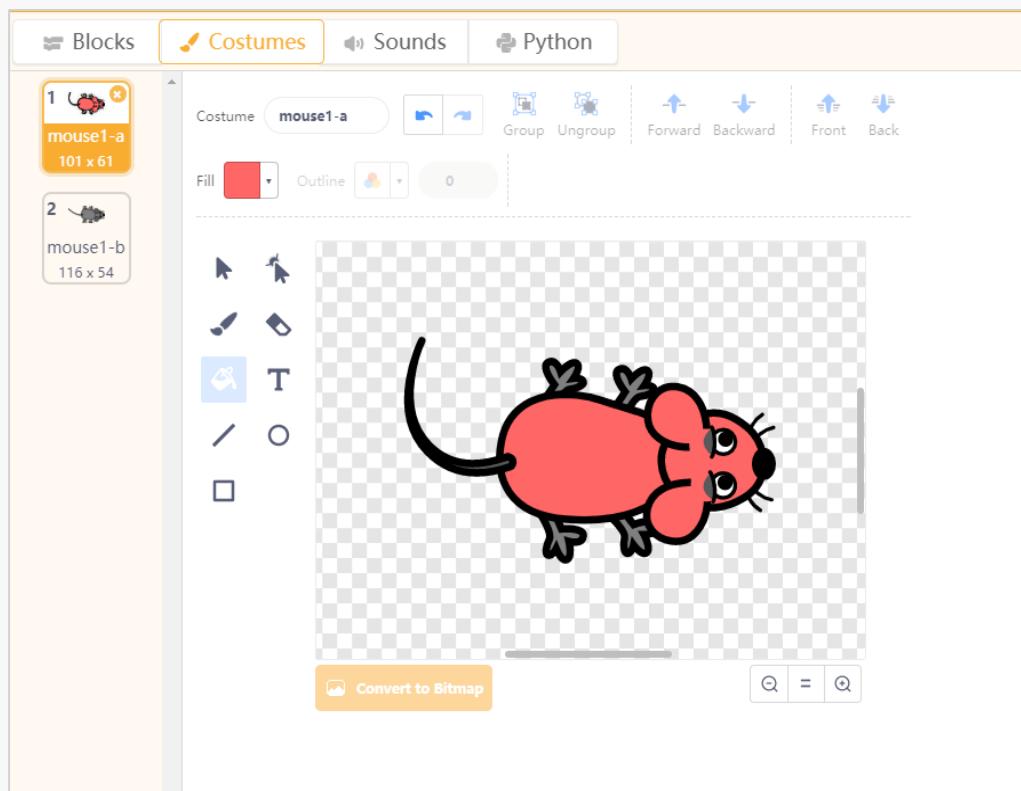


Figure 4-3

## 3. Program the mouse

Figure 4-4 is the program that the mouse runs on the floor. To multiply the routine of the mouse, we add a “pick random () to ()” block to make the mouse move randomly.

## 4. Program the cat initially

The cat's movement is controlled by the micro:bit board. Tilt the board to left, the cat turns left for 15 degrees. Tilt to right, turn right for 15 degrees. Shown in figure 4-5.

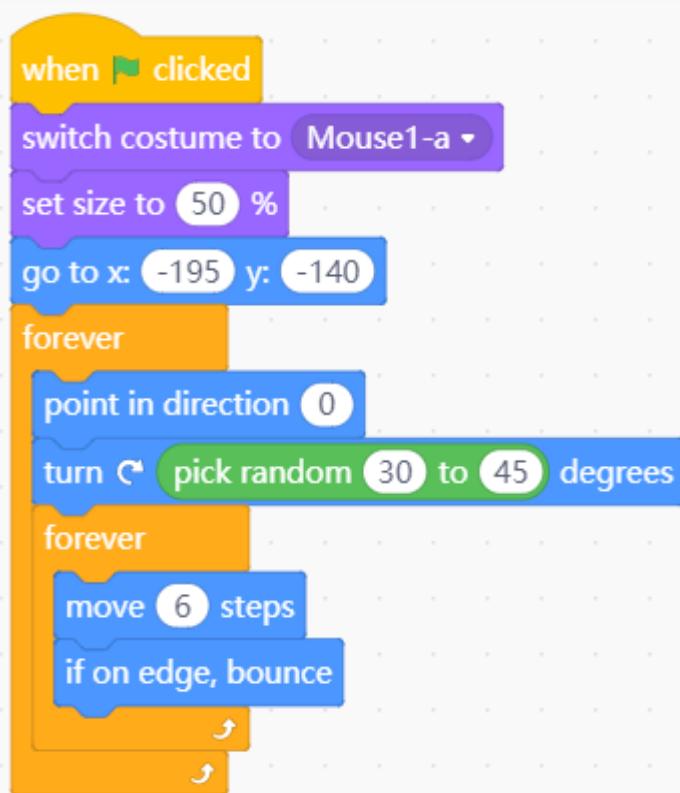


Figure 4-4

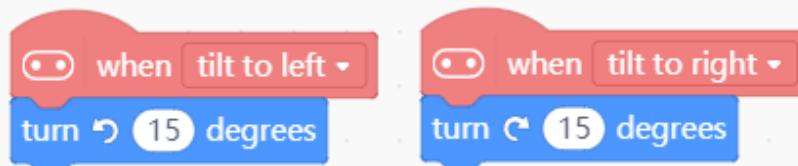


Figure 4-5

Except the gestures of tilting to left or right, micro:bit also can recognize several other gestures.

## 5. Program the ball initially

The initial state of the ball is hidden. The program of the ball will be run in a loop: hidden in the bottom, detecting if the button is pressed, ejecting the ball. As figure 4-7.

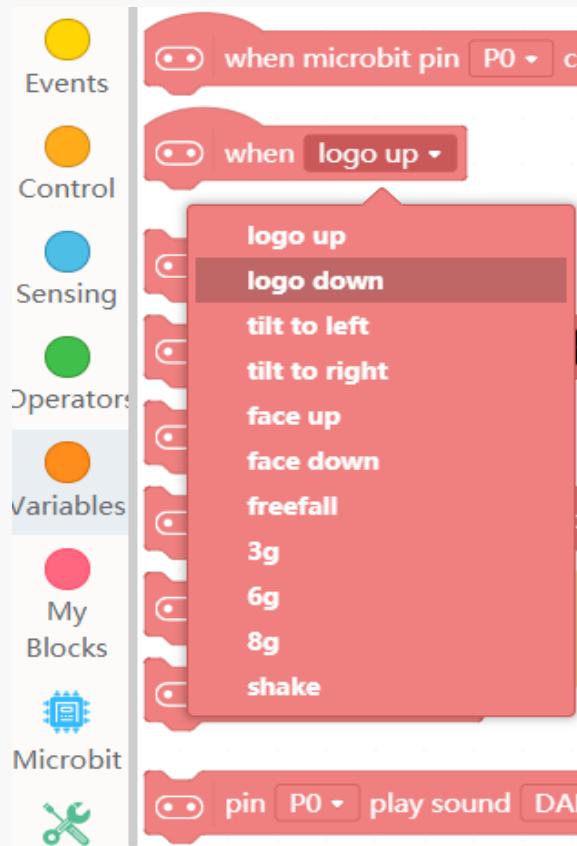


Figure 4-6

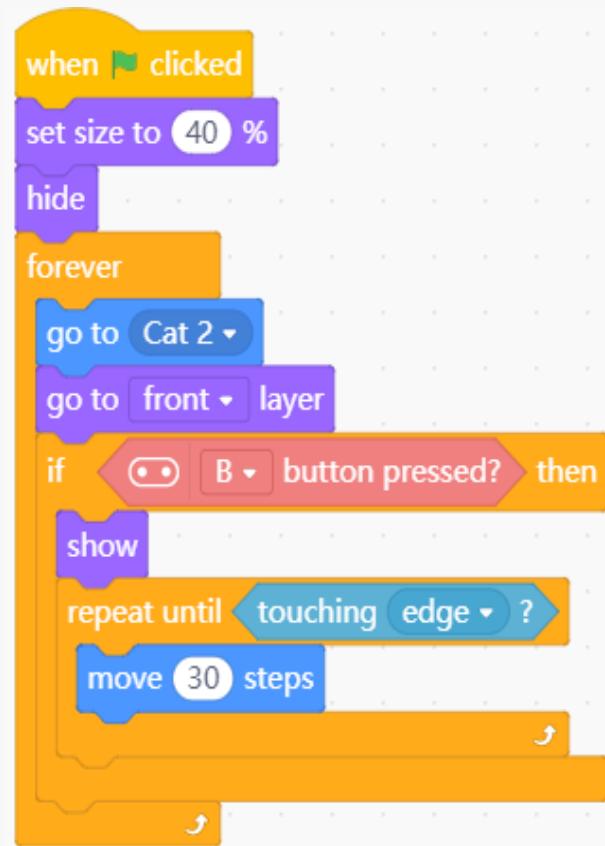
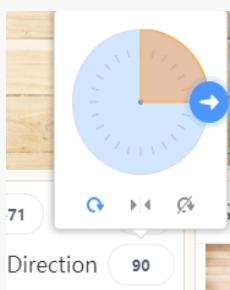


Figure 4-7

## 6. Complete the program

When we test the program, we found that there are two problems about the program of the ball: 1. The ejecting direction of the ball is unchanged but we want that changes with the cat turning around. 2. After the ball hit the mouse during its way, nothing changes. However, we want it like this way: when the mouse gets hit, add one score for the player, meanwhile, the ball broadcast a message “got hit”, then after the mouse received the news, it changes its shapes and makes screaming noise.

How to solve the first problem? —create a variable “direction”. When the cat turns around, we assign a value to the direction, then the ball points to that variable “direction”.



Let's get to know about the "point in direction ()" block. As shown in figure 4-8, the top is 0 degree, and 180 degrees is a semicircle, which points to the down. The left semicircle ranges from 0 to -180 while the right semicircle from 0 to 180. -180 and 180 point in the same direction.

Figure 4-8 The final script of the cat is shown in figure 4-9, tick the checkbox next to the variable "direction" block, then the value of it will be displayed on the left-upper corner of the stage area. Tilt micro:bit board to make the cat turn around, we can see the value of the direction equals to that shown below the stage. Now let's go to the second question. What will happen after the ball hits the mouse? Firstly, we need to add a score system. Set a variable "Score", and its initial value is 0. If the ball hits the mouse, add scores for the player. Meanwhile, broadcast the message "got hit". The program is shown as below.

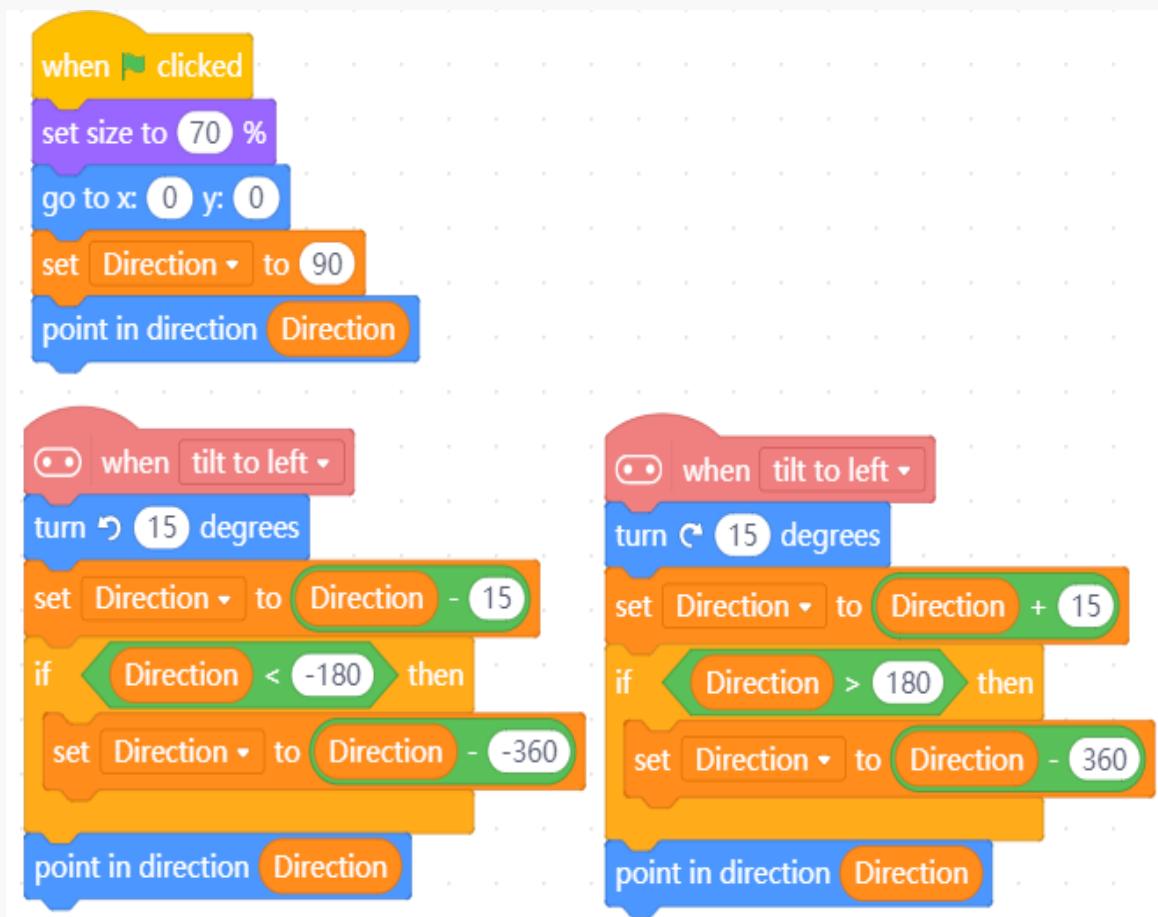


Figure 4-9

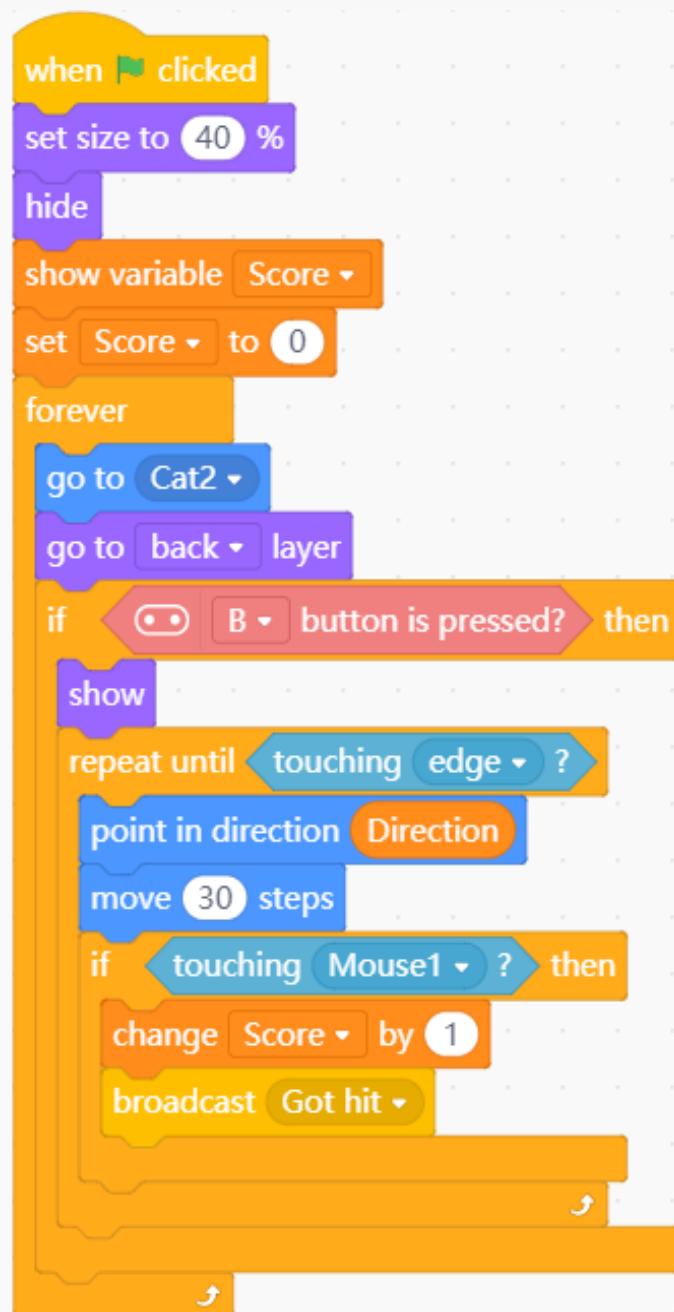


Figure 4-10

Then we need to program the mouse. The ball broadcast the message “got hit”. When the mouse received the message, it will make screaming noise. So, we need to add the sound play block here. Select “scream1” from the sound library. The final program of the mouse is shown in figure 4-11.

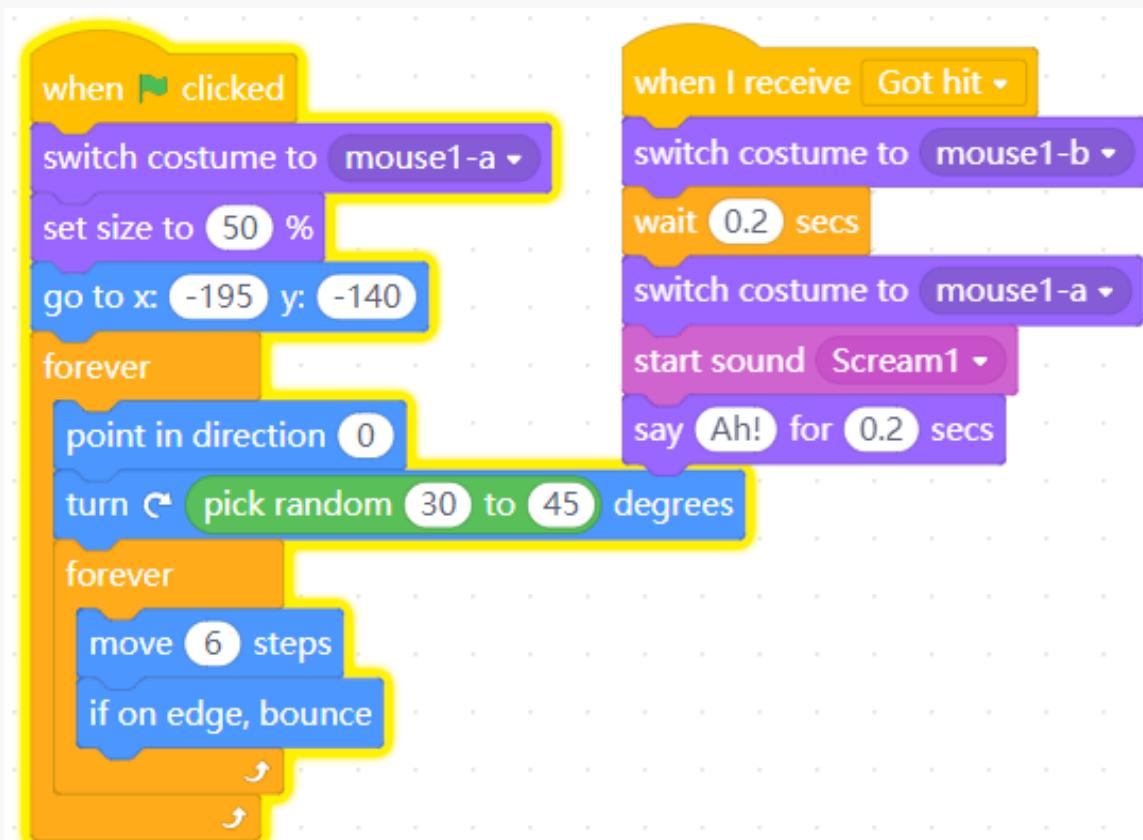


Figure 4-11

## 4.4 Variable and Broadcast

### 1. Variable

We may think of a variable as a container or box where we can store data that we need to use later, and each box can only hold one value (number, text and Boolean data) at a time. For example, use it to store an integer, after we put 1 into it, we put 2 into it, then we can only get 2 from this box. Similarly, if there has been a string “blue” in the variable, when we place another string “red” into it, only the latter can be stored. The name of the box is the variable’s name and the value of the variable is placed inside the box.

Mind+ supports three types of data: Boolean, number and string.

Boolean data type has two possible values: True and False. One of the most common uses for Boolean variable is inside if statement.

**Number type** includes integer and decimal.

**String** is a collection of a series of characters. It can be letter, number (0-9) or any symbols that can be output from the keyboard (+, \*, -, etc.). So, string can be used to store name, address, title and so on.

**List:** after we get known about variable, it would be so much easier to understand what a list is. If we think of a variable as a box with a name for storing data, then a list is a string of boxes, in which each box can hold one value. The string of boxes only has one name.

**Create a variable:** click “Variable” module, a default variable “my variable” can be found in its block area. Create a variable and name it. For instance, name it as “score”. Now we can design our variable: assign, increase by degree, hide or show the variable. Shown as figure 4-12.

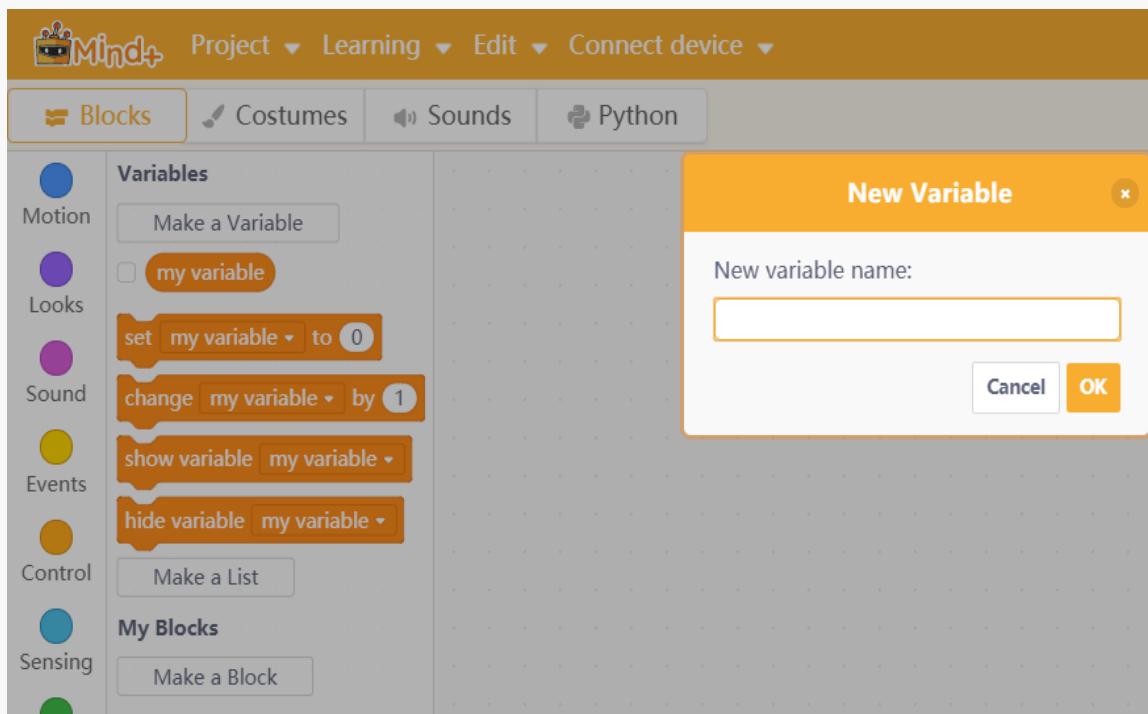


Figure 4-12

## 2. Broadcast

We can use the “broadcast” function of mind+ to order a sprite or several sprites to execute the preset scripts. The function related with “broadcast” can be found in “event” module (Three in total).

Use **broadcast message1 ▾** or **broadcast message1 ▾ and wait** block to send the message you want to broadcast. Click the drop-down box of the block, select “New Message” then input a content into the pop-up box “new message name”. You can input Chinese, English or numerical symbol.

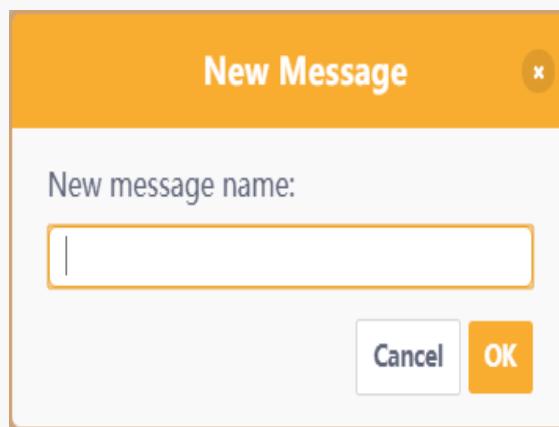


Figure 4-13

Different from the block **broadcast message1 ▾**, **broadcast message1 ▾ and wait** sends a broadcast throughout the whole project-any scripts that are halted with “When I Receive ()” block and are set to the broadcast will activate. It lets scripts send broadcasts, and have them wait until all scripts activated by the broadcast end.

Scripts that begins with this block **when I receive message1 ▾** will be invoked once the specified broadcast has been sent by a calling script.

## 4.5 Programming Result

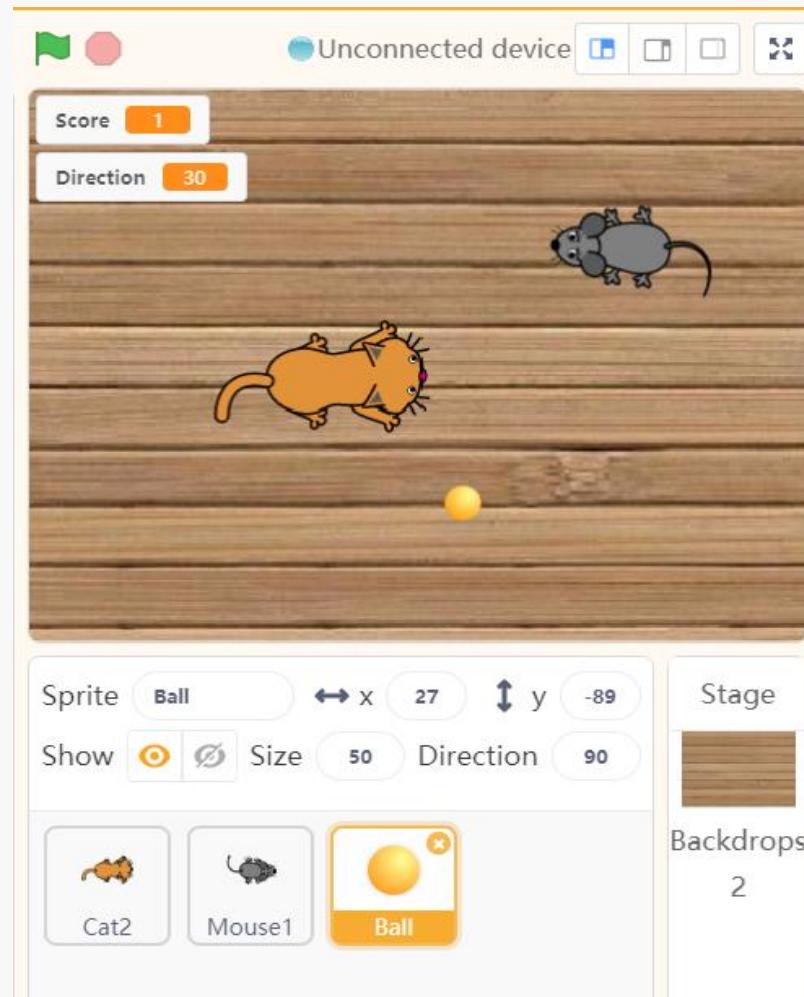


Figure 4-14

## 4.6 Training Camp

We have added score system for the game, but can you make some improvements based on that? For example, imagine what will happen if the mouse encounters the cat accidentally.

How about adding a timing system for the game? Just give it a try!

## Chapter 5. Stress Buster

Have you ever encountered something unpleasant in your life? how did you deal with it? If you want to kick stress out of your life, this stress buster may be a good choice for you. It is a motion sensing game that can help you relieve the negative emotions or stress. Visualize the character in the stage as the people you dislike or the things that made you unhappy, make actions like punch, scratch and pinch in front of the camera, or shake micro:bit board to strike the character and deplete its blood to death.

### 5.1 Algorithms Analysis

- Open your imagination, draw a sprite as shape 1, then draw another picture 2 showing the sprite collapsed, picture 3 and 4 display that sprite got punched;
- When the program starts, display the shape 1, add a variable “blood volume” and set its value to 100;
- When the variation of the camera’s position is over 50, the blood volume decreases randomly, play screaming sound and switch to shape 3. If the blood volume is 0, switch to shape 2 and display the sprite “KO!”;
- Shake the micro:bit board , then the blood decreases greatly, play the screaming sound and switch to shape 4. when the blood is depleted, change to shape 2 and display sprite “KO!”.

### 5.2 Programming

#### 1. Create sprites

Create two sprites: unknown guy and KO. As shown below.

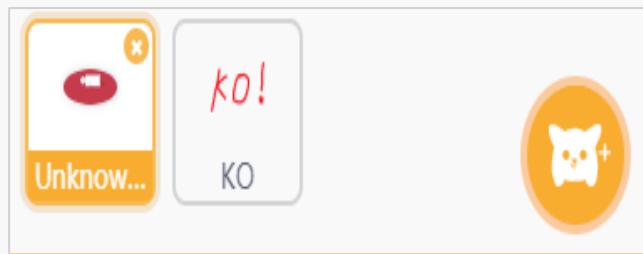


Figure 5-1

Select the character “unknown guy”, right-click “Costumes” to add four costumes (figure 5-2 to 5-5) for him. Costume 1 is the initial state, costume 2 shows the character is struck down. Costume 3 means the man is suffering from minor wound. And in costume 4, the man is severely injured. Shown as below.

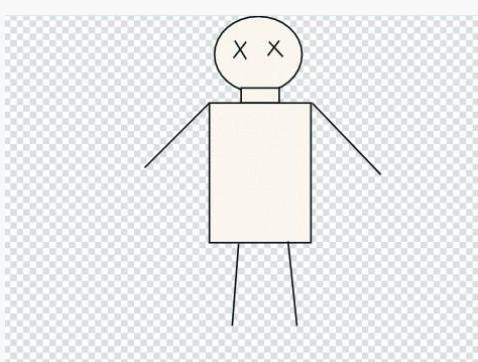


Figure 5-2 Costume 1

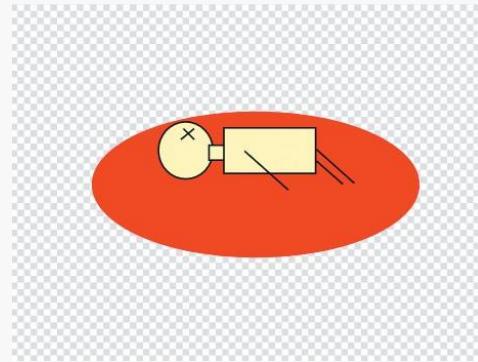


Figure 5-3 Costume 2

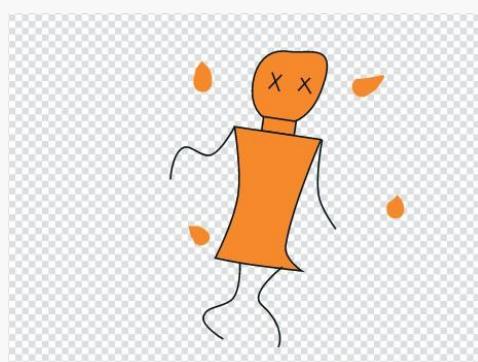


Figure 5-4 Costume 3

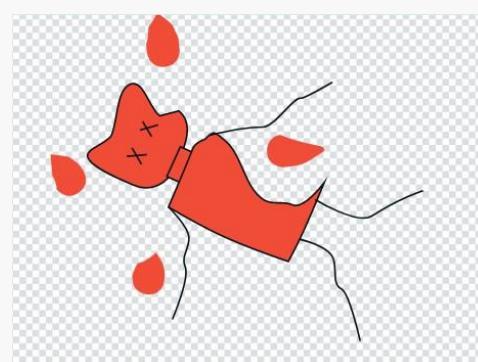


Figure 5-5 Costume 4

Use the paint to create a sprite “KO!” One shape is enough for him.

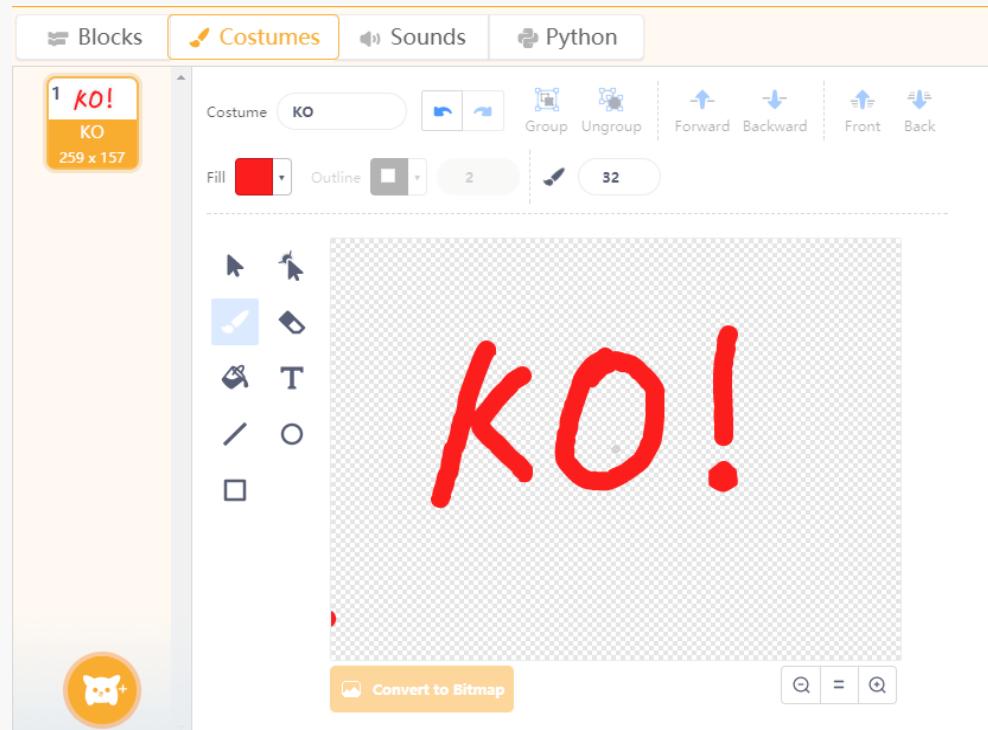


Figure 5-6

## 2. Add sound effect

Open the sound library, add two screaming sound effect. The method is shown below.

Tool	Description
	Import local file from the computer
	Pick a sound effect randomly from the sound library
	Record a sound
	Select a sound from the sound library

Here we record a sound and name it as recording 1. Then we choose another one screaming 1 from the sound library. In the sound interface, we can make revisions for the sound, for instance, make it faster, slower, or add specific features for the sound, even we can cut the sound to make it shorter.

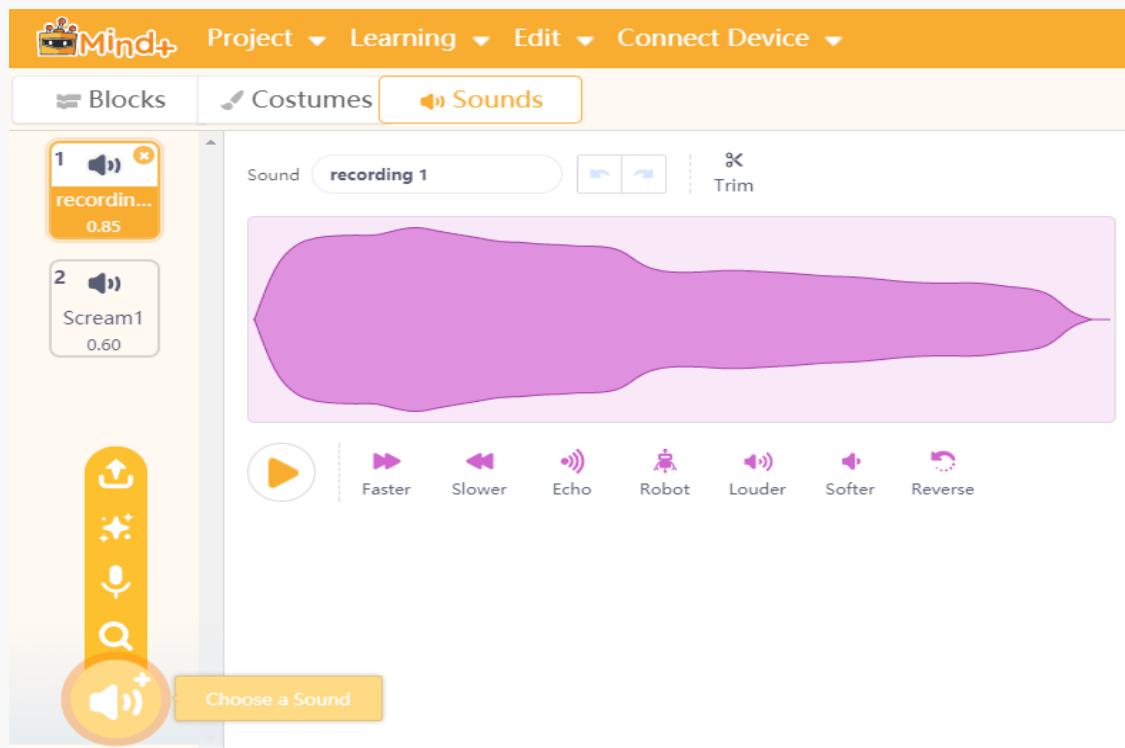


Figure 5-7

### 3. Load camera function

Click “Extensions”->“Function”->“Video Sensing”, back to the previous interface we can see the extended function in the module area. Shown as figure 5-8.

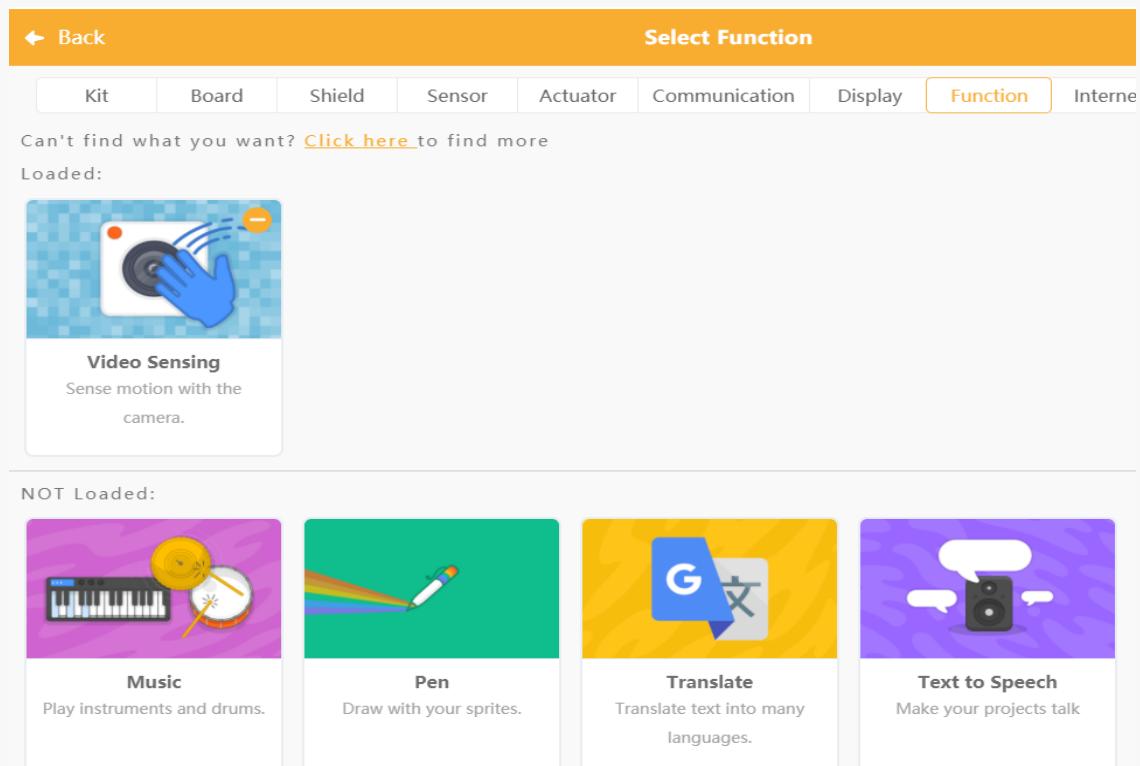


Figure 5-8

Sprite	Blocks
Unknown...	<p>The script consists of three parts:</p> <ul style="list-style-type: none"><li><b>when green flag clicked:</b> Sets costume to "shape 1", sets Blood Volume to 100, shows variable "Blood Volume", and shows the sprite.</li><li><b>when video motion &gt; 50:</b> Checks if Blood Volume &gt; 10. If true, changes Blood Volume by pick random -5 to -10, starts sound "recording 1", switches costume to "shape 3", waits 0.2 secs, switches costume to "shape 1", waits 0.5 secs, else sets Blood Volume to 0, switches costume to "shape 2", broadcasts "Fail", and stops the script.</li><li><b>when shake:</b> Checks if Blood Volume &gt; 15. If true, changes Blood Volume by pick random -10 to -15, starts sound "Scream1", switches costume to "shape 4", waits 0.2 secs, switches costume to "shape 1", waits 0.5 secs, else sets Blood Volume to 0, switches costume to "shape 2", broadcasts "Fail", and stops the script.</li></ul>



Chart 5-2

### 5.3 Program Description

Do actions like wave, punch, scratch, or strike the head in front of the camera. For the position changing range of the camera, we set the boundaries to 50. You may revise it to realize different game effects.

Except the above play modes, you can also shake the board to relieve the stress, by which to accelerate the decrease of the blood volume. The code is similar to that of the above playing modes. But the blood will be depleted more quickly by shaking. Besides, we set a different screaming sound for this mode and adopt shape 3 here.

We need to focus on the details when programming, for example, the value of the blood volume decreases by 5-10 randomly, but when the current blood volume is 4, the blood value will be negative after the last step. Besides that, after the blood of the sprite has been run out, can we still make him make screaming noise and switch shapes by punching? Absolutely not. Therefore, we set that if the blood value is less than 10, the blood volume is set to 0, and in this circumstance, stop playing sound and changing shapes. All programs stop running. We can use "if then... else" function to realize all of these.

Chart 5-3 Block Function Description

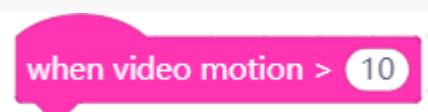
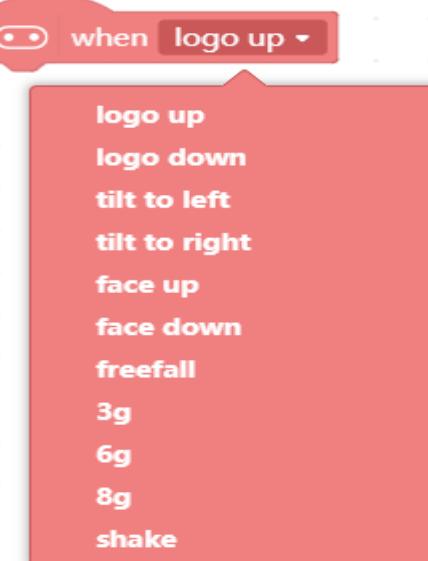
Blocks	Description
	Start the scripts below it when the current video motion on the stage is greater than the number entered.
	Play the recorded sound: recording1
	Select "Expansions"->"Board"->"micro:bit". The accelerometer on the micro:bit board can detect the movements of the board: shake, tilt, free fall and so on.

Chart 5-3

## 5.4 Programming Result

Right-click variable “blood volume” on the stage, and select slider then we can see the change of the blood volume visually. Shown as Figure 5-9.

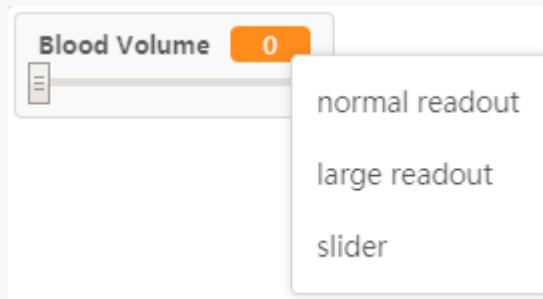


Figure 5-9

Start the game, the blood volume decreases gradually, as the figure 10 and 13 shows.

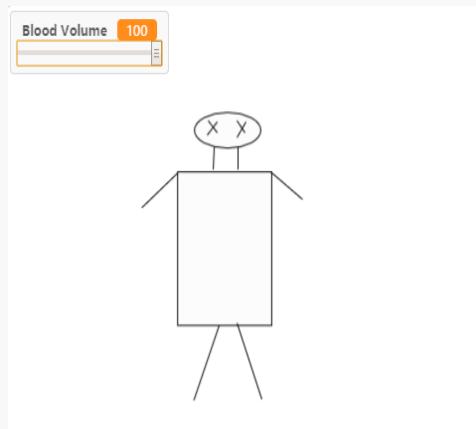


Figure 5-10 Initial State

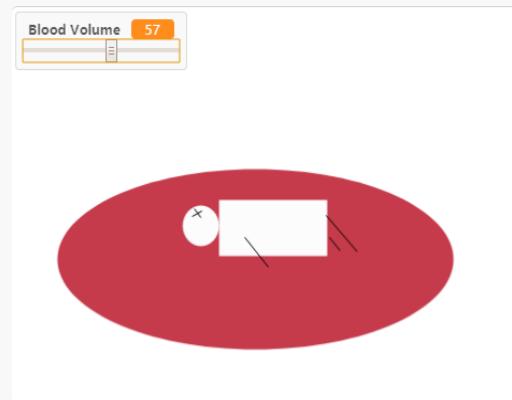


Figure 5-11 Shaking



Figure 5-12 Striking



Figure 5-13 Ending

## 5.5 Training Camp

How about making a survey to ask yourself what made you unhappy before playing the game? (detection->ask and wait). Create a variable, set its value as "answer" and display the variable, that's all! To add more playability, we can also design a part to add blood volume for the sprite, come and try it!

## Chapter 6. Chatbot

Have you ever chatted with iphone's Siri or interact with a smart robot? If not, that's ok, cause now we can directly program an intelligent chatbot by ourselves with mind+ and micro:bit. Isn't that sounds cool! Well, let's begin.

The AI translations will be involved in this chapter, also we will get to know about the temperature sensor on the micro:bit board.

### 6.1 Get to know list

We have already get known about variable, you may find that there is a "create a list" block in the variable module. What is a list ? we know that a variable can be used to store number, char etc and can be refereed repeatedly. Of course, we can also update a variable. In fact, a list is a collection variable, just like a queue. For example, we create a list and name it as "present", then place four items in the present list: crystal ball, music box, lego, model car. We can add items into the list, here we add an item to the end of the list, and then the list would be like this "crystal ball, music box, lego, model car, doll house". Besides, the items in the list can be revised, for instance, change the first item into dinosaur model, then the list would be "dinosaur model, music box, lego, model car, doll house". Apparently, we can delete items in the list. Well now, let's create a list and see what functions will be found in blocks. As figure 6-1 shows, we can add or delete a specified item to a list; replace an item, hide or show a list.

#### Case 1: Santa Claus gives gifts

In this case, we are gonna create three sprites: Santa Claus, a boy and a girl, as figure 6-2 shows. The boy and girl asked Santa Claus respectively "what can I get for my Christmas present?" Santa Claus thinks for a while and then determines to pick a gift randomly from his pocket. But here comes a problem: there are crystal

ball and Transformer in his pocket, so a situation like this could happen: the crystal ball was sent to a boy and the Transformer went to a girl, then it would be awkward! To deal with that problem, we need to classify our gifts into For Girl and For Boy. Cause it could be much reasonable to pick a present from For Girl list when a girl asked for her gift.

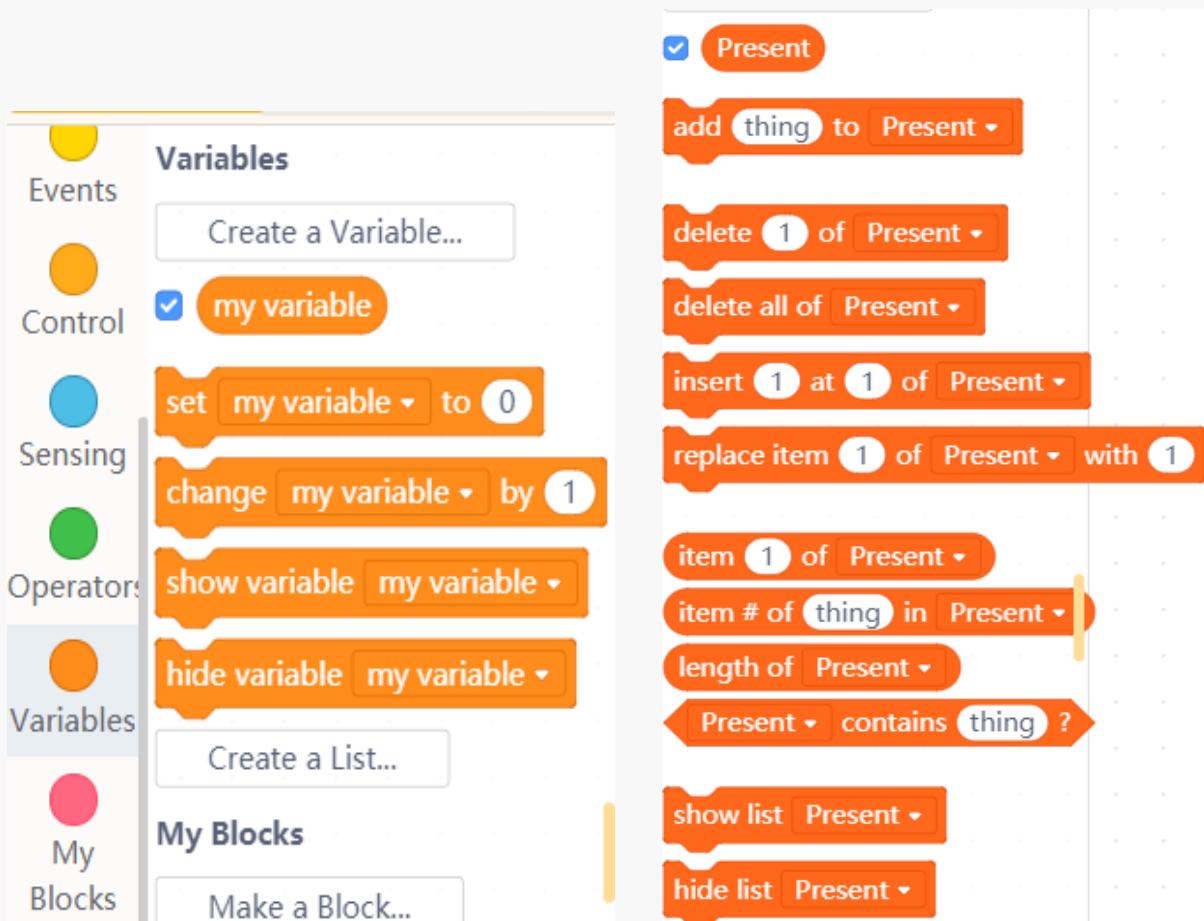


Figure 6-1



Figure 6-2

Sprites	Blocks
 Avery	<p>when green flag clicked</p> <p>say [Mr. Santa, what can I get for my Christmas present? for 3 secs]</p>
 Abby	<p>when green flag clicked</p> <p>wait [9 secs]</p> <p>say [Mr. Santa, what can I get for my Christmas present? for 3 secs]</p>
 Santa	<p>when green flag clicked</p> <p>add [Barbie doll] to [Present for girl]</p> <p>add [Crystal ball] to [Present for girl]</p> <p>add [Paint] to [Present for girl]</p> <p>add [Transformer] to [Present for boy]</p> <p>add [Lego] to [Present for boy]</p> <p>add [Book] to [Present for boy]</p> <p>hide list [Present for girl]</p> <p>hide list [Present for boy]</p> <p>wait [3 seconds]</p> <p>think [Hmm...] for [2 seconds]</p> <p>set [answer] to [item pick random 1 to 6 of Present for boy]</p> <p>say [join You will get: answer] for [4 seconds]</p> <p>wait [3 seconds]</p> <p>think [Hmm...] for [2 seconds]</p> <p>set [answer] to [item pick random 1 to 6 of Present for girl]</p> <p>say [join You will get: answer] for [4 seconds]</p>

Chart 6-1

In this case we learned how to create a list, add items for the list, hide and acquire a certain item from the list and so on. We will use a new block here **join** **apple** **banana**. It can be found in “Operators” module. The block concatenates or “links” the two values together and reports the result.

### Case 2: Chat Language Detection

We will use list to detect chat language in this case. The functions the programs need to realize: obtain a speech the user input, divide it into single character and put them into a list, then check every item in the list.

First of all, we need to learn how to get the number of characters.

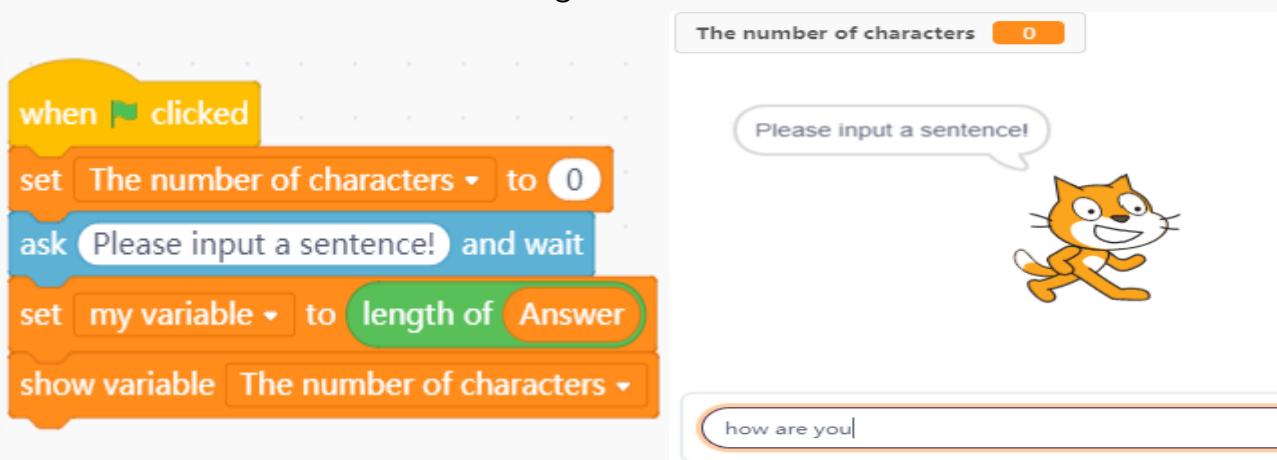


Figure 6-3

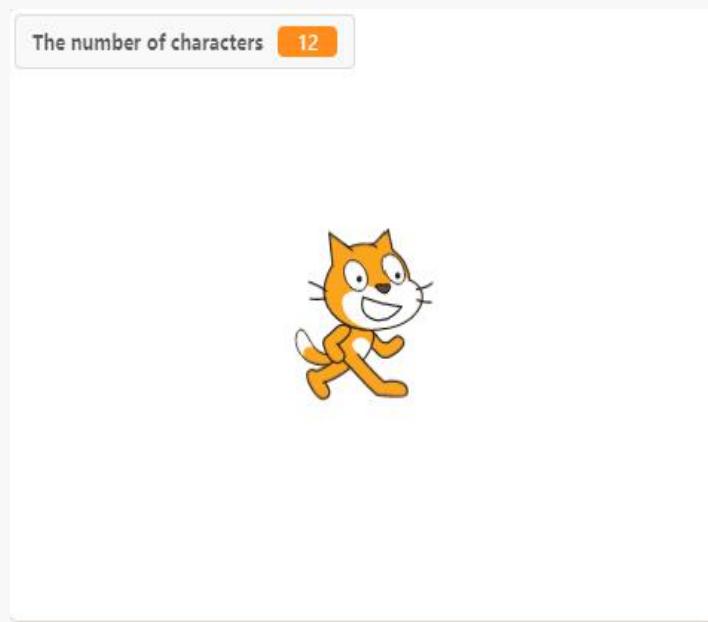


Figure 6-4

Secondly, divide a speech into single characters and place the characters into a list.

Input a speech "I am Micko", the number of characters is 3, and it needs three times to get all the characters. Set a variable "i" to represent the number of cycles. Get the first character in the first loop and so on. The value of i will increase by 1 in every loop. When it comes to the last loop, the value of i will be 4 after getting the third character. Then stops the loop and display the content of the list.

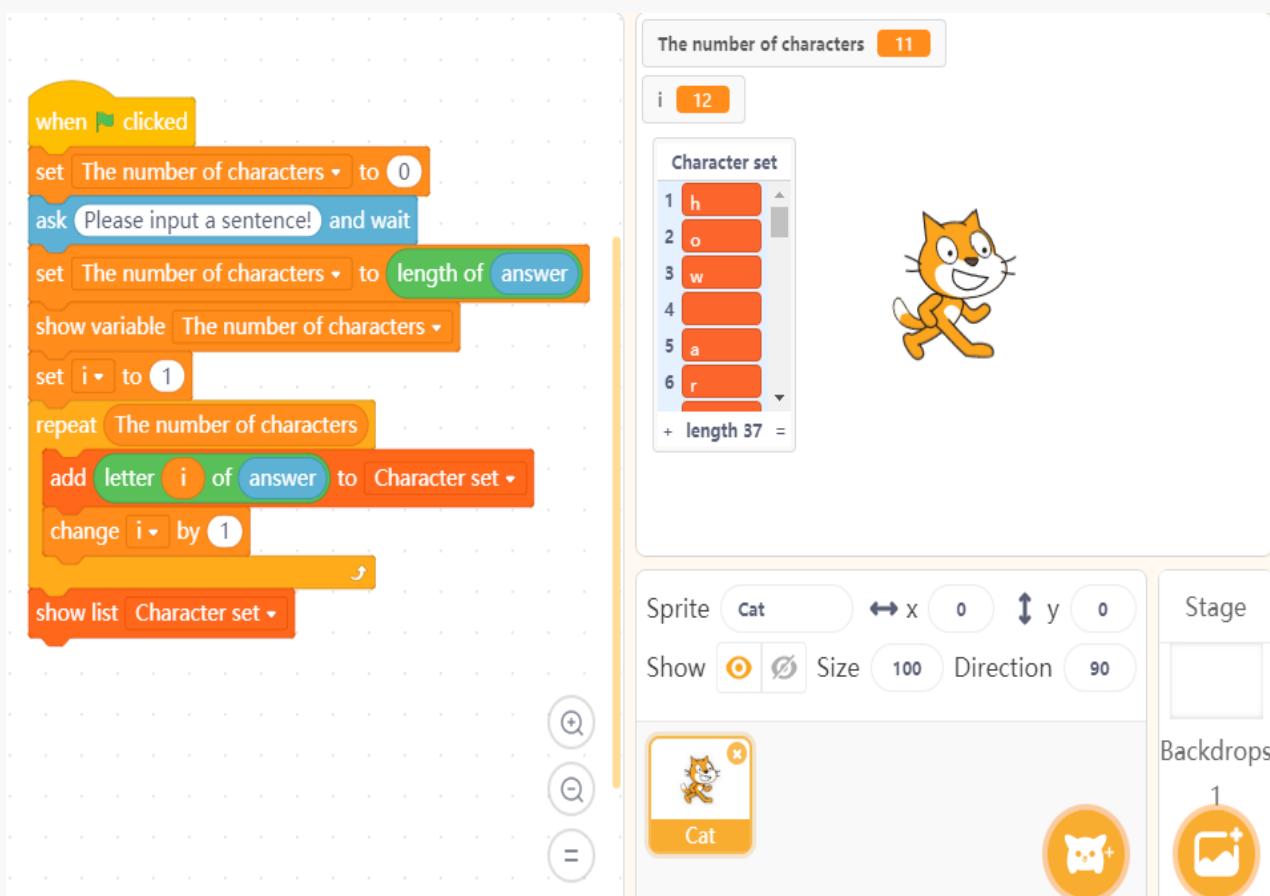


Figure 6-5

### Character Detection

Input "I am Micko" and check the result. So far, we can detect some particular characters of the chatting language.

With the knowledge of list we can make much more fun programs, detect language, chatting room and so on. Ok, now let's start to make the chatbot.

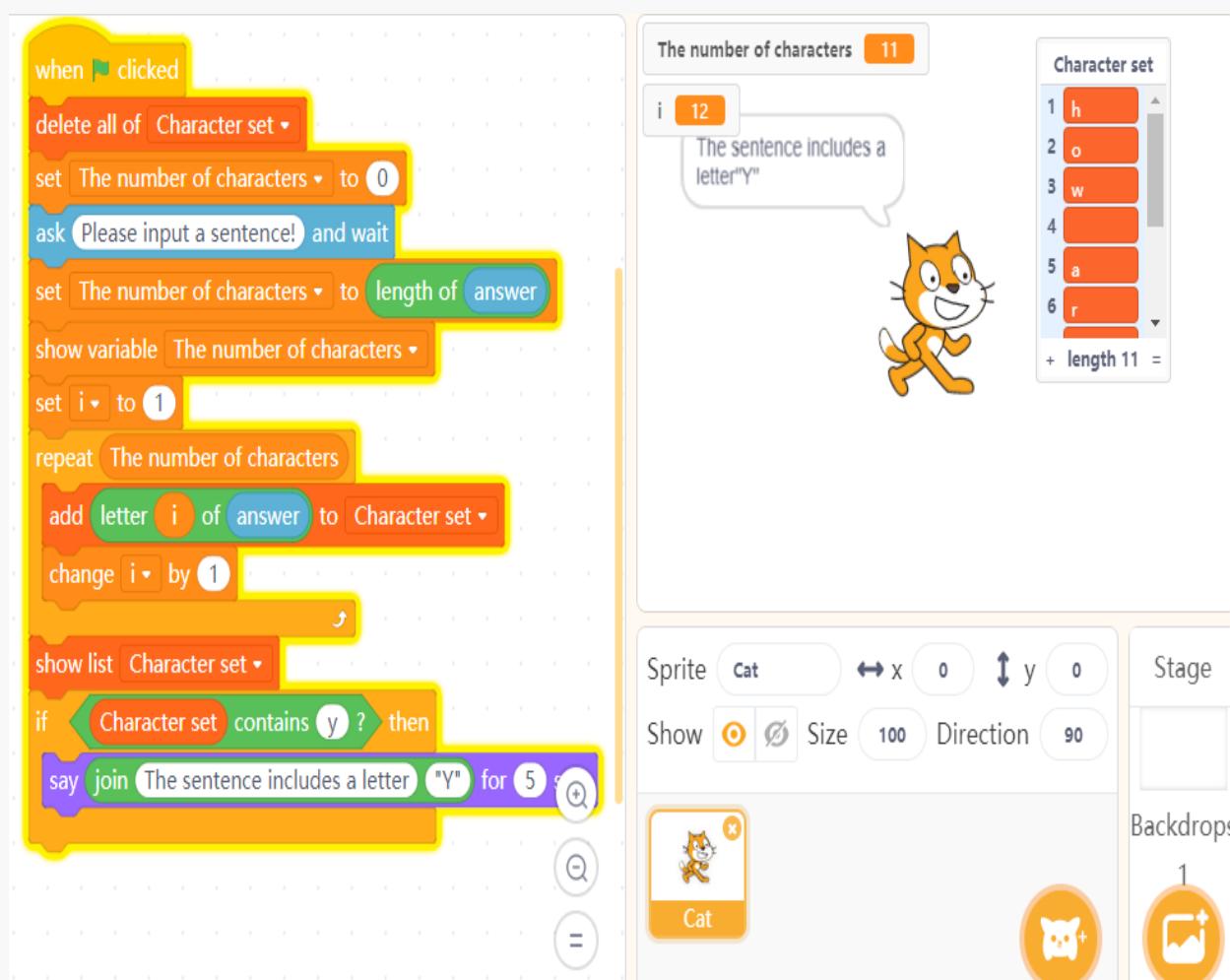


Figure 6-6

## 6.2 Algorithms Analysis

- Create a list “Chatbot” and show it;
- Repeatedly run the program “ask ... and wait”, and “answer”. Add “answer” to the end of the list to make it like an effect of asking question.
- Embed the “if then...else” repeatedly and detect if the words like “time”, “joke”, “song”, “translation”, and “temperature” includes in the “answer”.

## 6.3 Programming

We can ask the chatbot to tell us time, temperature, or let him tell jokes, play music and do translations. The scripts are shown in figure 6-7.

You can play the sound in the sound library, record your own sound or upload a sound file from your computer. We use the sound “dance around” in the loops of the sound library.

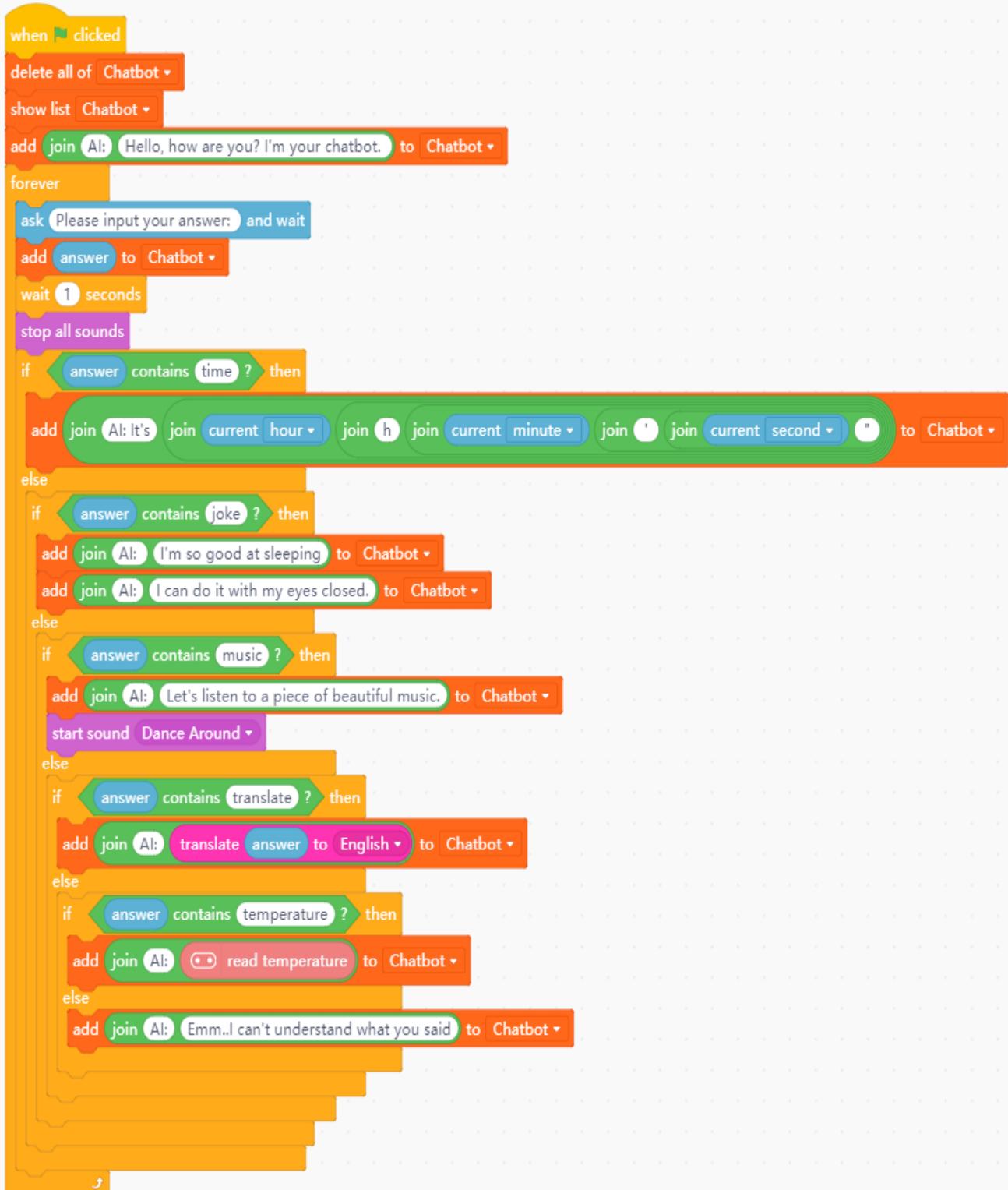


Figure 6-7

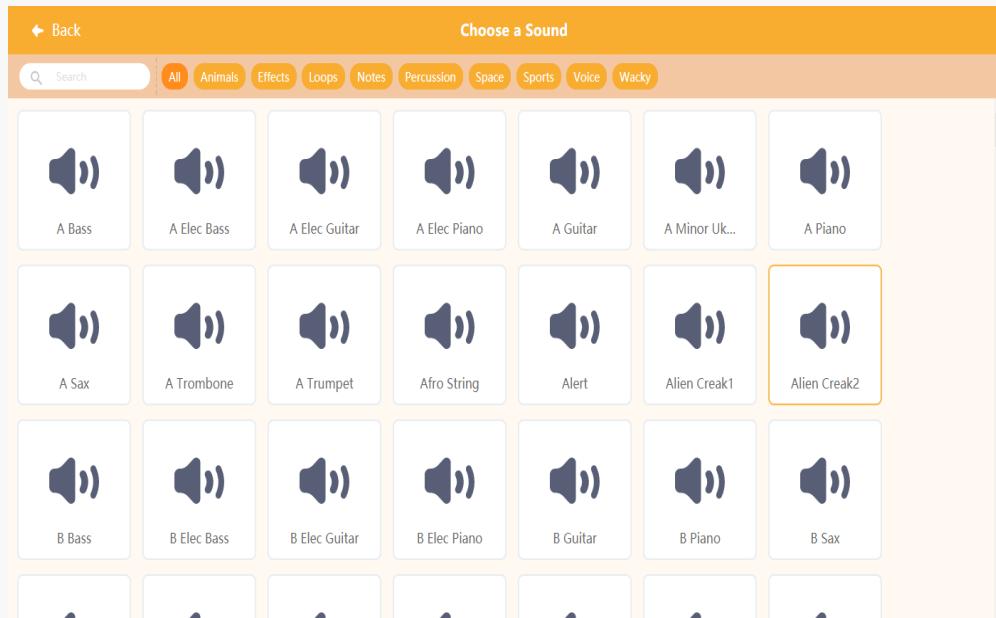


Figure 6-8

Load the google translate: click “Extensions”->“Function”->“Google Translate”. Then we can use the related google translate module.

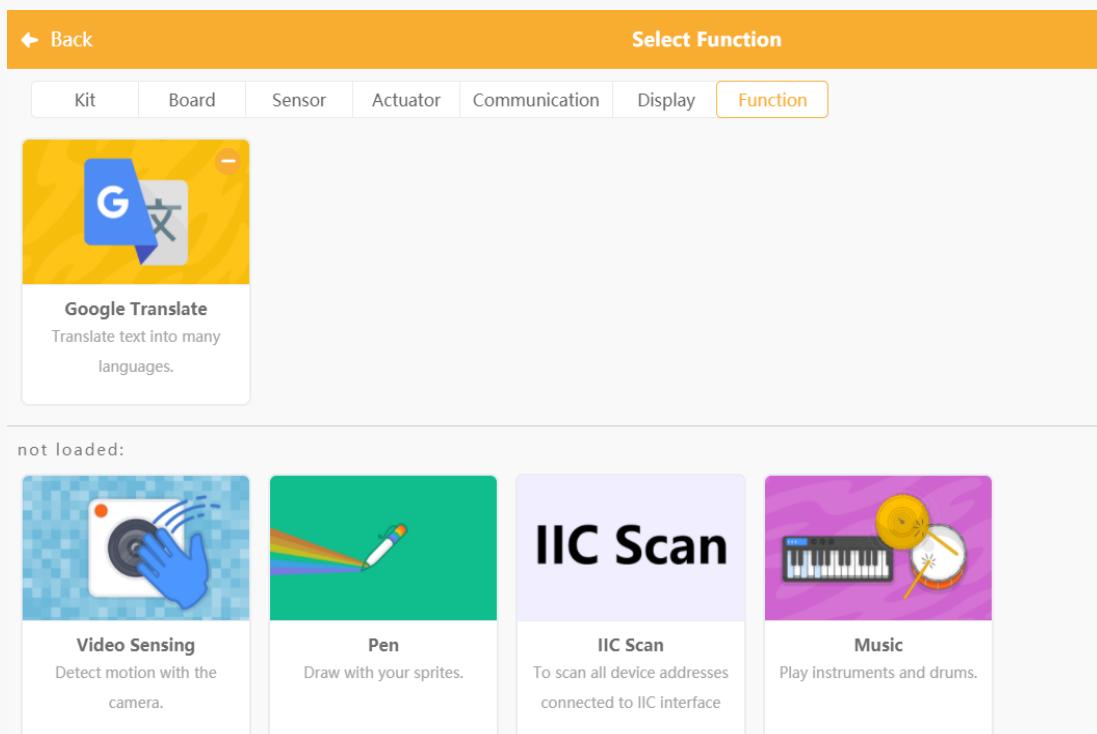


Figure 6-9

## 6.4 Temperature Sensor

Connect micro:bit and the computer with micro USB. The temperature sensor on the micro:bit board will detect the current temperature of the device. (unit: °/°C)

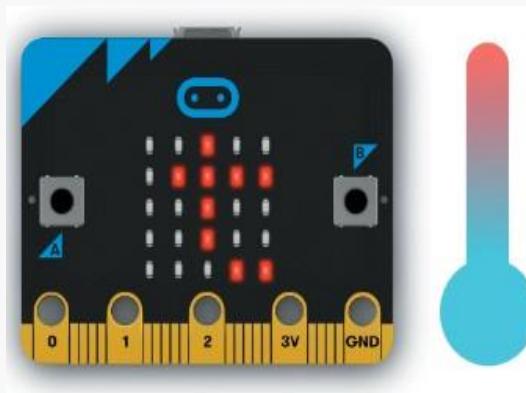


Figure 6-10

## 6.5 Programming Result

Click the green flag to start the program, and check the effect. We can drag the frame of the list to expand the display area.

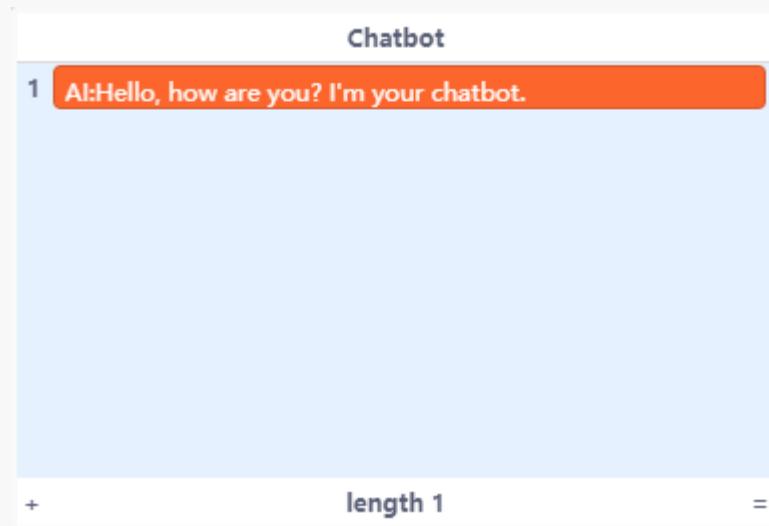


Figure 6-11



Figure 6-12

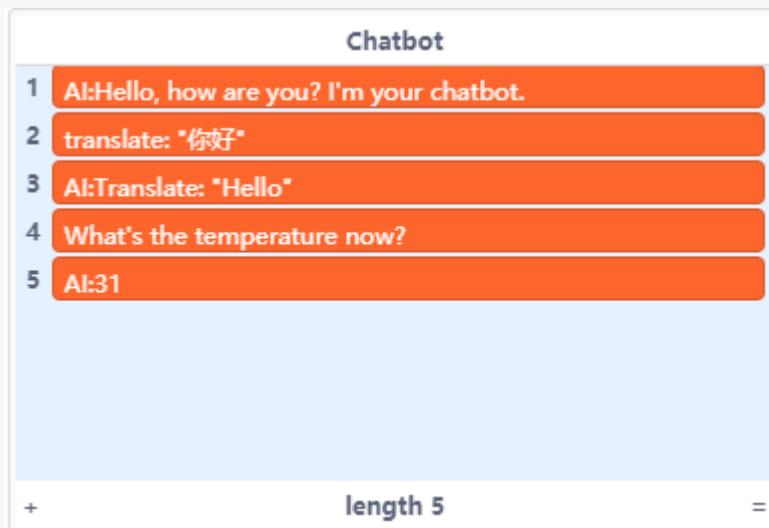


Figure 6-13

## 6.6 Training Camp

Now the chatbot can only realize the function of chatting, but we can combine it with the micro:bit to simulate the smart home, for example, to give orders like "turn on/off the light" to control the LED light on the micro:bit board.

## Chapter 7. Gorgeous Firework

Fireworks are so mesmerizing! The moment fireworks explode in the night sky, the whole sky comes alive with so many vibrant hues, starbursts, and showers of light along with ribbons of smoke, making us happy and awestruck. In this chapter, we are going to display an effect of firework exploding on the stage of mind+.

Gorgeous Fireworks always comes together with a backdrop of night sky. We will use light sensing function of micro:bit to simulate the firework exploding in the night sky.

We also will get to know how to use the light sensor on micro:bit.

### 7.1 Clone

In some projects, you want several identical characters, they must have exactly same motion scripts and shapes. The only difference is their positions. How to do that? Well, the first idea coming to us is to duplicate. Yes, we could first design the motion scripts and shapes of one sprite and then duplicate it as much as we need. But if we want to revise some functions of these sprites, we have to do it one by one or delete all the duplicated sprites, revise the original one and duplicate it again. So complicated, right? But now with the function of clone, this will be just as easy as a pie!

What's the difference between clone and duplicate? We can explain it in this way: duplicating is a step that happens in the period of editing and there will be more sprites appearing on the editing area of sprite and background when you start duplicating, but cloning is a feature that allows to create a clone, or semi-duplicate, of itself while the project is running and there will be only one sprite remaining on that editing area. The "when I start as a clone" block will perform in the clone once it is created. The blocks relating to clone can be found in "Control" module. As figure 7-1 shows.

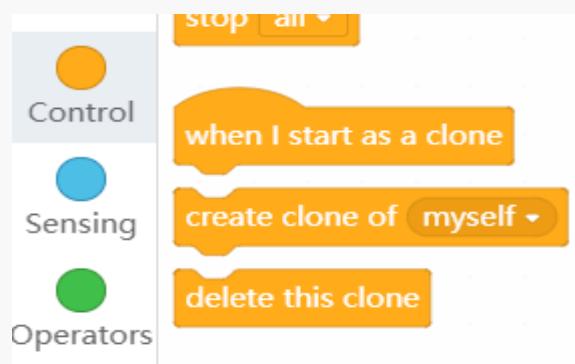


Figure 7-1

Next, in the case of Starry Night Sky, we will learn how to use clone in the actual application.

### Case 1: *Starry Night Sky*

We set the blue sky and star as the background and sprite respectively. Look at the scripts below, we can find that the star is cloned four times. When it starts as a clone, its size gets smaller, and it will move to the position (55, 44). Shown as figure 7-2. The four clones of the star will stay at the same place, drag and move it, the lovely stars will appear.

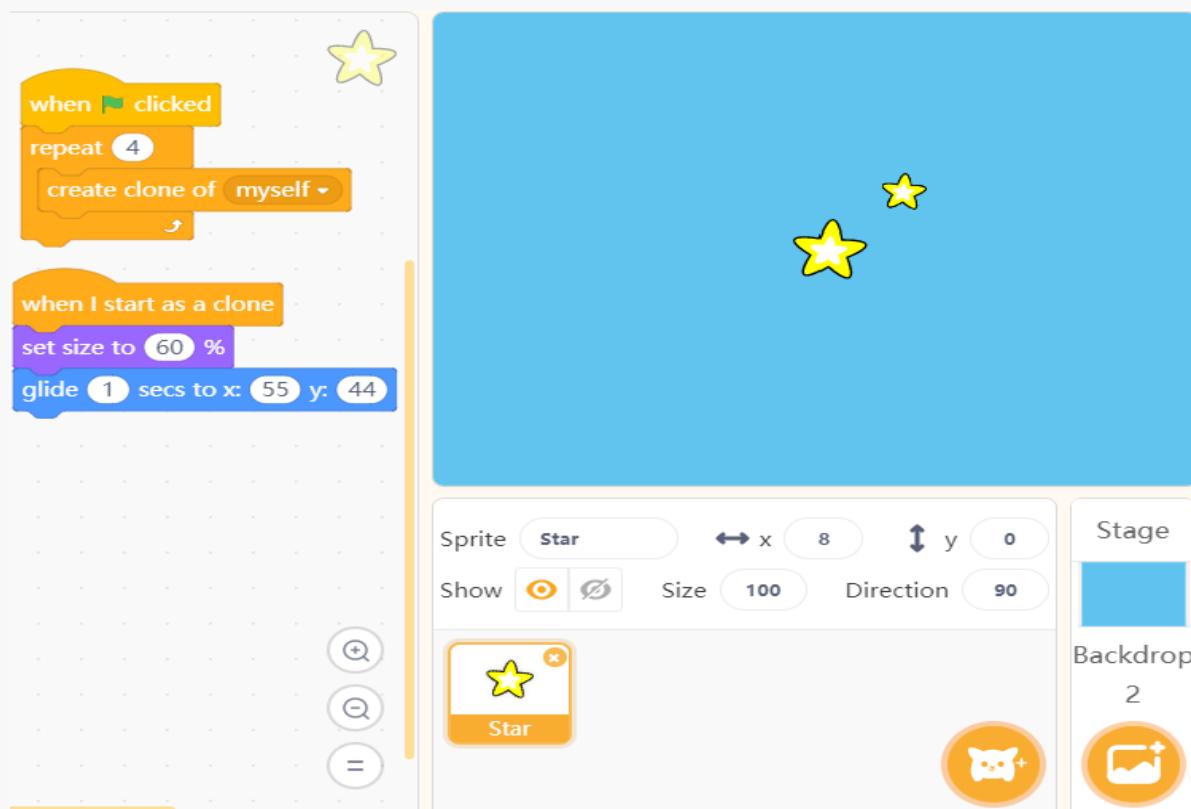


Figure 7-2

How to scatter the stars over the blue sky? Yes, a random function totally works here.

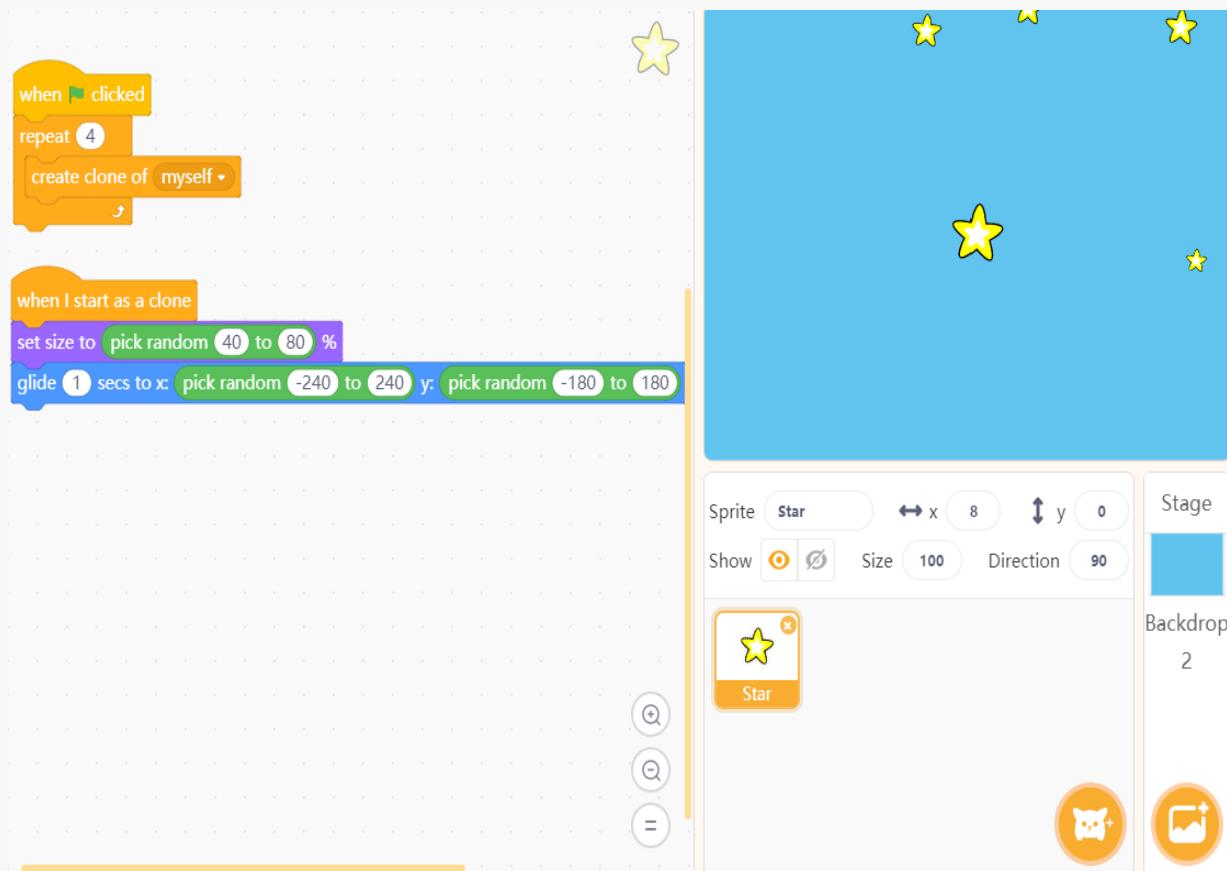


Figure 7-3

At last, we need to make these stars seem to twinkle. Where to place the scripts of this part?

Of course, it should be put under the block "when I start as a clone". Since the parent of clone. do not have the function of shining, we have to use the "delete this clone" or "hide" block to realize that effect.

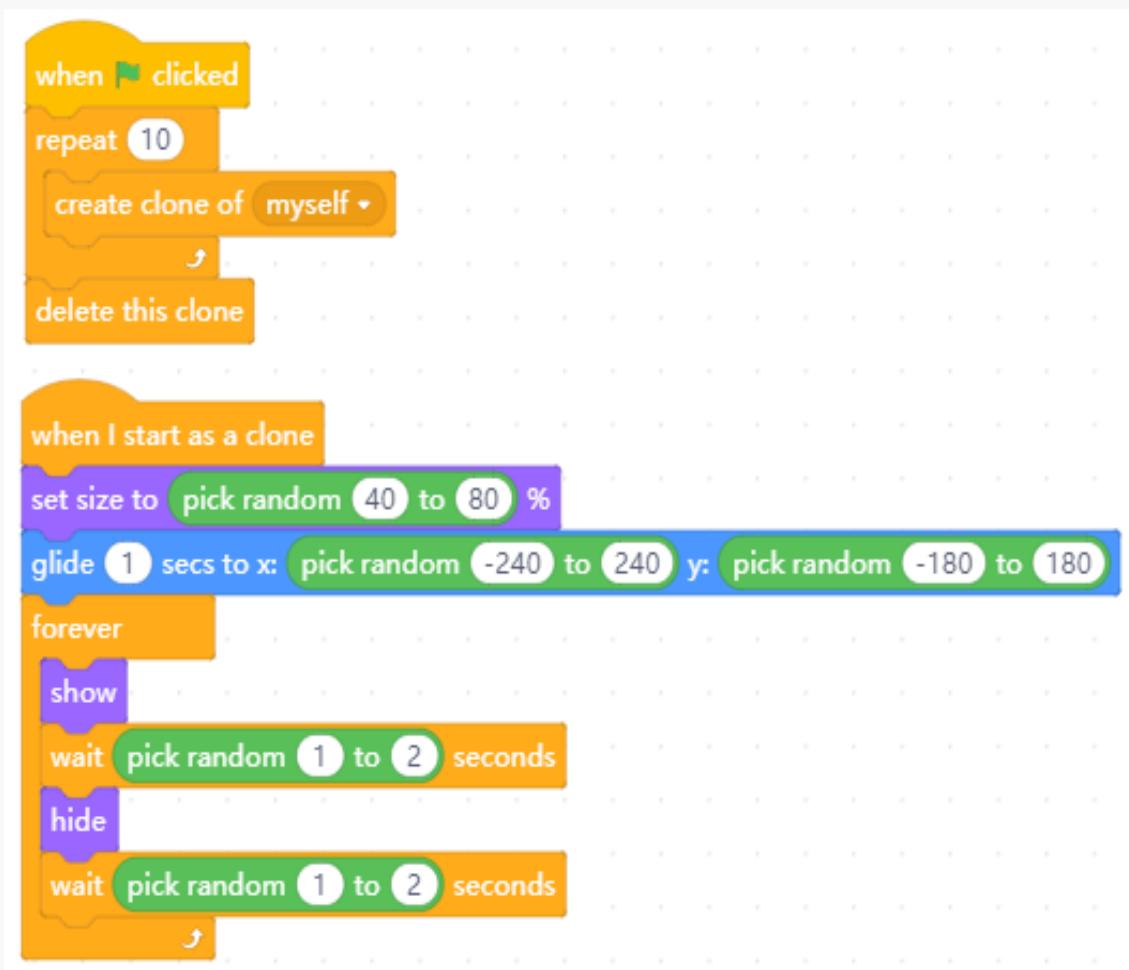


Figure 7-4

### Case 2: Draw flowers

The effect of the program will be like this: click the mouse, then the colorful flowers appears on the stage, as figure 7-5 shows. Well, now, consider if the clone function is needed here. If so, what is the parent of the clone? Take a look at the pictures below, some flowers have one layer of petal while the others have two, so obviously that is randomly chosen. To achieve an effect like that, we have to use the loop function.



Figure 7-5 Flowers

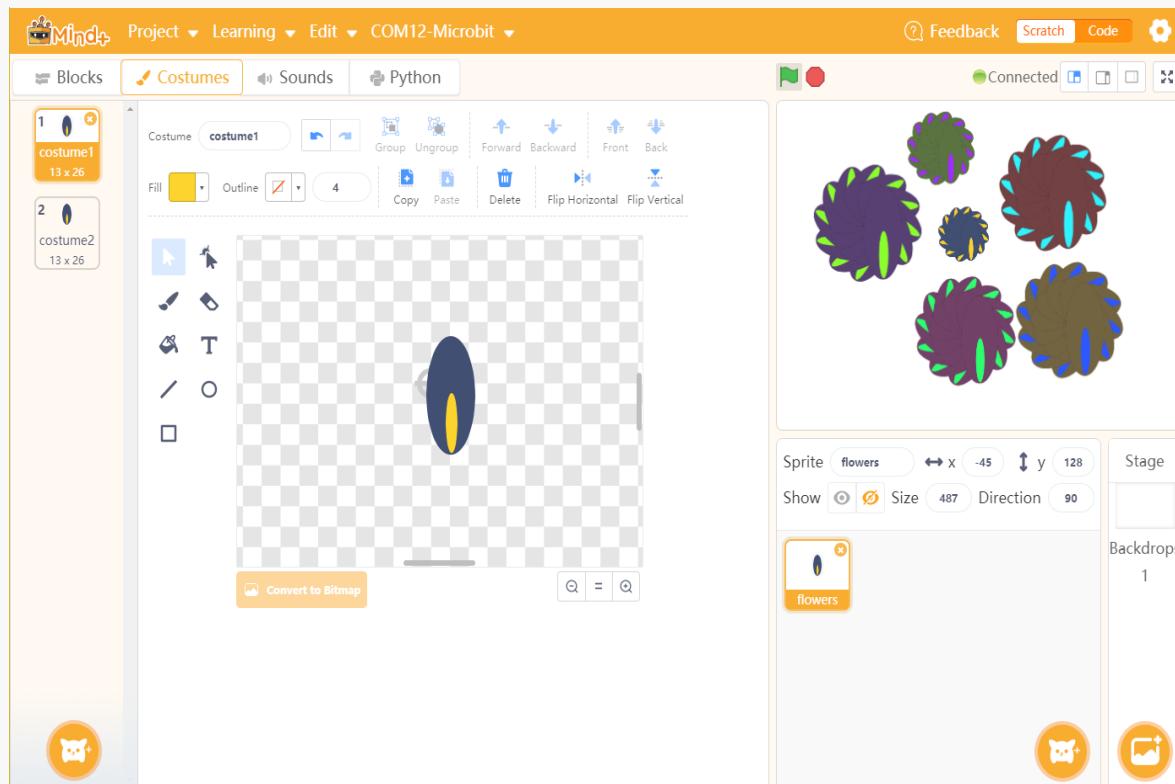


Figure 7-6

Revise the parameters in the blocks below, and check the result.

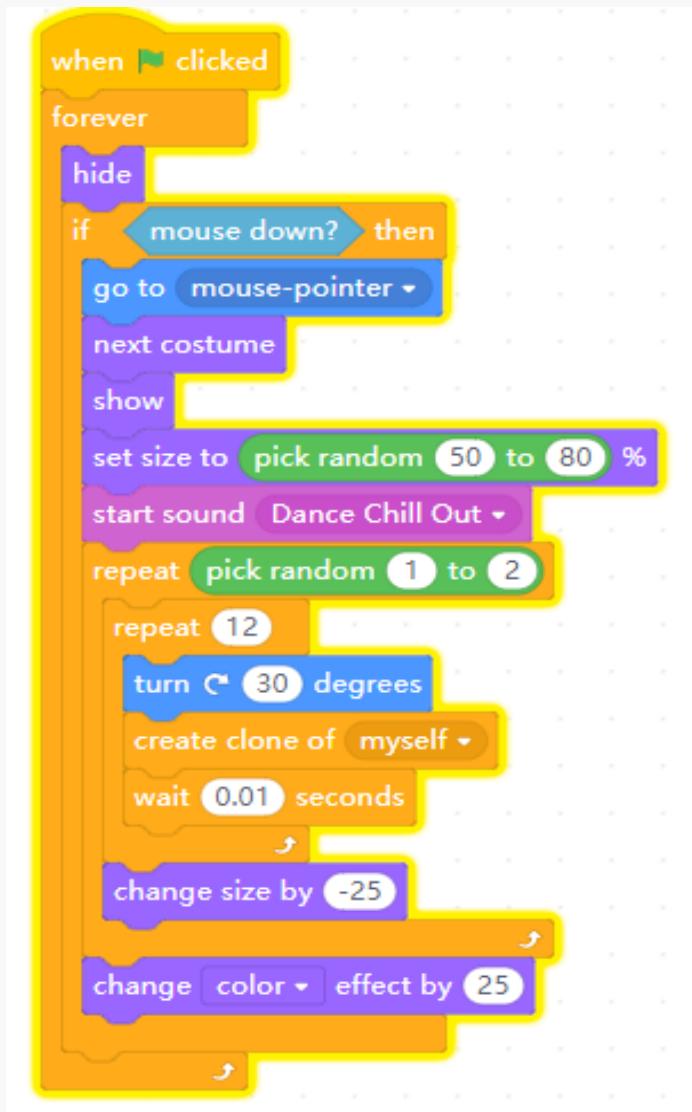


Figure 7-7 Draw flowers

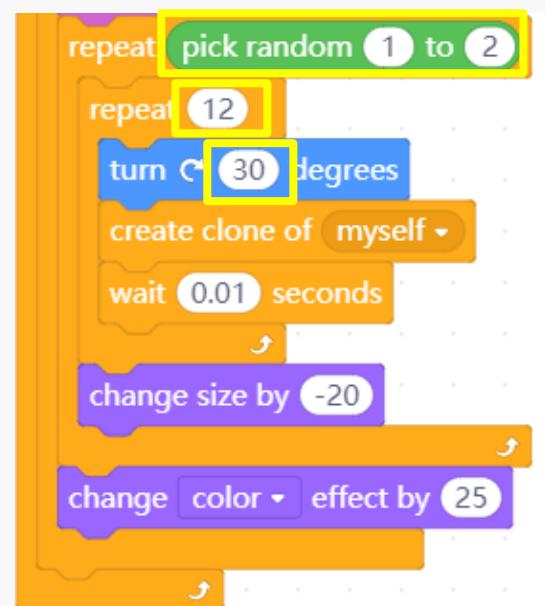


Figure 7-8 Revise parameters

## 7.2 Algorithms Analysis

- Create a sprite “red dot” as the initial shape (shape 1) of the firework. Upload the pictures of firework exploding as the other shapes of the sprite;
- Click green flag, determine if the ambient light is less than 1, if so, the sprite clones itself for 20 times;

- When starting as a clone, the shape 1 slides to a random position from the lower part of the stage, play the sound effect of firework exploding;
- Meanwhile, set size of the sprite to 20%, switch the shape to the rest shapes, repeatedly run the program to enlarge the sprite, add color and ghost effect for the sprite. After one firework exploded, delete the clone to realize an effect of randomly exploding.

## 7.3 Programming

Costumes design: paint a red dot (shape 1) as the initial state of the firework, then upload several fireworks pictures as the rest shapes. As figure 7-9 shows.

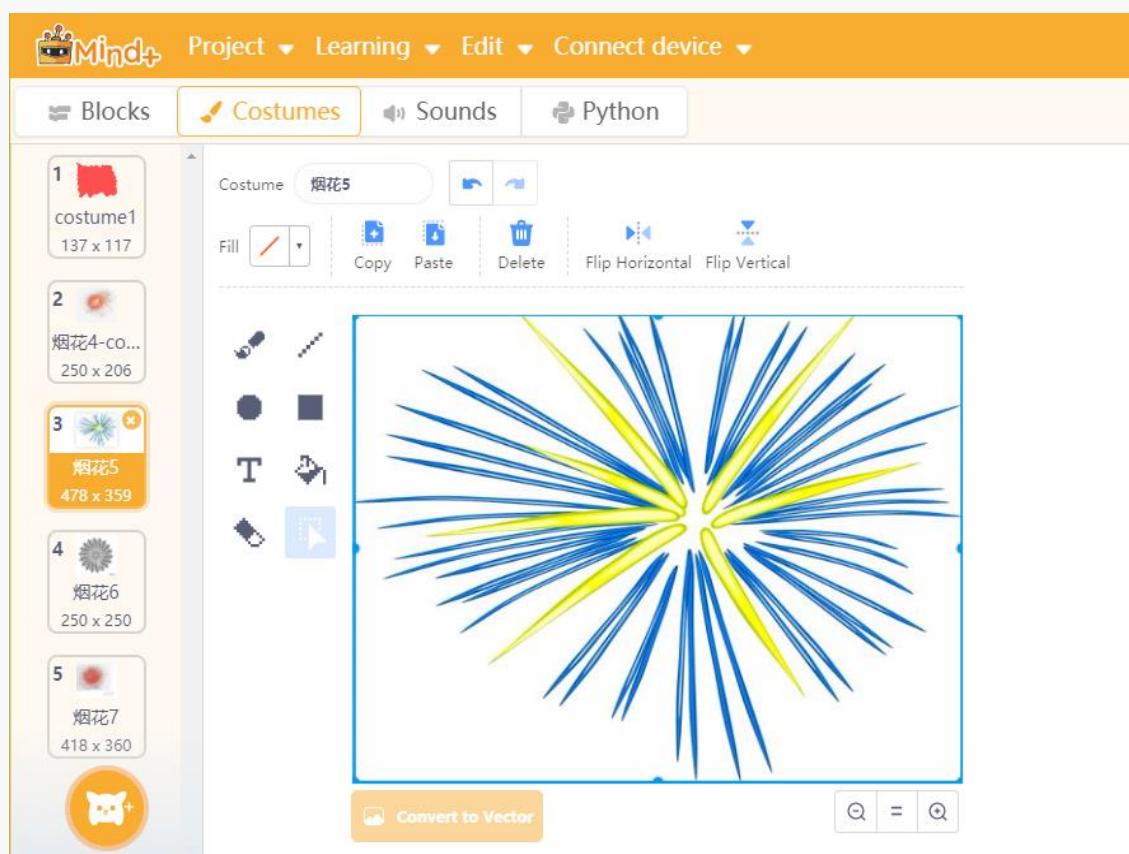


Figure 7-9

Sound Design: search for the sound of fireworks exploding on the internet, upload it into Mind+. You can trim the sound to make it shorter.

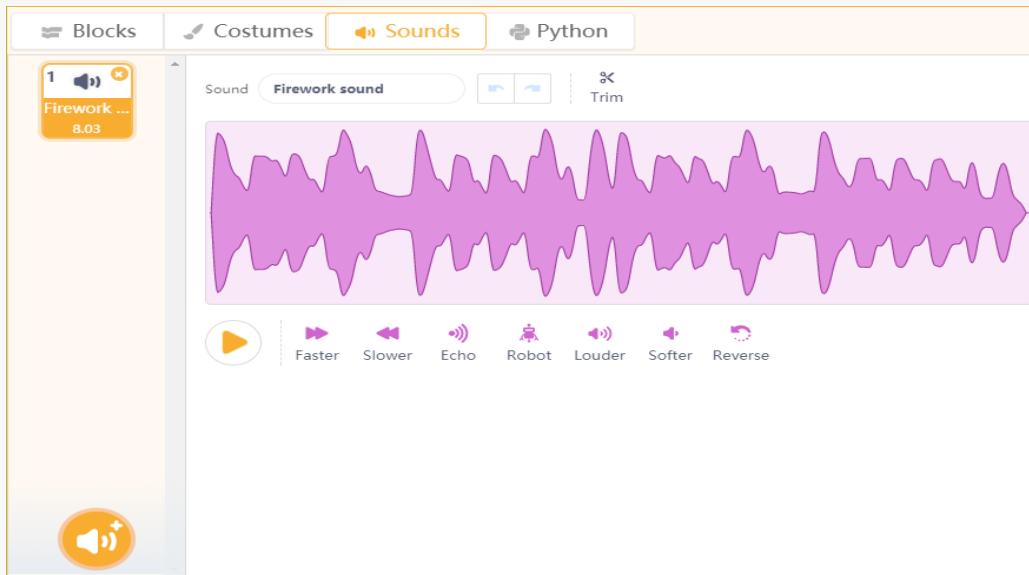


Figure 7-10

The details of the program are shown below.

Sprites	Blocks
 1 Firework 137 x 117	<pre>when green flag clicked   hide   switch backdrop to [Stars v] forever   if [read environmental brightness &lt; 10] then     repeat (20)       create clone of [myself v]       wait [pick random 0.5 to 1 seconds]     end   end end</pre>



Chart 7-1

## 7.4 Programming Result

Connect the micro:bit to your computer with micro USB, and select the corresponding COM port. Click the green flag to start the program, cover the LED panel of the micro:bit to simulate the night. Check the result.

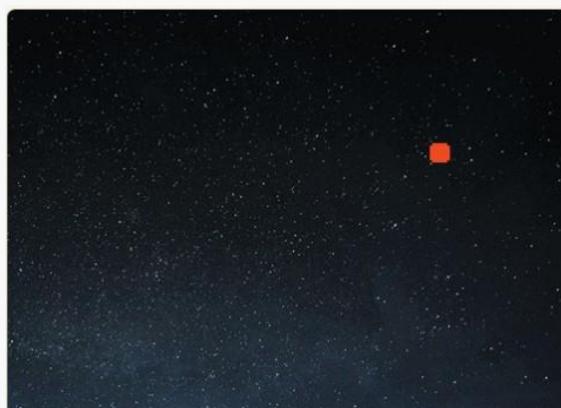


Figure 7-11

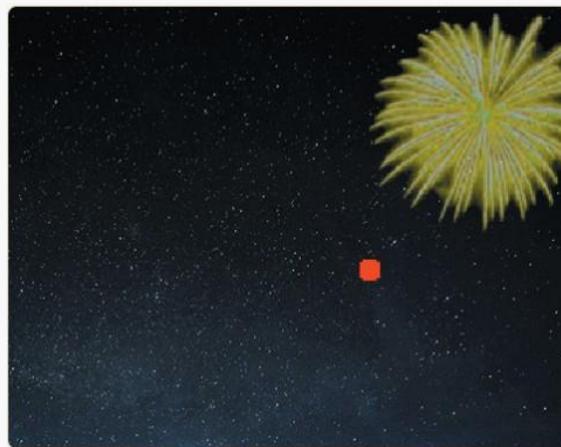


Figure 7-12

## 7.5 Light Sensor

The LED panel on the micro:bit can be used as light sensor to detect the light level around it.

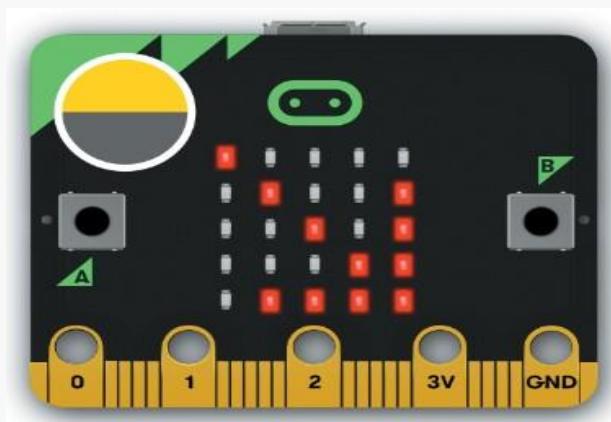


Figure 7-13 Light Sensor

## 7.6 Training Camp

This program is basically to use the light sensor to control the light level within a certain range, then make the fireworks explode when the ambient light is as dark as night. How about using “light sensor” to control the lights in the corridor? If you dare to dream, you can make it. Go ahead!

## Chapter 8. Sound Paintbrush

Hi, guys, have you ever thought that sound can be used as a paintbrush? By detecting the “loudness” of the sound, the sound paintbrush can draw a curve line of the sound; when the sound is too high, the buzzer will sound a beep to alert people. So cool, right!

We will learn how to use micro:bit expansion board and buzzer in the actual application.

### 8.1 Pen Module

**1. Click “Expansions”->“Functions”->“Pen” to load the pen module. As figure 8-1 shows.**

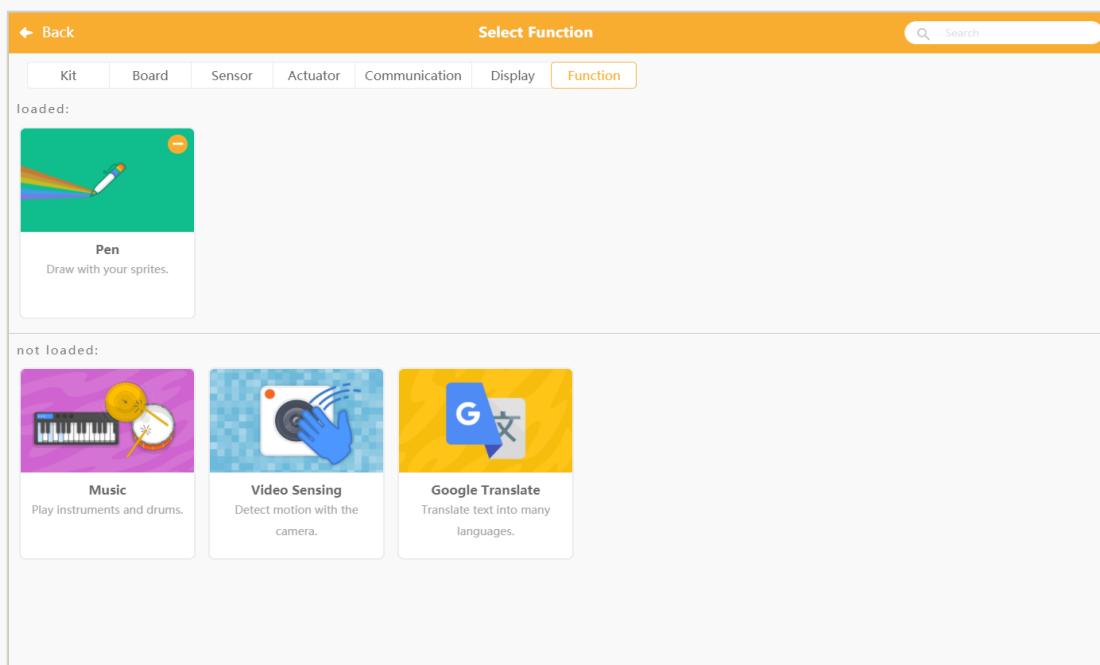


Figure 8-1

**2. Pen module function description.**

Chart 8-1 Pen Module.

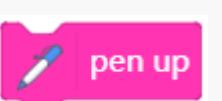
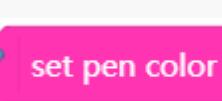
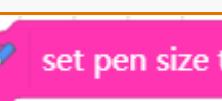
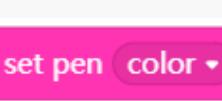
Blocks	Description
 erase all	Remove all the marks made by the pen or stamps on the stage
 stamp	When used in a script, the Sprite will produce a bitmap image of itself, which is stamped onto the stage.
 pen down	Start the function of the pen, make the sprite continuously pen a trail wherever it moves.
 pen up	Stop the function of the pen. If a sprite is currently using the pen feature because of the pen down block, this block will stop it from continuing.
 set pen color to 	The block sets the pen's color to the color chosen with block's color-picker.
 set pen size to 	Set the pen's size to the specified amount.
 set pen  to 	Set the pen's color to the hue chosen with the argument. The different numbers represent different colors such as 0 for red.

Chart 8-1

### Case 1: Pen down and pen up

#### 1. Chose the center point of the sprite

By default, the center point is right at the middle of the canvas, also is the centre of the paintbrush. If we don't adjust the position of the pen, the picture it draws will be like that showing in figure 8-2. We can use "select" function to select the pen, and drag the nib of the paint to the center of the canvas. Shown as figure 8-3.

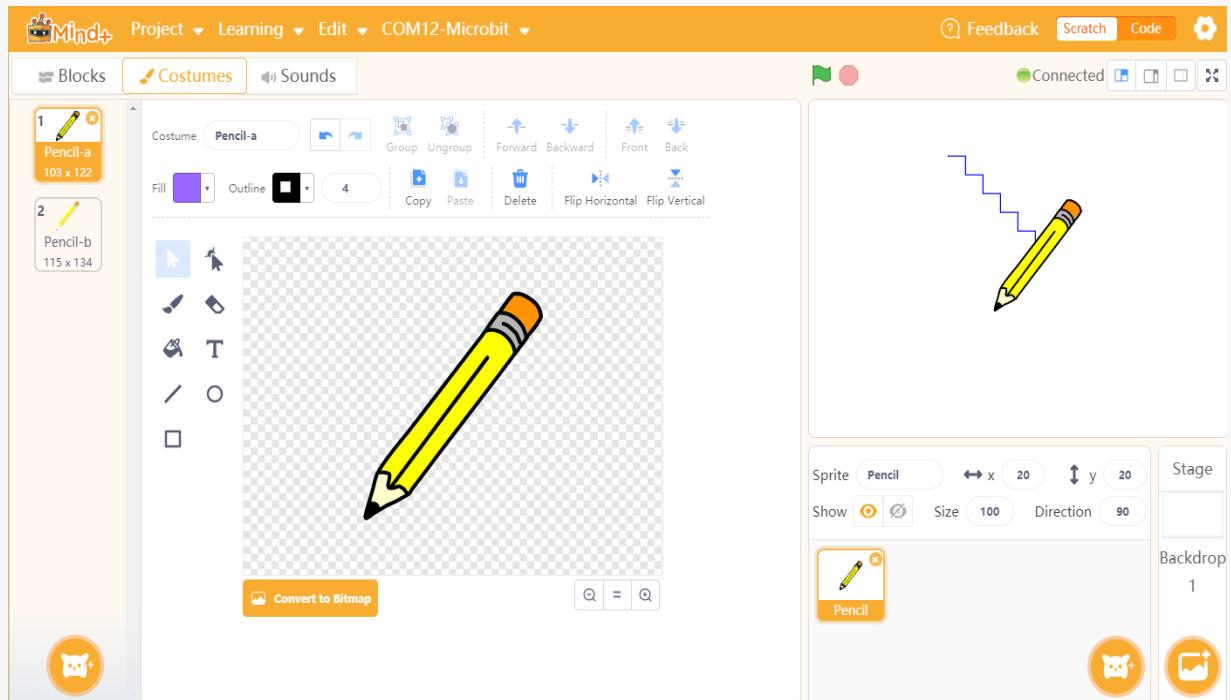


Figure 8-2

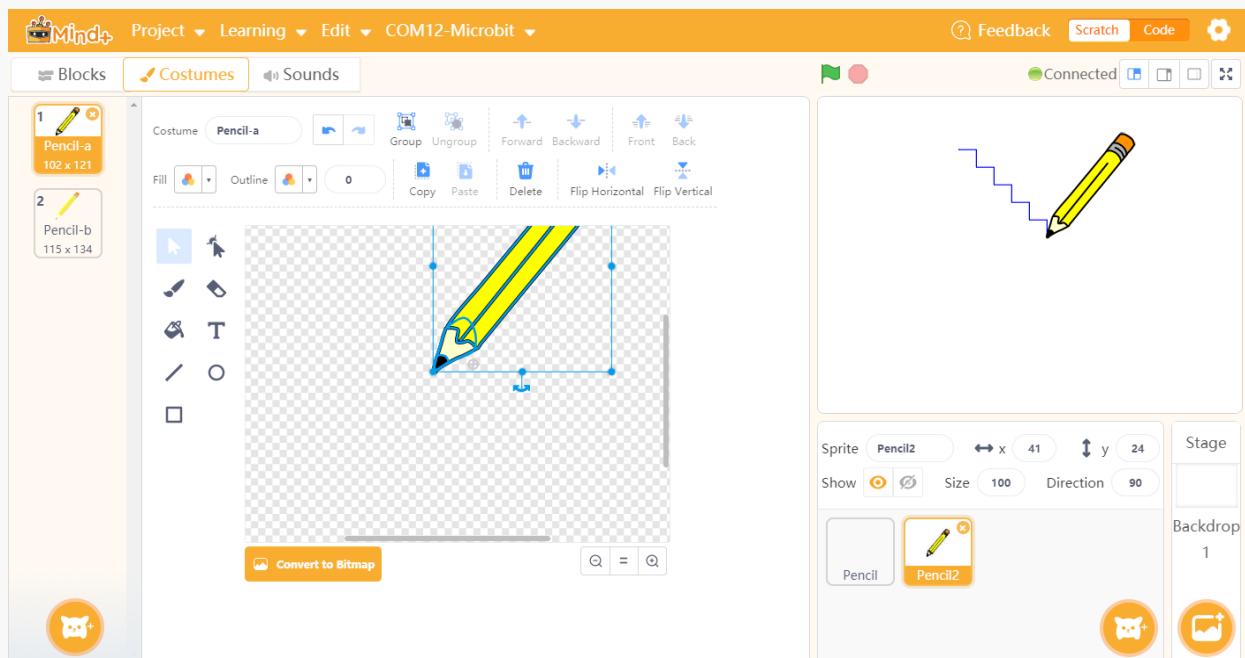


Figure 8-3

Next, we will go to the programming part. Please make the distinctions of the two pictures below and get to know the usage of “pen up and pen down”.

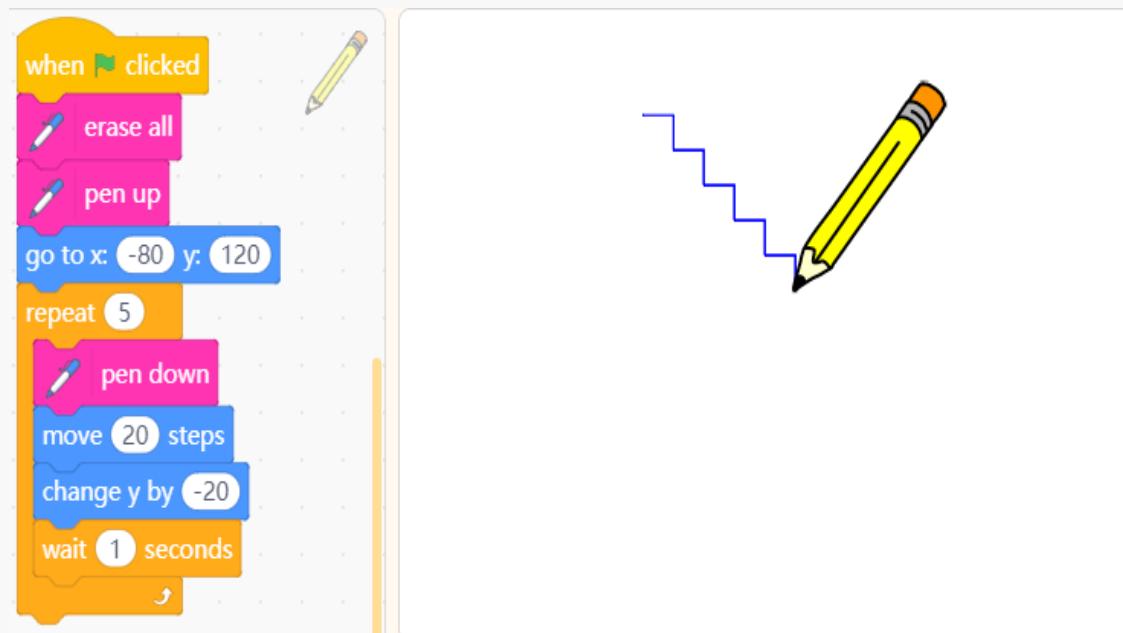


Figure 8-4

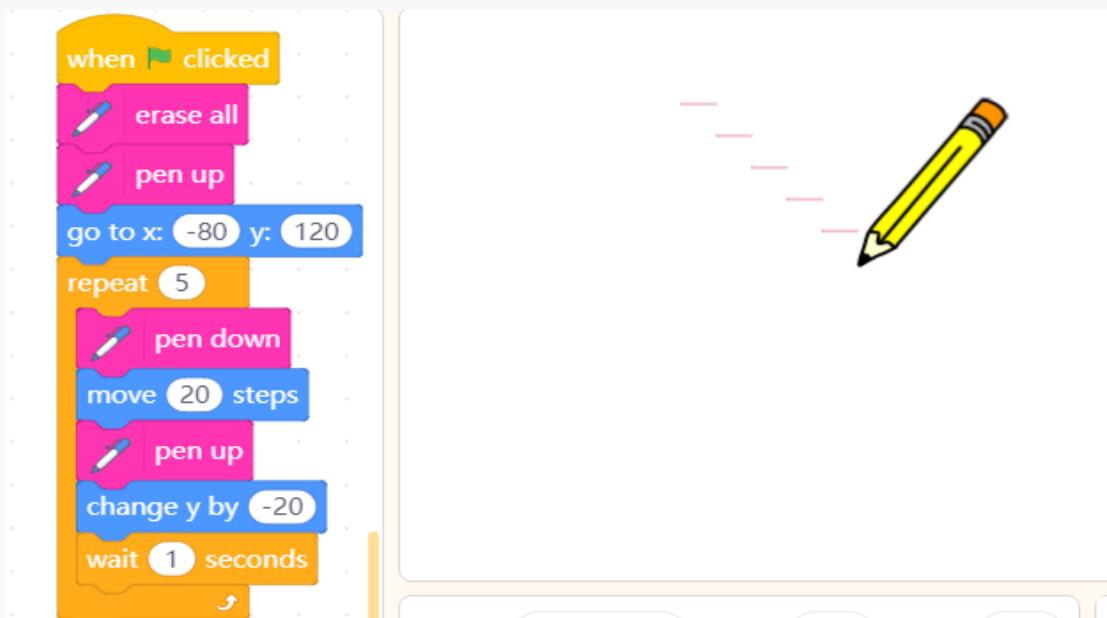


Figure 8-5

### Case 2: Draw Snowflake

All snowflakes have six sides. But we just need to draw one side of the snowflake and then run the loop body for 6 times to complete the rest. The entire programs are displayed below.

Add stamps at the bottom of the above programs. There will be one more butterfly on the stage.

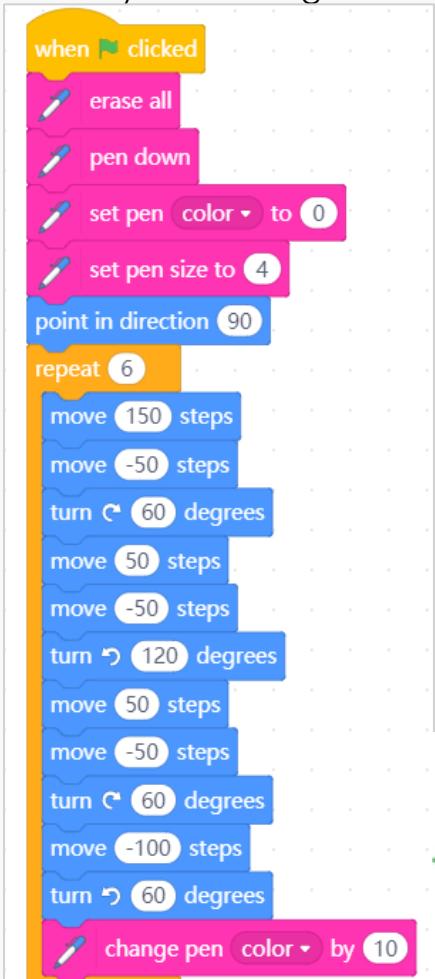
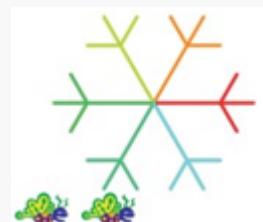


Figure 8-6



Figure 8-7



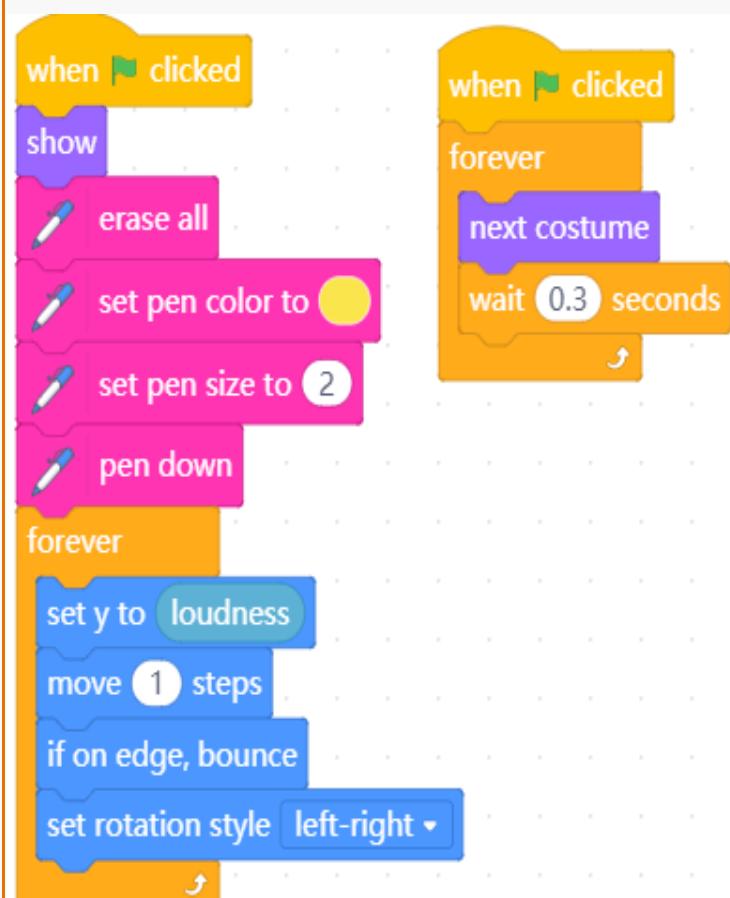
## 8.2 Algorithms Analysis

- Create a sprite as paintbrush, and click green flag to clear the stage. The value of X coordinate is random, and that of Y coordinate is the value of “loudness”. Run the program repeatedly to move 1 step, if on edge, bounce;
- Draw a sprite “disc”, and make it rotate to simulate the effect of playing music;

- Connect a buzzer to the pin of micro:bit expansion board. If the loudness is detected to be greater than a certain value, the buzzer outputs high level, if not, output low level;
- Add script blocks for the background: click green flag to increase the color effect, and play the music 1 to 10 randomly.

## 8.3 Programming

There are two sprites in the programs: a girl whose outline can be used as a paint to draw lines on the stage, and a rotating disc. Besides, we add a spot-light background for the stage, and we can also make the background to change color effect and play music randomly by programming.

Sprite	Blocks
 Ballerina	 <pre>when green flag clicked   show   erase all   set pen color to yellow   set pen size to 2   pen down   forever     set y to [loudness v]     move (1) steps     if on edge, bounce     set rotation style to [left-right v]</pre> <p>The script for the Ballerina sprite starts with a "when green flag clicked" hat block. It then performs a series of actions: "show", "erase all", "set pen color to yellow", "set pen size to 2", and "pen down". This is followed by a "forever" loop that repeatedly sets the y-position to the current "loudness" value, moves the pen 1 step, and bounces it if it hits an edge. Finally, it sets the rotation style to "left-right".</p>

 Empty	<pre>when green flag clicked   go to x: 0 y: -100 forever   change color by 5   turn C 2 degrees</pre>  <pre>when green flag clicked   set digital pin P8 output LOW forever   if loudness &gt; 20 then     set digital pin P8 output HIGH   else     set digital pin P8 output LOW</pre>
--	---

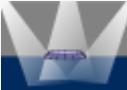
Background	Blocks
	<pre>when green flag clicked forever   change color by 5   play sound [pick random 1 to 10 until done]</pre>

Chart 8-2

We can tick the loudness in the sensing module to display the loudness value on the stage. Shown as below.

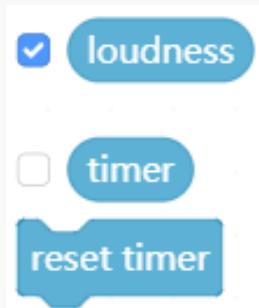


Figure 8-8

## 8.4 Expansion and Buzzer

### Micro: Mate Expansion Board

Micro:Mate is a tiny multifunctional IO expansion board designed for micro:bit. It expands 6 sets of 3 pin I/O interfaces and 3.5mm audio jack, which is capable to connect DFRobot Gravity series modules, servo motors, sensors and jumpers, freeing users from complex plugging steps.

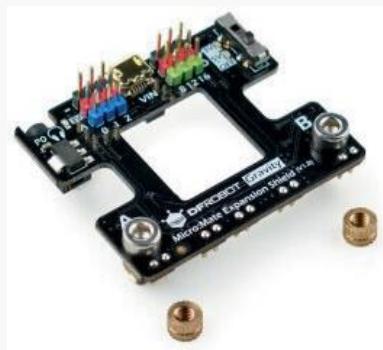


Figure 8-9

In addition, Micro: Mate is in the same dimension of micro:bit, utmost integrating the common functions of micro:bit. Connect it to micro:bit through contact pins (with spring loaded), easy and secure. Pin8, 12, 16 support voltage switches between 3V-5V, which is able to drive 5V 2A electronic device (external power is necessary). Servo or small motors can work well with this board.

There are rubber bumpers, silkscreen and power indicator on the back of the board that could keep the expansion installed stably on the board as well as prevent reverse connections. Besides that, Micro:Mate adopts standard Gravity colorful pins to help users to distinguish different pins. The expansion board is

perfectly suitable for electric experiments, small robot, interact device, wearable device and so on.

Micro: Mate Specification:

<b>Gravity IO expanded functions</b>	3 analog interfaces (only support 3.3V analog input)
	3 digital interfaces (support 5V/3.3V digital input and output)
	Servo driving current:1.2 A
	Input Voltage: 3.3V/5V
<b>Dimensions</b>	68.5 * 53.3mm/2.7 *2.1"
<b>Weight</b>	22.8g

Chart 8-3

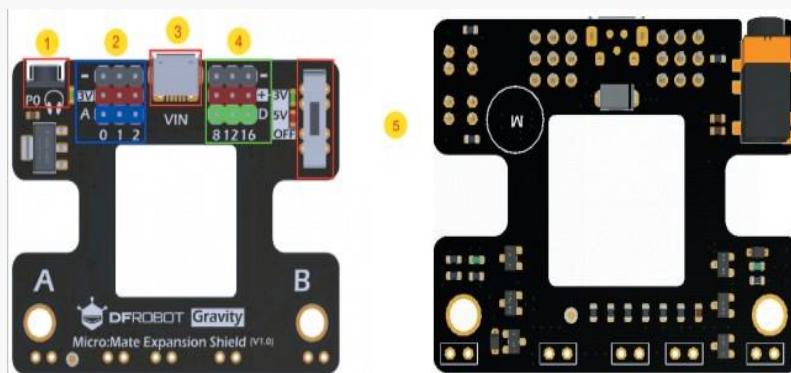


Figure 8-10

Micro: Mate multifunctional expansion board pinout:

Number	Name	Function Description
1	Audio jack interface	3.5mm Audio Jack
2	Analog output interface	Gravity series compatible (only support 3V)
3	USB external power interface	Connect external power supply (program uploading is not supportable)
4	Digital interface	Gravity Series compatible (3V and 5V)
5	Voltage Switch	Switch between 3V, 5V (Pin 8, 12, 16) and OFF

**Voltage Switch:** there is a voltage switch on the expansion board that can switch the power between 3V, 5V and OFF.

**D&A:** We can find the silkscreen “D” and “A” on the board. “D” represents the digital signal (digital pin). “A” is the analog signal (analog pin)

One of the most advantages of the IO expansion board is that it greatly expands the number of the power and GND interfaces. So, there would be no need to worry about the interface number problem when you want to connect more sensors.

There are two rows of pins in color red and black respectively over the digital pin and analog pin. They are the expanded power interfaces. Connect the red pin to the positive pole and black pin to GND.



Buzzer Specification Chart 8-5

Figure 8-11

Type	Digital Signal
Voltage	3.3V
Connector	PH2.0-3
Dimension	30mmx20mm/1.18x0.79inches
Principle	High level to play sound; low level to stop playing

Connect micro:bit to the computer with micro: USB. Micro:bit USB, and complete the installation. Connect the buzzer to the pin 8 of the expansion board, please note that the black wire should be plugged into the GND pin, as figure 8-12 shows. Run the program and check the result.

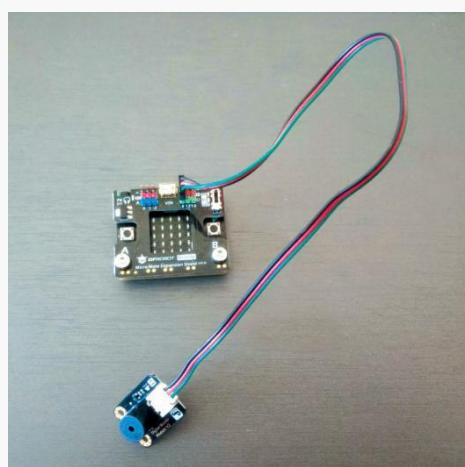


Figure 8-12

## 8.5 Programming Result

Click green flag to start the program, make noise like clapping hands, then check the result. Shown as figure 8-13.

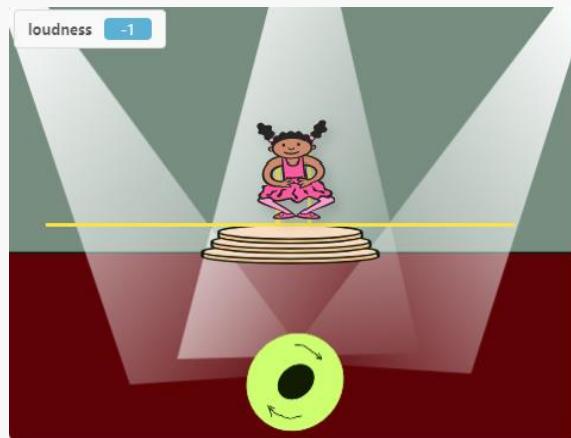


Figure 8-13

## 8.6 Training Camp

In this project, we take the loudness as the value of Y axis of the paintbrush, but can you guys try to make more fun playing modes?

## Chapter 9. Security Alarm

There are so many different types of security alarms on the market, but choosing one can be very frustrating and confusing. How about making a security alarm to keep your house and property safe or prevent your children from touching dangerous stuff like electronic device? The security alarm will integrate sound sensor, infrared motion sensor and touch sensor in one piece to detect threat condition, and to warn people through buzzer and indicator.

The programs in this chapter will be completed under the code mode. So, switch to code mode before getting started!

### 9.1 Algorithms Analysis

- Security alarm includes four input parts (sound sensor, IR motion sensor, digital touch sensor, digital push-button module) and two output parts (buzzer and LED light);
- When the loudness is over 100, the motion sensor senses a body's approaching or the touch sensor detects an object touched the sensor electrode, the buzzer makes noise and the LED lights up.
- If an adult receives the alarm, he needs to press down the button to stop the buzzer and turn off the LED.

### 9.2 Sensor and Multithread

1. Select the micro:bit board in Extensions, then Click “Extensions”->“Function”->“Sensor” to choose the sensors you need.

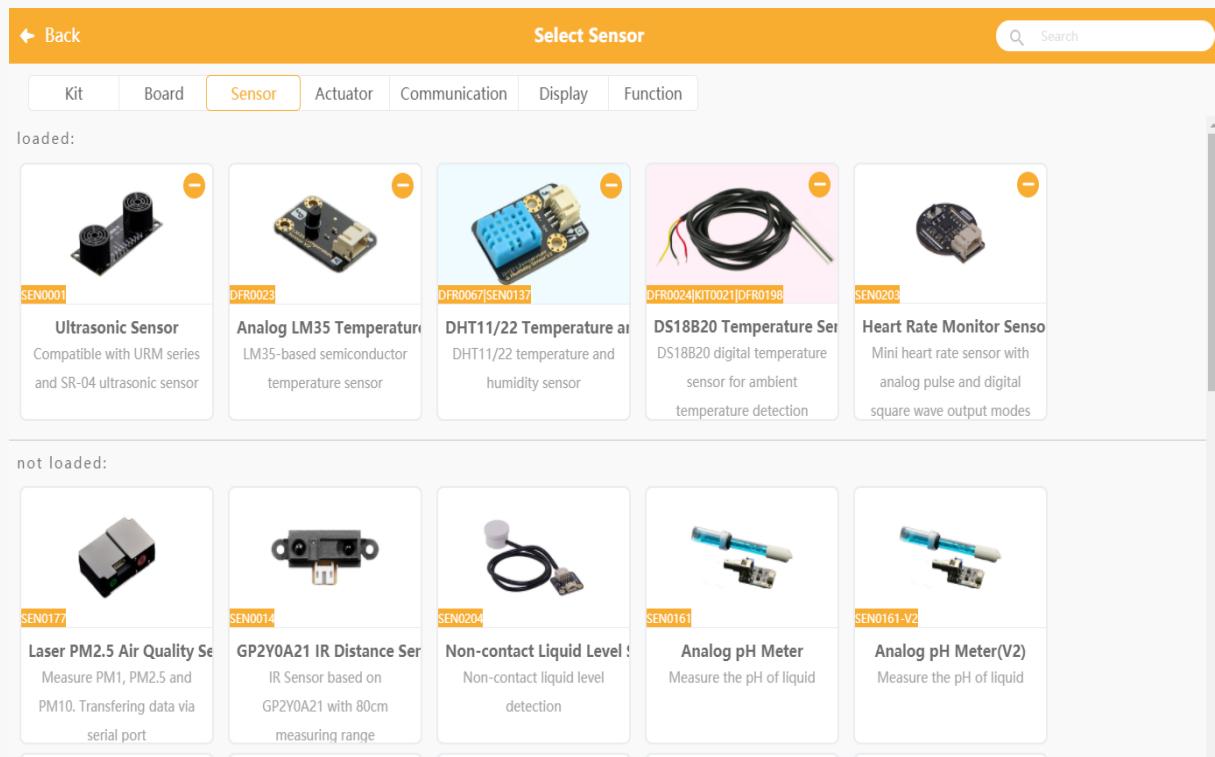


Figure 9-1

## 2. Load multithread module: click “Extensions”->“Function”->“Multithread”.

A thread is a path of execution within a process, and multi-thread can handle several different tasks at the same time.

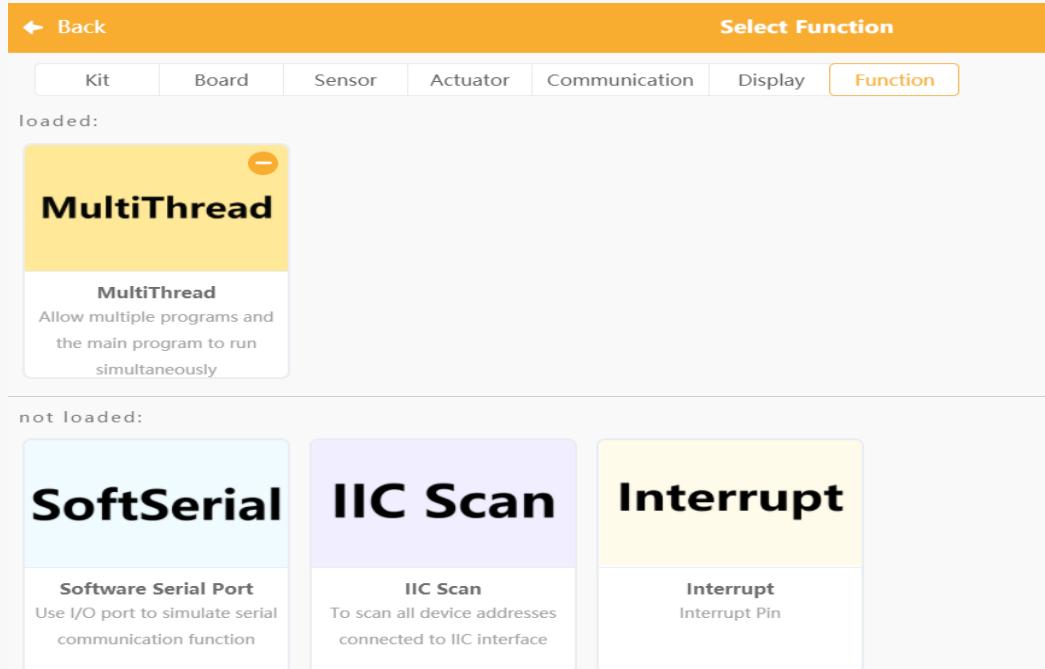


Figure 9-2

### 3. Blocks descriptions:

Chart 9-1 Block Description

Block	Description
	Click green flag to execute the scripts under the "loop 1"
	Start the block under "loop1"
	Stop the scripts under "loop1"
	Detect if there is motion of people or animal, if there is, the value is 1, if not, value is 0.
	Detect if there is sound in ambient. If there is, the value is 1, if not, value is 0.
	Press down: output high level; release: output low level
	Use capacitive touch-sensitive switch to sense the contact of body or metal object. If a touch is detected, the value is 1, if not, value is 0.
	If at least one of them is true, the block returns true; if neither of them are true, it returns false

### 4. Programming

Connect all parts to micro: bit expansion board. Output module: buzzer to Pin 8, LED to Pin 1. Input module: sound sensor to Pin 0, IR motion sensor to Pin 2, digital push button to Pin 12, digital touch sensor to Pin 16.

There are two programming ways to achieve the effect of security alarm.

Method 1: put all the detection parts in a main program, if any detection is triggered, the buzzer begins playing sound and the LED module lights up until the button is pressed down. Then the whole programs stop and all devices stop detecting, which is just similar to the function of the "stop ()" block in mind+.

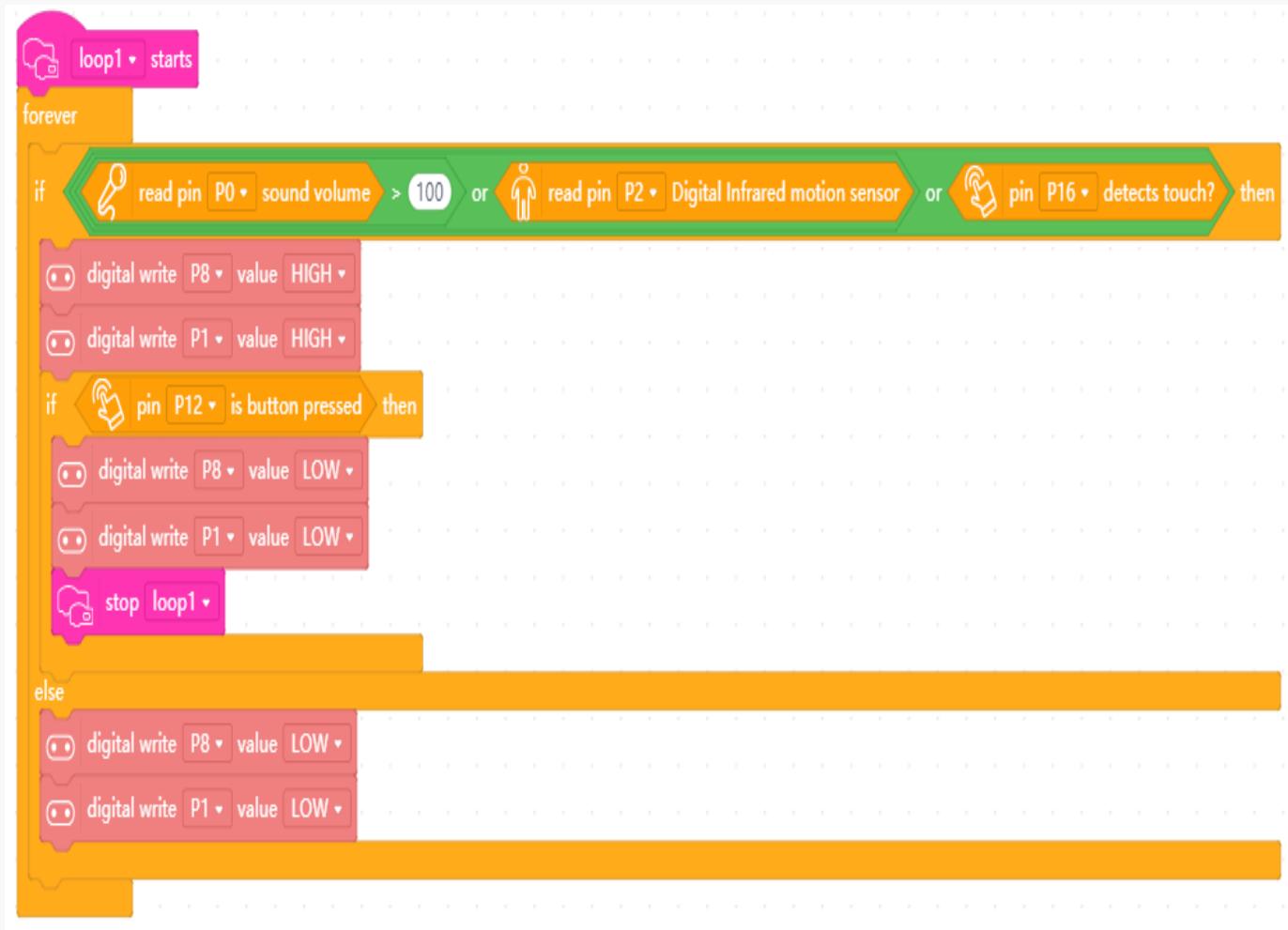


Figure 9-3

Method 2: put each kinds of detection in a thread independently, and the three threads execute concurrently. If the button is pressed down, turn off the buzzer and LED module. However, all sensors are still in detecting state. As figure 9-4 shows.

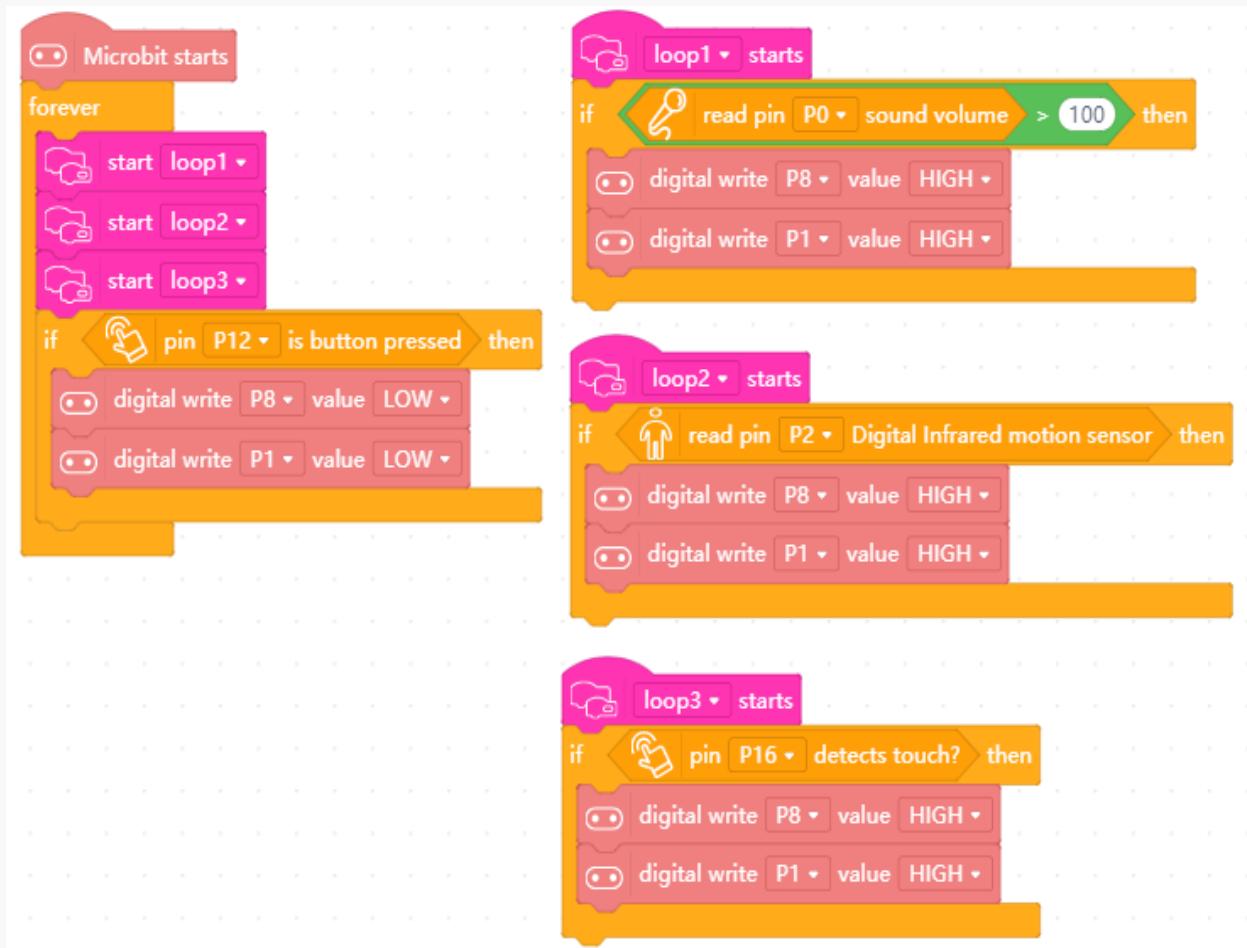


Figure 9-4

### 9.3 Output and Input Module

The specification of digital push button is in chart 9-2.

Type	Digital signal
Voltage	3.3V-5V
Connector	PH2.0-3
Dimension	22mmx30mm
Principle	Press down: output high level; release: output low level

Chart 9-2



Figure 9-5

The digital touch sensor is based on capacitive touch-sensitive principle. The direct contact of human or metal object on the metal surface of the sensor will be detected. Apart from that, it can also sense the indirect touch separated by

layer of material with certain thickness like plastic and glass, but the sensitivity will change with different thicknesses and touch areas.

**Note:** to reduce the sensing error, do not let the sensor touch any object that you are not going to detect in using.

Type	Digital signal
Voltage	3.3V-5V
Connector	PH2.0-3
Dimension	22mmx30mm
Principle	Based on capacitive touch-sensitive Can detect the contact of body or metal object

Chart 9-3



Figure 9-6

The analog sound sensor has an on-board microphone, through which Arduino can sense the loudness of a sound, then convert it to analog signal. That is to say, the loudness of a sound is presented by the returned voltage value.

Chart 9-4 Analog sound sensor specification:

Type	Analog signal
Voltage	+3.3V/5V
Connector	PH2.0-3
Dimension	22mmx32mm
Principle	Detect the sound in ambient

Chart 9-4



Figure 9-7

Infrared motion sensor can detect the IR light radiating from the moving objects (human or animal). It can be used in the occasion requiring body movement detection.

Application: motion detection, occupancy sense, IR security alarm system.

Chart 9-5 Infrared motion sensor specification:

Type	Digital signal
Voltage	3.3-5v, 6V(max)
Connector	PH2.0-3
Dimension	22mmx30mm
Principle	Detect the motion of human or animal. Output delay time (high level):2.3-3s; Detecting angle: 100 degrees; detecting distance: 7m

Chart 9-5



Figure 9-8

The specification of the LED module is shown in chart 9-6

Type	Digital signal
Voltage	+3.3V-5V
Connector	PH2.0-3
Dimension	20mmx30mm
Principle	High level: turn on; low level: turn off

Chart 9-6



Figure 9-9

The specification of the buzzer is in chart 9-7.

Type	Digital signal
Voltage	3.3V
Connector	PH2.0-3
Dimension	30mmx20mm
Principle	High level: play sound; Low level: stop playing

Chart 9-7



Figure 9-10

## 9.4 Programming Result

Connect the micro:bit to the computer with micro USB. Connect all parts to micro: bit expansion board. Output module: buzzer to Pin 8, LED to Pin 1. Input

module: sound sensor to Pin 0, IR motion sensor to Pin 2, digital push button to Pin 12, digital touch sensor to Pin 16.

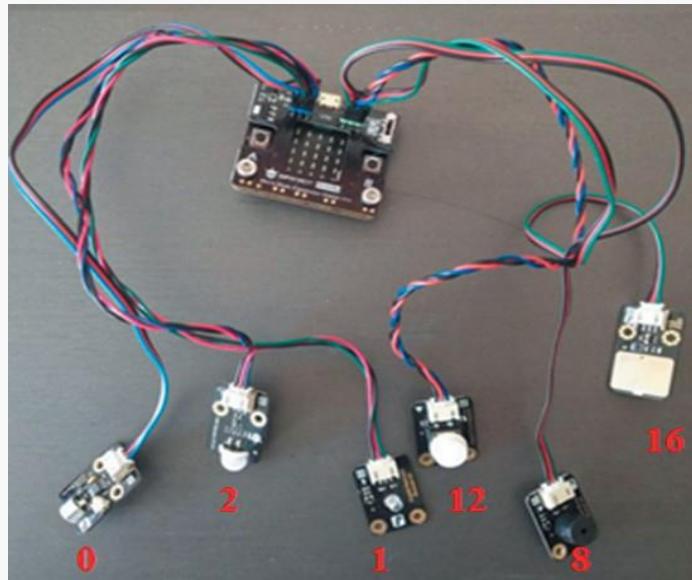


Figure 9-11

Drag and combine blocks to code, and upload the program to micro:bit, then the security alarm is done.



Figure 9-12

## 9.5 Training Camp

Add some decorations for the security alarm to make it more beautiful. Except the sensors we used in the project, there are much more sensors in mind+ waiting for you, you can create more fun and cool projects with them, come and explore!

## Chapter 10. Bubble Machine

Bubble machine are getting so popular among children recent years. It could be a fun way of encouraging children to play outside. The principle of bubble machine is not complicated, and it can be thought that it replaces human mouth with a machine to blow bubble. In this chapter, we will make our bubble machine by our hands!

### 10.1 Algorithms Analysis

We will code the project under both “Scratch” and “Code” mode. The former one can achieve to blow bubble on the stage of mind+, while for the latter, we have to upload the codes to micro:bit and use a servo to blow bubble.

#### ● Scratch Mode

Click green flag, repeatedly execute the program that the rabbit switch to next shapes to form an effect of running;

The bubble clone itself continuously, and run the programs: generate from the bottom, move up, disappear if it touches the edge, then delete the clone.

Upload a picture to mind+ as a background sprite, place it to the center of the stage, enlarge to cover the whole stage. Set a variable “background X”. When the green flag is clicked, run the program to increase the value of the variable by -2, so the background will move backward as the rabbit runs forward. While if the rabbit comes to the edge of the picture, also when background is less than -120, set the variable to 120 and start over.

#### ● Code Mode

When micro:bit main program starts, light up two LED as the eyes of the rabbit bubble machine, the servo repeatedly rotates from 0 to 60 degree to realize the waving arms function of bubble machine. Use a blower to blow bubble.

## 10.2 Scratch Mode Case

### 1. Scratch Mode

We are going to make a scene in this mode: a rabbit is running in a forest, and there are full of bubble around it.

Key point:

There we should know a little bit about relative movement. The background forest is moving left in relation to the rabbit. Hence, the rabbit keeps changing shapes at a fixed position. The value of X coordinate of the background sprite constantly decreases, so it will seem like the rabbit is running forward.

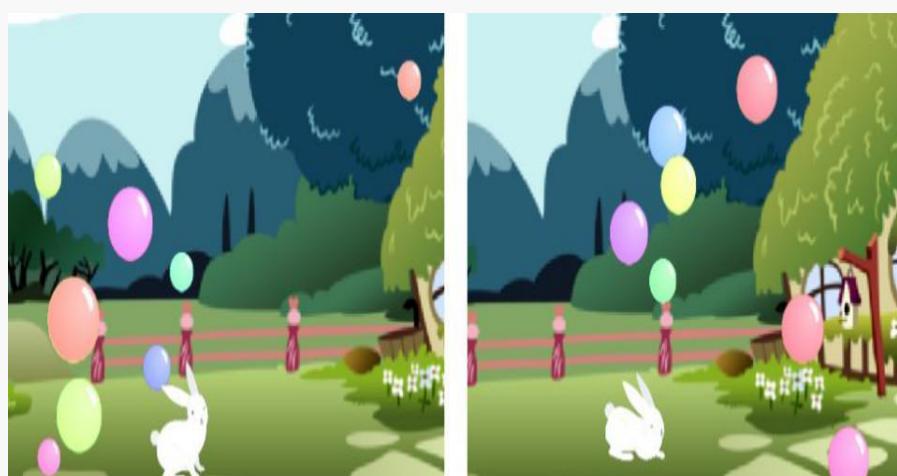


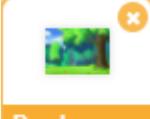
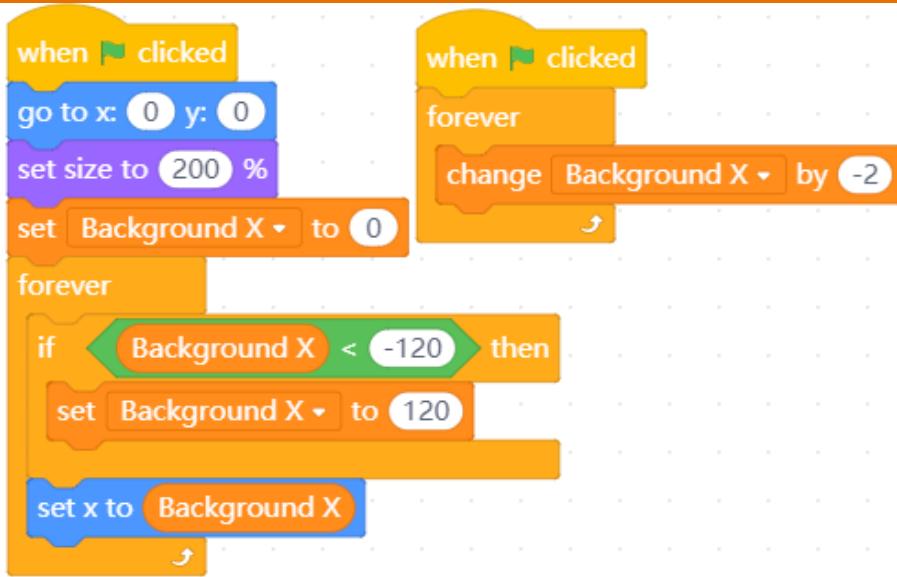
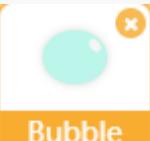
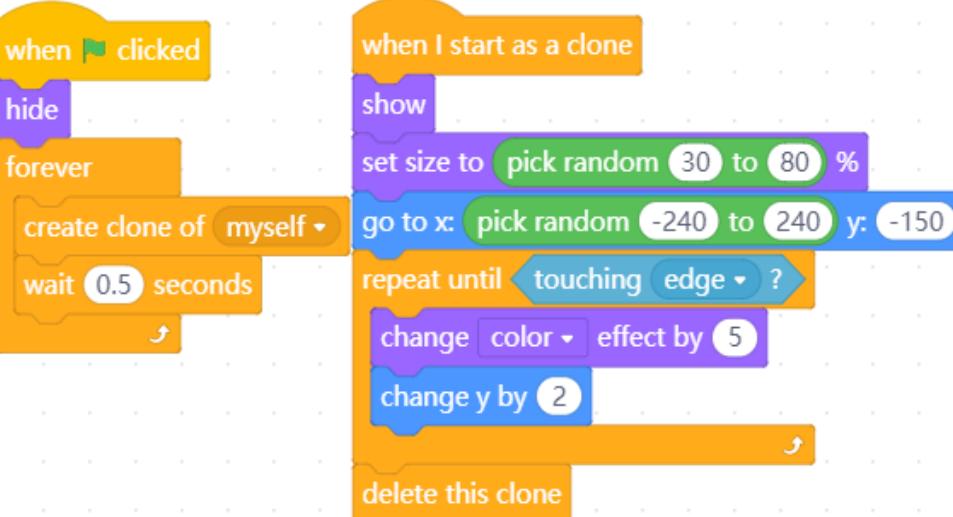
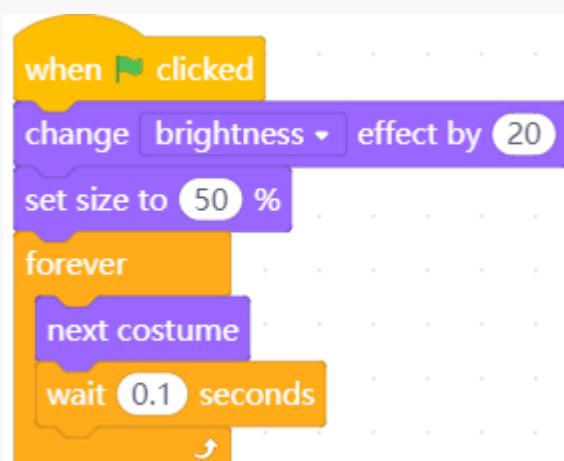
Figure 10-1

### Clone

We have learned about the clone before. Go to chapter 7 if you forgot. The clone the bubble constantly appears, and disappears when touching the edge, then delete the clone.

### 2. Programming under scratch mode

Chart 10-1 Scripts of all sprites

Sprites	Blocks
 Background	
 Bubble	
 Rabbit	

## 10.3 Code Mode Case

### 1. Load the servo module

Switch to Code module, click “Extensions”-> “Actuator”-> “Servo”.

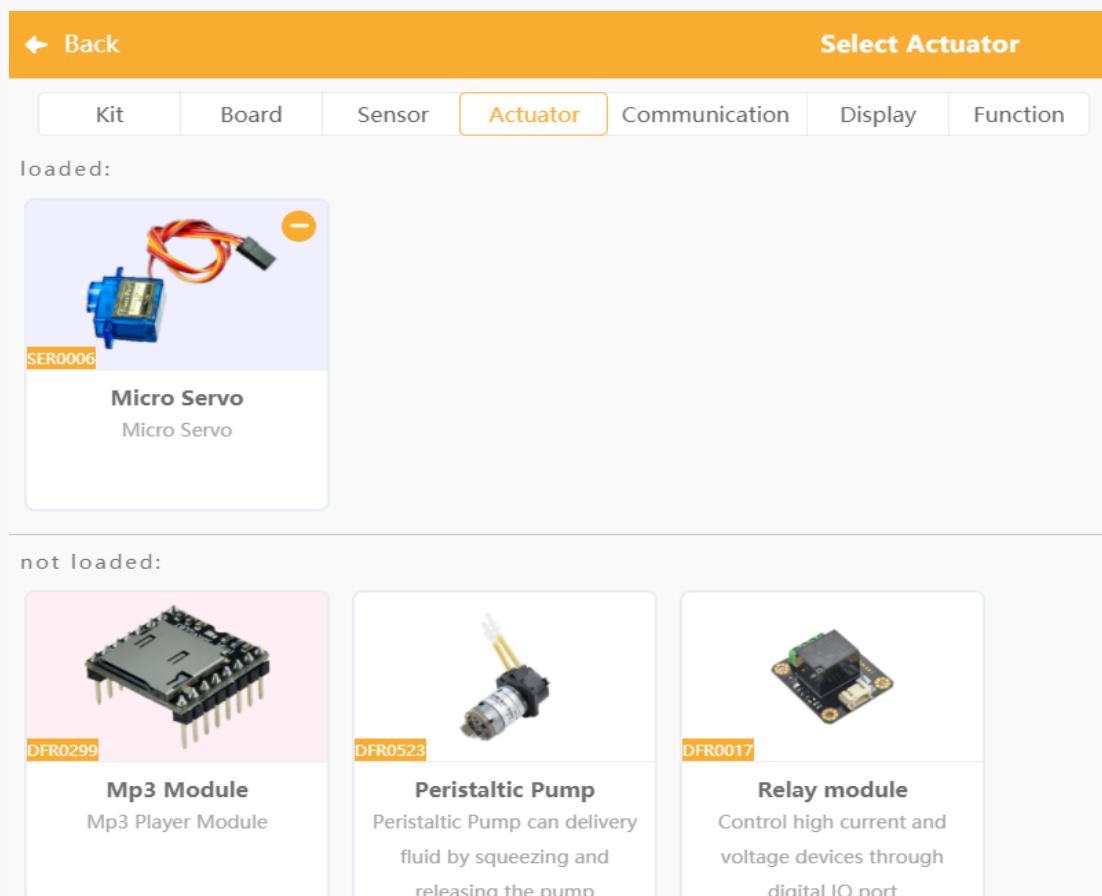


Figure 10-2

### 2. Connection and shape design

#### 1) Connection

**Components:** 1 micro:bit mate expansion board, 2 LED module, 1 Servo.

Connect servo to pin P8 of the expansion board, two LED module to P0 and P1.

Use external power for the device.

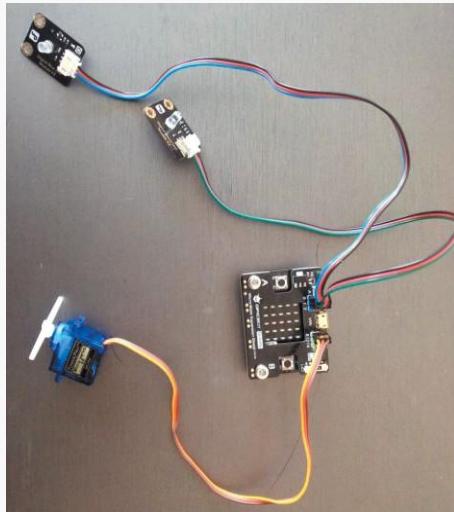


Figure 10-3



Figure 10-4

2) Where does the wind comes from? —Blower;

We can shop a blower online, and it usually comes with power supply and switch. Shown as figure 10-5. Or we can make it by ourselves. Thus, we need a servo, fan blade, and a device like a funnel for gathering wind.

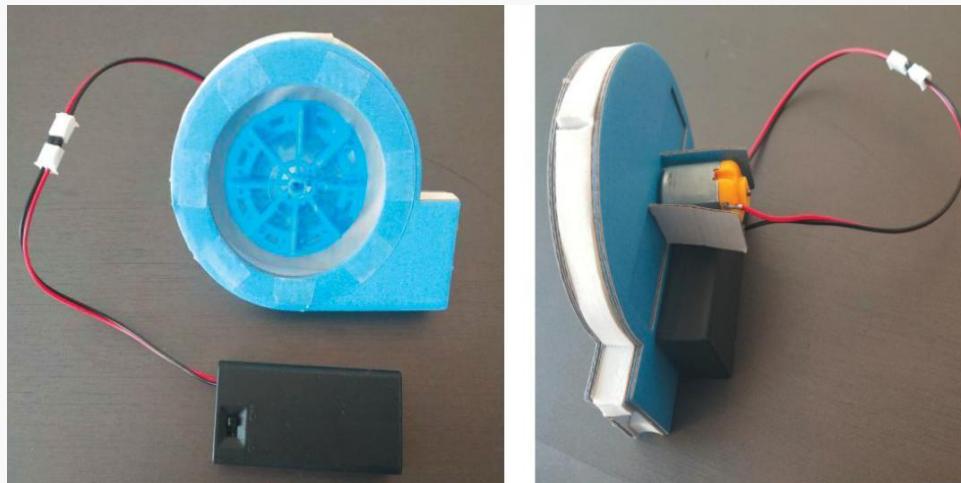


Figure 10-5

3) Making the bubble machine

First, we cut a rabbit shape with a piece of cardboard, then arrange all parts together inside a box. Tie up the servo's shaft and bubble ring with iron wire.

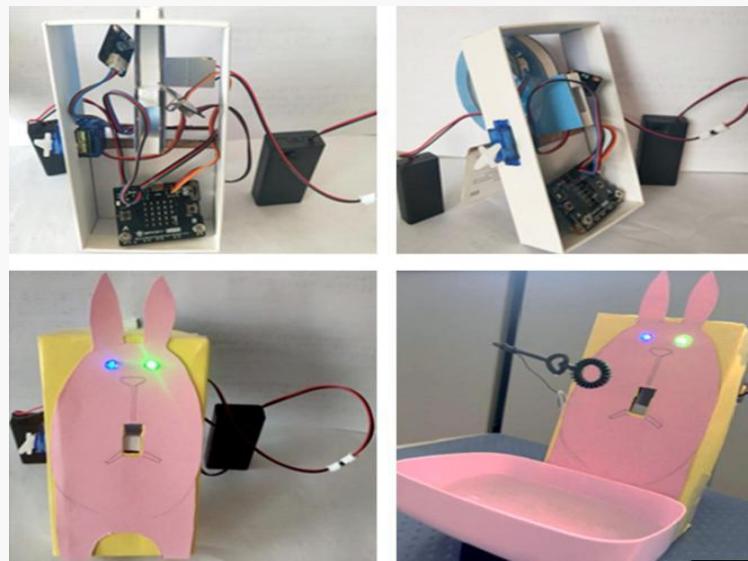


Figure 10-6

### 3. Programming under code mode

Switch to “Code Mode”, drag and snap blocks to code, and upload to the micro:bit.

### 4. Programming effect

Prepare the bubble solution, turn the blower and power up the expansion board, the bubble machine will start to work.

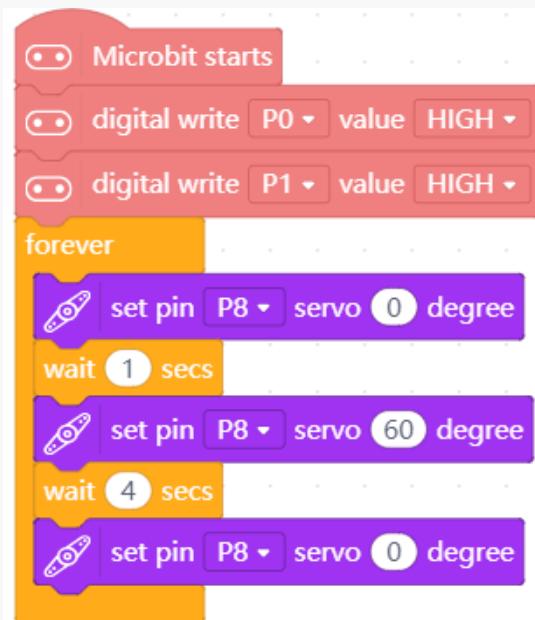


Figure 10-7



Figure 10-8

## 10.4 Servo



Figure 10-9

Servo is a rotary actuator and linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It is known as servomotor.

Red wire to positive pole, brown wire to negative pole and orange wire is signal line.

Chart 10-2 Servo specification

Type	180° Servo
Voltage	4.8V
Connector	PH2.0-3
Dimension	23x12.2x29mm
Principle	Torque: 1.6kg/cm (4.8V) Speed: 0.14 Second /60degree (4.8V) Operating Temperature:30~+60°C Dead Zone Width: 5 micro-second

Chart 10-2

## 10.5 Training Camp

If you want to make colorful bubble, add several drops of food coloring in the bubble solution and stir well. The bubble machine we made today basically depends on the blower for blowing air and servo for waving the bubble ring. Actually, the servo is also very commonly used in our daily life, for instance, the fortune cat with waving arms in the front window of a shop or restaurant, merry-go-round in the fairground, parking garage railings and so on. Give rein to your imagination, and make more interesting project with servo.

## Chapter 11. Get to know micro: Maqueen

Today I would love to introduce a new friend, Maqueen. Have you suspected that car, lightning McQueen, a super hero in the movie Cars. No, not it. At here, what we get to know is the robot micro: Maqueen, an excellent educational robot.

What is a robot in your imagination? Does a robot must look like people?

### 11.1 What is a robot?

A new restaurant chain called Easta changes our view of traditional fast-food. Easta is almost automated, functioning like a vending machine that spits out freshly-prepared quinoa bowls. In Easta, customers order at an iPad kiosk then sit to wait their food where will be show in the glass cubbies. You will meet no one in the whole process. However, there are still kitchen staffs in the back of the cubbies. In China, a future restaurant was opened in October 29th, 2018. Different from Easta, this future restaurant is fully worked by robots. The robots are not just waiters, but cooks. So, you can really enjoy the robot service!

Look at these robot cooks, they are familiar with traditional Chinese 8 cuisines: Shandong cuisine, Sichuan cuisine, Cantonese cuisine, Chinese Cabbage, Fujian Cuisine, Huizhuo cuisine, Hunan cuisine and Zhejiang cuisine. They can cook more than 40 dishes. Their programs to cook strictly follow the standards about temperature, time, materials that given by famous cooks. When the food is cooked, the cook robot passes it to waiter robot, then the waiter robot could serve customers. In the restaurant, waiter robots can drive by themselves and avoid obstacles automatically, and auto optimize the path. They can pass dishes and recover dishes.

There are different function robots in life, but what is the robot? Its components are similar to human. In fundamental, human body are composed by 5 main

parts: body and muscle to move; sensor to receive environment information and control muscle to move; energy source to provide energy to muscle and sensor. Brain to process sense-information and command muscle movement. A typical robot should be equipped with a moveable body, a device similar to motor, a sense system, a battery and “brain” to compute and control all. Essentially speaking, robot is the “animal” made by human. It is a machine that imitate human activities and animal activities. In our life, it can help or even replace human to work.

**Execute Device:** first of all, almost all robots have one moveable body. Some only has motorized wheel, some equip with a large amount of moveable parts. Similar to bones, these independent parts are connected by joints. Normally, the number of joints represents the degree of freedom.

**Drive:** according to the commands from main control, the drive motivates robot based on the power part. Input electronic signal, then the power output the line and the angular displacement. Majority of robot drives are electricity drive device, such as stepper motors, servo motors and so on. Besides, hydraulic and pneumatic drives are also adopted.

**Detect:** monitor the robot movement and work in reality, and feedback the necessary information to the control(brain). There are two categories of detect device, one is the internal sensor and the other is the external sensor. The internal sensor can detect all internal part of the robot, for an example, the joint position, speed, acceleration and so on. Then it will pass the detected information to the controller in the form of feedback signals, forming a close-cycle control. The external sensor is used to get information about object, environment and so on. In this way, the robot activity can adjust the outside changes, and can reach the high-level automation, and even make the robot be sensitive, be smart. Specifically, with the help of information about object and environment given by the eye sensor and sound sensor, the robot can form a big feedback cycle, then the robot activity is more precise.

**Control:** similar to the human brain, process the information and make decisions. It can be a small single chip, it also can be a microcomputer.

## 11.2 Robot Application

Based on the application environment, the Chinese robot expert classify robots to 2 categories, industry robots and special robots. Industry robot, just as its name implies, are designed to serve the industry, so majority are multi-joints manipulators or high freedom degree robots. They can replace workers to do dangerous, repeat, mass and qualitative jobs. The robot machines can work overnight. They are widely used in making cars and household appliances. Take the welding robot as an example, it can weld complex parts precisely.

Except industry area, the special robots are used in all areas to serve human.

**Service robot:** robots around our lives such as restaurant robot, sweep robot and so on.

**Medical robot:** classified as surgery robot, recovery robot, assistant robot, medical service robot... With the rapid growth of medical robot market, the surgery robot takes the lion's share and the recovery robot are growing at the fastest pace. In branch, exoskeleton robot is going to boom, and the train robot and bionic artificial robot are both growing with high speed.

**Entertainment robot:** have you ever seen robot band? So as paint robot, toy robot, dance robot. Entertainment robot makes our lives more joys.

**Underwater robot:** guess what is used for? Yes! The underwater robot can detect, research and rescue. In artist works, Tibetan Plateau waters are called as tear of earth. People has adopted smart underwater robot to detect and collect water-quality information about tear of earth waters. Moreover, the underwater robot has already put into emergency rescues. It can observe, analyze and judge the object position and situation in water, then pass data in real-time.

**Army robot:** have you heard about explosive-removal robot, ammunition disposal robot, mine-sweeping robot car, UAV? Now, the modern army pay more attention to digital, information and intelligent. Robot system can do almost all kinds of works. It has a place in army for its talented skills, such as detection and explosive-removal. Adopting army robot system can advance the efficiency to take missions, reduce loss, and protect soldiers.



Robotic Arm



Medical Robot



Robot Band



Underwater Robot



Army Robot



Rescue Robot

## 11.3 Hello, Maqueen!

Maqueen is a cute graphical programming robot for STEM education. It is small but powerful. Simple but Simple in design, smart in function. Maqueen is compatible with BBC micro:bit.

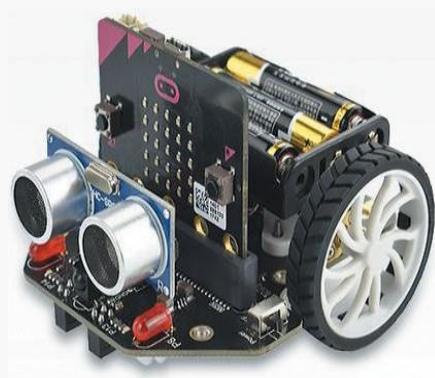


Figure 11-1

The function diagram is shown as below:

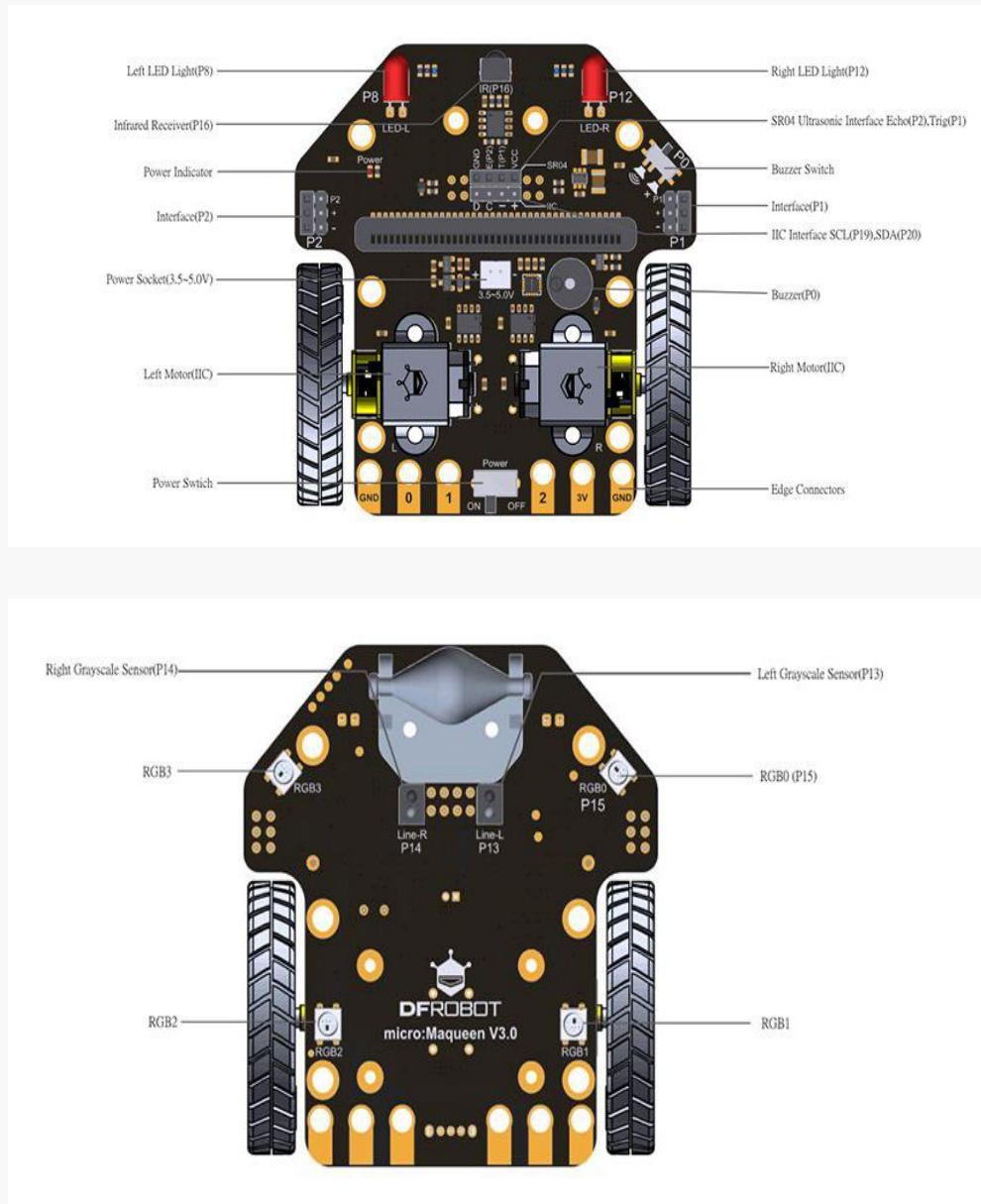


Figure 11-2 Function Diagram

Here we can see, there are many sensors in Maqueen, line-tracking, ultrasonic, infrared input. And 2 motors, 2 LEDs and 4 RGBs, and a buzzer. BBC micro:bit is an ARM single chip, equipped with many onboard resources, such as Bluetooth, wireless, acceleration, e-compass and buttons, and 5 x 5 matrix. The combination of Maqueen and micro:bit can realize many functions such as infrared remote control, line-tracking, light-tracking, obstacle avoidance,

distance detection and temperature measurement. It is free from complex wiring, plug and play.

In programming, micro:bit and Maqueen robot has already supported Scratch. Based on Scratch3.0, mind+ comes into a tool combining micro:bit and related products. Mind+ supports online and offline geographic programming, supporting Arduino and Python programming. It also supports both explanation programming and download programming.

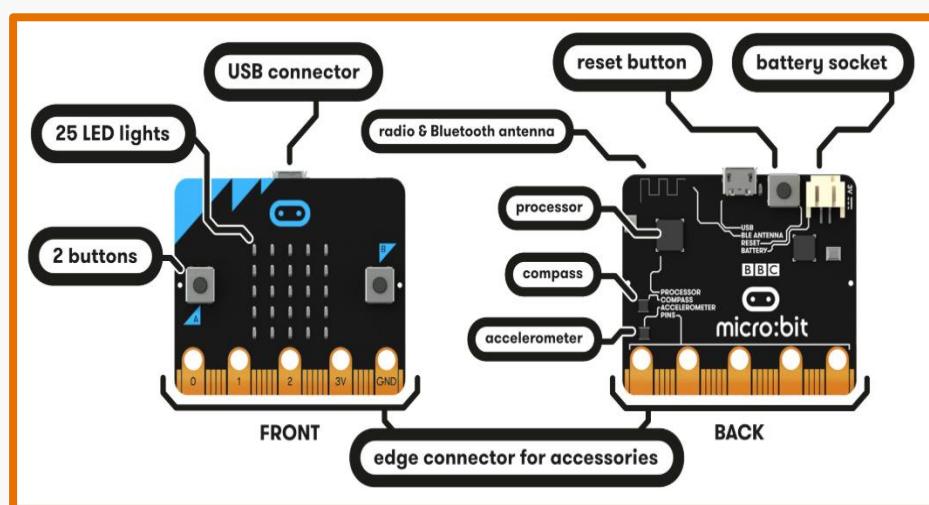


Figure 11-3

## 11.4 Have a Try!

Let's assemble Maqueen, please refer to the details as below.





Figure 11-4

## Chapter 12. Maqueen, let's move!

Here Maqueen is already, wait, the body is ready, but we totally forget one important thing, make Maqueen intelligent. What should we do now? Yes, we need to programming the brain. My great engineers, come to work!

Better tools make for good work. Firstly, I would love to introduce Mind+, a programming software developed based on Scratch3.0. It is suitable for middle high school education; the download website is <http://www.mindplus.cc/>.

Let's explore mind+ and find its features and make Maqueen a smart robot.

### 12.1 Programming

Mind+ is designed based on Scratch, but beyond. Through mind+, you can control robot, know more about artificial intelligent, IOT(internet of things) in extensions. It is one of the best choices to begin programming.

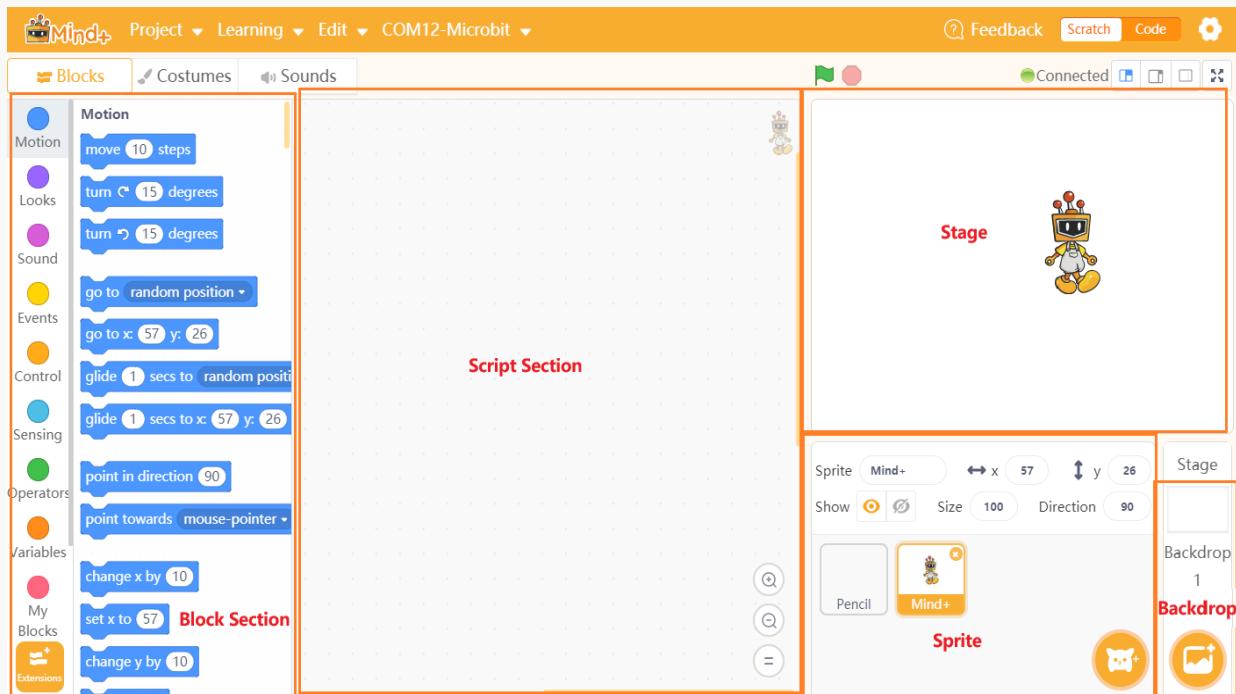


Figure 12-1

**Block section:** This is the “tool section” of the stage. We need all kinds of tools to give an excellent performance on the stage, and we can get more extra tools in “Extensions” to support different hardware.

**Script section:** show your program.

**Sprite:** adjust the size, color and position of your main character.

**Backdrop:** adjust the stage background.

After a successful installation, Mind+ will open scratch in default. We can check the mode in the right top. Click the icon to switch to code. The mode code aims at hardware, including kits, main boards, sensors, actuators, communication modules, displays, modules... Which means scratch is mainly for software programming, and code is mainly for hardware programming.

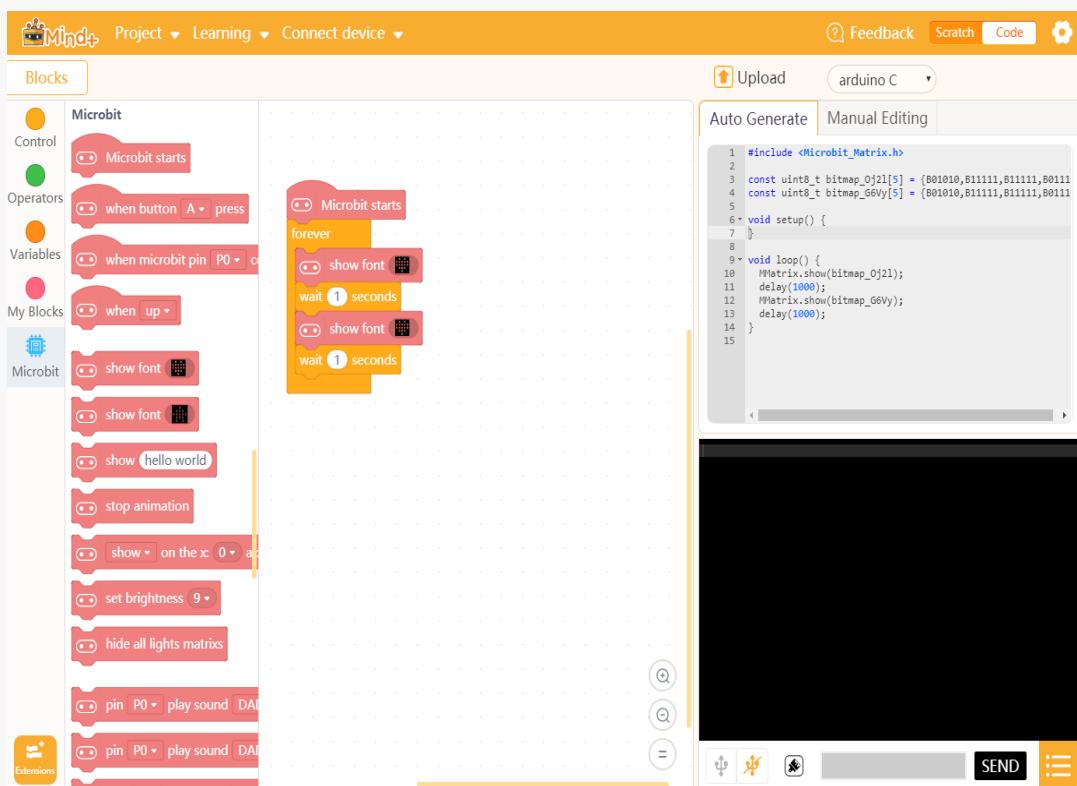


Figure 12-2

Switch to Code mode, upload to device. The lower right black area is the serial port printing area. Serial port is also called as COM. It is expansion port that adopts serial communication protocol.

In mind+, it can be used to receive signals and values from sensors. You can check Figure 12-3 for details.

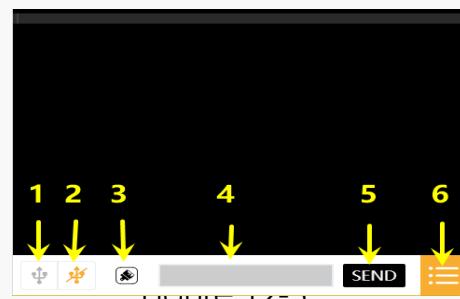


Figure 12-3

1.	Open serial port
2.	Close serialport
3.	Clear data
4.	Command box
5.	Click to send

Chart 12-1

Select code because we need to drive Maqueen(hardware). We need to load Maqueen module. Click the Extensions and choose Maqueen in kit, shown as in Figure 12-4. Then all functions belong to this kit will be loaded.

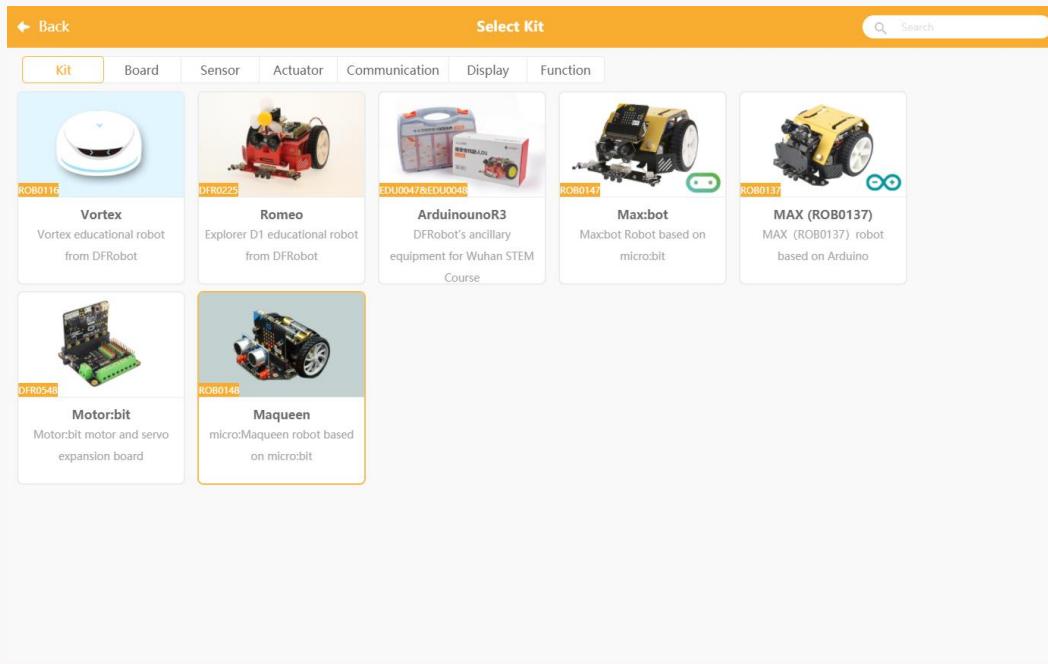


Figure12-4 Load Maqueen Module

Install drive, shown as below:

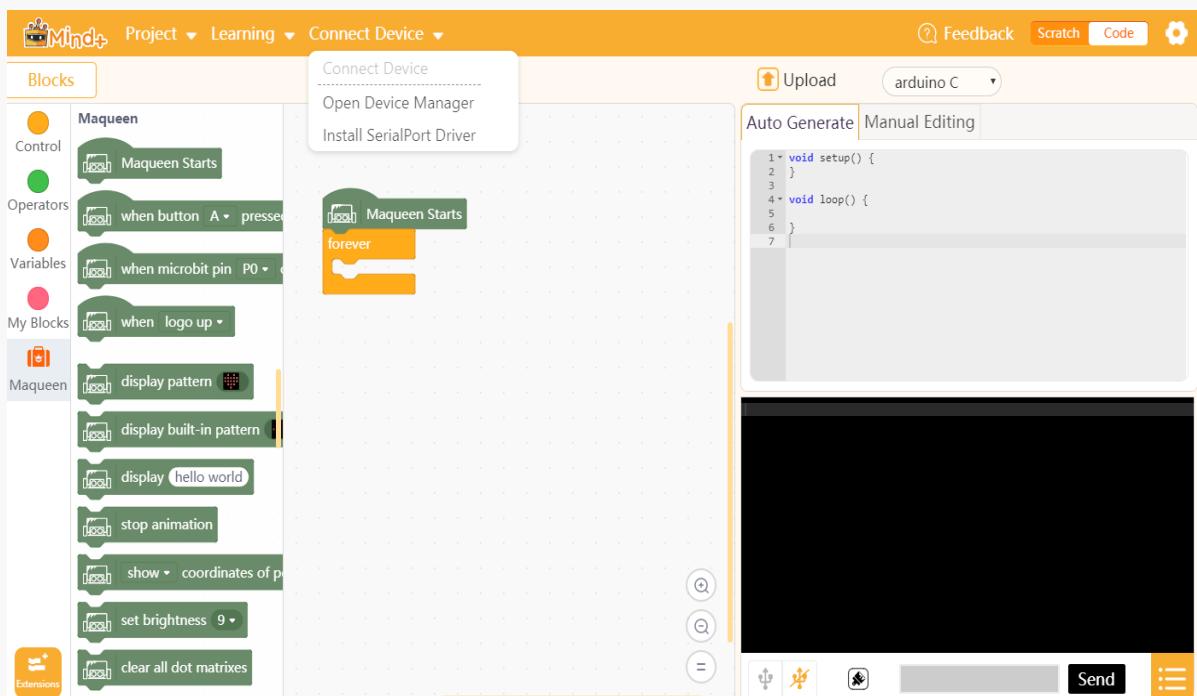


Figure 12-5

Please uninstall and re-install if the serial port you had installed the serial port.

So the preparation process is Extensions>Kit>Maqueen>Connect device (Menu Bar)>Open Device Manager>Micro:bit COM.

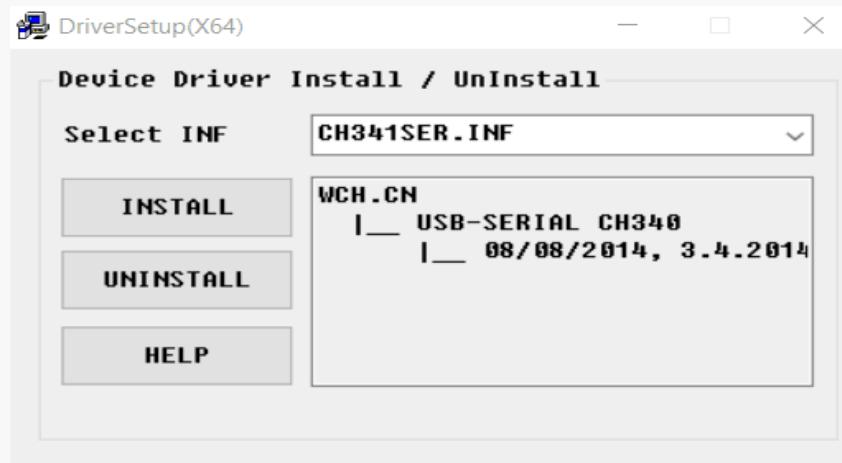


Figure 12-6

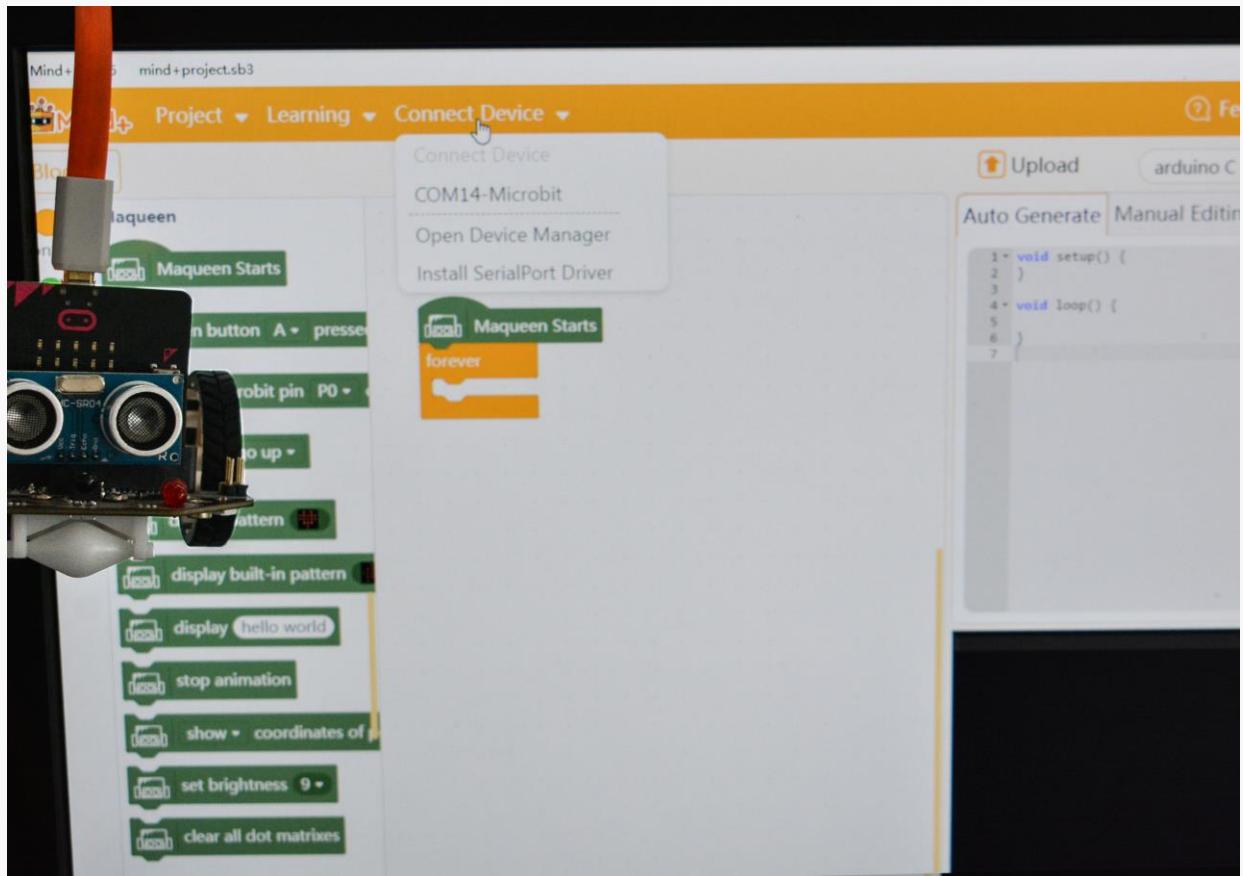


Figure 12-7

## 12.2 Maqueen, let's move!

Welcome Maqueen, how about welcome Maqueen with a program Inspection.

Imaging, Maqueen is walking at a steady step with his back straight. Then, he walks in front of the platform, stops and turns to audience, and smile. Finally, he turns back and moves on. We expect all his excellent performance.

### 1. Analyze Program

It is a simple program, the original idea is to move forward for 1s and turn left for 0.5s, then show smile and turn right for 0.5s, and keep going forward for 1s.

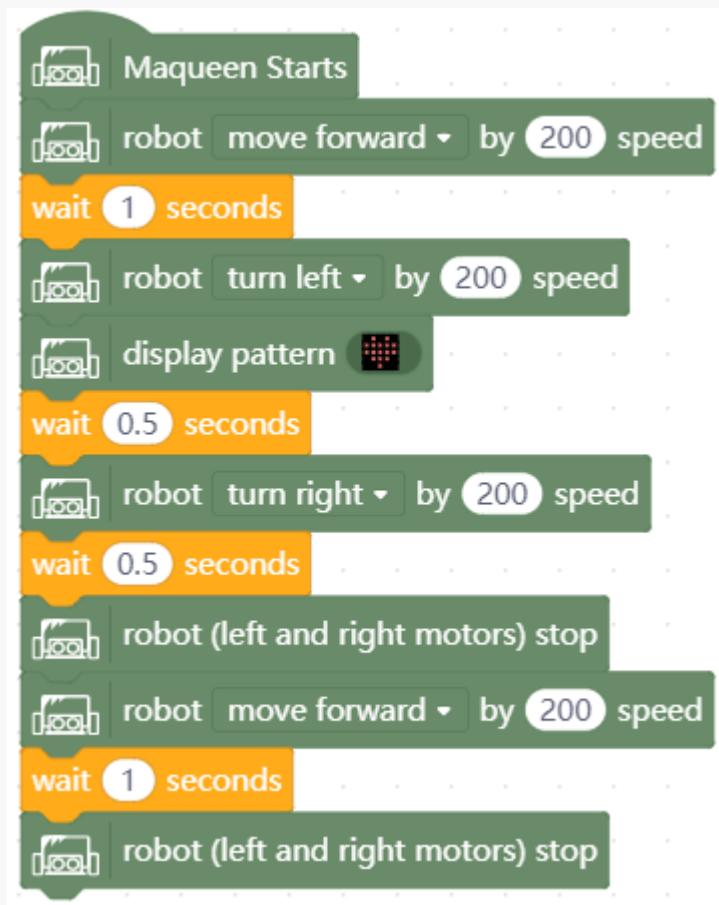


Figure 12-8

Mind+ keeps Scratch programming features, a trigger even is needed to run a program. The events like activate, press button, connect P0, logo upwards and

so on; they all could be trigger events. And the trigger event block should be put in the program begins. So, we just need to drag blocks from the blocks to the scripts and combine, then the program is done.

## 2. Optimize Program

How about your thinking? Is that similar to this program? Try uploading to the device, is there any problem?

Yes, the robot does not move as expected. What's the problem?

We find the left wheel does not turn when turning left, while the right wheel is still running. At this time, Maqueen running in circle and the left wheel is the Centre. How about make Maqueen rotate on its axis?



Figure 12-9

After repeating test, you will find that we need to obey the rule to control motors. That is to control port, speed and direction in priority, then control time and sensor conditions, finally set to the rotation speed to 0 and stop move. In this way, the motor will not keep running. The finally program is shown as below:

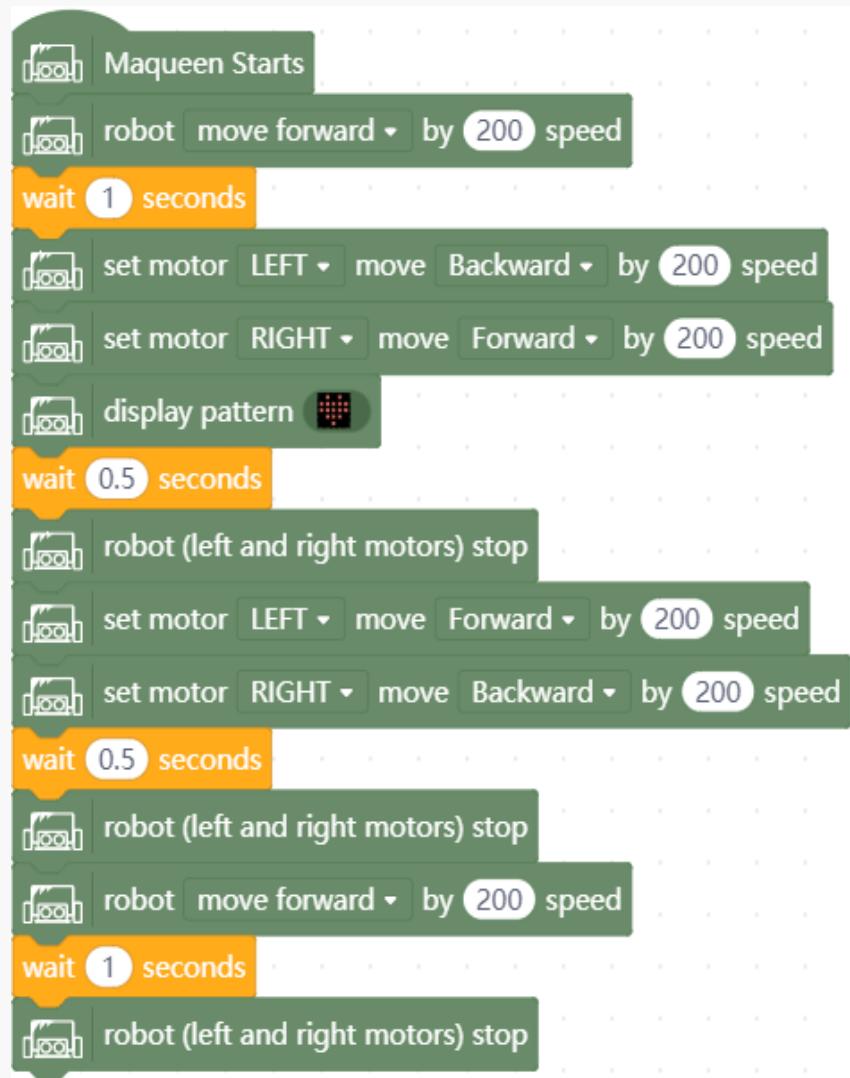


Figure 12-9

## 12.3 Training Camp

Try adding LEDs to Maqueen. Turn left, the left LED(P8) is ON. Turn right, the right LED(P12) is ON. Go straight, the LED is OFF. Then you may need to use the blocks as below. But remember to turn off the LEDs. Just have a try!

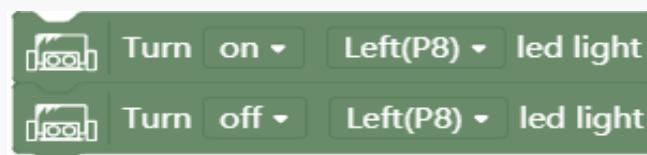


Figure 12-10

## Chapter 13. Turn on RGB Lights

There are 4 RGBs in the bottom of Maqueen: RGB0, RGB1, RGB2, RGB3. This class we will make a light show. And put Maqueen into a colorful stage.

### 13.1 Programming

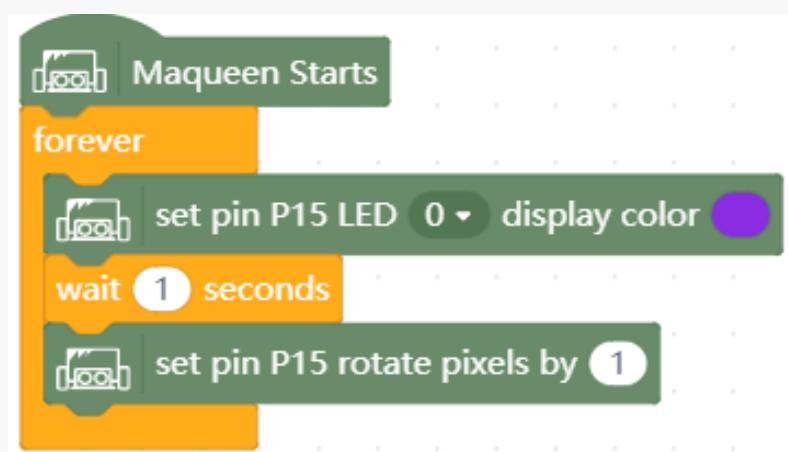


Figure 13-1

How the effect? The program will light up 4 RGBs one by one every second. Let's make a little change to switch lights.

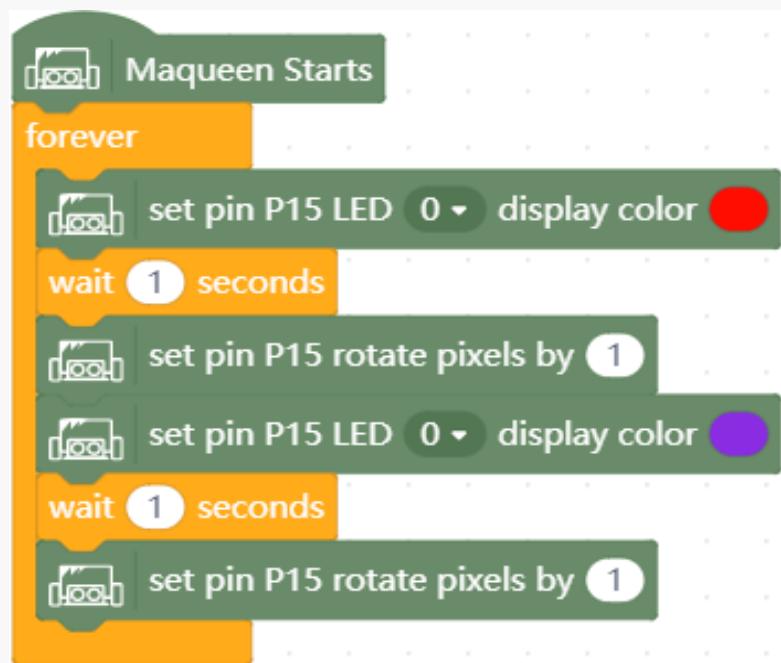


Figure 13-2

## 13.2 Time for Music

Maqueen is equipped with a buzzer in P0. And here is a buzzer switch to control the buzzer port, always ON in default. It must be cool to add sound when the color is changing! The program below is for your reference.

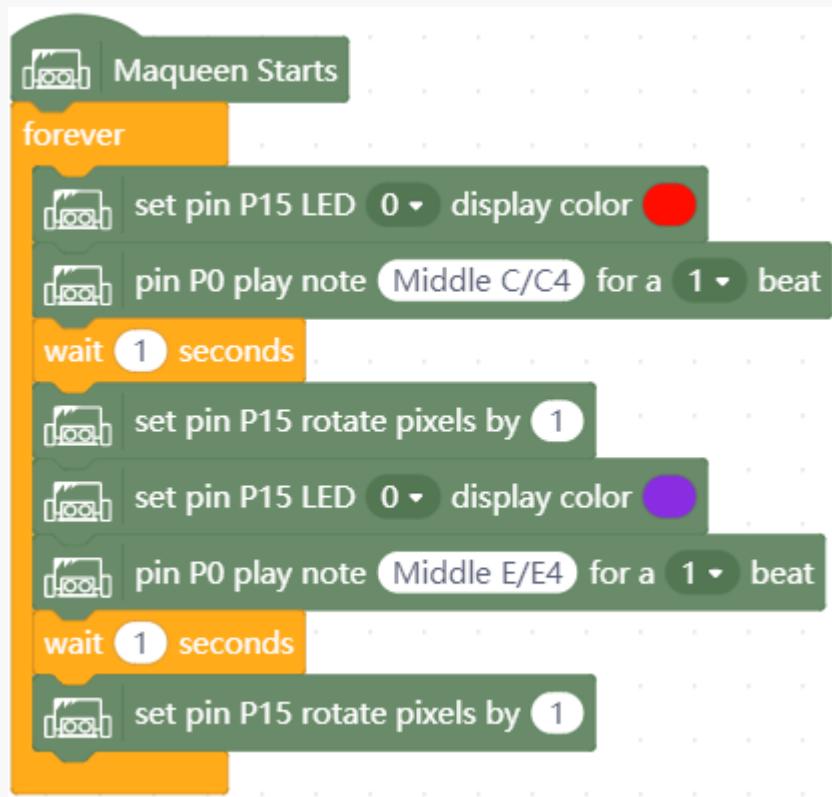


Figure 13-3

## 13.3 Training Camp

Can you change the example code to present different color shows?

Be a director, make a fantastic lighting show!

## Chapter 14. Ultrasonic Obstacle Avoidance

Here we will use an ultrasonic sensor in Maqueen. Different from human beings, Maqueen does not have eyes to see the world, but he has an ultrasonic sensor to feel the world.

### 14.1 Principle of Ultrasound

The frequency of sound waves that can be heard in the human ear is 20~20000Hz. When the vibration frequency of sound waves is greater than 20000Hz, the human ear cannot hear it. Ultrasound is named for its lower frequency limit, which is approximately equal to the upper limit of human hearing. Ultrasound is named for its lower frequency limit, which is approximately equal to the upper limit of human hearing. So we name the sound waves over 20000Hz as ultrasound. Ultrasound is widely used in Fishery. It can be used for ranging, speed measurement, obstacle detection, cleaning, welding, crushing, sterilization, inspection of metal product defects, welding of aluminum metal, washing clothes, drilling holes in glass, and looking for sunken ships. The principle of ultrasonic ranging is to use the speed data of ultrasonic waves in the air, then measure the time of the acoustic waves reflected by the obstacles, and calculate the actual distance from the launching point to the obstacle according to the time difference between the transmitting and receiving.

Check the figure below, the ultrasonic distance sensor transmits ultrasonic waves in a certain direction, and starts timing when emitting. The ultrasonic waves propagate in the air, and return to the emission area once encountered on the way, and the ultrasonic receiver stops timing immediately upon receiving the reflected wave. The ultrasonic wave travels at 340 m/s in air at 15 ° C. Based on the time recorded by the timer, the distance from the launch point to the obstacle can be calculated.

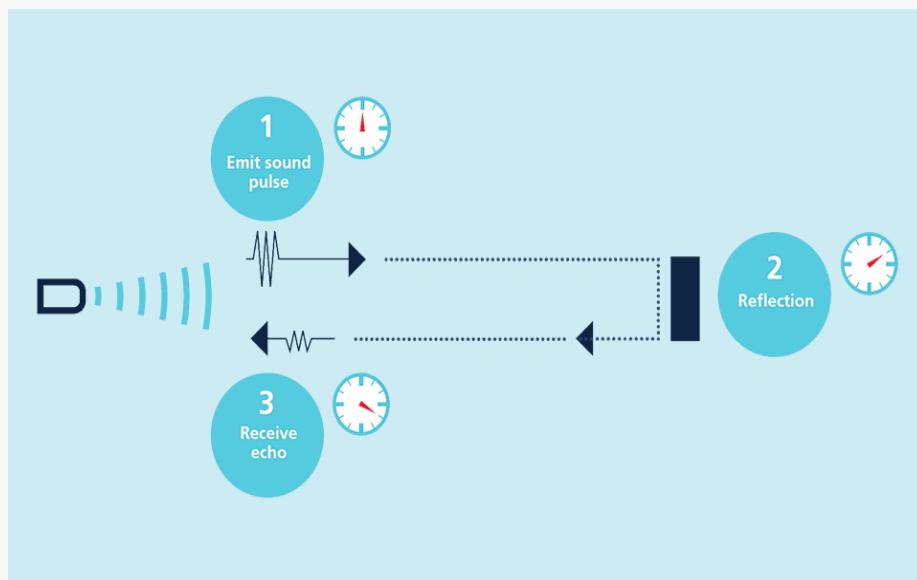


Figure 14-1

The onboard ultrasonic sensor in Maqueen is HC-SR04. It can offer non-touch ranging from 2cm to 400cm, the precision is as high as 3cm. After the block trigger signal, the ultrasonic wave is emitted. When the ultrasonic wave reflected back for reach object, the module outputs an echo signal. Then we can determine the distance of the object by the time difference between the trigger signal and the echo signal. VCC is 5V, GND is grounding, Trig is trigger signal input, and Echo is the echo signal output.



Figure 14-2

## 14.2 Obstacle Avoidance Program

Maqueen turns right when detected an obstacle.

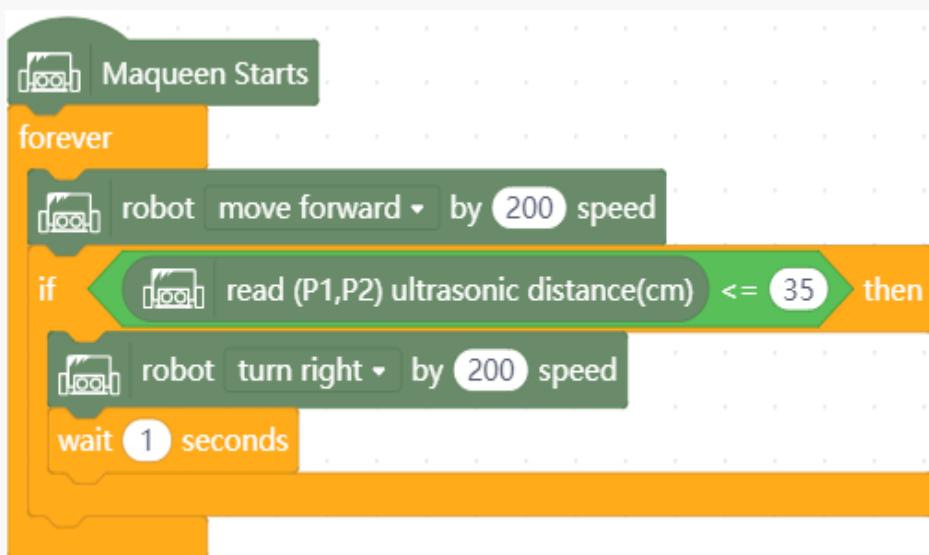


Figure 14-3

Make a test, you will find the robot is obedient. It really turns right to avoid obstacles. How could it achieve? Let's analyze the program.

In the process, once an object is detected, Maqueen turns right; otherwise, it keeps going straight. The reason why Maqueen can change status again and again is the block forever. Look the blocks in the figure 14-3, the blocks covered by forever are repeatable to execute.

Maqueen keeps going straight, only change status when meets obstacles. There are two movements, go straight and turn right. Execute 2 conditions but without forever cannot let Maqueen keep running. So, the block forever is needed at here. The simple movement go straight can be realized by a simple block that given in mind+.

The hard part is to realize the way to turn right. Take it easy, let's ask and answer. When we walk straight and need to turn, how could we do? We may stop and turn, or just turn and keep walking. But a response time is needed at here. Especially, a robot little Maqueen. So, we need to leave some time like 1s for

Maqueen to restart. Ok, now we just need to combine turn right with space time together, then the other condition is finished. But how could we set the effective distance range of obstacle be found? The answer is the blocks if... then..., and  $\leq$ . Then we combine all, it works.

## 14.3 Training Camp

Hi there, try programming a program when the distance between Maqueen and obstacle is  $\leq 35\text{cm}$ , Maqueen turns left; And when the distance  $\leq 10\text{cm}$ , Maqueen turns back and turns right.

## Chapter 15. Light-chasing Maqueen

The dark night has given me a pair of black eyes, but I use them to find the light. This sentence is from a Chinese poetry wrote by Gu Cheng. Humans love bright, so does robots, so does Maqueen. When the bright is strong enough, Maqueen will track the light. With the growth of light value, he runs faster.

Firstly, we should know how could Maqueen knows the environment light strength? Is there a light sensor in micro:bit? Where is it?

In fact, there is no real light sensor in micro:bit, but how could it sense the light strength? LED matrix. The LED matrix in micro:bit is used to sense the environment lights. The principle is to convert LED drivers to inputs and sampling the voltage weaken time, which is roughly in direct proportion to the environment light level.

### 15.1 Get Data

We need to use serial monitor to print the environment light strength data.

So, to begin with, we need to install a serial port driver.

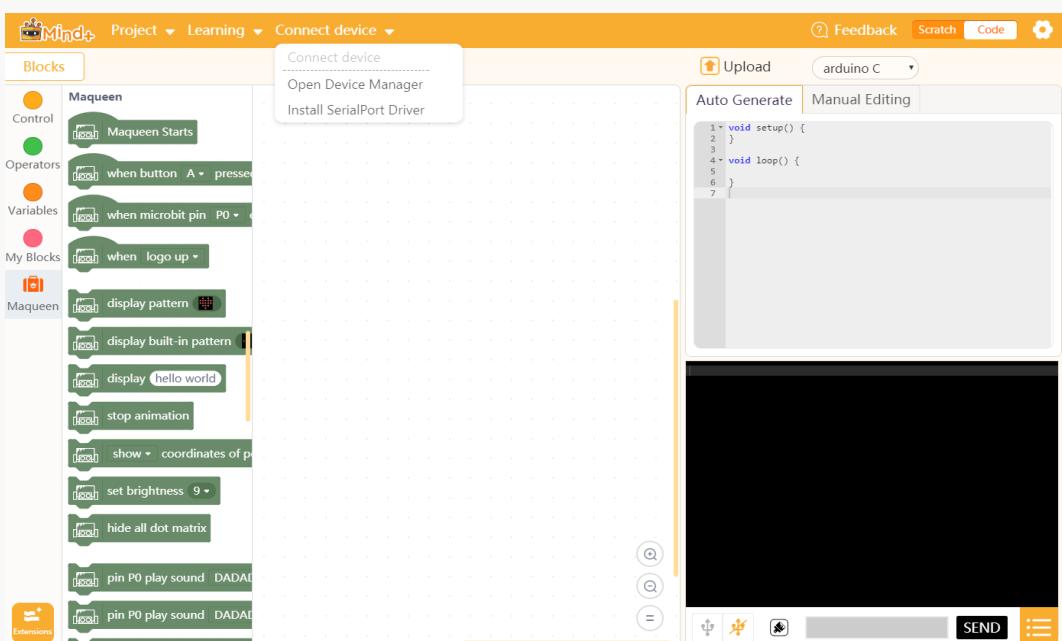
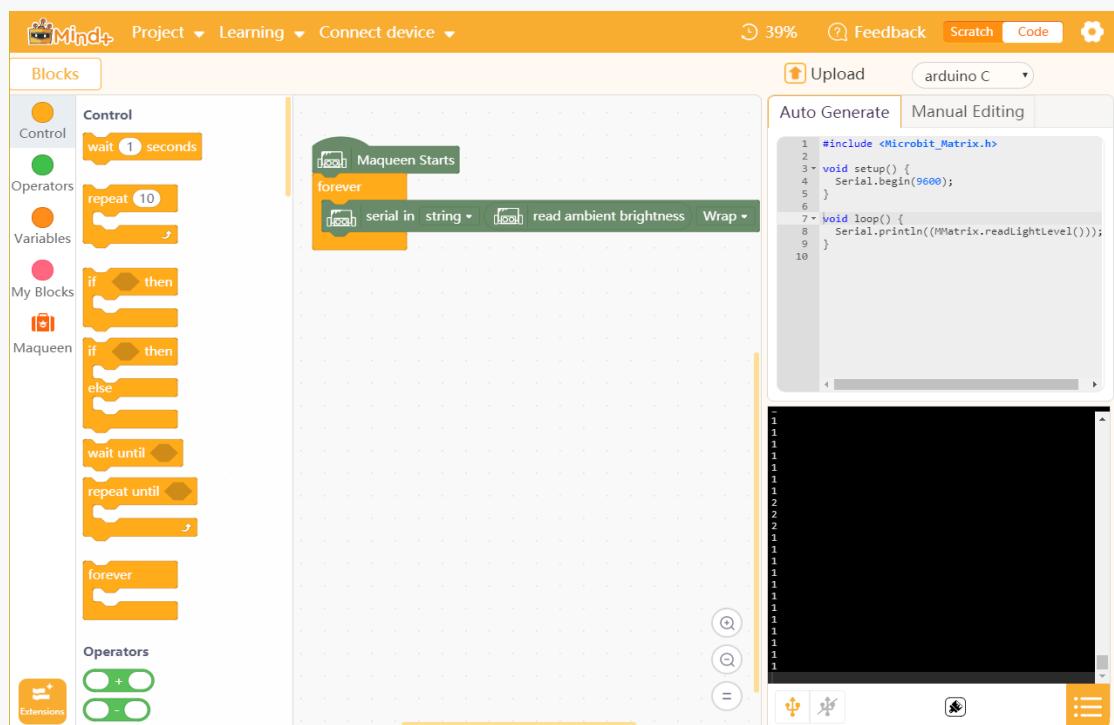


Figure 15-1

When the driver is installed, click Connect device to connect hardware, then click the serialport icon in the black serialport area. And the icon “open Serial” will turn orange.

Programming to get the environment light value. From the data below we know the current light is low. When light it with flashlight, the light strength is close to 200.



```
#include <Microbit_Matrix.h>
void setup() {
    Serial.begin(9600);
}
void loop() {
    Serial.println(MMatrix.readLightLevel());
}
```

Figure 15-2

## 15.2 Light-chasing

Thinking draft: Maqueen keeps turning left in low light. While, when the light strength is over 100, Maqueen follows the light direction to move. The speed relates to the light strength. The larger the light, the faster the speed. And in other conditions, Maqueen still executes turning left to find lights.



Figure 15-3

### 15.3 What is Forever?

There are 3 blocks related to repeat in control, infinite loop, finite loop and loop with conditions

Forever (infinite loop)	repeat for limited times	loop with conditions

Chart 15-1

#### 1. Forever (infinite loop)

Forever, as the name suggests forever is to repeat forever, again and again. The lack of the block Forever will lead to short movement in the project above.

#### 2. Repeat for () times: means specify the robot to repeat for concrete times

E.g. Enable Maqueen to walk square

Of course, we can proceed this via sequence structure, like move forward for 2s, then turn left for 0.75s, then turn left again for...

In fact, we can use the block infinite loop to realize it. The key point is the part to loop and loop time. Move square, which means moving forward and turns direction and repeat for 4 times. And the repeat time regards to power supply and friction, so you can make test and change it.

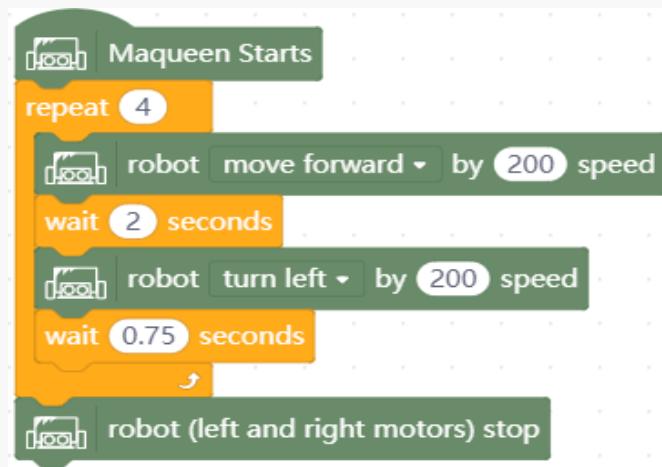


Figure 15-4

### 3. Repeat until: loop with conditions

Judge the condition after executed loop body for once, keep executing until the condition is satisfied. Flow chart is shown in the Figure 15-5.

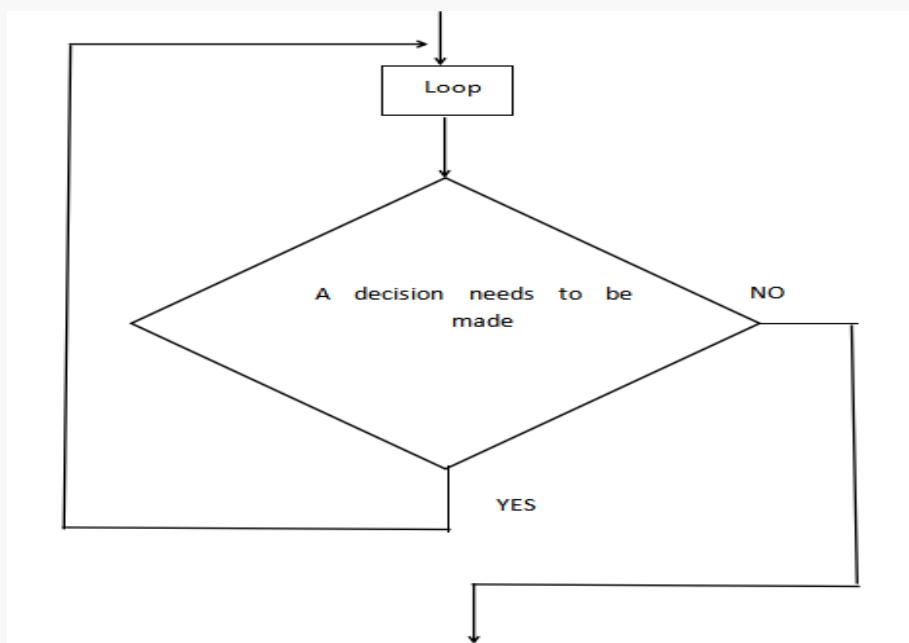


Figure 15-5

Have a try: try modifying the light-tracking program to a repeat until structure, the result is shown in the figure 15-6:

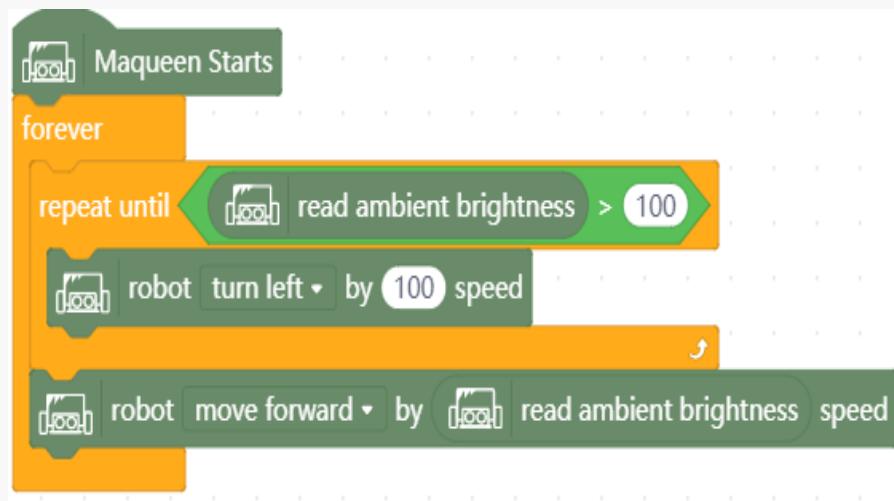


Figure 15-6

## 15.4 Training Camp

Maqueen can track light now, but how could he judge 2 lights in different directions?

## Chapter 16. Line-tracking Maqueen

Line-tracking is a classical robot sport. It's a self-feedback process. We can see line-tracking in many robot competitions. Through line-tracking study, we can have a deeper thinking of robot movement principles.

In the bottom of Maqueen, there are 2 line-tracking sensors Line-L and Line-R. Each sensor consists of an infrared transmitter and an infrared receiver, the detect range is from 10mm to 20mm.

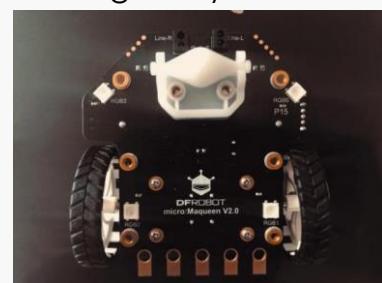


Figure 16-1

The infrared transmitter keeps emitting infrared light to the ground. When the ground (white and light colors) reflect IR light, the receiver receives and the output is 1(the blue LED will light on). When the ground (black or deep colors) observes or cannot reflect the IR light, the receiver cannot receive the IR signal, the output value is 0. In line-tracking, the sensor will keep detecting reflect value from the ground to distinguish black and white line.

### 16.1 Single Line-tracking Sensor

Single line-tracking is a very basic operation. It is common in Lego robot learning. What is the principle of single line-tracking sensor and double line-tracking sensor? Come to see.

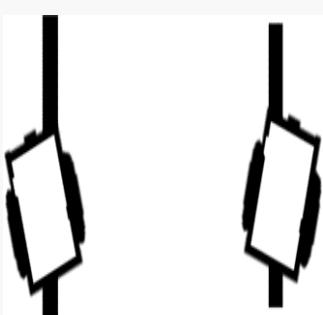


Figure 16-2

Single line-tracking is the fundamental line-tracking way, the light sense sway back and forth from write lines and black lines. So the single line-tracking self-movement can only be realized by alternative moving of motors in two sides, the path way is a "Z". This line-tracking way is simple in construction and easy to master. But for the only 1

sensor, it cannot accomplish complex line-tracking task and the running speed is slow.

**Design Idea:** put the sensor to the left of the black wire and turn right, if the sensor detects black line, turn left; and if the sensor detects white line, turn right. And run it in loop.

**Result:** the robot keeps turning right and turning left, and finally moving with the path “Z”.

## 16.2 Double Line-tracking Sensor

Firstly, we need to focus on the line rule. We can draw or use adhesive tape to make a black path. Please do ensure two sensors are in the black part.

Imagining what will happen to 2 sensors in moving? What actions should be designed to react to different situations?



Figure 16-3

Value of left line-tracking sensor	Value of right line-tracking sensor	State of the two sensors	Maqueen's movement
0(black)	0(black)	Both are on the blackline	Move forward
0(black)	1(white)	Right sensor deviates from the line, car body inclines to right side.	Only right motor rotates, the car turns left
1(white)	0(black)	Left sensor deviates from the line, car body inclines to left side.	Only the left motor rotates, the car turns right.
1(white)	1(white)	Neither are on the blackline	Move backward

Chart 16-1

### 1. Read data from Serialport

To intuitive understand double line-tracking sensor values, let's program to test and verify.

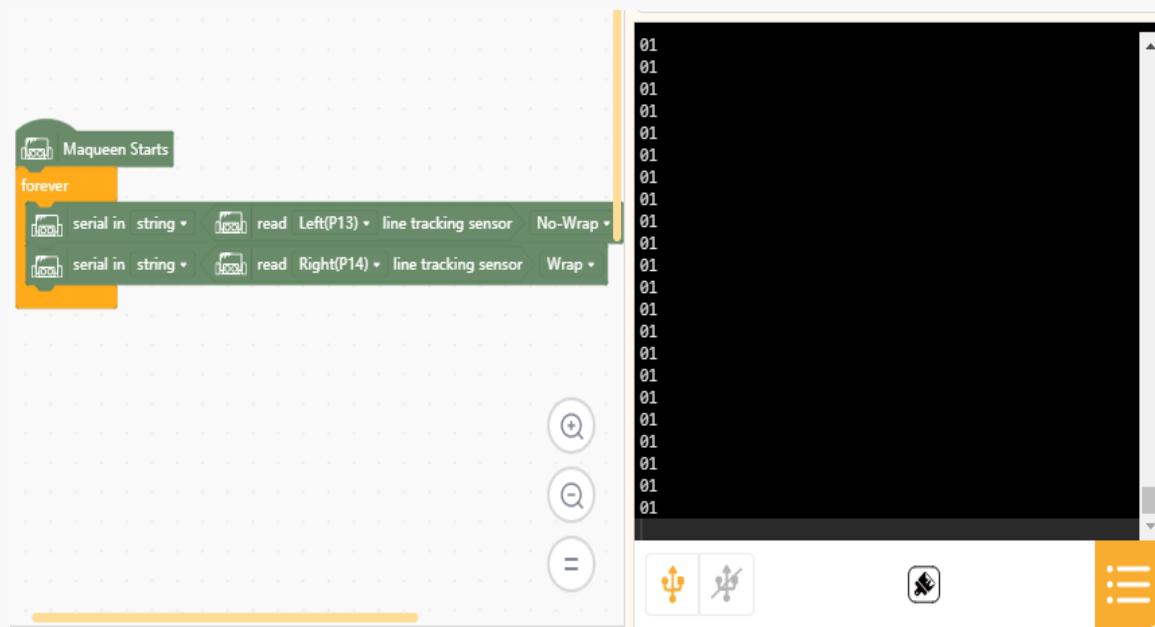


Figure 16-4

## 2. Program flow chart

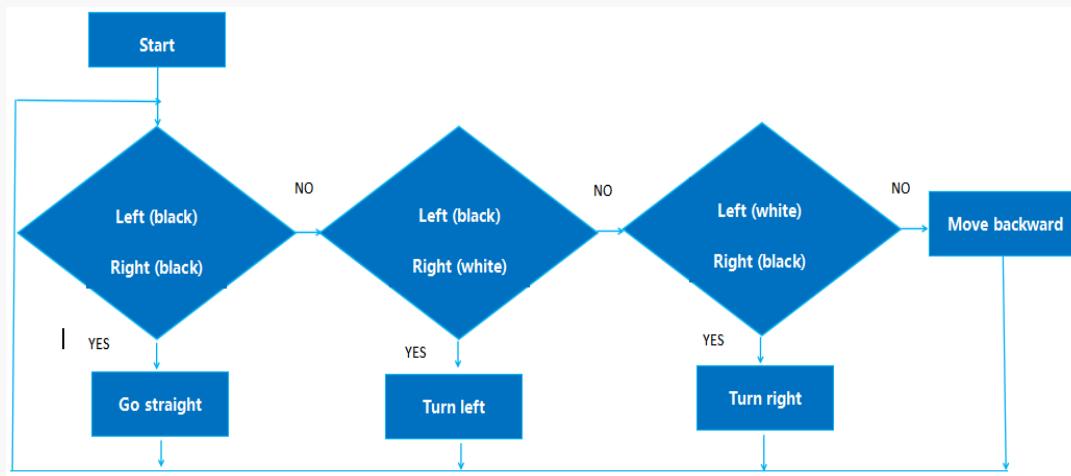


Figure 16-5

## 3. Add user-defined module

Function is a set of code segment to achieve a function. Which means we need to transform functions to modules to realize movement, actions, controls and so on. Function is the algorithm set of some function, and advances program reusable ability.

Create a new function: My Blocks -> Make a Block->Block Name.

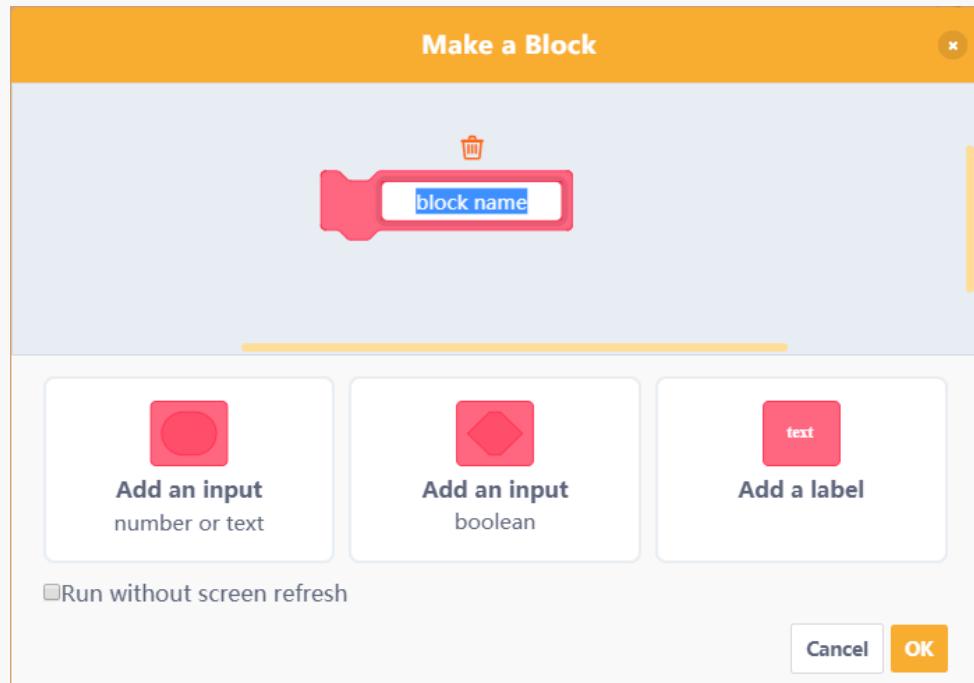


Figure 16-6

Use functions to present the robot motion. And we just need to call functions in programming.

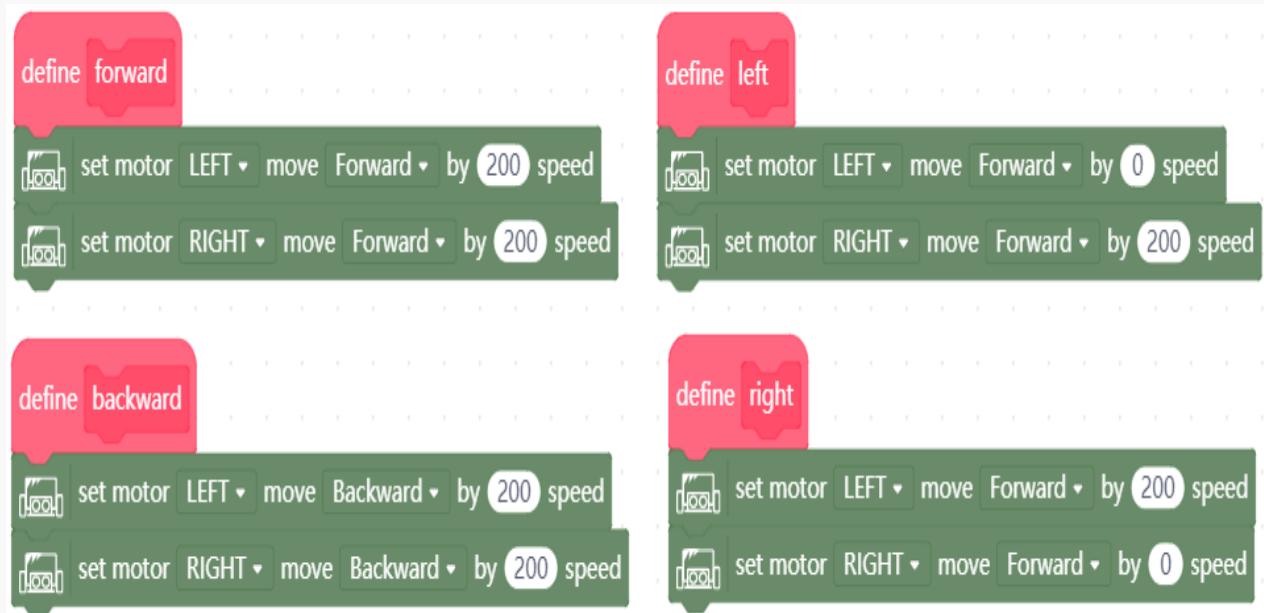


Figure 16-7

#### 4. Coding the main program

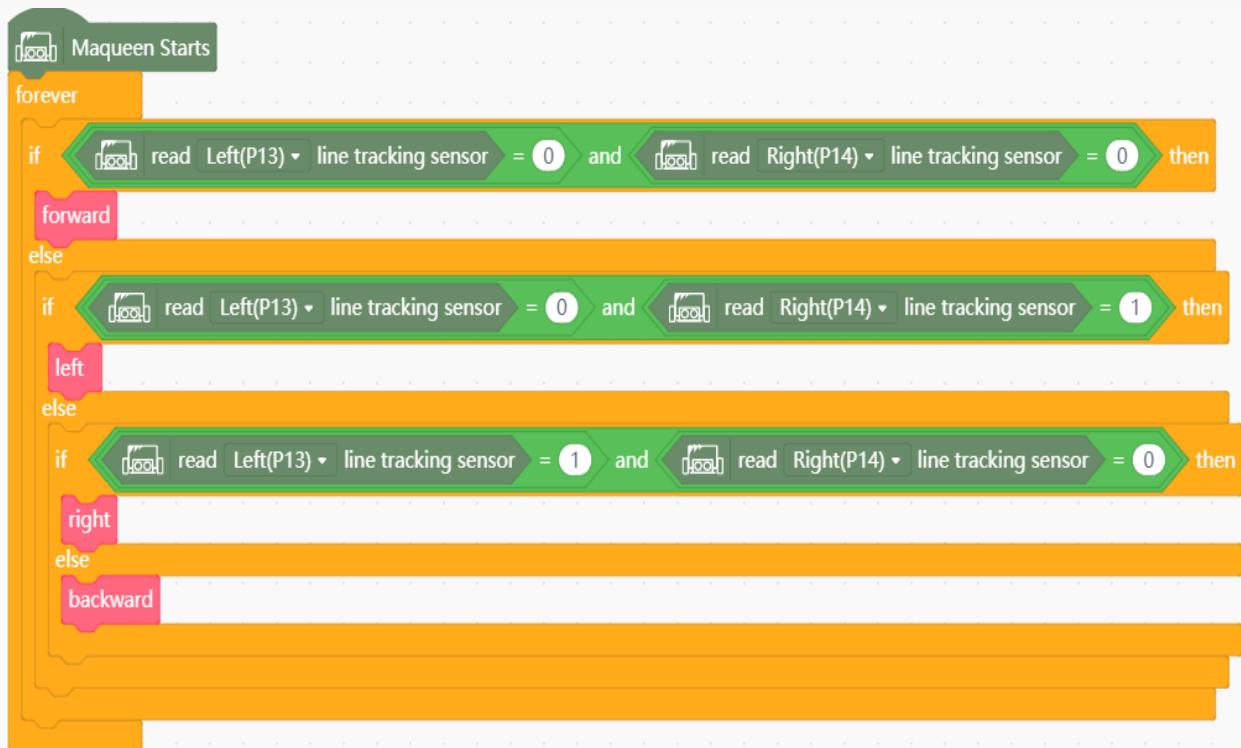


Figure 16-8

## 5. Test

The line-tracking test path is shown in the figure 16-9, what's the disadvantages? Can we modify these disadvantages by decreasing the speed? Besides, will the environment influence the robot judgement?

After learned the principle of line-tracking, can you realize line-tracking by other programming methods?

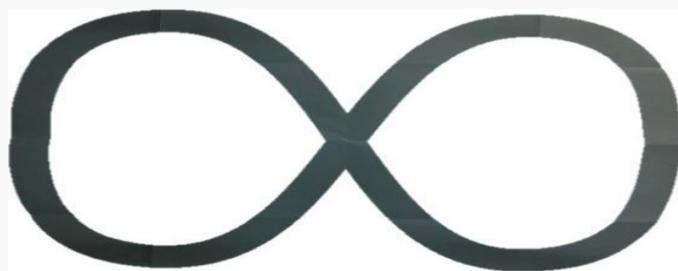


Figure 16-9

## 16.3 Training Camp

Try programming to make Maqueen move in the light flat and stop in the edge.  
Shown in the figure16-10.



Figure 16-10

## Chapter 17. Infrared Remote-control Robot

The visible light we can see is the visible light field of the naked eye ranges from 400 nm (violet light) to 700 nm (red light). Light with a wavelength between 760 nm and 1 mm is called infrared light, and it is a light that is invisible to the naked eye.

Infrared remote-control device is common to see in our life. For low produce cost, it is widely used in electric device like TVs, air conditioners, CD players, telephones and so on. In general, infrared remote-control system is divided to 2 parts, emitter and receiver. The main part of the emitter is the infrared emitting diode (LED). In fact, LED is a special emitting diode, because its inner material is different from common emitting diodes, so it emitting infrared lights when be powered. In practical applications, the LED only work when reverse bias is added, which means LED should be used in reverse in circuit to get high sensitivity.

The infrared remote-control has many advantages. Its signal will not influence environment and interface other electric equipment. Because the infrared light cannot go through wall, so a common remote controller can control machines in different rooms without any interference. In general, you just need to connect circuit in a right way and need not to do any adjustment to work. It is easy to program; multipath remote control is available. Therefore, infrared remote-control is widely applied to household applications and indoor close-range control (within 10meters).

Let's welcome the remote-control tool shown in the figure 17-1.



Figure 17-1

## 17.1 Get Key Value

### 1. Programming to get key value from the emitter and upload to device.

Drag the plastic strip in the battery to power-on. Toward to Maqueen and press keys, the serialport will show related key value in real time.

### 2. Set variables

Variables are named storage zones in computer. Once you create a new variable, the program will set a storage zone to store the variable and name the zone at the same time. Therefore, you can get all stored in this zone by using the variable name.

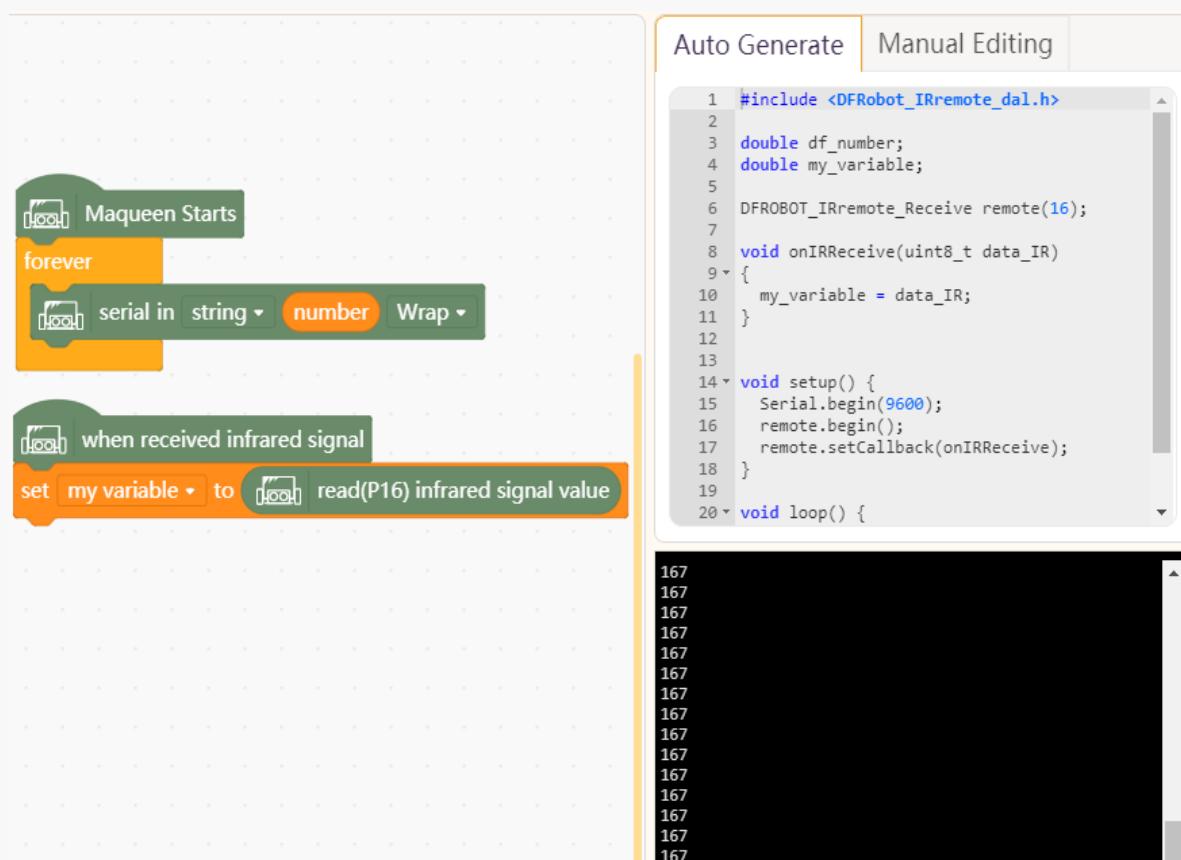


Figure 17-2

There are 3 types of variables are provided in Mind+, Numbers, Strings and Lists. The program above uses Numbers, try name the variable as number. Create a new function: Create number -> Name the variable. Then you can find the variable number in Variables.

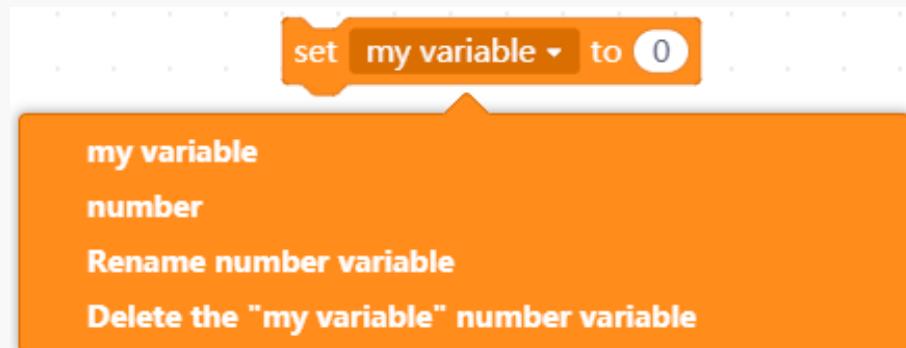


Figure 17-3

## 17.2 Remote-control Maqueen

Through test, part of the key value result is shown as blow. Please check the other functions if in need.

Key	Value	Key	Value
VOL+	127	0	207
VOL-	111	1	247
Left arrow	223	2	119
Right arrow	159	3	183
Pause	95	4	215
Power switch	255	5	87
6	151	8	103

Chart 17-1

Task: remote-control Maqueen to move forward, move back, turn left and turn right, then pause. Take keys like VOL+ as an example:

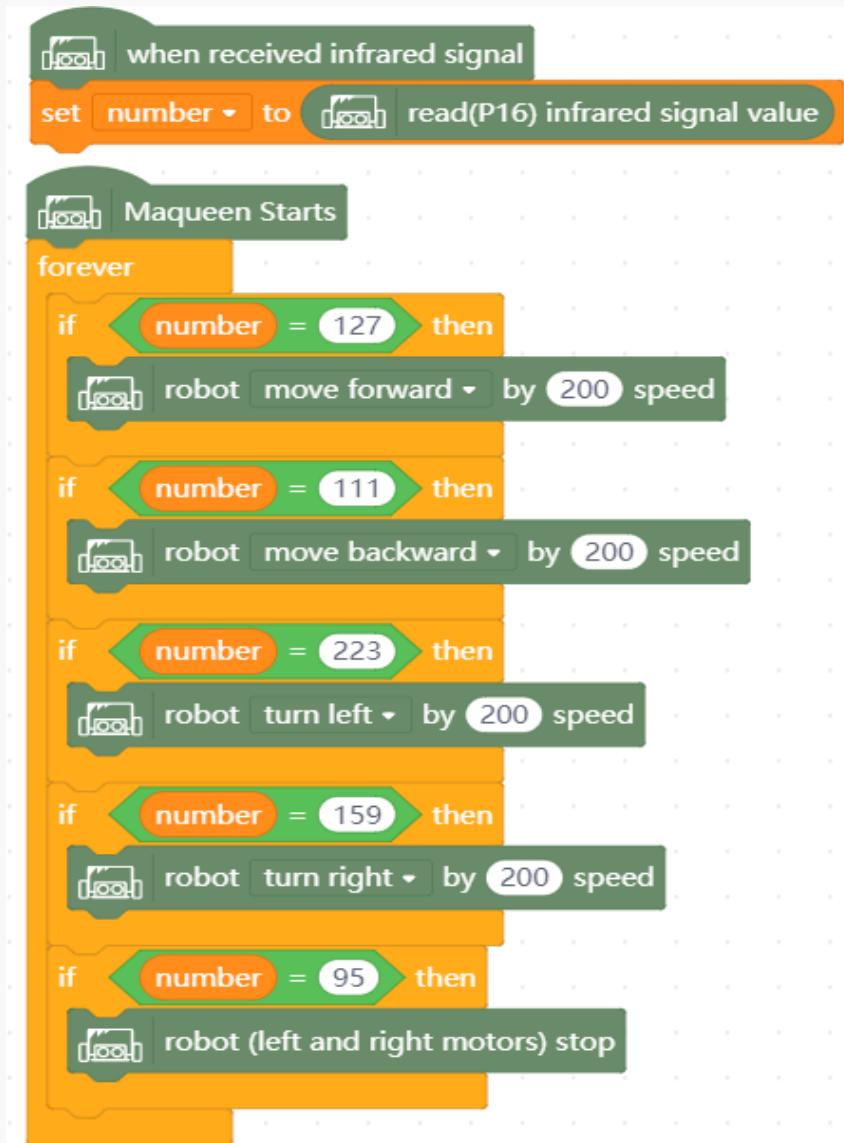


Figure 17-4

## 17.3 Training Camp

Try programming a remote-control program to perfect the buttons in the controller. Like you can use 2,8,4,6,5 to make Maqueen execute more interesting actions!

## Chapter 18. Radio Remote-control Maqueen

We have already learned infrared remote control, but what's the difference between IR and RF Remote Control?

Infrared remote-control is belongs to light transmission wave. It transmits electricity signals by infrared rays, but it cannot go through obstacles. The infrared remote control is strict in environment requirement, the emitter must face to the receiver, and should not have any obstacle. And there is no frequency limit. But the radio remote-control is depends on radio signal to spread in the air. It can be controlled by the radio frequency and can go through obstacles. So you can choose IR control and RF control according to the practical necessities. By the way, infrared control circuit is simpler than radio control.

The difference between IR and RF Remote Control is different from carrier. The infrared remote-control is directional, cannot be obstructed, short work distance (usually 7meters at most), free from electromagnetic interference. While, radio control transmits signal via radio wave. It is unidirectional, not need face to face control, long distance (more than 10meters, even several miles) but easy to be interfered. Radio remote-control can be applied to long distance wild and other unidirectional areas, such as industrialized control.

The rf remote controller is a device that control remote objects by radio signals. When signals are accepted by the receiver, you can drive and control machines, electricity device to work, such as turn on/off circuit, control motor and so on. As a complementary controller with IR controller, RF controller is widely used in garage door, electronic door, barrier gate remote-control, alarm, wireless smart home and industrial control.

## 18.1 Gamepad



Figure 18-1

Micro:bit gamepad is based on micro:bit mainboard expansion pad. Plug in Micro:bit then you get a wireless remote control pad or a wireless game console. The gamepad has a total of 8 buttons, the left side have up, down, left, right four buttons, the right side has X, Y two buttons, and two buttons A, B are in the front of the gamepad. The gamepad also has programmable built-in vibration motor, buzzer, and LED. Using graphical programming, it will turn into a multimedia vibration controller or multimedia interactive game console instantly.

### Pinout description:

Pin	Button/key	Pin	Button/key	Pin	Device
P1	Button X	P14	Key Left	P0	Buzzer
P2	Button Y	P15	Key Right	P16	LED
P8	UP key	P5	Key A	P12	Vibrator
P13	Down key	P11	Key B		

Chart 18-1

**Task:** code to control LED matrix by gamepad, when up is pressed, show arrow UP in micro:bit; when down is pressed, show arrow Down; when left is pressed, show arrow left; when right is pressed, show arrow Right.

**Idea:** try making a test. Press up in gamepad, show arrow UP in micro:bit, otherwise show music note.

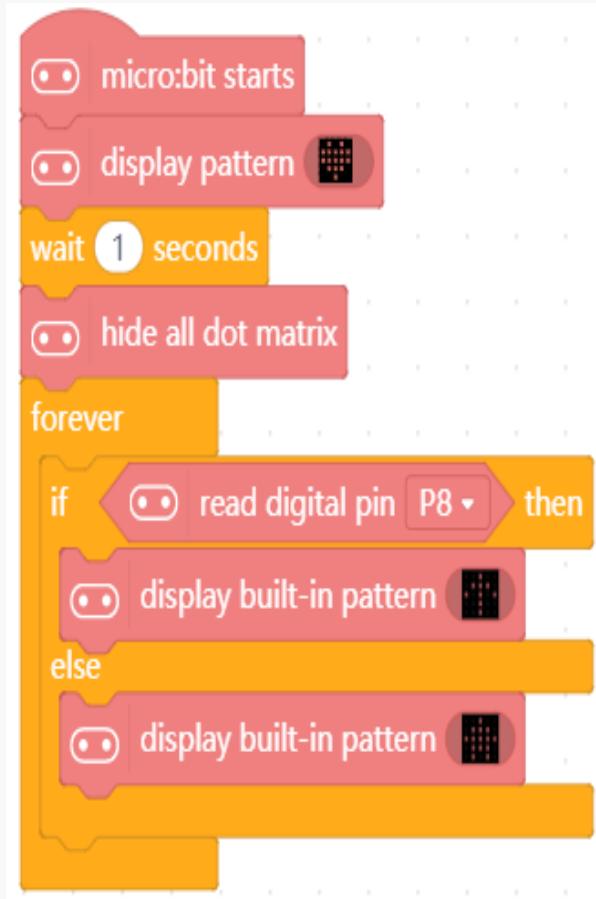


Figure 18-2

**Result:** when up is not pressed, the arrow UP shows. So, let's make some change, change conditions to logic NOT.



Figure 18-3

The final result is in figure 18-4:

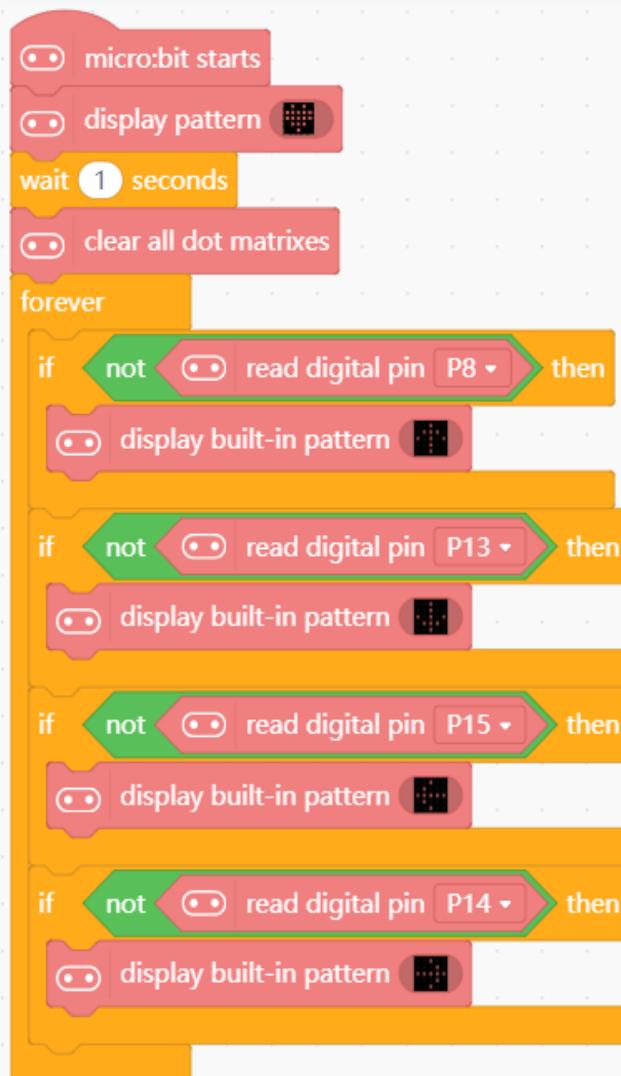


Figure 18-4

**Have a try:** try adding functions up and down, press X and the buzzer is ON, press Y and the gamepad vibrates.

## 18.2 Control Maqueen with Gamepad

Micro:bit has onboard 2.4G wireless communication, when put 2 micro:bit boards together, they can communicate with each other by transmitting signals and receiving signals.

**1. Task: remote-control Maqueen to move forward, move back, turn left, turn right and pause by up, X, A, B, Y in micro: gamepad.**

**2. Analysis:** we need to control micro:bit in Maqueen by the gamepad, so Maqueen should be a receiver and micro: gamepad be a transmitter. Then we can program to make the transmitter communicates with the receiver.

Firstly, 2 micro:bit boards must in a same group, so that they can communicate with each other without interference from other groups. Secondly, the transmitter (Client) can send different types of information when different keys are pressed. Information could be strings and numbers. Thirdly, the receiver can execute corresponding actions when different numbers are received.

**3. Programming:** the transmitter program (micro: gamepad):

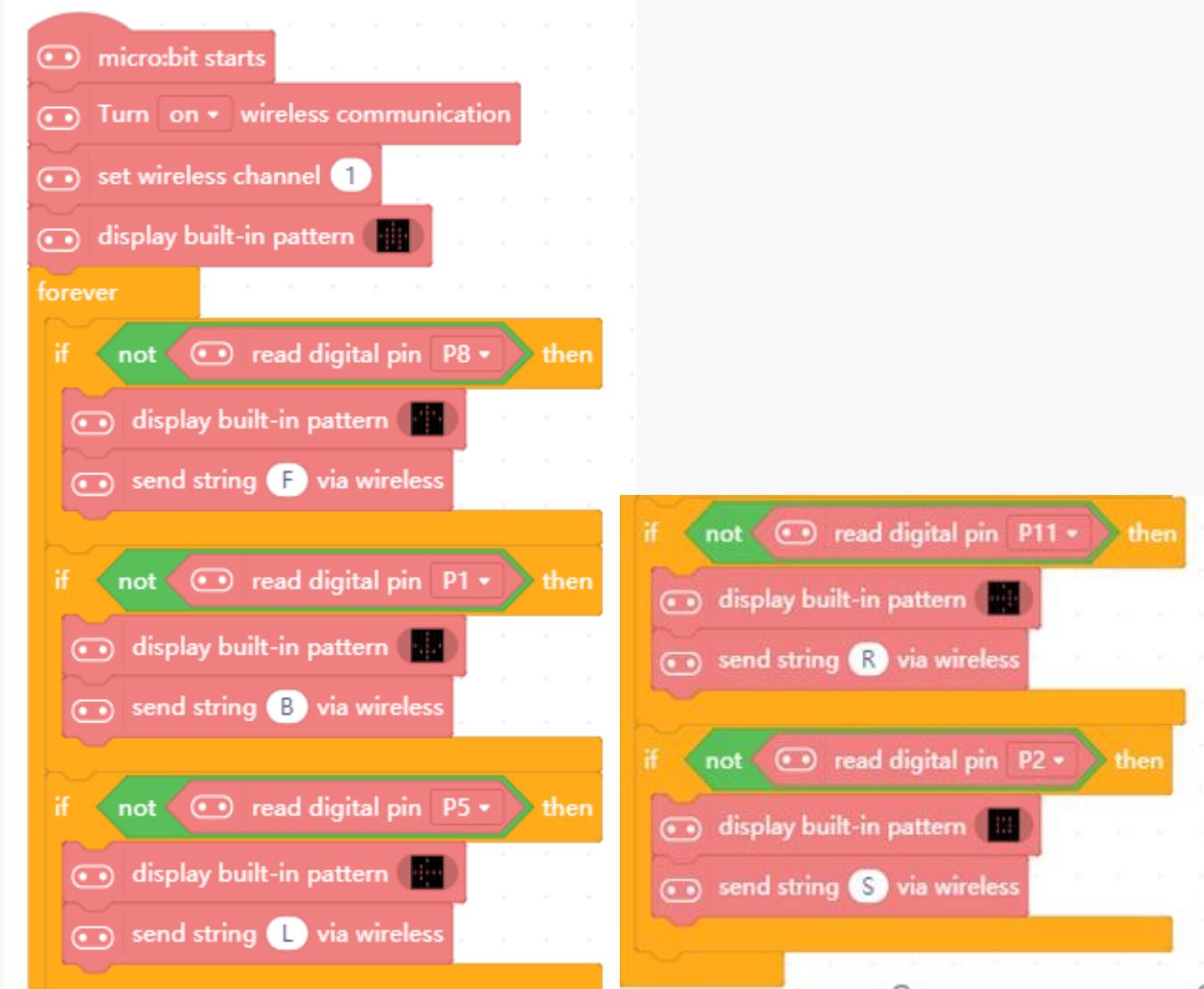


Figure 18-5

The receiver program (Maqueen:)

```
Maqueen Starts
  turn on wireless communication
  set wireless channel to 1
  display built-in pattern

when received wireless data
  if receive data via wireless = B then
    turn off Left(P8) LED light
    turn off Right(P12) LED light
    robot move backward at speed 200
    pin P15 the RGB0 (0) LED display color
    pin P15 the RGB3 (3) LED display color
    pin P15 the RGB1 (1) LED display color
    pin P15 the RGB2 (2) LED display color

  if receive data via wireless = F then
    turn off Left(P8) LED light
    turn off Right(P12) LED light
    robot move forward at speed 200
    pin P15 the RGB0 (0) LED display color
    pin P15 the RGB3 (3) LED display color
    pin P15 the RGB1 (1) LED display color
    pin P15 the RGB2 (2) LED display color
```

```
if receive data via wireless = B then
  turn off Left(P8) LED light
  turn off Right(P12) LED light
  robot move backward at speed 200
  pin P15 the RGB0 (0) LED display color
  pin P15 the RGB3 (3) LED display color
  pin P15 the RGB1 (1) LED display color
  pin P15 the RGB2 (2) LED display color

if receive data via wireless = L then
  pin P15 the RGB0 (0) LED display color
  pin P15 the RGB3 (3) LED display color
  pin P15 the RGB1 (1) LED display color
  pin P15 the RGB2 (2) LED display color
  turn off Right(P12) LED light
  robot turn left at speed 200
  turn on Left(P8) LED light
```

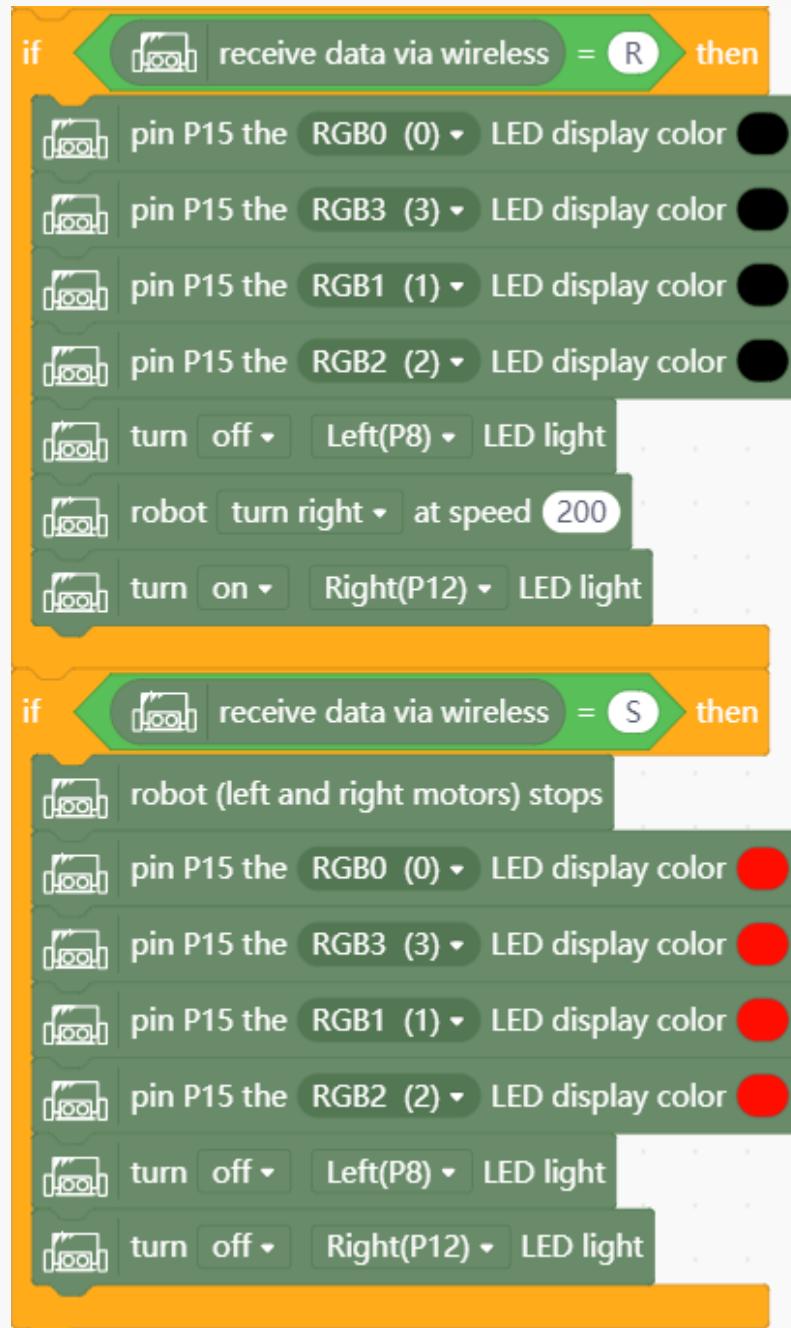


Figure 18-6

### 18.3 Training Camp

Besides actions below, can you control Maqueen do other actions with micro:gamepad?

## Chapter 19. Research and Rescue Maqueen

Today, robot has widely applied to many fields. In war, robots go first and soldiers follows. Scout robots not just can avoid casualties but also can feedback the front-line situations in real time. About research and rescue robot, even in the early 911 event, robots from most of American robot companies and research departments had joined the on-site rescue in the New York City's World Trade Center. Compared to human, robots have many advantages, such as reach to narrow and dangerous zones with shorter time.

This class, we will take advantage of common INPUT and OUTPUT pins in Maqueen to realize a simple research and rescue robot. According to the requirements, Maqueen need to satisfy two conditions, research and rescue.

So the robot must integrate temperature sensor and smoke sensor. It can feedback the on-site situation and remote transmit information to inform searchers.

And Maqueen must equipped with a thermoelectric infrared sensor and a wireless camera to detect vital signs in long distance and notify the rescue place to researchers. Meanwhile, the robot should avoid obstacles by itself. It can handle relief materials like nutrition liquid to gain time before researchers arrive. With a manipulator, the robot can help remove heavy objects to rescue suppressed people.

Nowadays, research and rescue robots become more and more important in earthquake, fires, floods.

### 19.1 Thinking Draft

Take advantage of a thermoelectric infrared sensor, Maqueen can detect vital signs. Then transmits information to the receiver by the wireless module. When

data received by the receiver, it would drive the servo rotate and animate rescue operations, such as providing nutrition liquid and so on.

## 19.2 Connection Diagram

Micro: bit IO pins mainly contain GPIO, ADC, DAC, IIC, SPI, PWM. And P0~P16, P19, P20 can be set to input and output. When set to output, it can output high and low levels. When set to input, it can read input level.

But only P0~P4 and P10 can analog read voltage. Below is the chart about all pins and pin functions in micro:bit.

Pin function	Pin
GPIO	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P19, P20
ADC/DAC	P0, P1, P2, P3, P4, P10
IIC	P19(SCL), P20(SDA)
SPI	P13 (SCK), P14 (MISO), P15 (MOSI)
PWM	P0, P1, P2, P3, P4, P10
Occupied	P3 (LED Col1), P4 (LED Col2), P5 (BUTTON A), P6 (LED Col9), P7 (LED Col8), P9 (LED Col7), P10 (LED Col3), P11 (Button B)

We can use P1 and P2 to connect a thermoelectric infrared sensor and servo, because these two pins can read high and low levels, read analog voltage and get values. P0 has already occupied by buzzer in Maqueen, so if you want to use P0, please switch the buzzer to OFF first.

### 1. Connect servo

Method 1:

Connect servo and Maqueen with 3 pins Dupont wires. Connect one side of the Dupont to a servo, the red wire to red wire of servo, VCC.

Connect the black wire to brown wire of servo, GND. Connect the blue wire to yellow wire of servo, digital wire. The other side, plug the red wire to 3v, black wire to GND, blue wire to P1. To fix these connectors, you can dismiss these wires and tie to metal.

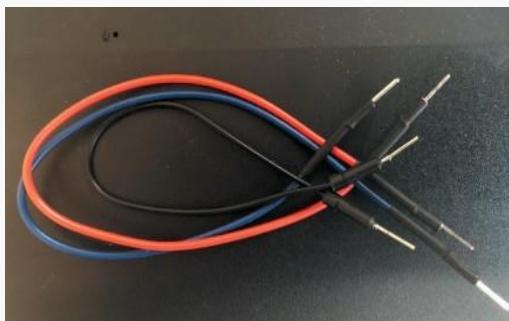


Figure 19-1



Figure 19-2

Method 2:

We also can use the port that ultrasonic sensor uses. Remove the ultrasonic sensor from Maqueen, then plug the red wire to VCC and the black wire to GND, blue wire to E(P2).

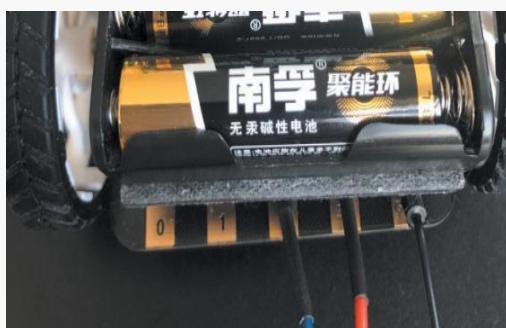


Figure 19-3

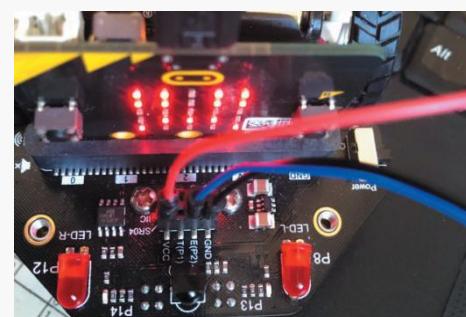


Figure 19-4

## 2. Connect thermoelectric infrared sensor

To distinguish it more easily, here we also use red wire, black wire and blue wire to connect. As the figure 19-5 shows, we connect P1 to get data.

The reason why we connect the data wire to P1 is that the thermoelectric sensor belongs to input device.



Figure 19-5



Figure 19-6

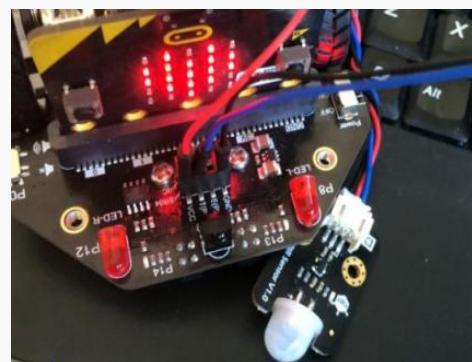


Figure 19-7

## 19.3 Programming

### Algorithm:

Put the detect transmitter and the receiver into the same wireless channel 7. Open wireless functions of both sides.

Detect transmitter:

Once the detect transmitter finds vital signs, its thermoelectric sensor will get signal and LED matrix will show image 'Heart' and send string 'signal' to the

receiver. Otherwise, LED matrix just shows image '---' which suggests no object is detected.

Receiver: show the image 'Square' when no signal is received, otherwise, show image 'Smile'. Press the button a, the receiver will send 'ok' to the detect transmitter.

When the detect transmitter receives the signal 'OK', it will enable the servo to 90° to animate nutrition liquid transportation. Then the servo returns to the original place in 1s.

### 1. Programming the detect transmitter.

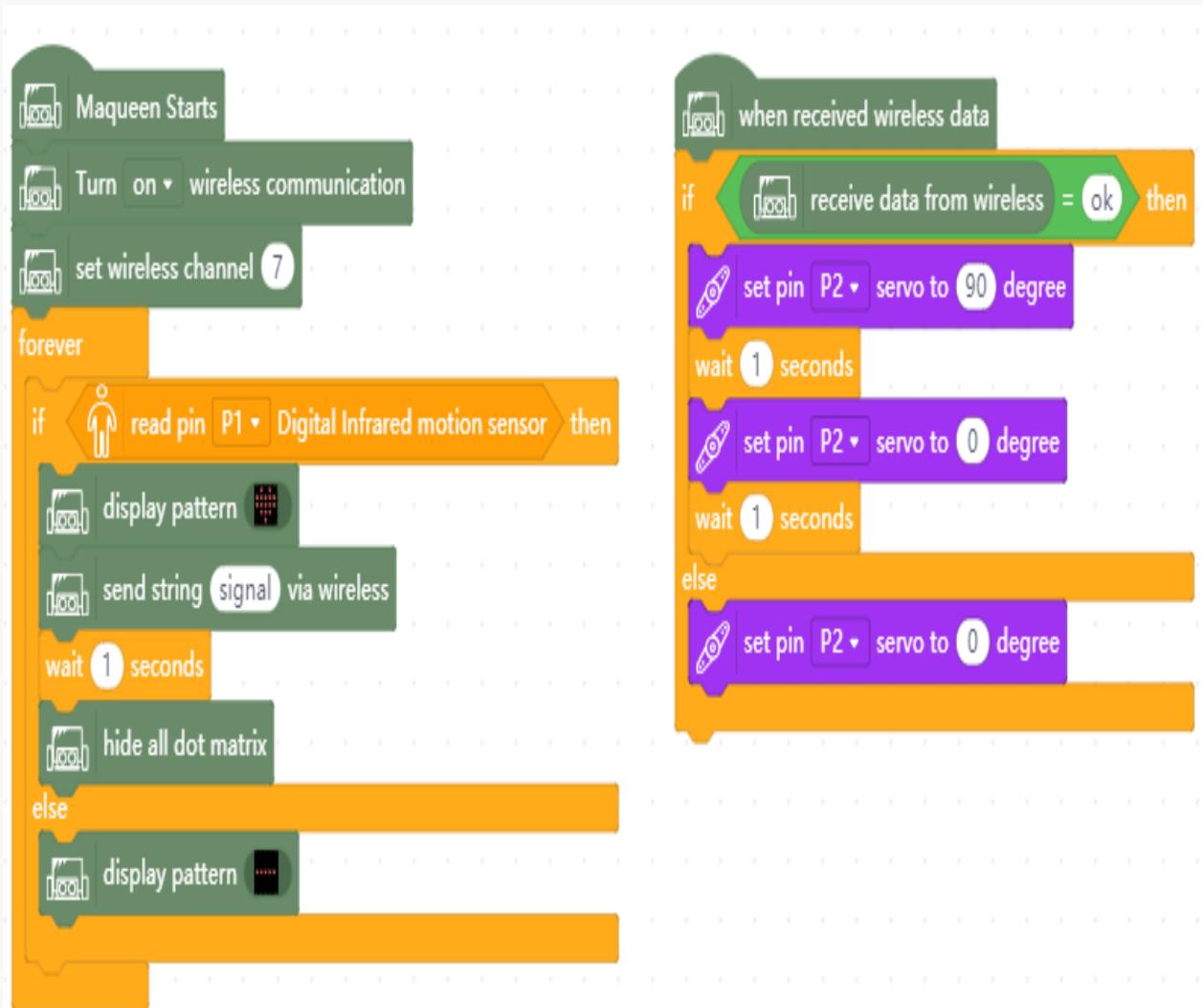


Figure 19-8

## 2. Programming the receiver

### 3. Connection Diagram

Connect the servo of the detect transmitter to the P2 port in the rear end, and the thermoelectric sensor to P1 in the front.

The receiver only has one micro:bit board, and supplied by external battery.

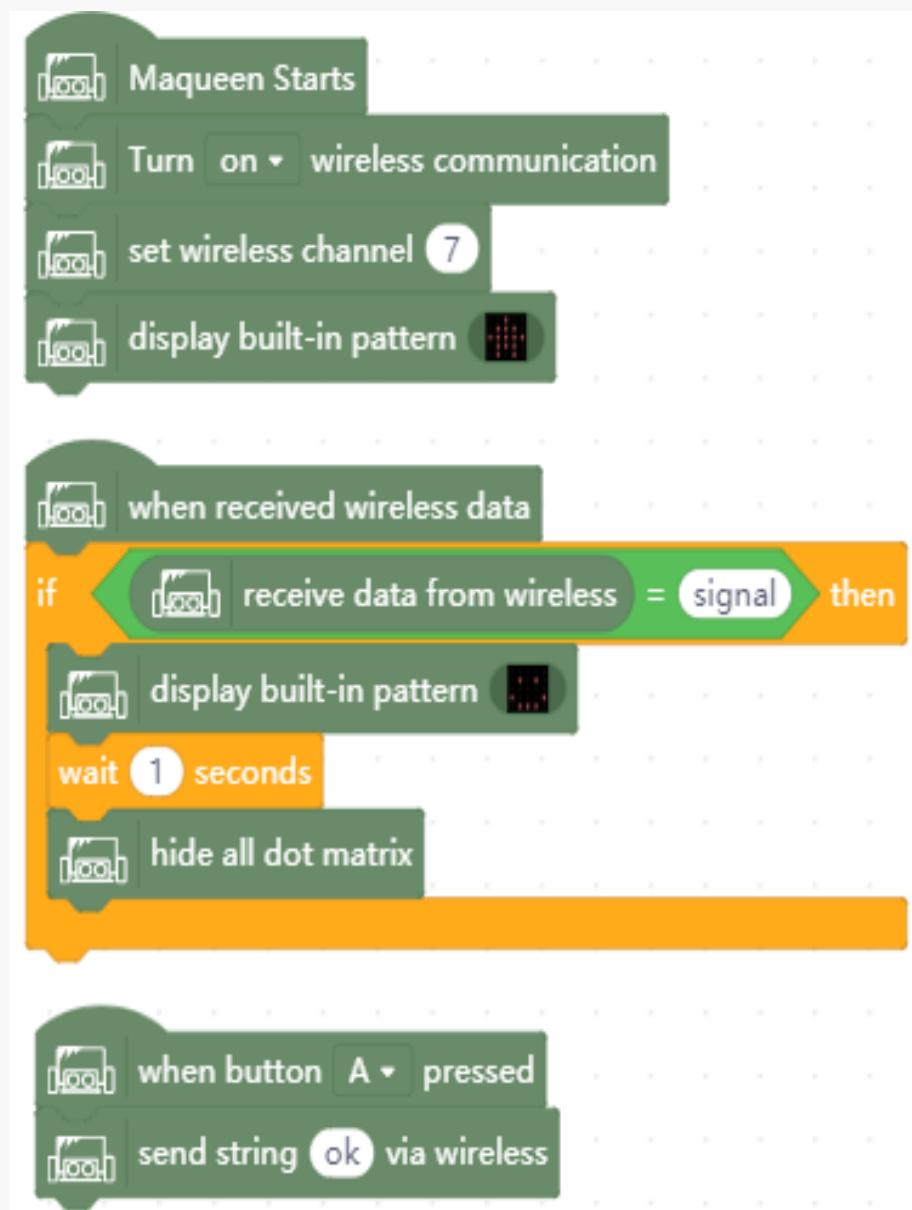


Figure 19-9

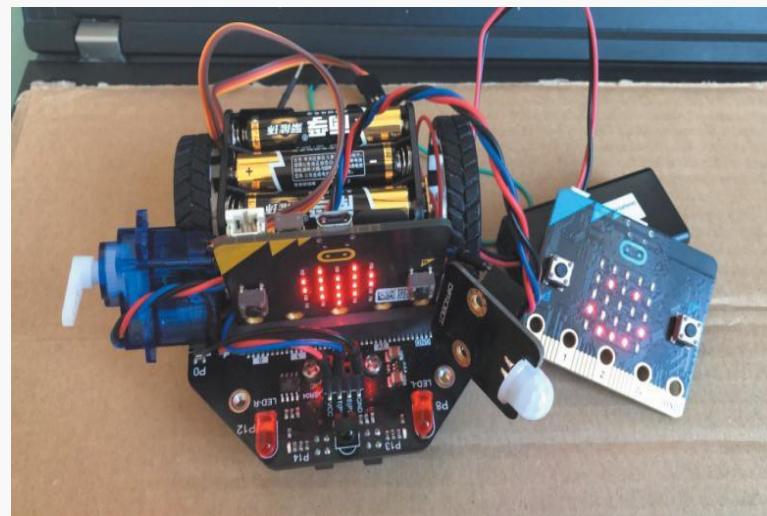


Figure 19-10

## 19.4 Training Camp

Here we only mentioned about servo and thermoelectric sensor, can you try making research and rescue Maqueen get more functions?

## Chapter 20. Maqueen and IoT

The internet of things has already stepped into our life, but what is the internet of things? Is there any practical application? To begin with, we will talk about popular Bike Share.

Bike-share make our life more convenient, just open your telephone and scan the QR Code, then you can unlock and ride the bike. The IoT principle used in Bike-share is a simple construction Telephone-Cloud-Bike. Through the APP installed in smart phone, we can check bikes around, add credit and reserve and so on. Cloud is the server controls all bike share system. It can communicate with all bikes, collect command data, respond to operations from users and managers. And the bike is the receiver to receive data and execute commands, like GPS and unlock...

The smart lock represents the practical use of Internet of Things. It is a typical IOT and Internet application. By communication technologies like smart sense, identify technology and computing, it can transmit information between users and bikes. The smart lock integrates GPS system and SIM, which can upload position and e-lock status to cloud. It has many functions, such as buzzer alarm (make sound like di-di-di before unlocking), vibration detection (alarm when the bike falls or crashed), lock status detection (end billing automatically when locked)

IOT is an important part of new information technology, Internet of things, as the name suggests, it is a internet to connect different objects. Think deeply, there are 2 meanings.

1st the core and foundation of IoT is internet, but expand and extent.

2nd the client expands and extends to objects with objects to exchange information and communicate.

Application is the biggest part of IOT development. The soul to develop IOT is the innovation that based on users experience. Nowadays, IOT has widely used in industry monitor, city manage, smart home and smart traffic and so on.

## 20.1 OBLOQ Module

If you want to connect lights or other electricity applications to Internet, then monitor and control them in remote, you can realize it by OBLOQ module. With Wi-Fi and OBLOQ module, the device could connect to Internet, send data and receive remote-control command.



Figure 20-1

Here we can regard OBLOQ module as a network card. Once the device(contains main control) plugin this network card, it can communicate with the Internet environment. You can choose Arduino main control boards and micro:bit as a main control board.

Let's check the interfaces. TX is the transmitter, RX is the receiver, GND is the cathode (-), VCC is the anode. The signal lights show different colors in different situations. RED: abnormal status; GREEN: normal status; BLUE: Wi-Fi connecting; Purple: MQTT disconnect; WHITE: Obloq firmware upgrade Please check the signal light to get the module status before any adjustment.

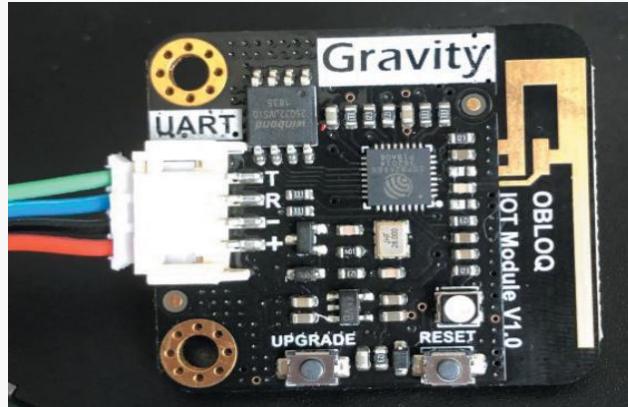


Figure 20-2

## 20.2 Functions

We want to remote control Maqueen by Internet. Maqueen moves forward when received 1 from the transmitter, move back when received 2. Of course, you can design your code and realize more actions.

### 1. Connect IOT module

Connect Maqueen to OBLOQ module to link to network. With Wi-Fi, Maqueen can realize remote communication. But how to connect OBLOQ module?

Using pin-pin DuPont wire to connect the module to the port that used by ultrasonic wave sensor. RED to VCC, BLANK to GND, GREEN to T(P1), BLUE to E(P2).

Surely, you can also connect the OBLOQ module to the mental port which is in rear end of Maqueen.

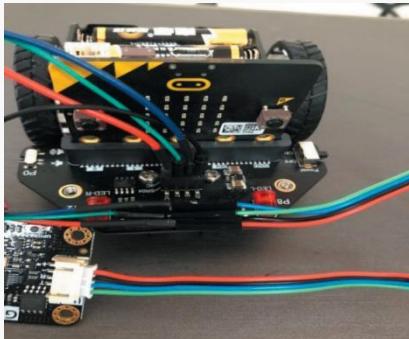


Figure 20-3



Figure 20-4

## 2. IOT Platform

Log in IOT platform to set, now we take the network IOT as an example.

Click to open <http://iot.dfrobot.com.cn> and register. The telephone number will be need for confirmation. After a successful login, enter to Workshop and you can add more device at here. Try adding a new device Maqueen, then a Topic pops up. The left column contains ID information like user name and password.

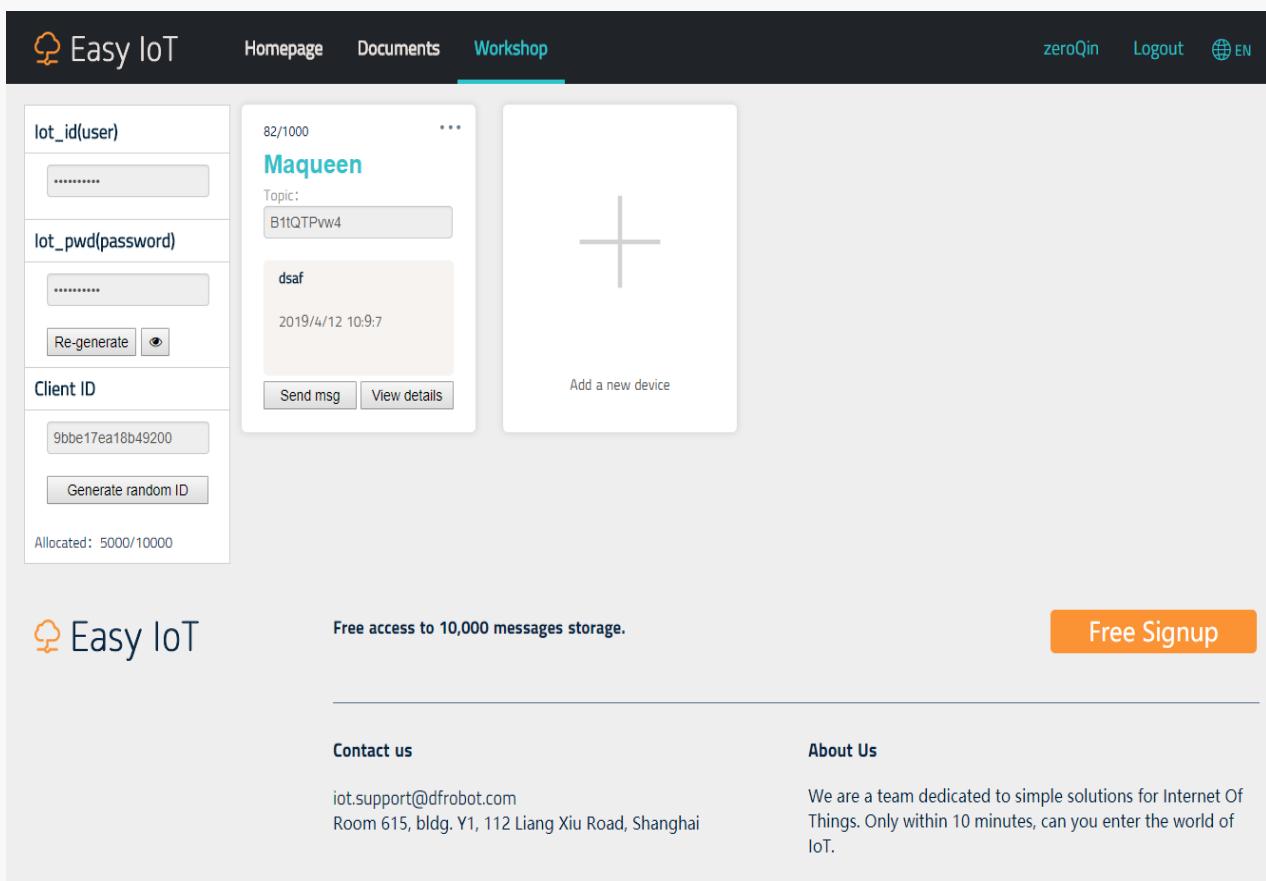


Figure 20-5

## 3. Programming

Firstly, load OBLOQ IOT module, then you can see blocks with different functions

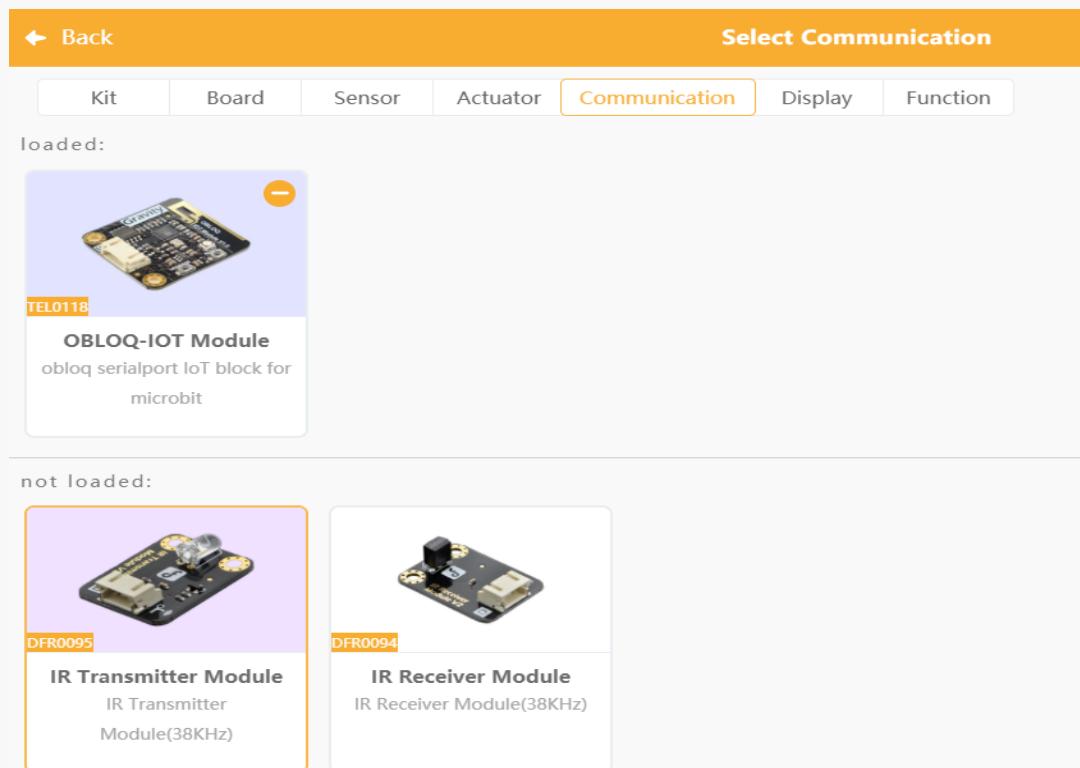


Figure 20-6

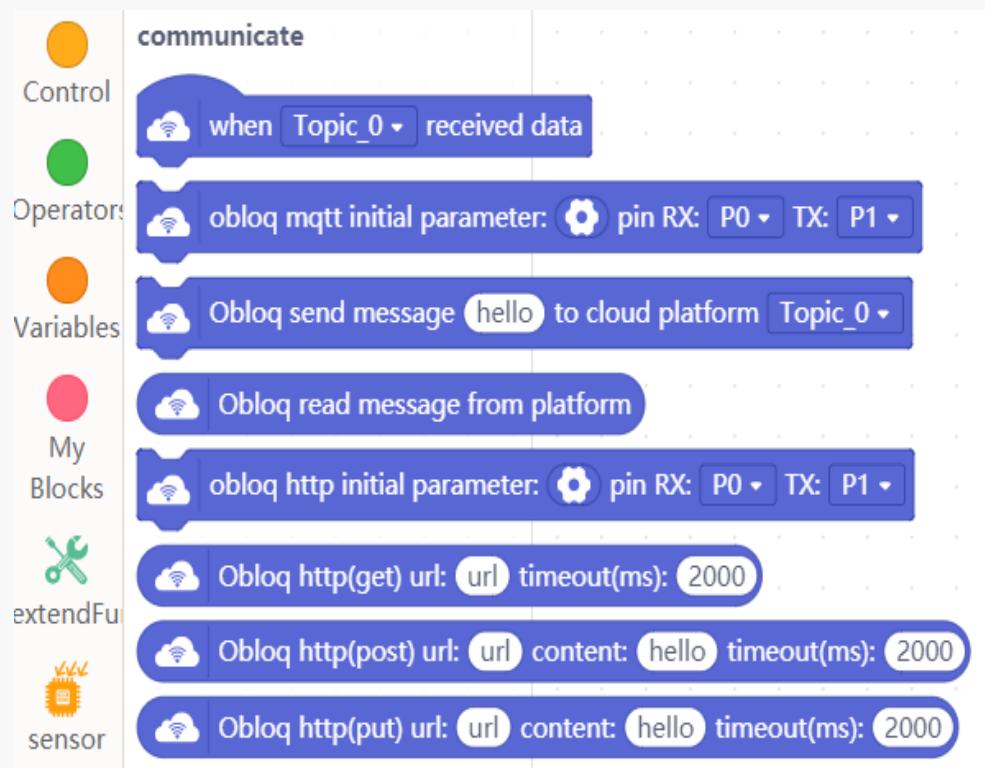


Figure 20-7

In the beginning of program starting, we need to initialize the IOT setting. To start with, we need to set the wi-fi environment. Please choose the available Wi-Fi to guarantee communication between Maqueen and receiver. Secondly, the IOT platform parameters lot\_id, lot\_pwd and topic in Mind+ must consistent with IOT platform. You can copy and past to avoid wrong input. Finally, set GREEN wire to P1 and BLUE wire to P2. Click the icon eye can show and hide password.



Figure 20-8

IOT platform can send data to Maqueen and receive data from Maqueen. But please notice the data is string, so do not forget to add “ ”, otherwise mint+ will report error.



Figure 20-9

## 20.3 Test Result

Download program to the development board in the mode coding. Switch ON Maqueen and we find the indicator light in Maqueen changes from RED to BLUE and to GREEN. The green light indicates a successful connection with network. Start the program, the OBLOQ module send message "start" to the platform. Check the platform and we will find it receives a message "start". Now Maqueen can communicate with the IOT platform.

Then, let's try send command in IOT platform and check how Maqueen responds.

### Test result:

Send 1, Maqueen make sound do and move forward for 1 second;

Send 2, Maqueen make sound do and move back for 1 second;

Send 3, Maqueen make sound do and standstill.

Let's check the information received and sent in platform. We can also realize it by telephone IOT app.

Time	Message	Operate
2019/4/25 12:6:46	2	trash
2019/4/25 12:6:43	1	trash
2019/4/25 12:6:39	3	trash
2019/4/25 12:6:34	2	trash
2019/4/25 12:6:29	Start	trash

Figure 20-10

This program uses MQTT agreement to connect IOT. And OBLOQ also can use HTTP agreement. Different from MQTT agreement, HTTP agreement does not need to use parameters iot\_id, iot\_pwd but use the methods get, post and put.

## 20.4 Training Camp

We can see IOT can help us control device. IOT applications are waiting for our exploration. Come join us to make IOT creative and powerful!

## Bibliography

1. 《小创客轻松玩转 micro:bit》化学工业出版社 2018 出版, 刁彬斌、余静、桑圆圆编著
2. DFRobot Community: [www.dfrobot.com](http://www.dfrobot.com)