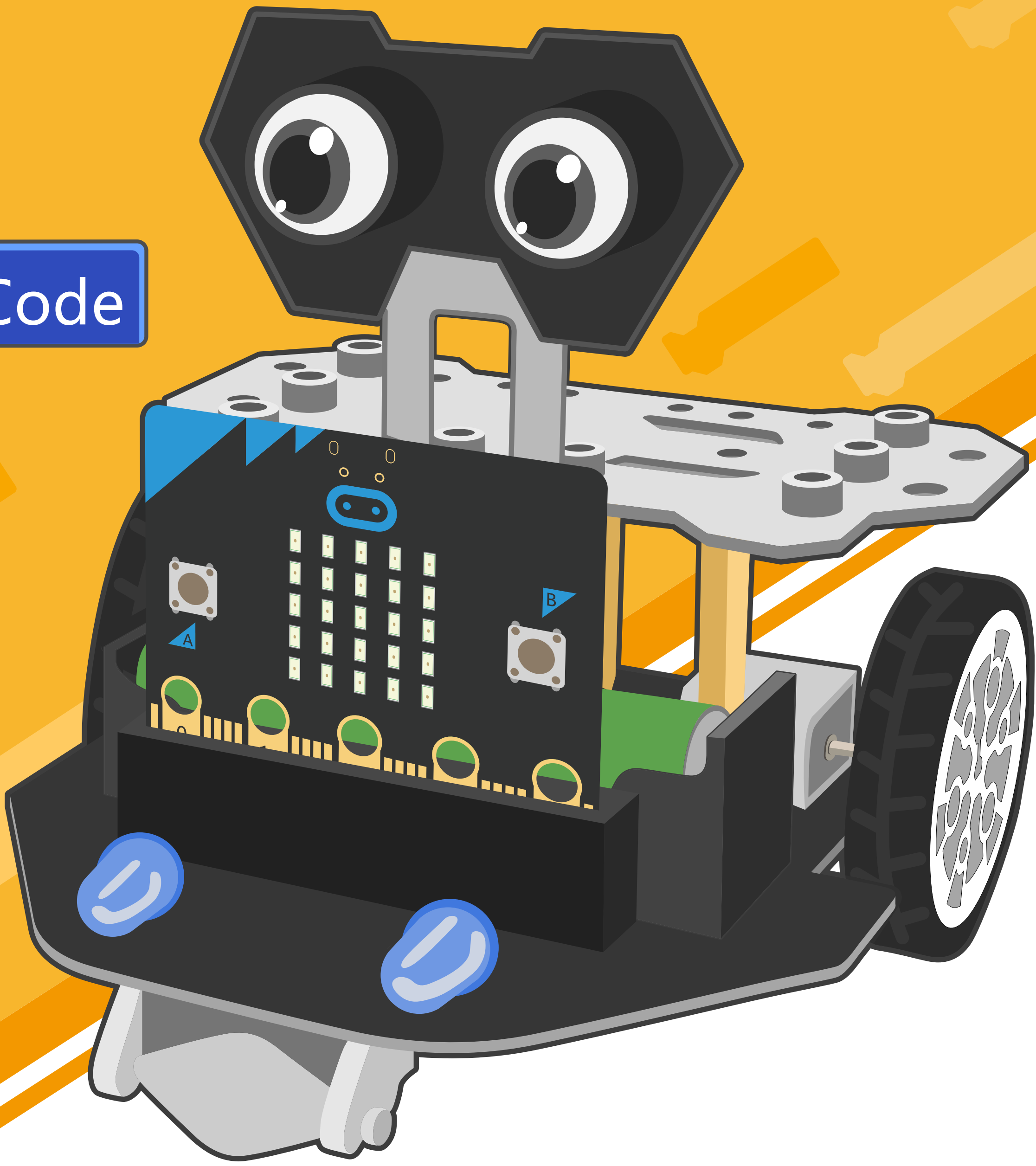
DFROBOT
DRIVE THE FUTURE

# Maqueen Plus

## Getting Started Tutorial

MakeCode



# Contents

# Chapter 1

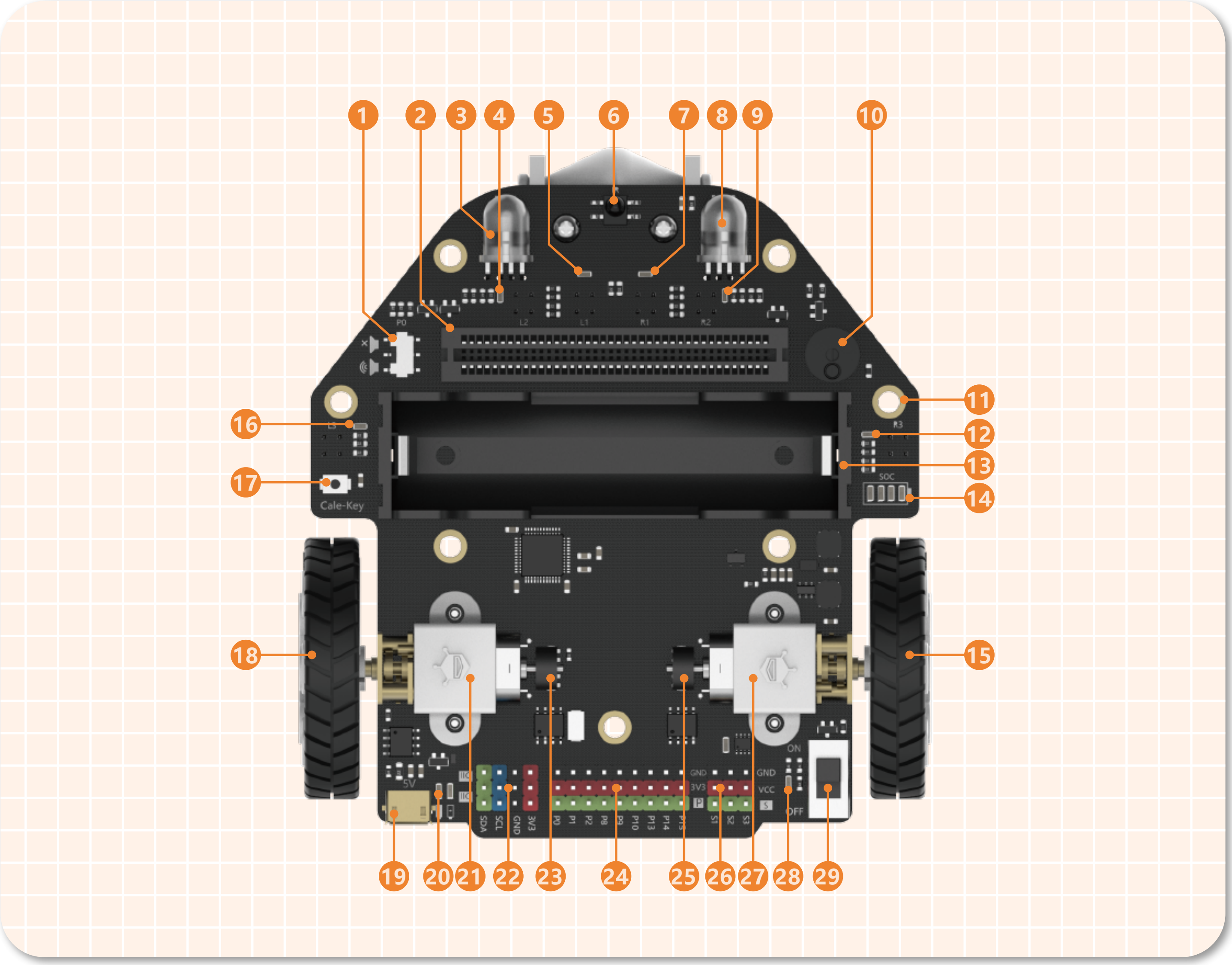
## Introduction to Maqueen Plus

# Introduction

## What is Maqueen Plus?

Maqueen Plus is a smart programmable educational robot designed for beginners. It supports Mind+ and Make-Code programming platforms, on which we can program Maqueen Plus to realize awesome functions by simply dragging and snapping the graphical blocks. Follow Maqueen Plus to enter the world of robotic, and while learn something about coding in playing!

Before we get started, let's see what Maqueen Plus has got there.



1 Buzzer switch

2 micro:bit socket

3 RGB-LED-L

4 L2 indicator LED

5 L1 indicator LED

6 Infrared receiver

7 R1 indicator LED

8 RGB-LED-R

9 R2 indicator LED

10 Buzzer

11 M3 Mounting holes

12 R3 indicator LED

13 Battery case

14 Electricity indicator LED

15 Right wheel

16 L3 indicator LED

17 Line-tracking Calibration Key

18 Left wheel

19 Charging port

20 Charging indicator LED

21 Motor-L

22 IIC expansion port

23 Encoder-L

24 GPIO port

25 Encoder-R

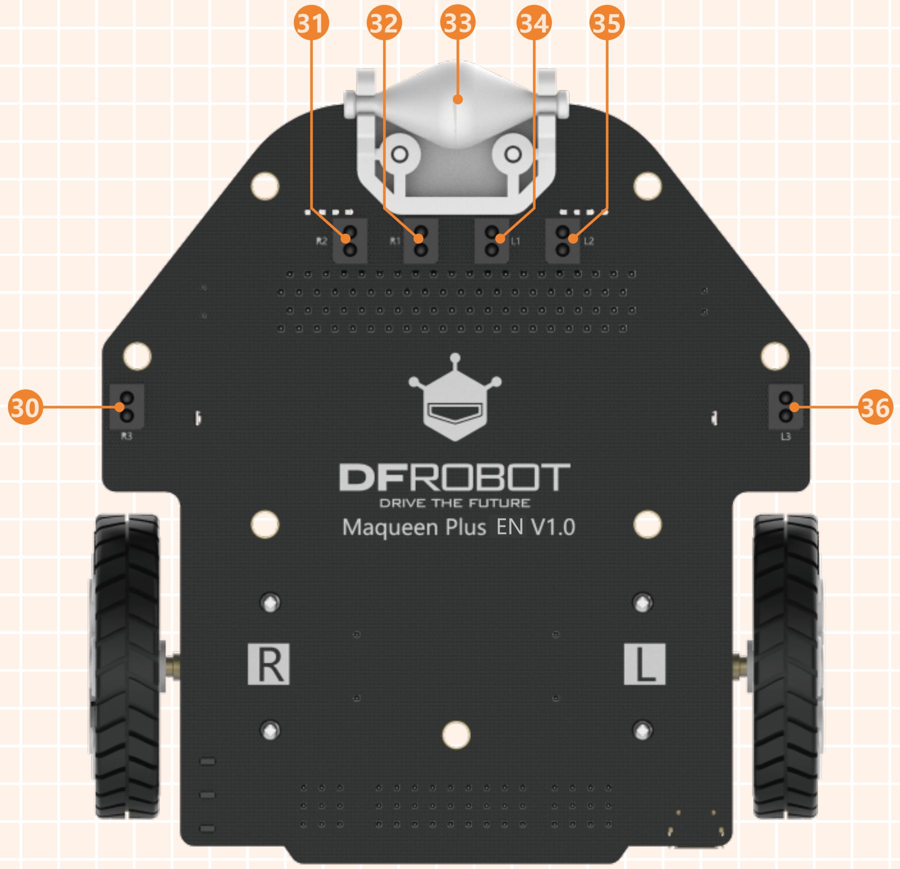
26 Servo port

27 Motor-R

28 Power supply indicator LED

29 Power supply switch

## Back view for Maqueen Plus main-board



30 R3 line-tracking sensor

31 R2 line-tracking sensor

32 R1 line-tracking sensor

33 Support wheel

34 L1 line-tracking sensor

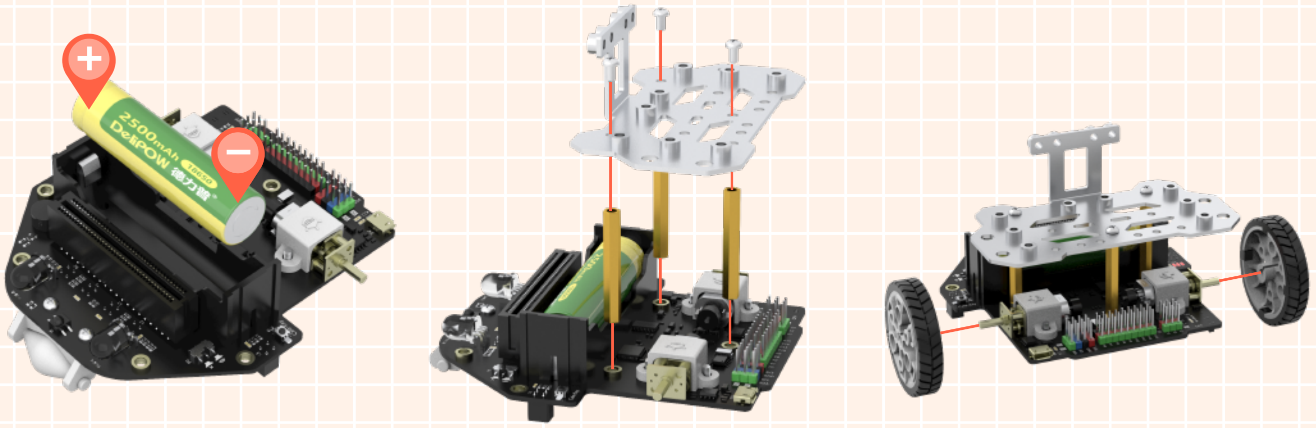
35 L2 line-tracking sensor

36 L3 line-tracking sensor

We can see that Maqueen has equipped with so many functions, and now you must can’t wait to try them. OK, here we go!

# Assembly

## Maqueen Plus Assembly Diagram

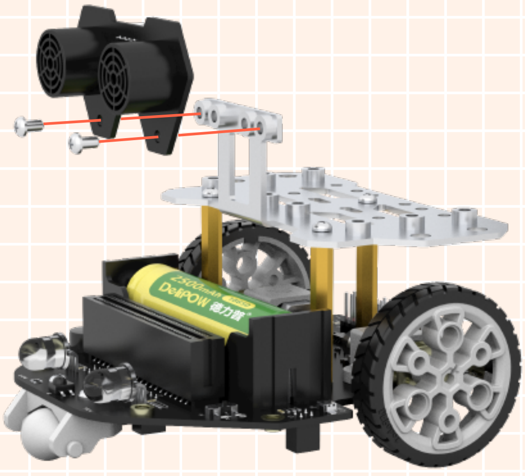


1 Install 18650 battery

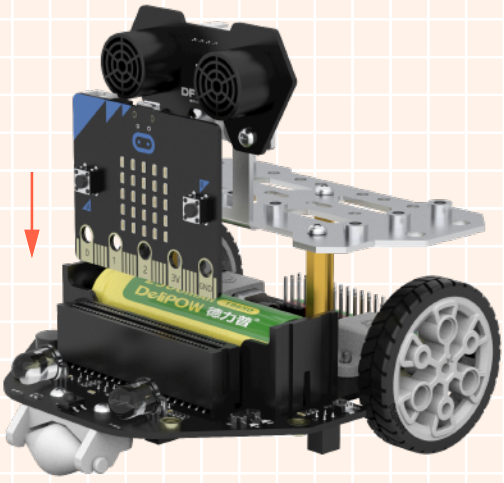
2 Install the expansion bracket

3 Install wheels

Maqueen Plus Assembly Diagram

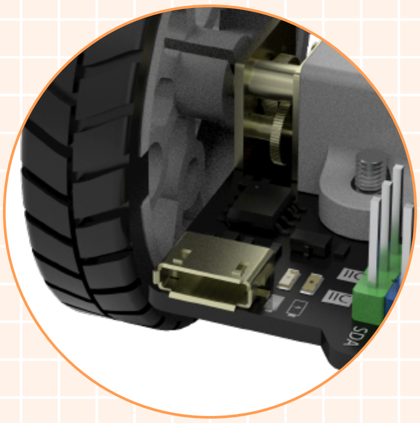


4 Install ultrasonic sensor



5 Plug in micro:bit board

Battery indicator



Charging port

Note: when the battery is fully charged, all LEDs will be on. The LEDs will be off one by one as the power gradually decreases. If all lights go out, the battery needs to be recharged.

After we assembled Maqueen Plus, put it aside because first, we need to get familiar with it's most important controller device---micro:bit. Just like the CPU in a computer, micro:bit is Maqueen's "brain" for storing and processing data, which also is the key to make Maqueen Plus "alive".

## Introduction to micro:bit

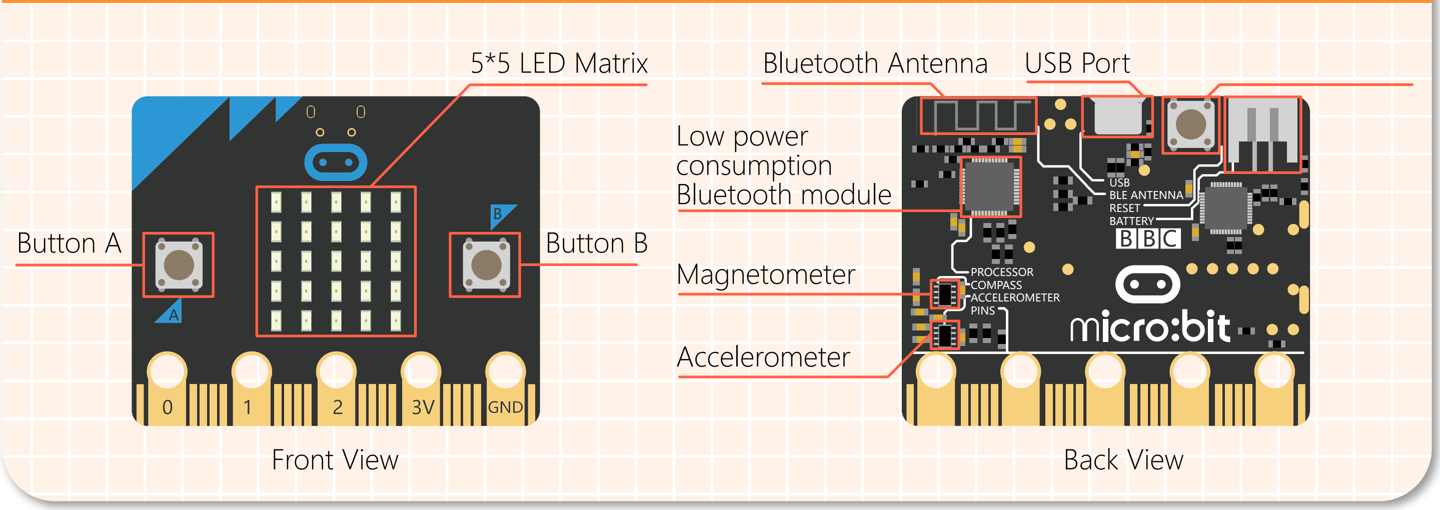
### What can micro:bit do?

The micro:bit can be programmed to do various interesting things, it can be a digital watch, fitness tracker, or a game console. The device features 25 LED lights and two programmable buttons, which can be used in game-play or to skip through tracks in a playlist. It also features an on-board compass to track the direction of the wearer...
In addition, micro:bit is equipped with commonly-used sensors like light sensor, temperature sensor, etc. It can be widely used in computer games, acousto-optic interactions, robotics, scientific experiments, wearable device and so on.

## micro:bit Function

On the credit card size board, it has 25 LED lights, two programmable buttons, light sensors, accelerometer, compass, temperature sensor, and Bluetooth module and so on.

**micro:bit Function Overview**



| | |
|---|---|
| 25 programmable LED Lights | Display patterns, words, and numbers |
| 2 programmable Buttons | Used separately or together to make things happen. For example, press down A to display a heart pattern. |
| Light Sensor | The 25 LEDs can act as sensors to measure how much light is falling on the micro:bit. |
| Accelerometer and Compass | Measure the gestures or forces in 3 dimensions. |
| Temperature Sensor | Detect the temperature in the current environment. |
| Bluetooth & Radio | Your micro:bit can communicate with other micro:bits by radio, and with other devices using Bluetooth. |

## micro:bit Programming

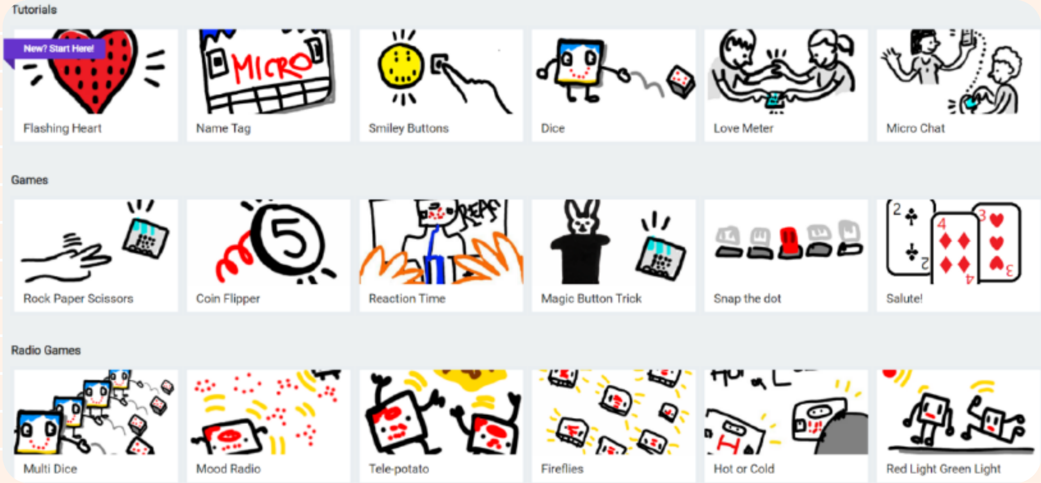
We can directly program micro:bit online without installing any software. There is a huge community of people making tools for programming and interacting with the micro:bit which means you can program your micro:bit in Python, C++ and other languages, including all kinds of block editors like Mind+, Blocks, Javascript, Python, Scratch. In this tutorial, we will use MakeCode to program, click the link https://makecode.microbit.org/ to enter MakeCode online programming environment.

# MakeCode Online Programming

## MakeCode Introduction

Microsoft MakeCode is a framework for creating interactive and engaging programming experiences for those new to the world of programming. The primary goal of MakeCode is to introduce programming in a way that is approachable and inviting. MakeCode uses the blocks programming model to let the users learn coding concepts in a more tangible fashion. Once the user becomes comfortable with the coding elements and structure, they can progress to create more complex programs.

## Projects in MakeCode

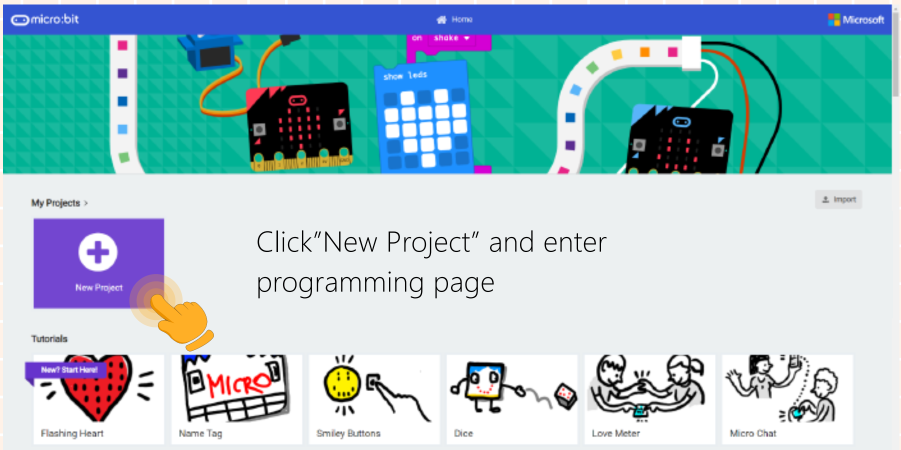


## Build Up Programming Environment

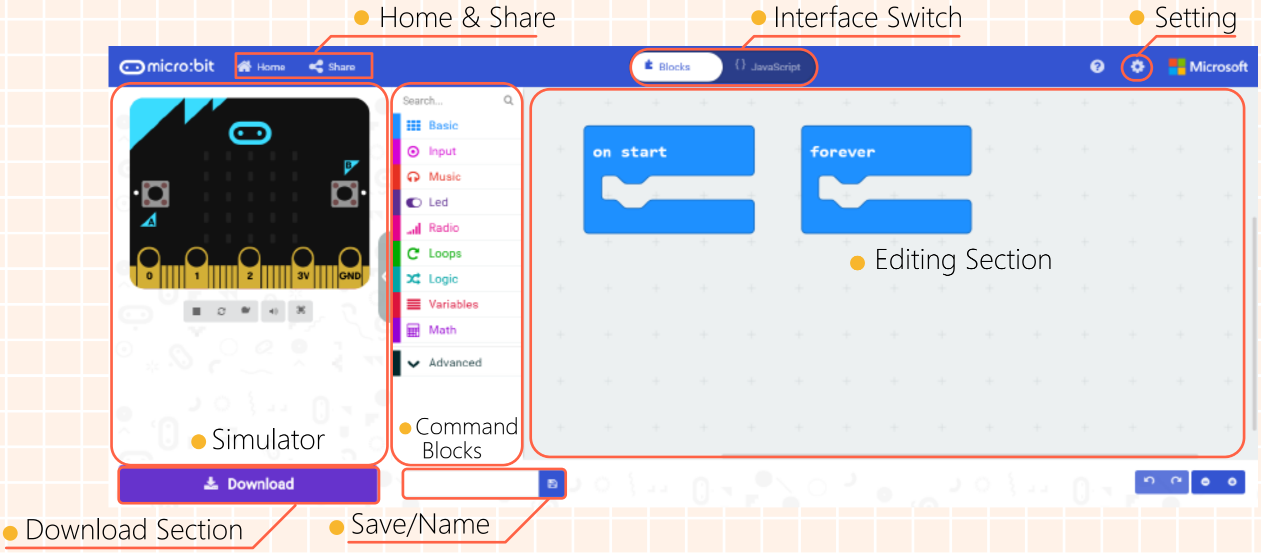
1.Input [https://makecode.microbit.org/](https://makecode.microbit.org/) to your browser to enter the MakeCode programming environment.

Note: it should be operated on the computer with a good Internet connection. If it cannot be loaded properly, please try it again using Google browser.

## Enter MakeCode



2.Open MakeCode editor and create a new project to enter MakeCode programming interface.

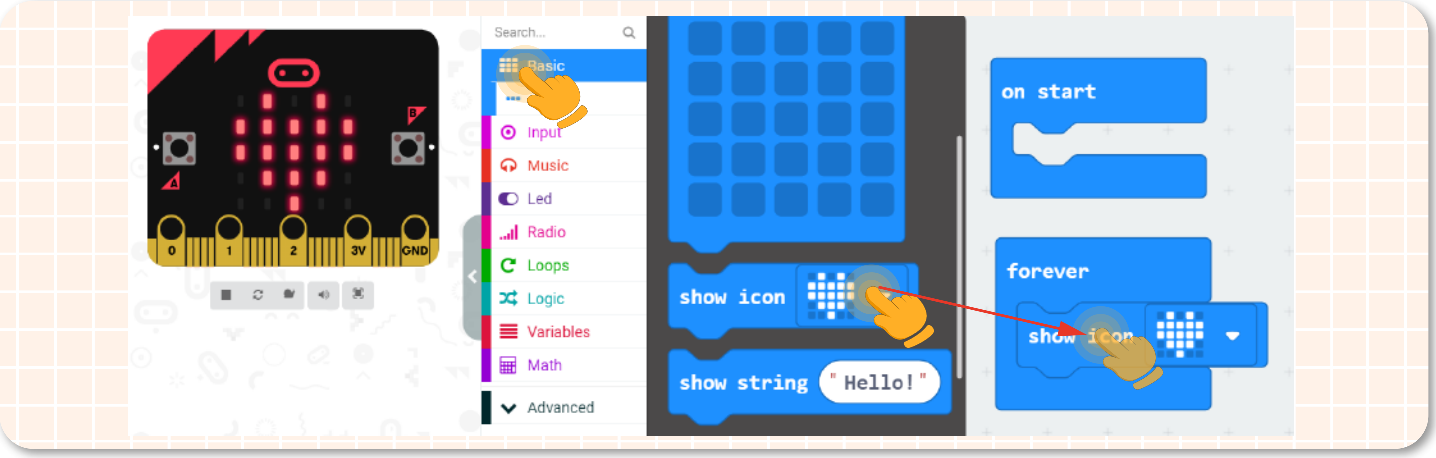


| Name | Function |
|---|---|
| Home & Share | Home: create a new project, open tutorial, game, music, fashion<br>Share: publish your project to share it or embed it in other web pages. |
| Interface Switch | Blocks: graphical programming, suitable for beginners, primary and middle school students.<br>JavaScript: code in JavaScript, suitable for high school, college students, and above. |
| Setting | Setting: project setting, extensions, language, delete project, reset, etc. |
| Simulator | Simulator: test the result of your program.<br>Preview Control: start, restart, stop the simulator, slow motion, mute audio, full-screen |
| Command Blocks | Blocks: 17 categories of programming blocks and more extended blocks. |
| Editing Section | Programming Editor: construct your program by dragging and snapping colorful blocks |
| Download Section | Download: download the program you edited into micro:bit. |
| Save/Name | Name: name your project.<br>Save: save your project.<br>Note: when you click "save", it will download the program at the same time. |

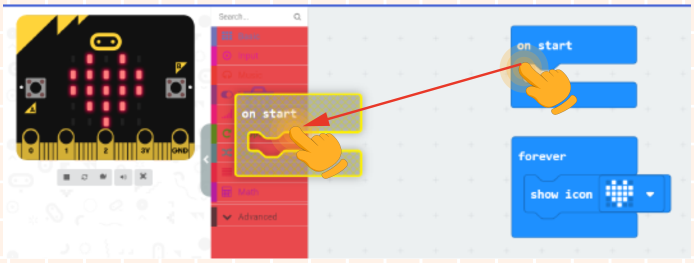
After we got a general understanding of MakeCode, let's step on our journey to code! In the first example, we will learn how to write and download a program.

## 1.Write a program

Step 1: drag the block we need to the editing section.



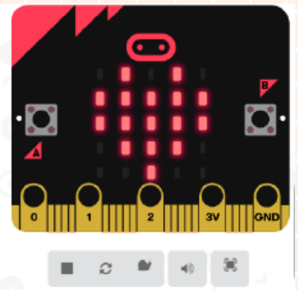
Step 2: to remove a block, drag it from the editing section to the command section, or right-click to delete.



Knowledge Expansion

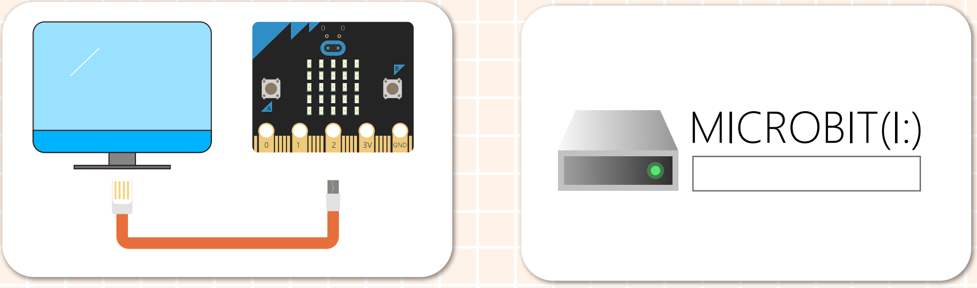
Hover your mouse pointer over the block, then the pointer will become "✋"; left-click to select the block, the pointer becomes "✊"; when you drag the block to the command section, it will become "✋" , then release the mouse to remove the block.

Step 3: after we completed a program, check its effect via the simulator.



## 2.Prepare to download

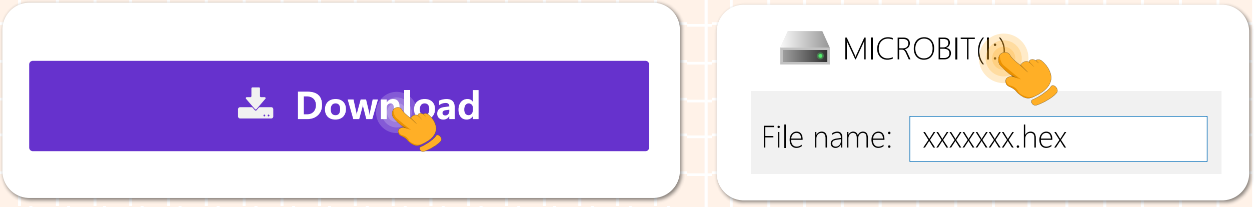
Connect the micro:bit board to your computer by a USB cable. There will be a hard-disk named micro:bit appearing in your computer when the connection is successful.



## 3.Download

Click "Download" to download the program into your micro:bit.

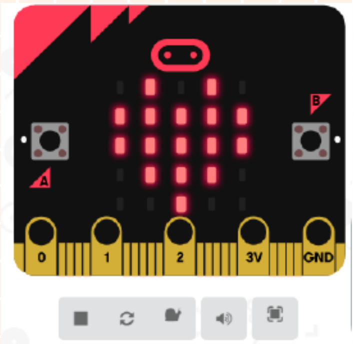
Note: when downloading a program, the micro:bit power indicator will keep flashing, and please do not disconnect the USB cable.



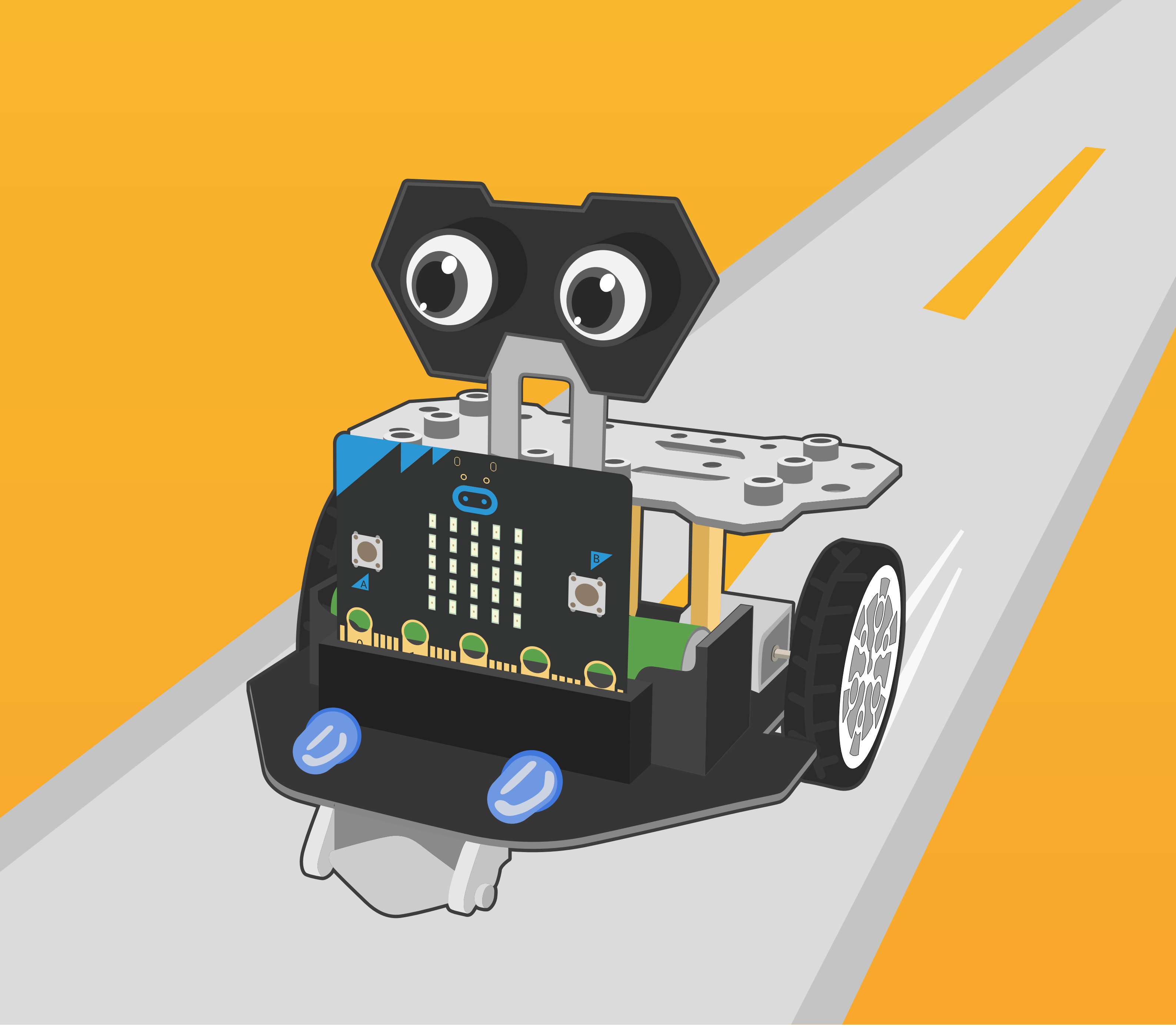
File name: xxxxxxx.hex

## 4.Download completes

Once the downloading is completed, the micro:bit LED screen will show a heart pattern.



Since we have learned the basics about Maqueen Plus, micro:bit and MakeCode programming, so for the next chapter, we are going to write a program to let micro:bit to drive Maqueen Plus.



# Chapter 2

## Let’ s move, Maqueen!

Here is our Maqueen Plus, look at this cool guy! You must be wanna play with him right now. Ok, let’s get started.

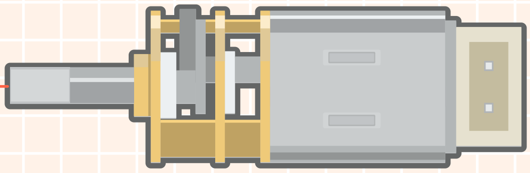
# Goal

Learn how to drive a motor.

# Electronic Component

## Motor Brief

Motor

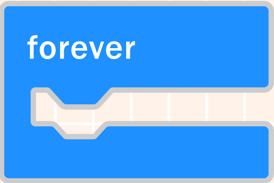


Motors can be used to drive Maqueen Plus to move left, right, backward, or go straight.

# Command Learning

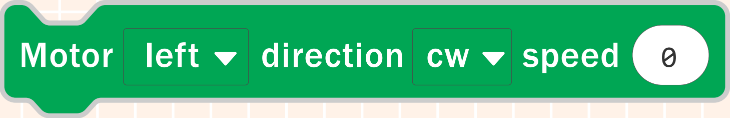
## Block Brief

Forever



Repeats the code forever in the background. On each iteration, allows other codes to run.

Motor Controlling

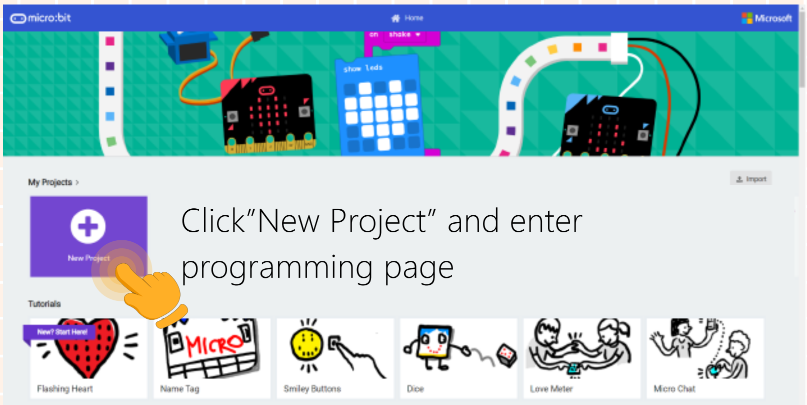


Control the speed and direction of the motor.

# Hands-on Practice

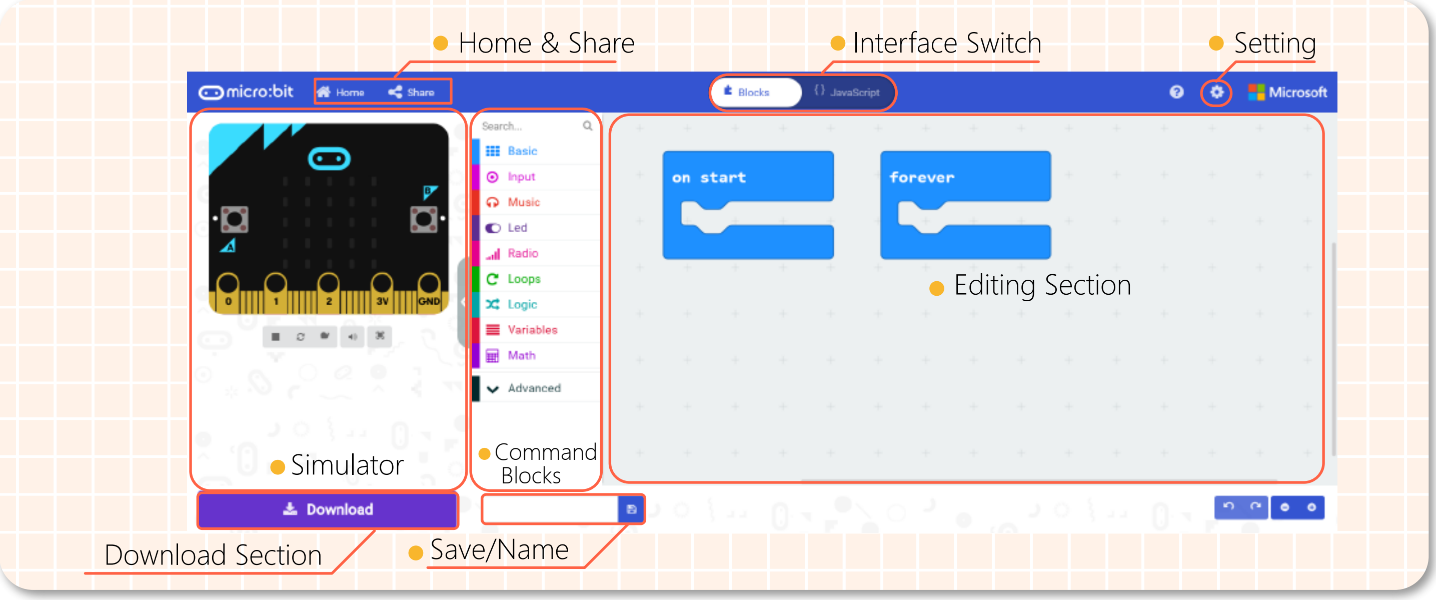
## Step 1: Create a new project

1.Input [https://makecode.microbit.org/](https://makecode.microbit.org/) into your browser to enter MakeCode editor.



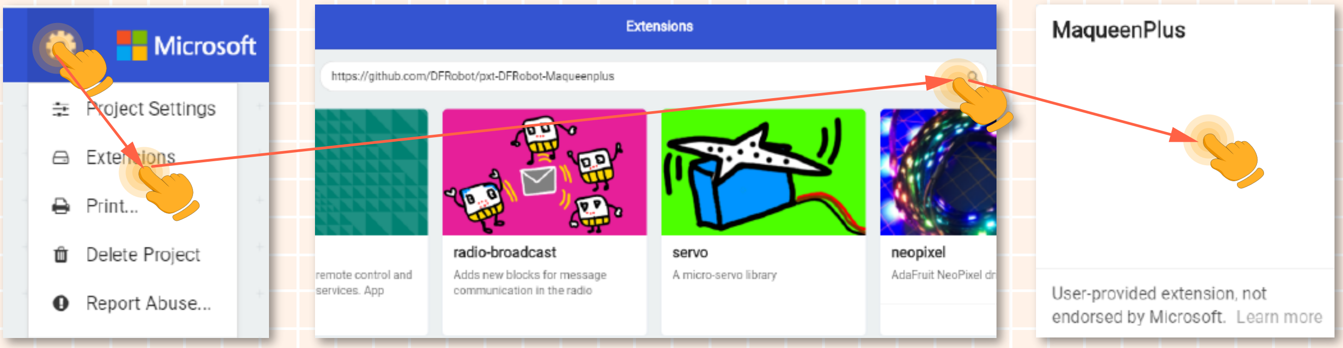
Click"New Project" and enter programming page

2.Click “new project” to enter MakeCode programming interface.



## Step 2 Add the Maqueen Plus library

1.To program Maqueen Plus, we have to find the related blocks in MakeCode  Extension.
Maqueen Plus Library: https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

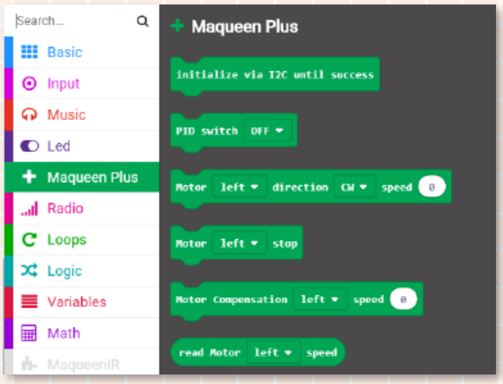

Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

1.Click Setting->Extension

2.Input https://github.com/DFRobot/pxt-DFRobot-Maqueenplus into the search bar to search

3. Click Maqueen Plus

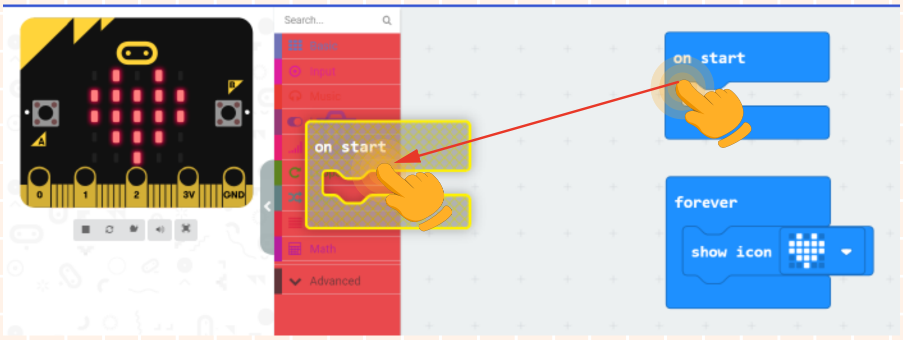
2.When the Maqueen Plus library is loaded successfully, there will be a icon”✚ Maqueen Plus” appearing in the command block section. Click the icon then you will see all the related blocks.
Refer to the attached document to check the detailed description of these blocks.



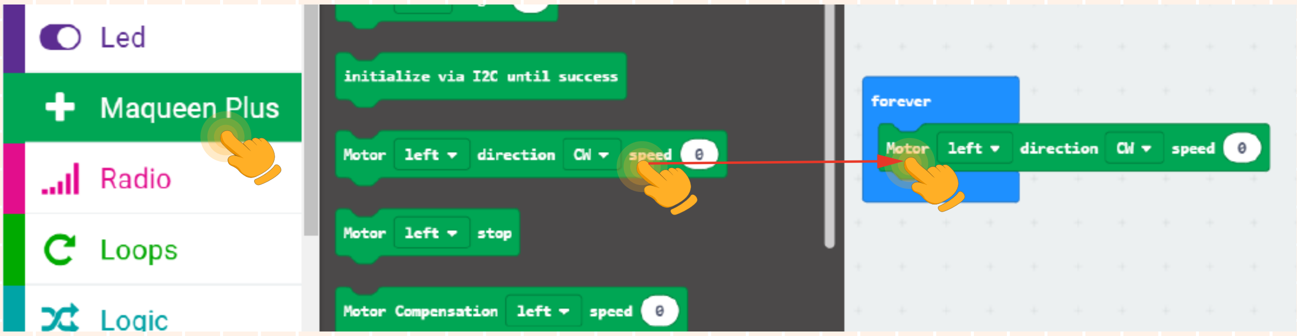
```
initialize via I2C until success
PID switch OFF ▾
Motor left ▾ direction CW ▾ speed 0
Motor left ▾ stop
Motor Compensation left ▾ speed 0
read Motor left ▾ speed
```

## Step 3 Programming

1.Delete the block we don’t need currently.



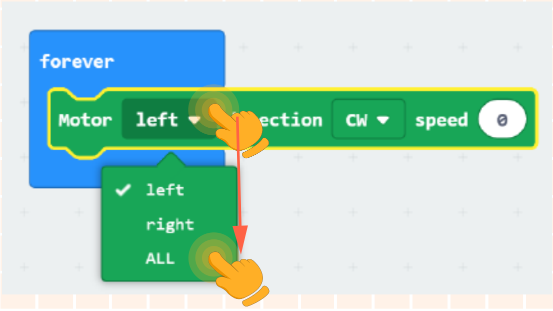
2.Embed the motor control block into the “forever” block.



```
initialize via I2C until success
Motor left ▾ direction CW ▾ speed 0
Motor left ▾ stop
Motor Compensation left ▾ speed 0
```

```
forever
    Motor left ▾ direction CW ▾ speed 0
```

3.Change the “left” to “all” in the motor block to let Maqueen’ s both wheels to move at the speed of 200.

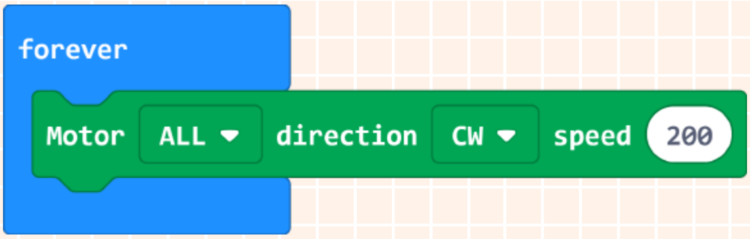


```
forever
    Motor left ▾ direction CW ▾ speed 0
```

Knowledge Expansion

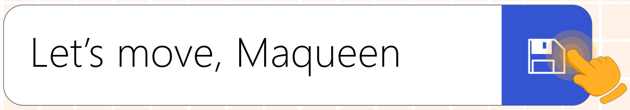
Most blocks can be used repeatedly, and the block with “ ▼ ” has multiple options for users to choose from. Besides, you can change the number in the“ 200 ” by typing or dragging the slider.

4.The complete program is shown below.



```
forever
    Motor ALL ▾ direction CW ▾ speed 200
```

5.Save and name the project as “Let’s move, Maqueen”.



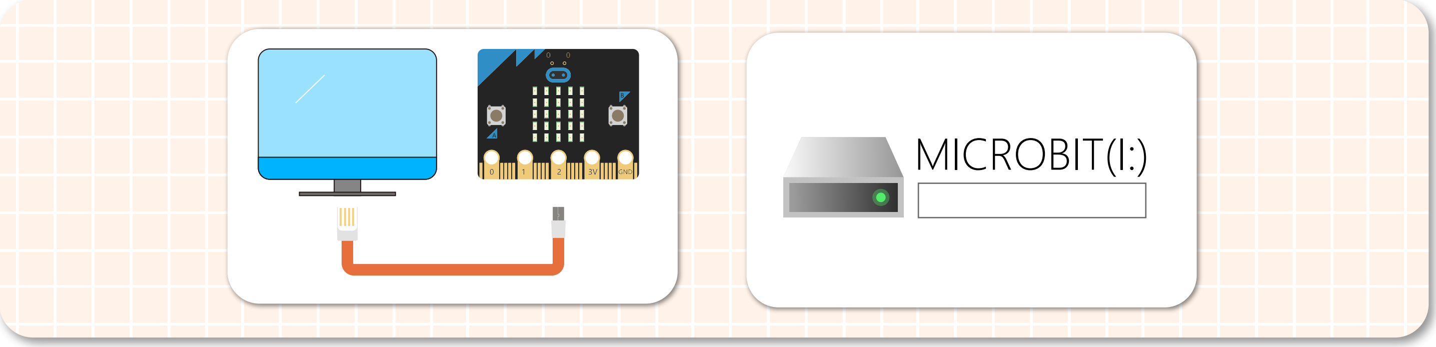
Let’s move, Maqueen

Knowledge Expansion

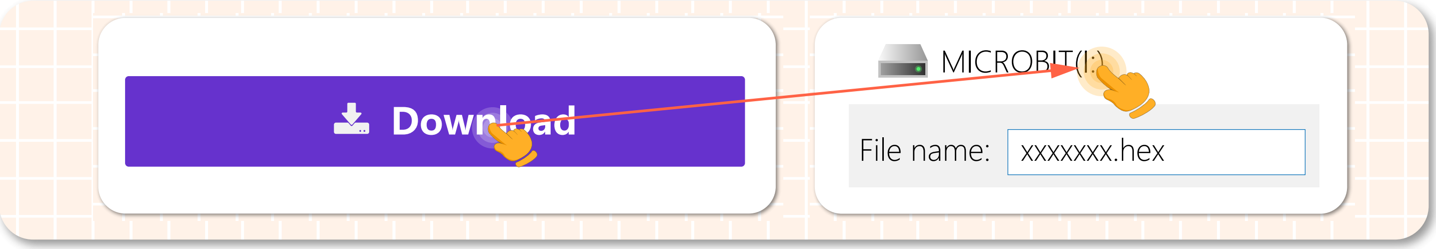
Name your project and click save.
Make it a habit to name your project!

## Step 4 Download a Program

1.Connect to a Computer: Connect the micro:bit to your computer with a USB cable before downloading. There will be a hard-disk named micro:bit appearing in the computer when the connection is successful.



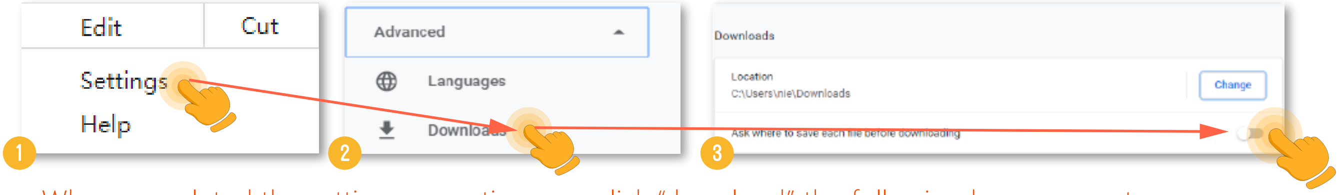
2.Download the program: Download your project into the micro:bit hard-disk.



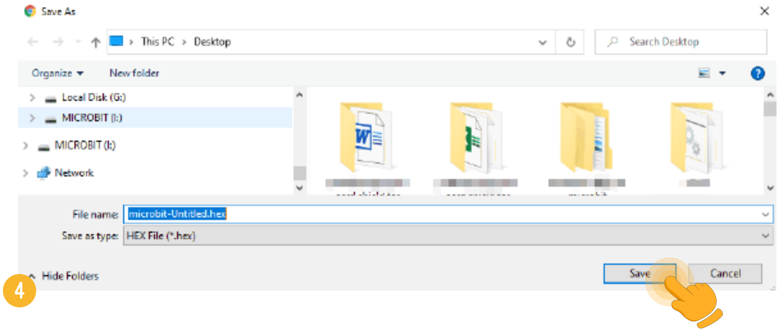
### Knowledge Expansion

Note: there will be no box popping out in Google browser since your files are directly downloaded into the default download folder. You have to change it in the setting of Google browser.

Step1: enter the setting interface in Google browser, scroll down to find the download setting part, enable"Ask where to save each file before downloading"



When completed the setting, every time you click "download" the following box pops out.

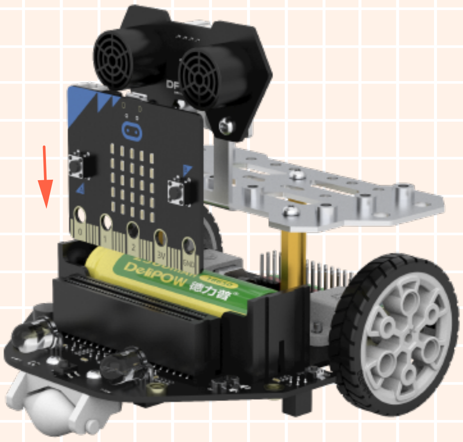


File name: microbit-Untitled.hex

Save as type: HEX File (*.hex)

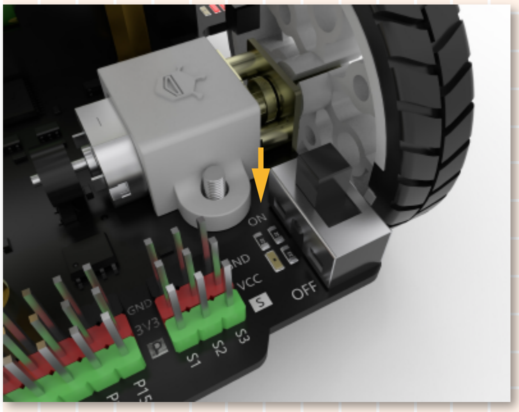
When writing data, the power indicator on the back of micro:bit board keeps flashing. Please do not disconnect the USB during downloading.

**3.Install the micro:bit Board：** After downloading the program, plug the micro:bit board into Maqueen Plus. Keep the LED matrix face forward, shown as below:



## Step 5 Effect Display

When you completed all the above steps, turn on the power switch of Maqueen Plus, then it starts running!

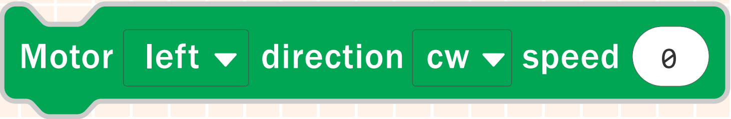


Turn on the power switch to awaken Maqueen Plus.

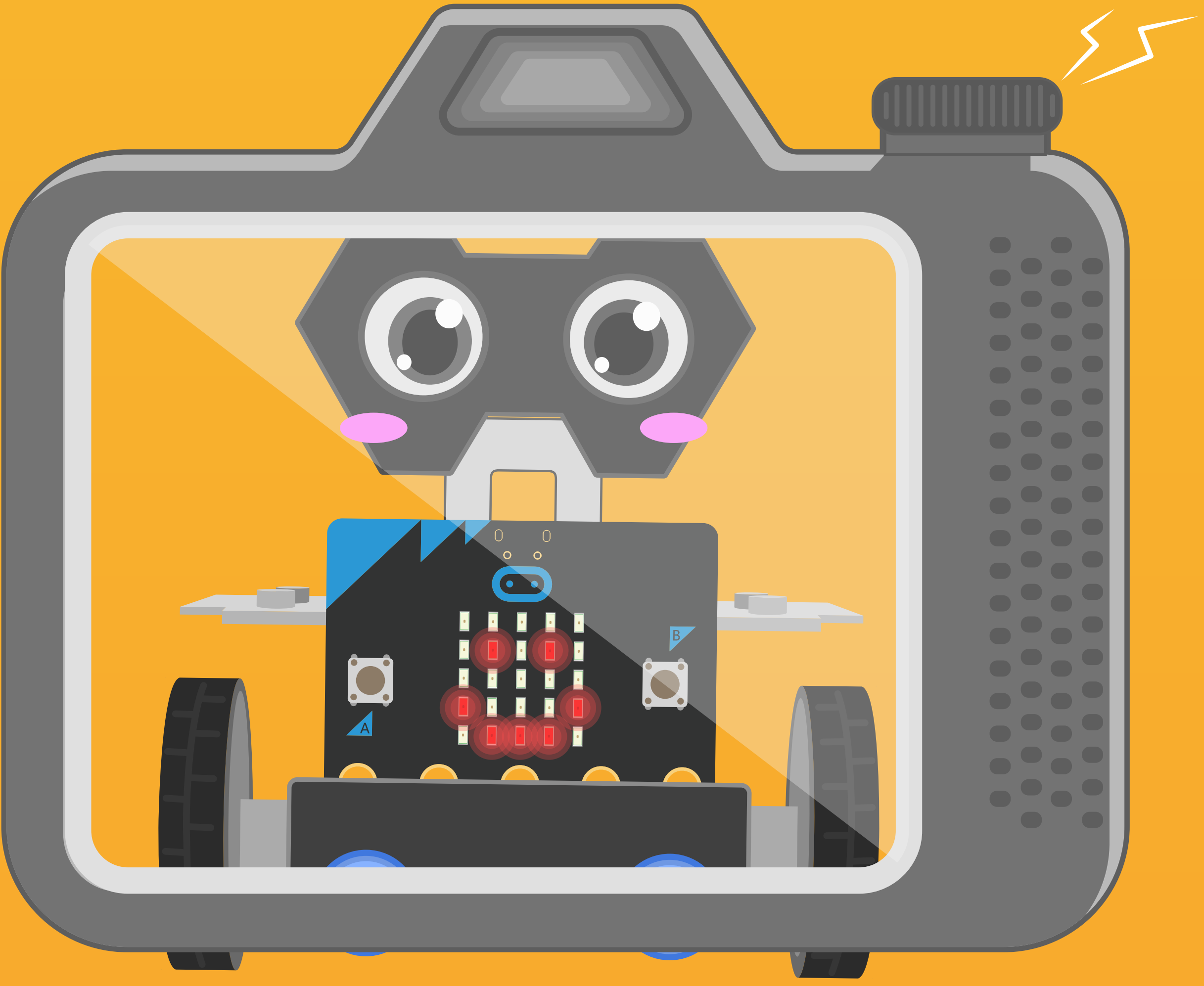
## Think & Explore

We have learned the program to make Maqueen Plus move forward, but in our daily life, a car is also able to move backward, do you know how to realize that on Maqueen Plus? Can you program Maqueen Plus to go backward at the speed of 100. Give it a try!

Tip: Just do a little bit changes in the motor control block!



Motor left ▾ direction cw ▾ speed 0



# Chapter 3

## Walking Emoji

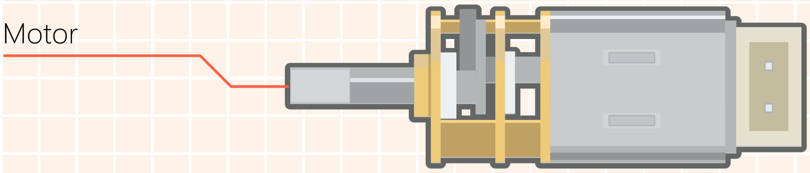
Emojis are now considered to be a large part of popular culture these days. Maqueen Plus also has his emojis. In this chapter, let’ s control Maqueen Plus to walk along a circle while displaying emojis on its LED screen.

## Goal

1.Learn the differential steering principle
2.The function of "pause"block

## Electronic Component

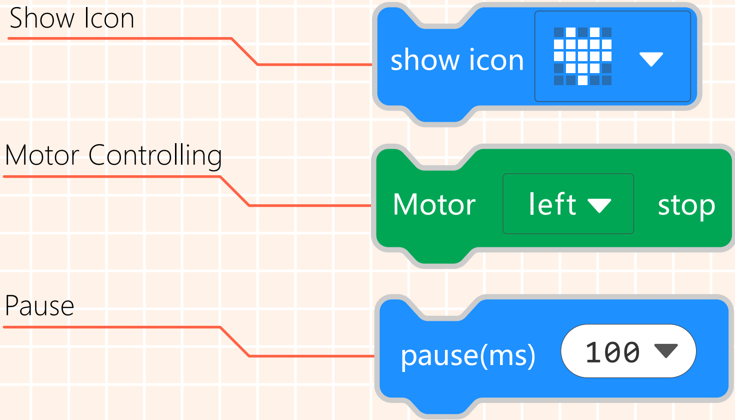
### Motor Brief



Motors can be used to drive Maqueen Plus to move left, right, backward, or go straight.

## Command Learning

### Block Brief



Draw the selected icon on the LED screen

Control the speed and direction of the motor

Pause for the specified time in millisecond

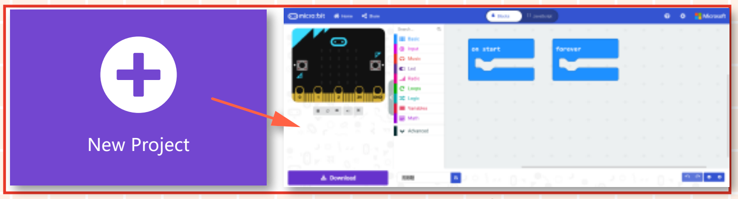
## Hands-on Practice

### Step 1 Create a New Project

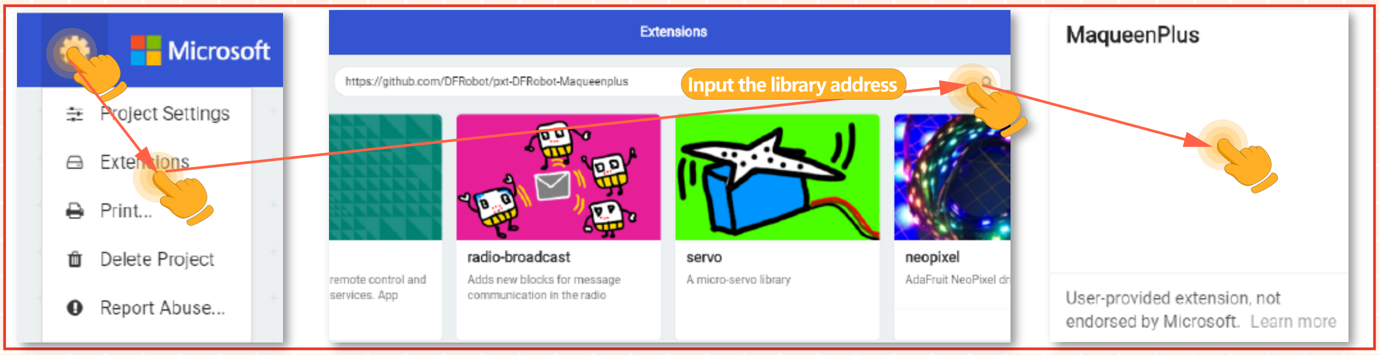
1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface



https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

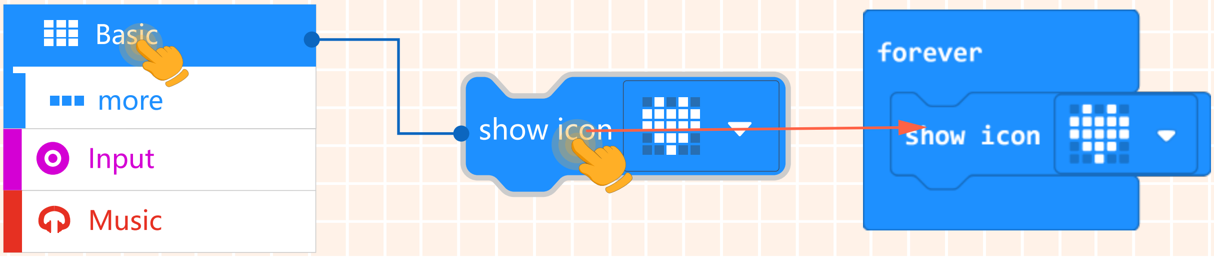
Input the library address

3.Add the extension library

## Step 2 Programming

1.Display emojis

Drag the “show icon”block into a “forever” block, then a heart pattern will be displayed on the micro:bit LED matrix.



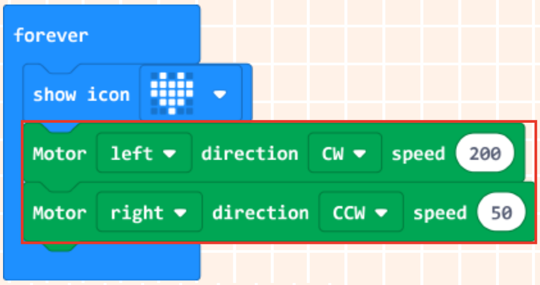
```
show icon
```

```
forever
  show icon
```

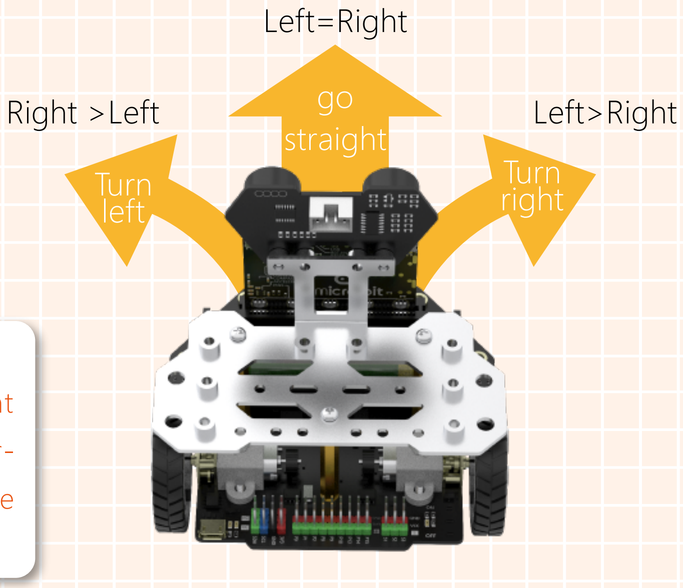
Note: click the drop-down arrow to select other patterns.

2.Maqueen Plus moves clockwise

Program Maqueen Plus to drive clockwise along a circle. According to the differential steering principle, make the left motor rotate forward at the speed of 200, and the right motor rotates forward at 50.



```
forever
  show icon
  Motor left direction CW speed 200
  Motor right direction CCW speed 50
```



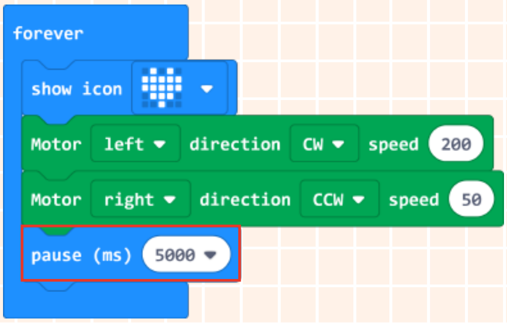
Knowledge Expansion

### Differential Steering Principle:

When the speed and direction of the left and right wheels are all the same, the robot car will move forward or backward. If the two wheels rotate in the same direction at different speeds, it will turn left or right.

3.Maqueen Plus drives along a circle

(1)Let Maqueen Plus move along a circle. Set the pause time via the pause module to make it drive a perfect circle.

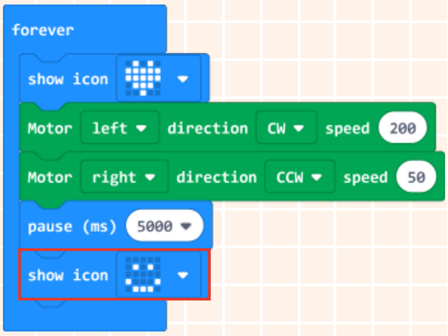


```
forever
    show icon
    Motor left direction CW speed 200
    Motor right direction CCW speed 50
    pause (ms) 5000
```

Knowledge Expansion

The length of time it takes for Maqueen Plus to drive a complete circle is related to factors such as friction on the ground, battery power and so on. So you may have to do some adjustments according to the actual situation!

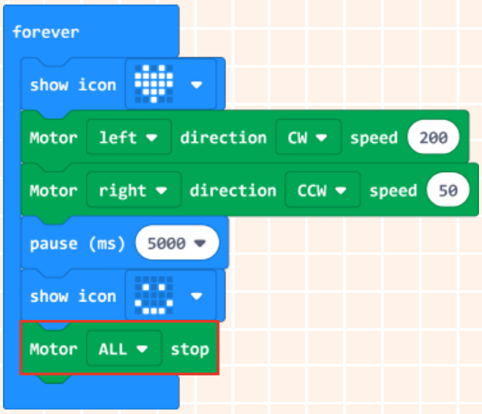
(2)When Maqueen Plus has walked a complete circle, display a smiley face on its LED screen.



```
forever
    show icon
    Motor left direction CW speed 200
    Motor right direction CCW speed 50
    pause (ms) 5000
    show icon
```

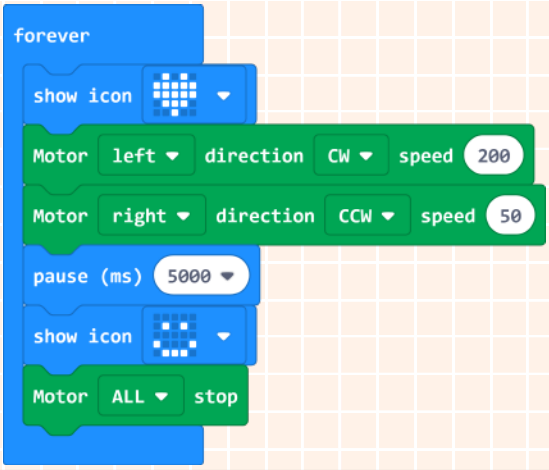
4.Maqueen Plus stops

Maqueen Plus will drive along a circle repeatedly. Now we can use another motor control block to make it stop. As shown below, change the “left” to “all”, then both motors will stop rotating.



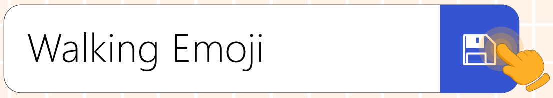
```
forever
    show icon
    Motor left direction CW speed 200
    Motor right direction CCW speed 50
    pause (ms) 5000
    show icon
    Motor ALL stop
```

5.Complete Program



```
forever
    show icon
    Motor left direction CW speed 200
    Motor right direction CCW speed 50
    pause (ms) 5000
    show icon
    Motor ALL stop
```

6.Name your project as "Walking Emoji" and save it.

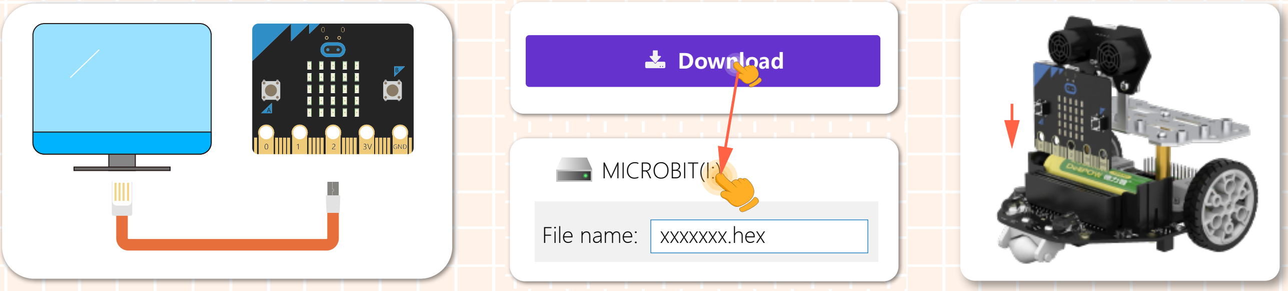

Walking Emoji

## Step 3 Download Program

1.Connect to a computer: connect the micro:bit to your computer with a USB cable before downloading. There will be a hard-disk named micro:bit appearing in the computer when the connection is successful.

2.Download the program: download your project into the micro:bit hard-disk.

3.Plug in the micro:bit board: after downloading the program, plug the micro:bit board into Maqueen Plus.



1.Connect to computer

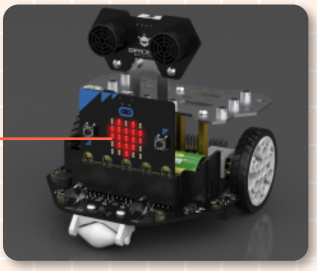
2.Download program

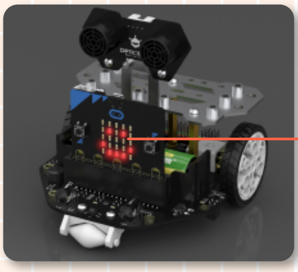
3.Plug in micro:bit

## Step 4 Effect Display

Turn on the power switch, then Maqueen Plus will start to run along a circle while showing a heart pattern on its face. When he stops, a smiley pattern will be displayed on his screen.

Walking:
heart shape





Walking completed:
smiley face

## Think & Explore

How is the movement state when the two motors are rotating at different speeds and directions? Program Maqueen Plus to explore, and complete the form below.

| | Left Motor | | Right Motor | | Movement |
|---|---|---|---|---|---|
| | Speed | Direction | Speed | Direction | |
| 1 | 200 | Forward | 200 | Forward | Forward |
| 2 | 200 | Forward | 50 | Forward | |
| 3 | 50 | Forward | 200 | Forward | |
| 4 | 200 | Backward | 200 | Backward | |
| 5 | 200 | Backward | 200 | Forward | |

# Chapter 4

## City Defender-A Police Car

There are so many city defender heroes in movies, and have you ever considered being one of them? Now let' s turn this Maqueen Plus into a city defender-a police car to make your dream come true!

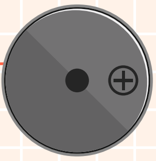
## Goal

1.Learn how to use the buzzer module
2.Learn how to use the RGB light module

## Electronic Component

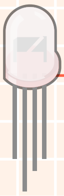
### Buzzer and RGB Brief

Buzzer



Work as an output module, can be used to play sounds.

RGB Light

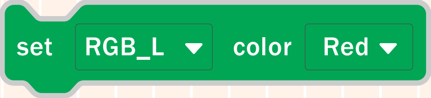


Display various colorful lighting effects

## Command Learning

### Block Brief

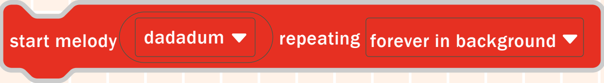
Control RGB LED



```
set RGB_L ▾ color Red ▾
```

Control the color of light

Play sound



```
start melody dadadum ▾ repeating forever in background ▾
```

Play MakeCode built-in music

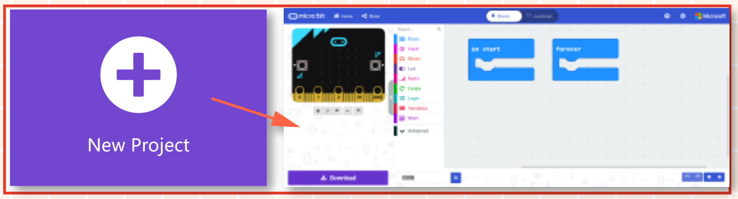
## Hands-on Practice

### Step 1 Create a New Project

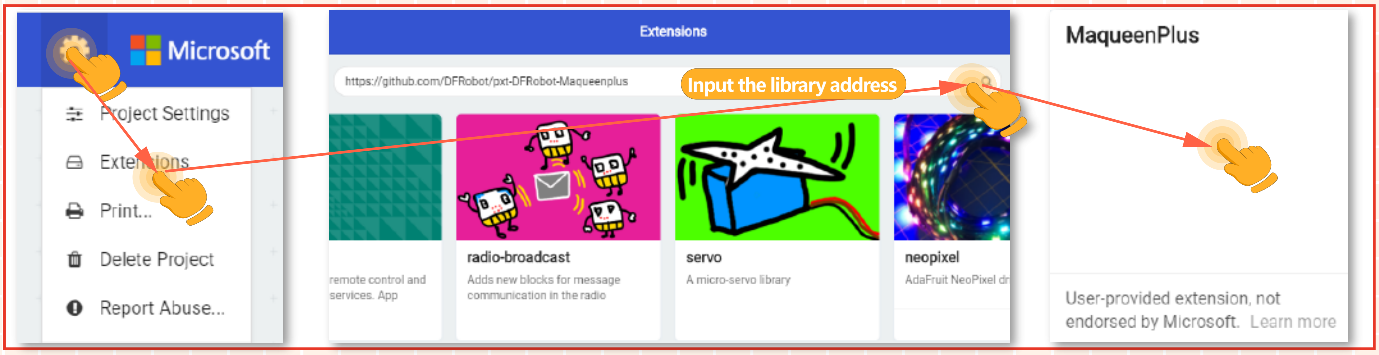
1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface

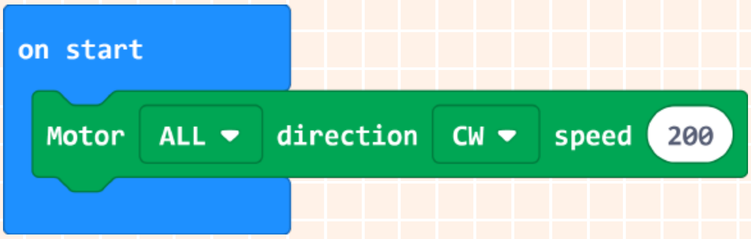

Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

3.Add the extension library

## Step 2 Programming

1.Program the car to drive

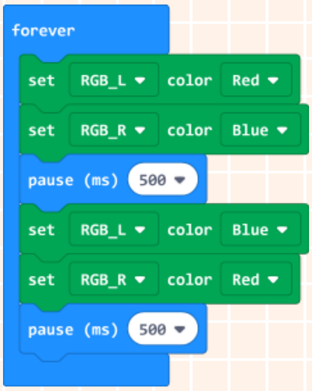
Both motors move forward at the speed of 200.



```
on start
    Motor ALL ▾ direction CW ▾ speed 200
```

2.Program the lighting effect

Since the lights on a police car emit red light and blue light alternatively, the next step here is to exchange the lighting color. App



```
forever
    set RGB_L ▾ color Red ▾
    set RGB_R ▾ color Blue ▾
    pause (ms) 500 ▾
    set RGB_L ▾ color Blue ▾
    set RGB_R ▾ color Red ▾
    pause (ms) 500 ▾
```

### Knowledge Expansion

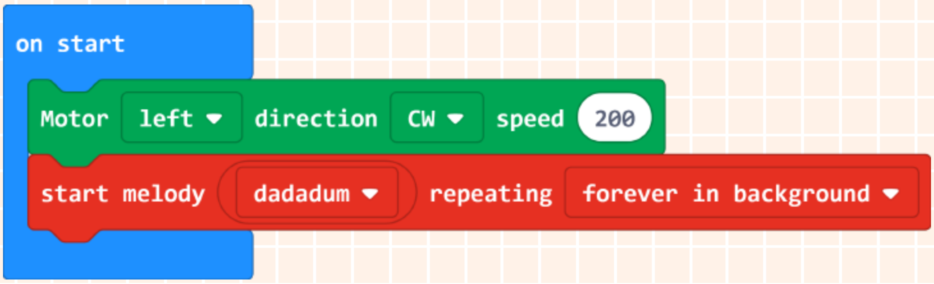
**Why are the lights on police car red and blue?**

1.Alert other drivers of its presence, so that they can maneuver out the way.

2.The color red is associated with stop and warning, but most tail lights are also red, so blue lights really stand out and help to alert others in these situations.

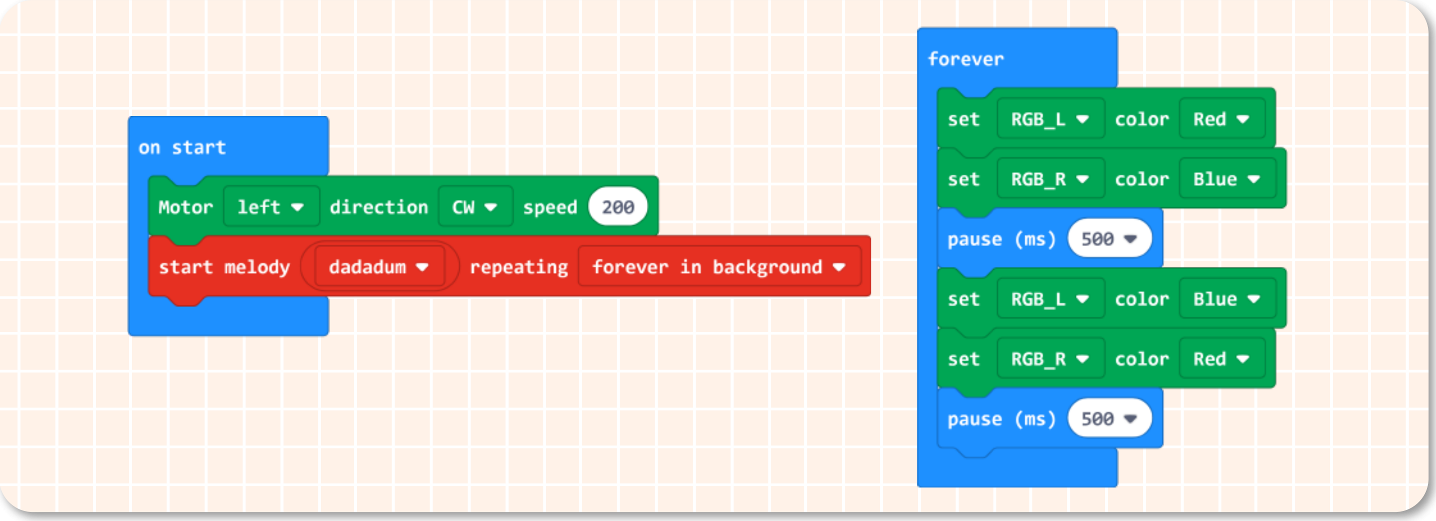
3.Program the siren

Play the built-in sound "dadadum"repeatedly to simulate the sound of siren.



```
on start
    Motor left ▾ direction CW ▾ speed 200
    start melody dadadum ▾ repeating forever in background ▾
```

4.The whole program is shown below.



```
on start
    Motor left ▾ direction CW ▾ speed 200
    start melody dadadum ▾ repeating forever in background ▾

forever
    set RGB_L ▾ color Red ▾
    set RGB_R ▾ color Blue ▾
    pause (ms) 500 ▾
    set RGB_L ▾ color Blue ▾
    set RGB_R ▾ color Red ▾
    pause (ms) 500 ▾
```

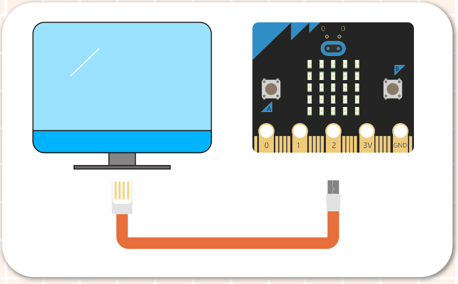
5.Name your project as "City defender-A police car" and save it.

## Step 3 Download Program

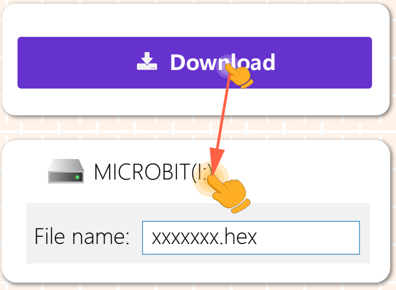
1.Connect to a computer: connect the micro:bit to your computer with a USB cable before downloading. There will be a hard-disk named micro:bit appearing in the computer when the connection is successful.

2.Download the program: download your project into the micro:bit hard-disk.

3.Plug in the micro:bit board: after downloading the program, plug the micro:bit board into Maqueen Plus.

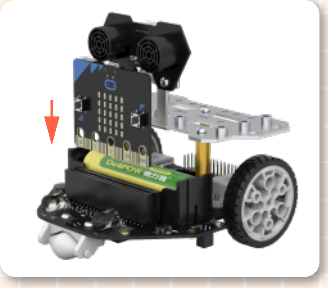


1.Connect to computer



File name: xxxxxxx.hex

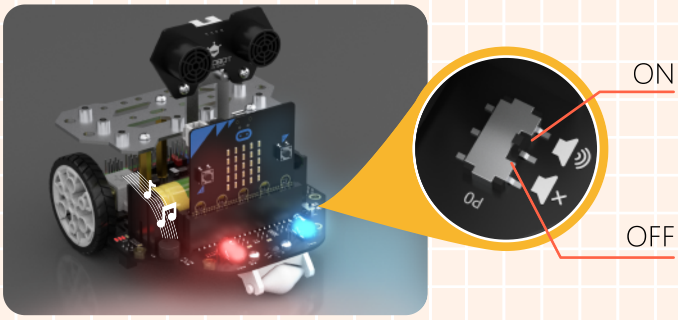
2.Download program



3.Plug in micro:bit

## Step 4 Effect Display

Turn on the power switch, then Maqueen Plus moves forward with siren wailing and lights flashing, just like a police car.



Note:

1.There is a buzzer switch on the left side of Maqueen Plus, and you need to turn it on when using the buzzer.

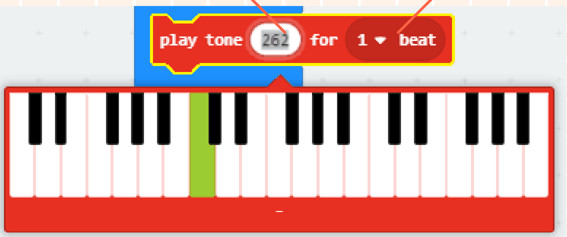
2.The buzzer and P0 are shared, so when you need to use P0 as input/output port, turn off the buzzer.

## Think & Explore

We have learned how to play the built-in music in MakeCode. Do you want to make a piece of your music? Try it with Maqueen Plus. The block shown below is used to play notes.

Click here to display keyboard, then select note.

Select beat for the note.

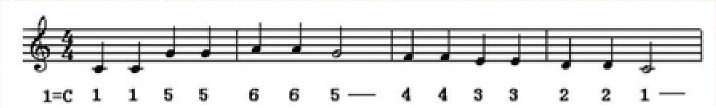


```
play tone 262 for 1 beat
```

The letter "C" represents the note "do", so middle C is middle do. "1" is the beat of the note, the basic time unit, also specified by the time signature. The relation between note and notation is shown below:

| C | D | E | F | G | A | B |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Create a piece of music according to the numbered musical notation below!



# Chapter 5

## Light Sensing Robot

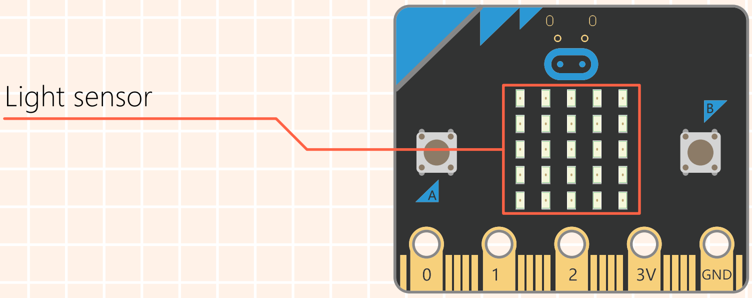
Without light, there would be no sight. We are able to see because light from an object can move through space and reach our eyes. But human eyes are very sensitive to light, both too strong and weak lights are harmful to our eyes. How do we know the changes in light brightness? Maqueen Plus can help us achieve that.

## Goal

1.Learn the light sensor
2.Learn to use “show number”block

## Electronic Component

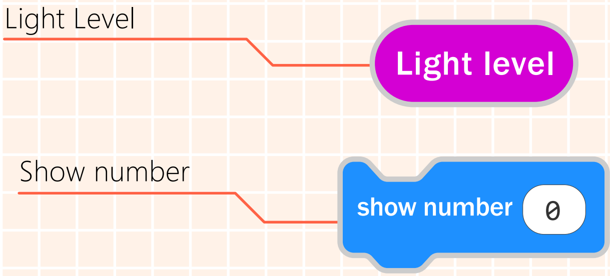
### Light sensor Brief



The micro:bit LEDs can be used to estimate the amount of ambient light, and output the light level as electric signal.

## Command Learning

### Block Brief



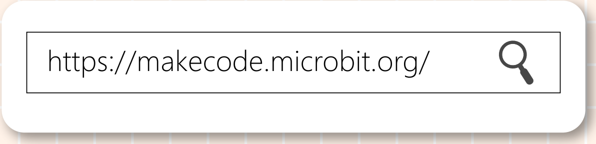
Reads the light level applied to the LED screen in a range from 0 to 255.

Scroll a number the micro:bit LED matrix.

## Hands-on Practice

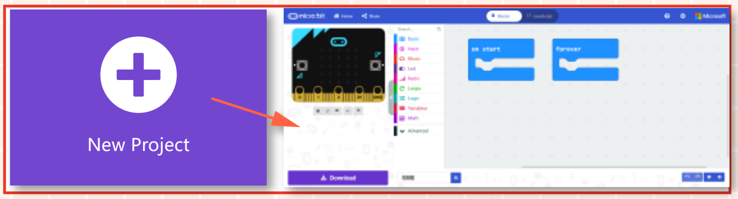
### Step 1 Create a New Project

1.Input [https://makecode.microbit.org/](https://makecode.microbit.org/) into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:[https://github.com/DFRobot/pxt-DFRobot-Maqueenplus](https://github.com/DFRobot/pxt-DFRobot-Maqueenplus)



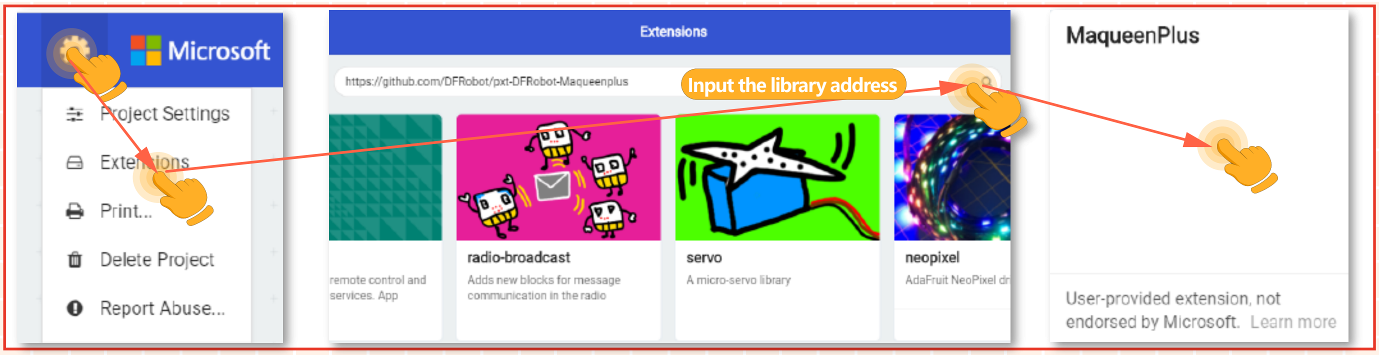
https://makecode.microbit.org/

1.Enter MakeCode editor



New Project

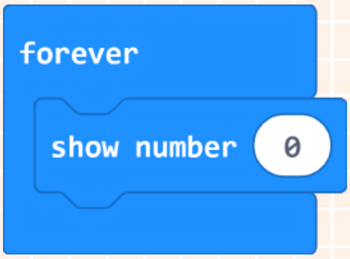
2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

3.Add the extension library

## Step 2 Programming

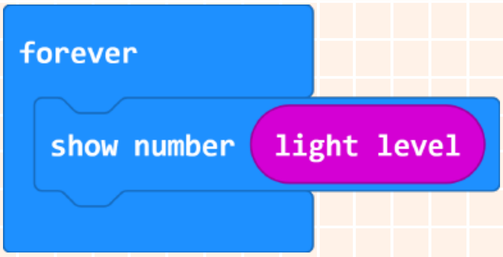
1.Embed the "show number" block into the "forever" block, then the robot will read the light level constantly.



```
forever
    show number 0
```

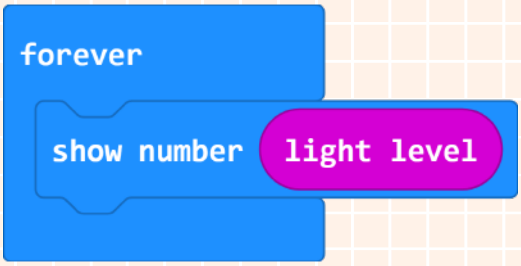
Note: the "show number" block displays 0 by default.

2.To display the ambient light level on the micro:bit LED screen in real time, we have to put the "light level"block into the "show number" block.



```
forever
    show number light level
```

3.The complete program is shown below.



```
forever
    show number light level
```

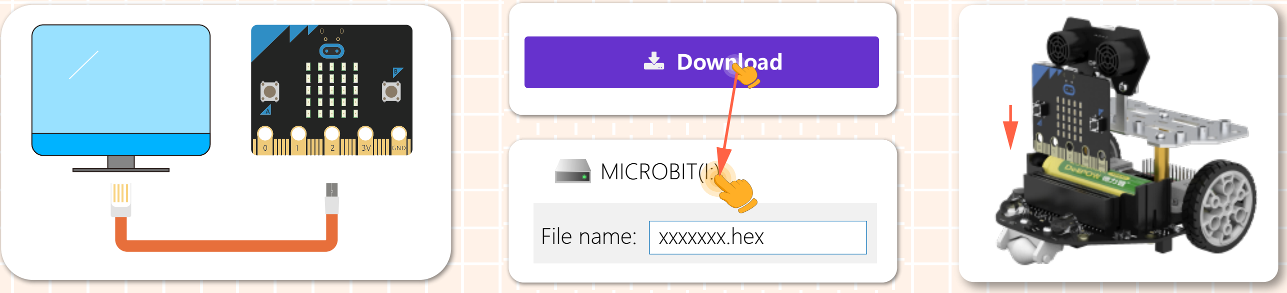
4.Name your project as "Light sensing robot" and save it.

## Step 3 Download Program

1.Connect to a computer: connect the micro:bit to your computer with a USB cable before downloading. There will be a hard-disk named micro:bit appearing in the computer when the connection is successful.

2.Download the program: download your project into the micro:bit hard-disk.

3.Plug in the micro:bit board: after downloading the program, plug the micro:bit board into Maqueen Plus.



1.Connect to computer 2.Download program 3.Plug in micro:bit

## Step 4 Effect Display

Turn on Maqueen Plus' s power, the LED screen will constantly display the current light level! The following are the light levels measured at the office and photo studio.

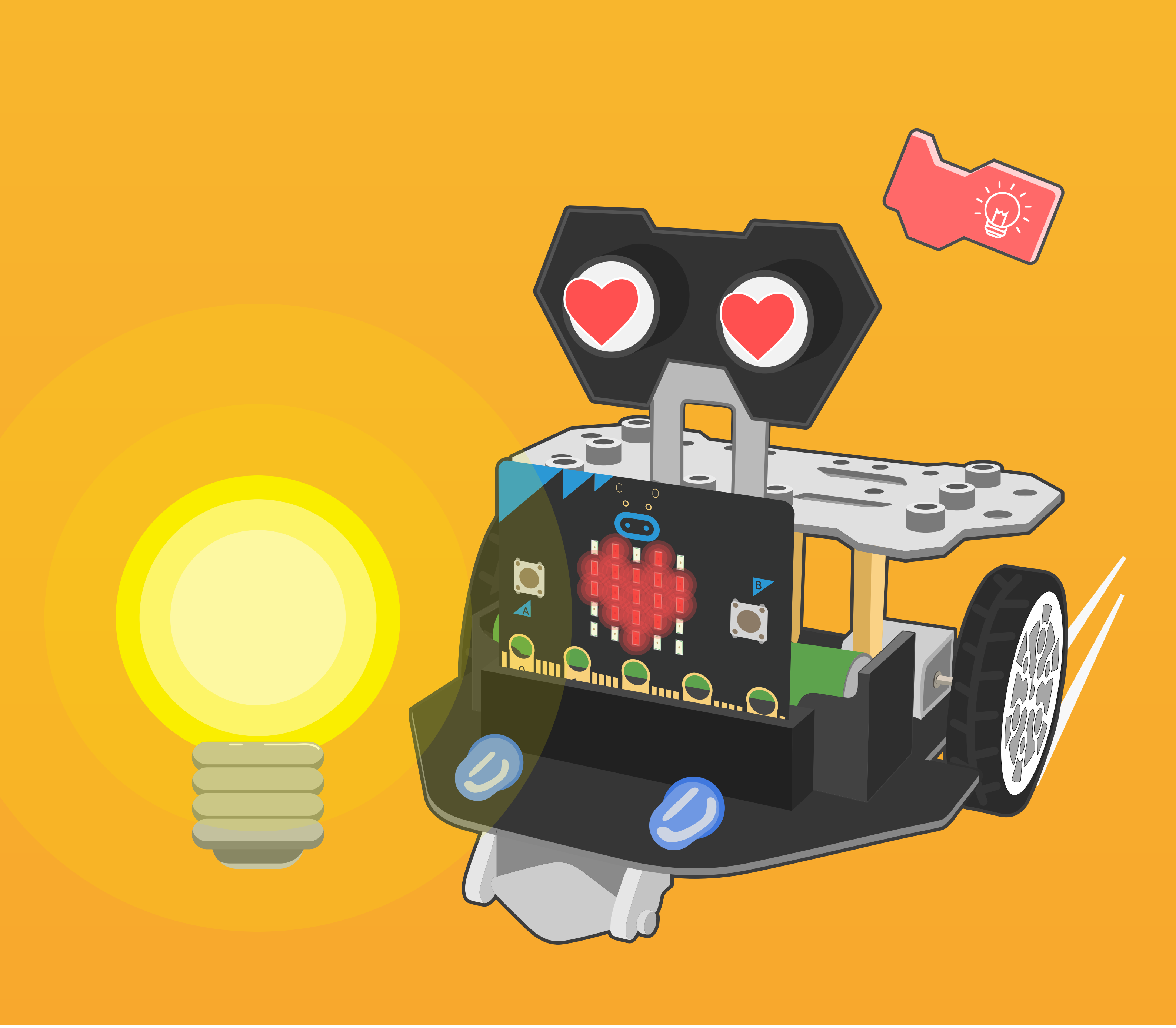
| Place | Light Level |
|---|---|
| Office | 125 |
| Photo studio | 255 |

Note: data in the experiments are for reference only.

# Think & Explore

Light level varies from places, let our Maqueen Plus explore! After that, please think if the light level of each place is reasonably designed.

| Place | Light Level |
|---|---|
| Kitchen | |
| living-room | |
| Bedroom | |
| Study | |



# Chapter 6

## Moth Robot

You must have seen that at summer night, plenty of moths fly around the streetlight, flame and any places with bright light. Why are moths attracted to flame? One idea is that moths are able to find their way partly by using light as a compass. You know what, Maqueen Plus can change into a moth robot because it has a pair of light-sensitive eyes.

# Goal

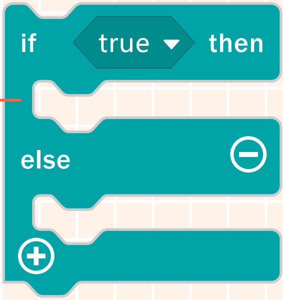
1.Learn condition block
2.Program flowchart

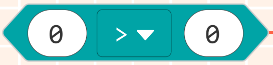
# Command Learning

## Block Brief

If...then...else

If a value is true, then do the first block of statements. Otherwise, do the second block of statements.





Comparison operator

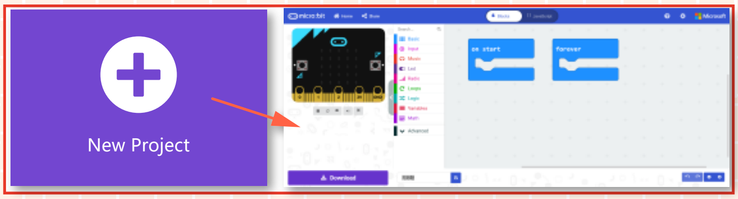
Return true if the first input is greater than the second input.

# Hands-on Practice

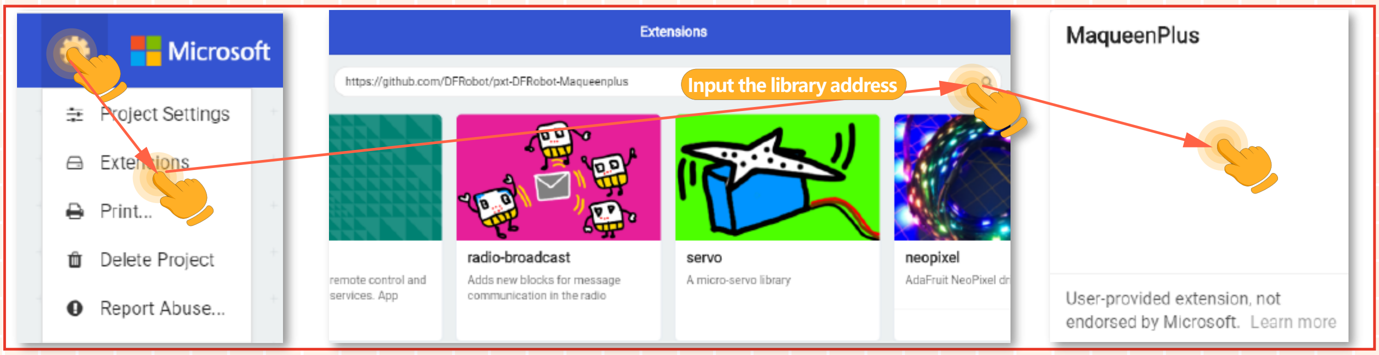
## Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

1.Enter MakeCode editor



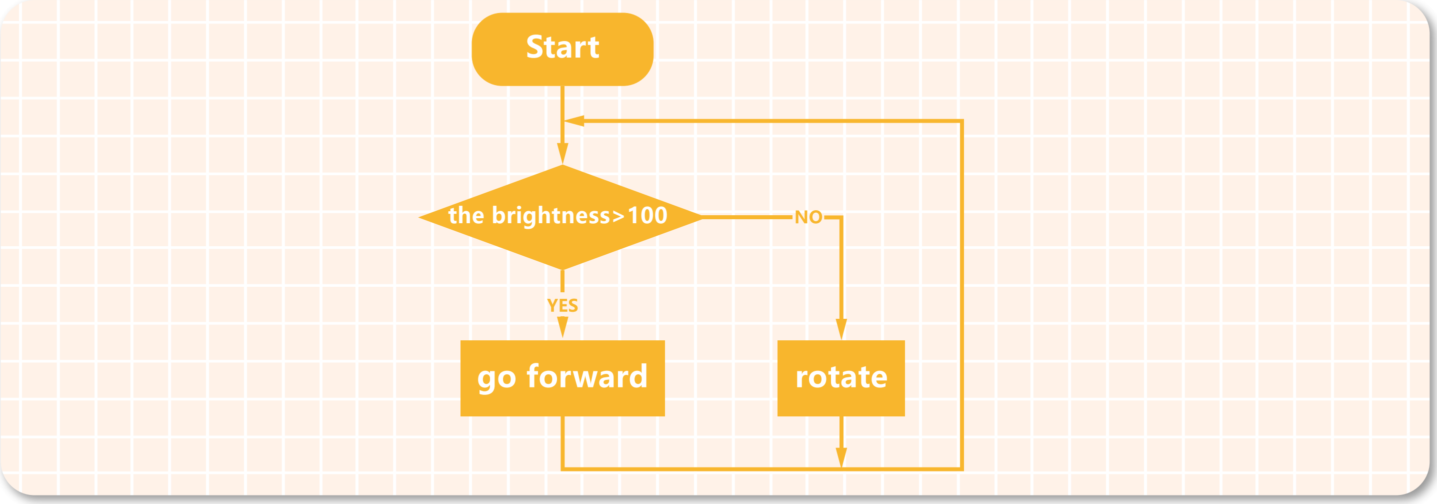
2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

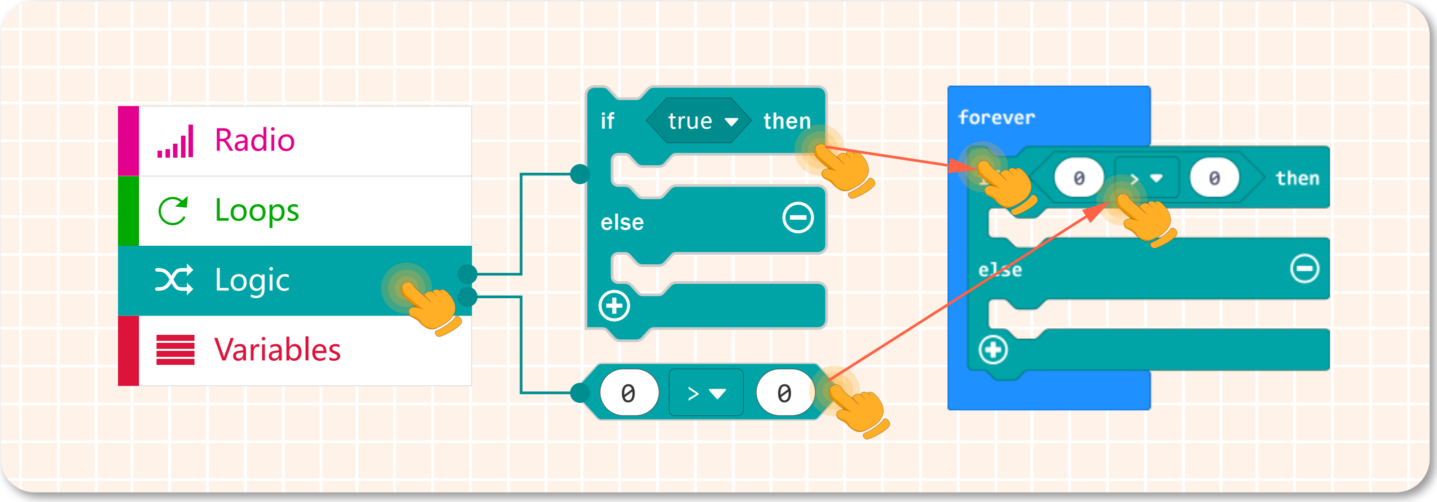
3.Add the extension library

## Step 2 Programming

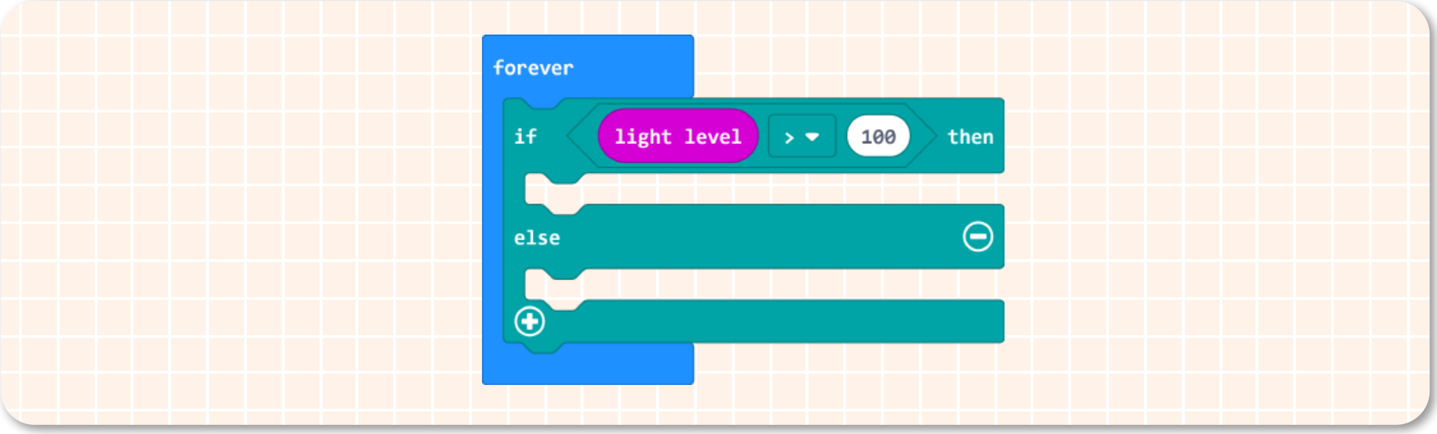
1.When the light level is more than the given value (100 in the example), the moth robot moves towards the light source; when less than that value, the robot revolves around its center. Drawing a corresponding flowchart according to the above functions is gonna help us a lot with programming!



2.Embed the condition blocks into the “forever” block, then the condition judgment can be processed in real-time.



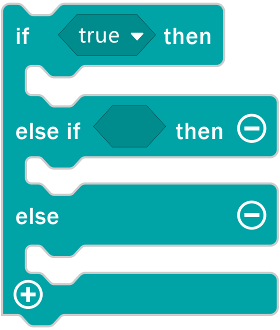
3.The key point of the whole program is the condition statement “light level >100”. Different operations will be executed according to the result of the condition block.



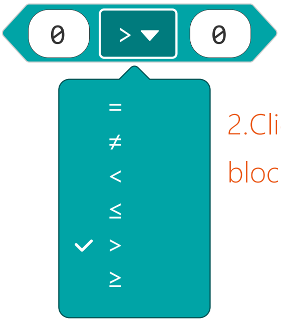
```
forever
  if light level > 100 then
  else
```

Knowledge Expansion

1.Click the " ⊕ " in the condition block to add a condition, click " ⊖ " to delete a condition.



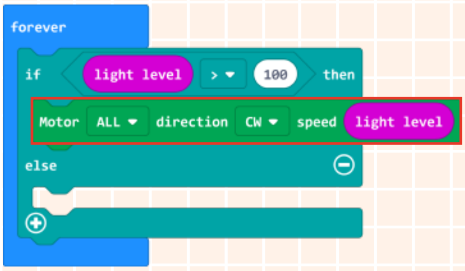
```
if  true ▾  then

else if  ⬡  then ⊖

else ⊖

⊕
```



```
0  > ▾  0
```

2.Click the " > ▾ " in the comparison block to select different operators.

4.When the condition "Light level>100" is true, the robot car moves forward at the speed of the current light level.

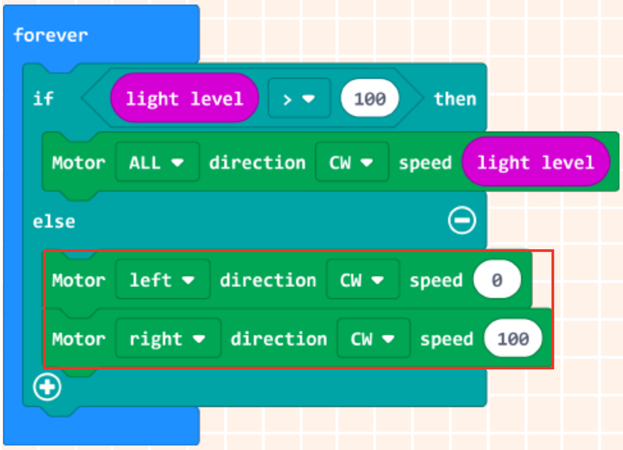


```
forever
  if  light level  > ▾  100  then
    Motor  ALL ▾  direction  CW ▾  speed  light level
  else ⊖

  ⊕
```

Knowledge Expansion

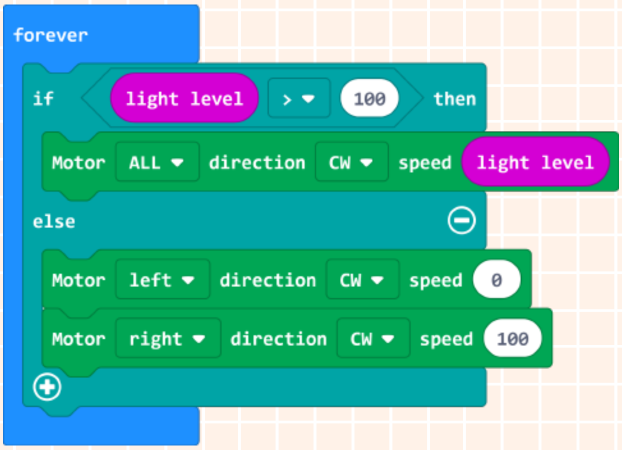
Set light level as speed, the higher the light level, the faster the speed.

5.When the condition"Light level>100" is false, the Maqueen Plus rotates around its center.



```
forever
  if  light level  > ▾  100  then
    Motor  ALL ▾  direction  CW ▾  speed  light level
  else ⊖
    Motor  left ▾  direction  CW ▾  speed  0
    Motor  right ▾  direction  CW ▾  speed  100
  ⊕
```

6.The complete program is shown below.



```
forever
  if  light level  > ▾  100  then
    Motor  ALL ▾  direction  CW ▾  speed  light level
  else ⊖
    Motor  left ▾  direction  CW ▾  speed  0
    Motor  right ▾  direction  CW ▾  speed  100
  ⊕
```

Knowledge Expansion

Condition to be judged: light lecel

Here we need to find a suitable critical value. Since if the value is too large, the moth robot will not move at all till a relatively strong light is given; if the value is too small, the robot will not likely to stop. So we have to set the critical value reasonably according to different conditions.

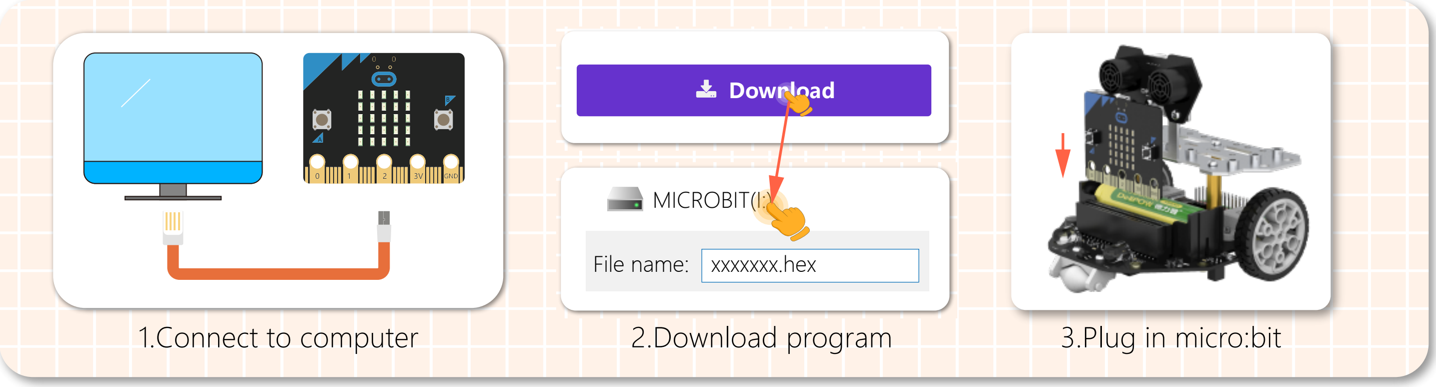
7.Name your project as "Moth robot" and save it.

## Step 3 Download Program

1.Connect to a computer: connect the micro:bit to your computer with a USB cable before downloading. There will be a hard-disk named micro:bit appearing in the computer when the connection is successful.

2.Download the program: download your project into the micro:bit hard-disk.

3.Plug in the micro:bit board: after downloading the program, plug the micro:bit board into Maqueen Plus.



File name: xxxxxxx.hex

1.Connect to computer

2.Download program

3.Plug in micro:bit

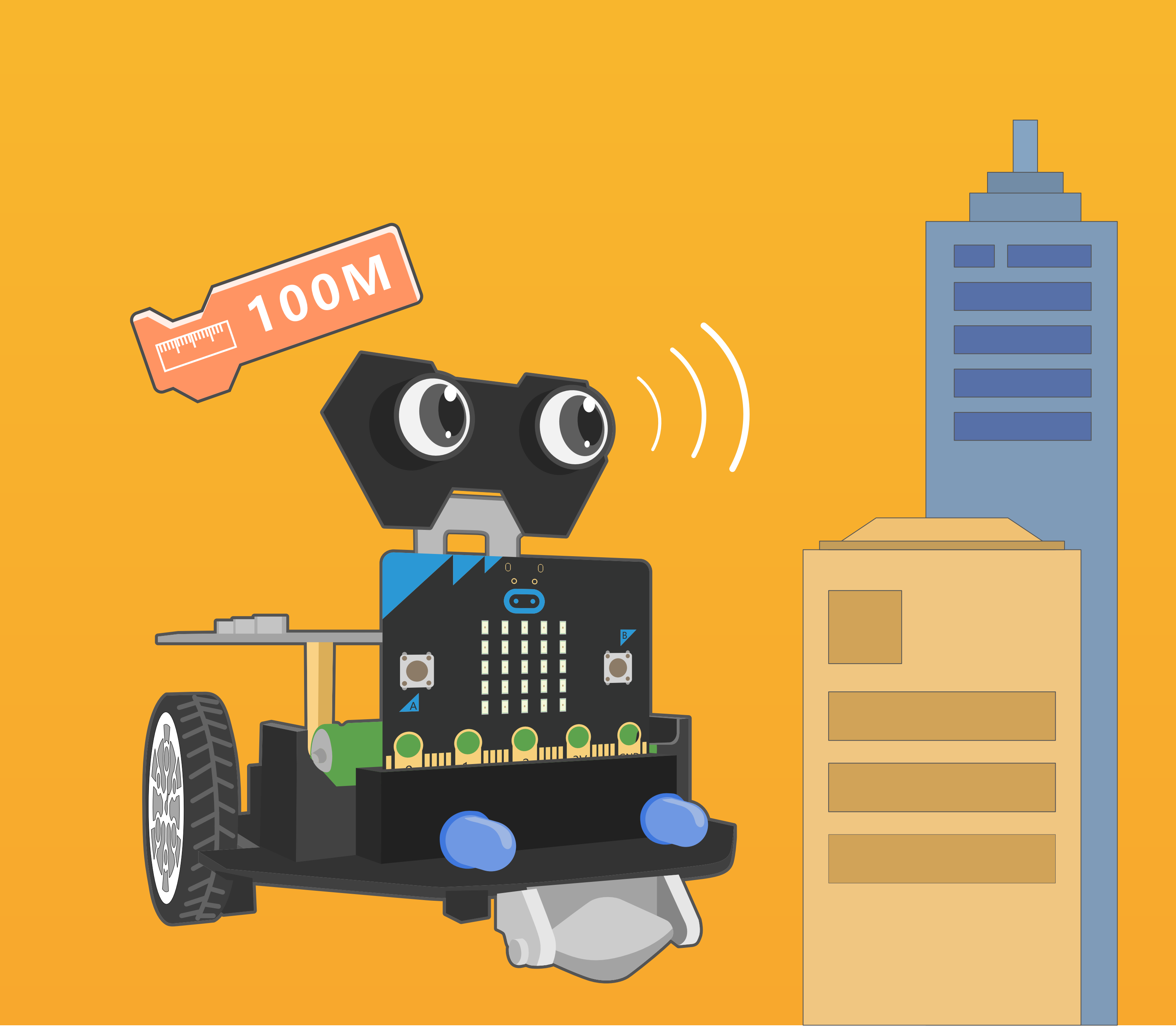
## Step 4 Effect Display

Turn on the power switch, then Maqueen Plus will turn into a moth robot. When the light level is over 100, our moth robot moves towards the light, the brighter the light is, the faster Maqueen Plus runs. When the light is less than 100, the robot will get lost and rotate around. So funny, right ? come to play with this moth robot!

## Think & Explore

Let’s do a robot running competition! Use a flashlight to lead Maqueen Plus to run forward, the one who uses the least time to finish the game will be the winner. Remember, do not cross the line. Invite your friends to join the game!

Tips: maintaining the speed within a reasonable range holds the key to success.



# Chapter 7

## Little Ranging Expert

We have known that Maqueen Plus is such a changeable robot with various functions, but more surprisingly, he can measure distance using his ultrasound eyes. With this buddy, you can say goodbye to your measuring tools.

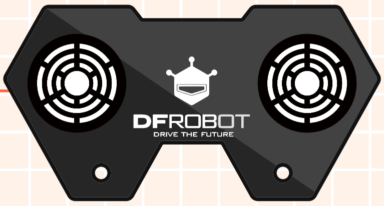
## Goal

1.Get to know ultrasound
2.Learn the principle of ultrasound

## Electronic Component

### Ultrasound Brief

Ultrasonic sensor

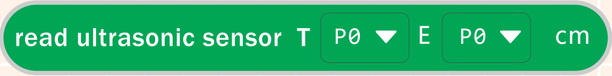


The transmitter sends out ultrasound, and when hitting the object, the ultrasound reflects as echo and will be sensed by the receiver.

## Command Learning

### Block Brief

Read ultrasonic sensor



read ultrasonic sensor T P0 ▼ E P0 ▼ cm

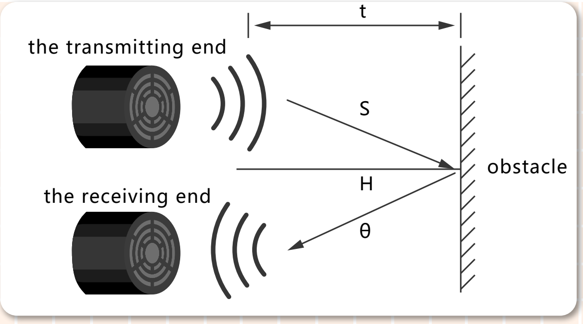
Read the value the ultrasonic sensor detects, unit: cm. (Preset the transmitter and receiver on the sensor)

### What is ultrasound?

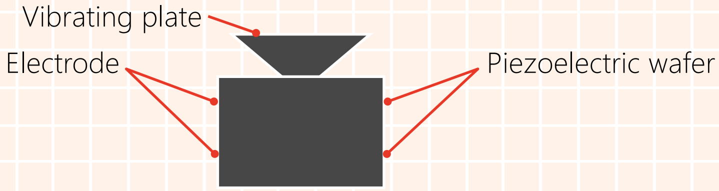
As we know, when vibrating, objects produce sound waves. Some of them can be heard by human ears, while others cannot. Scientists named the vibrating times per second as the sound frequency with its unit named as Hertz. Almost all human beings are able to hear the sound frequencies ranging from 20 to 20000Hertz. Sound frequencies out of that range are inaudible to humans. For those sound, the scientists name it as "Ultrasound".

### How does an ultrasonic sensor measure distance?

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The sensors determine the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.



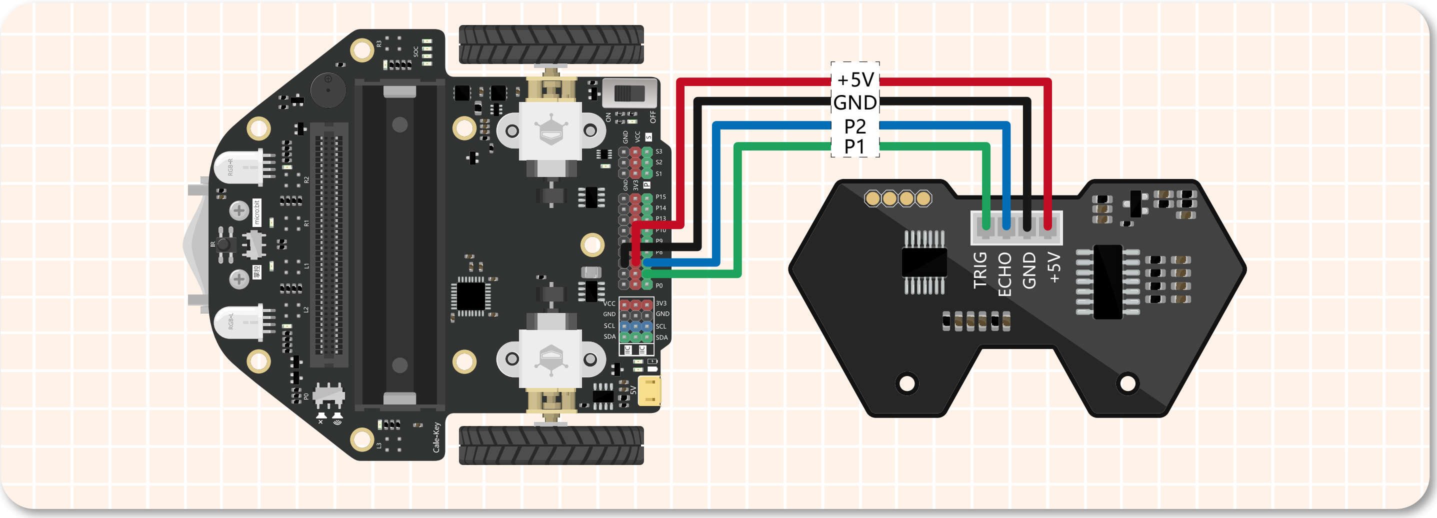
Ultrasonic sensors will convert the reflected sound into an electrical signal. A commonly used sensor consists of a piezoelectric wafer that can emit ultrasonic waves.



# Hands-on Practice

## Hardware Connection:

The pin connection is shown below:

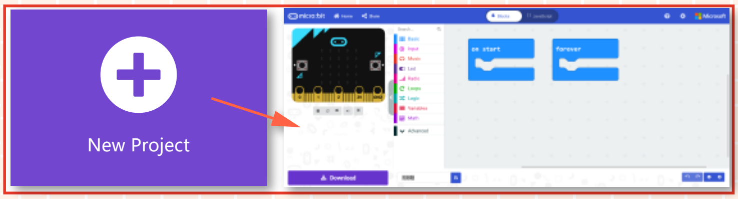


## Step 1 Create a New Project

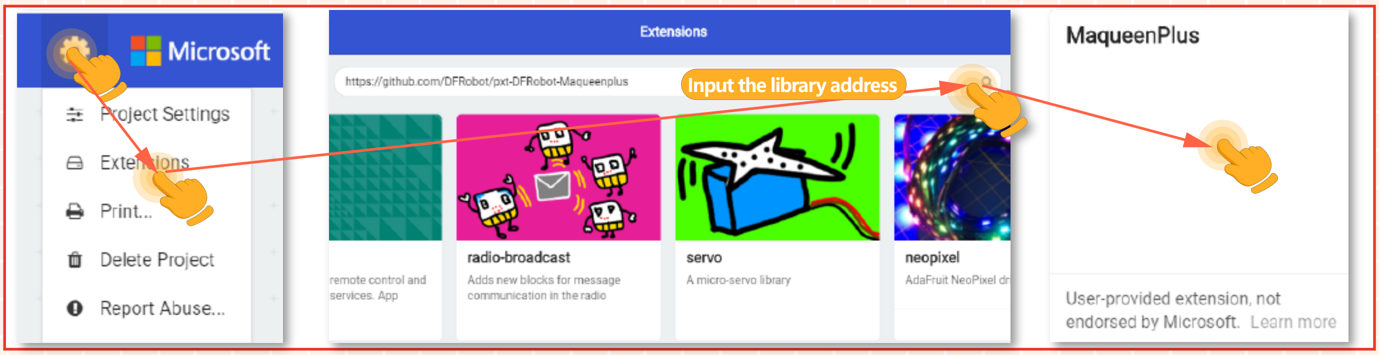
1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface



https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

remote control and services. App

Adds new blocks for message communication in the radio

A micro-servo library

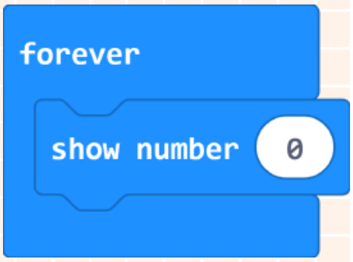
AdaFruit NeoPixel dr

User-provided extension, not endorsed by Microsoft. Learn more

3.Add the extension library

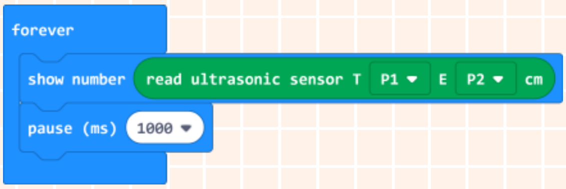
## Step 2 Programming

1.Place the "show number" block into the "forever" block, shown as below.



```
forever
    show number 0
```

2.Put the ultrasonic sensor value inside the “show number” block to display the current distance. Add a “pause” block to prevent the serial reading the value too fast.

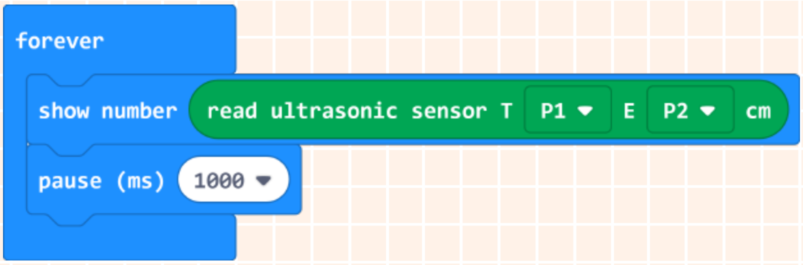


```
forever
    show number  read ultrasonic sensor T  P1 ▾  E  P2 ▾  cm
    pause (ms)  1000 ▾
```

**Knowledge Expansion**

“T” is the transmitting end, corresponding to the “trig” on the ultrasonic sensor; “E” is the receiving end, corresponding to the “echo” . On Maqueen Plus, trig is the P1, echo is the P2, so we need to set the pins in the ultrasonic block to P1, P2.

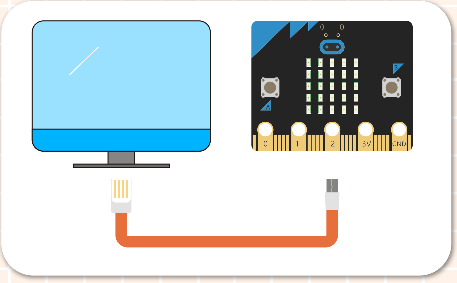
3.The entire program is shown below.



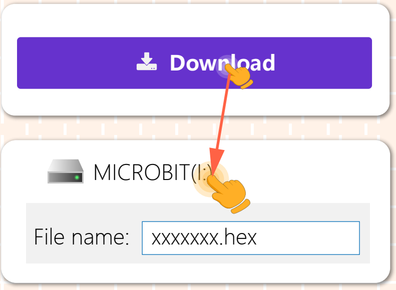
```
forever
    show number  read ultrasonic sensor T  P1 ▾  E  P2 ▾  cm
    pause (ms)  1000 ▾
```

4.Name your project as “Little ranging expert” and save it.

## Step 3 Download Program

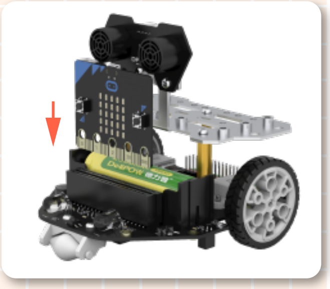


1.Connect to computer



File name: xxxxxxx.hex

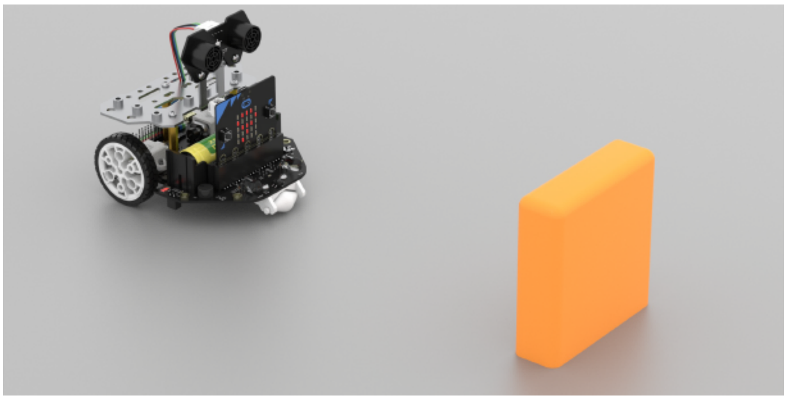
2.Download program



3.Plug in micro:bit

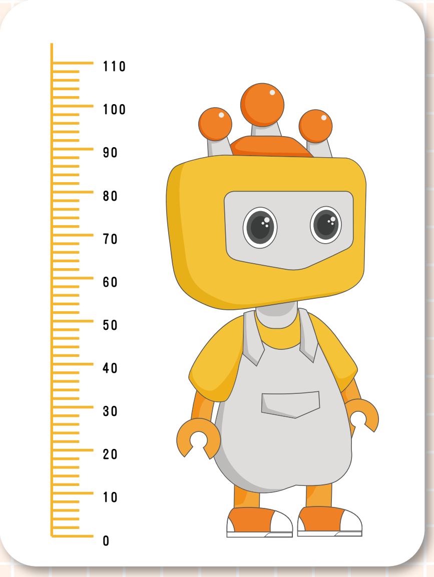
## Step 4 Effect Display

Turn on the power switch, then we can use Maqueen Plus to measure distance. The detected distance will be displayed on micro:bit. We can use two different measuring ways to test the accuracy of Maqueen Plus. As shown below:

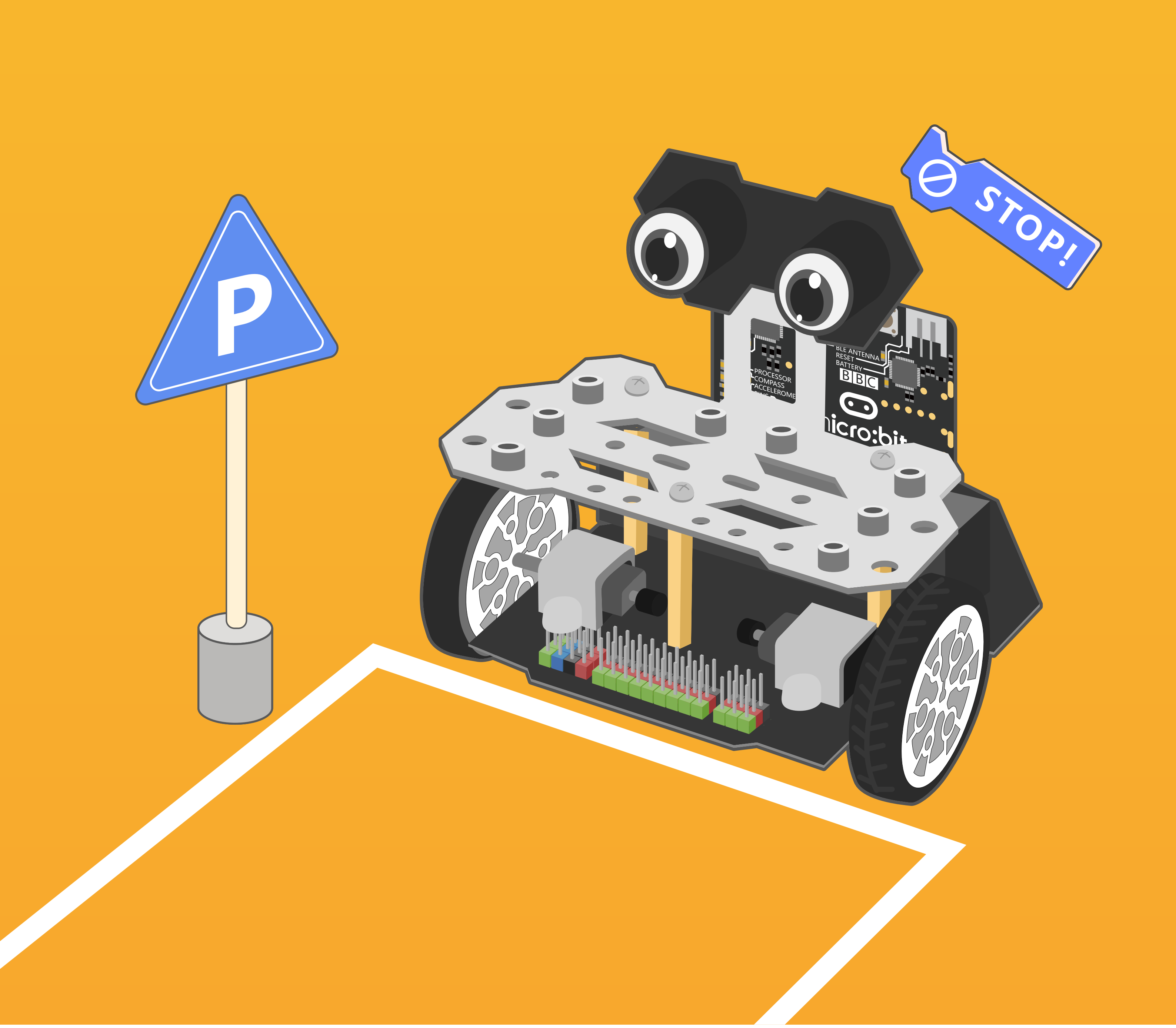


## Think & Explore

How to use an ultrasonic sensor to measure the height of a person?



To make the measurement more accurately, you have to:
Pay attention to the position and direction of the ultrasound.
Calibrate the sensor within 10cm to prevent large error.



# Chapter 8

## Car Reversing Helper

To help drivers back up and park more safely and easily, there are a lot of  car reversing radar alarming systems emerging on the market. Based on ultrasonic measurement principle, the reversing radar system can detect the distance between the car and obstacle, then the drivers can be alerted by beeps or the dashboard display, which could be very helpful for new drivers.

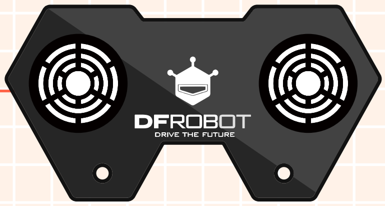
## Goal

1.Learn how to use variables
2.Learn how to embed a condition block inside another one.

## Electronic Component

### Ultrasound Brief

Ultrasonic Sensor

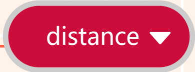


The transmitter sends out ultrasound, and when hitting the object, the ultrasound reflects as echo and will be sensed by the receiver.

## Command Learning

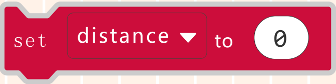
### Block Brief

Variable


distance ▾

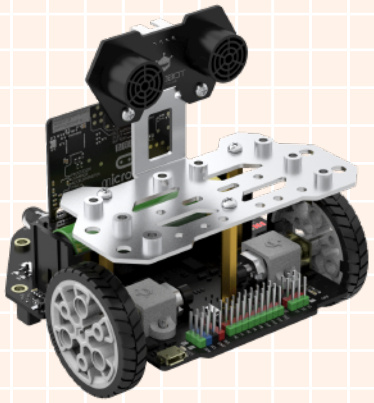
Changeable value: number, string, list

Assign a value to variable


set distance ▾ to 0

Assign a value to the variable.
Default setting:  distance=0

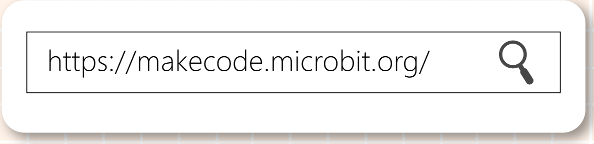
## Hands-on Practice



In this project, our aim is to measure the distance between the end of the robot and obstacles, so we need to inversely install the ultrasonic sensor onto the expansion bracket, as shown below.

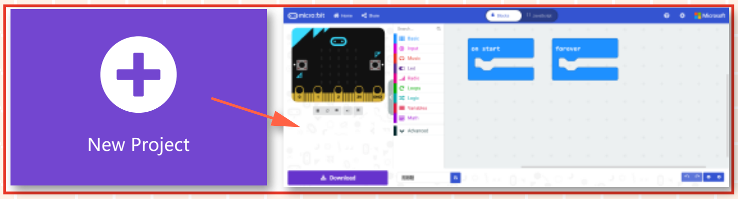
### Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

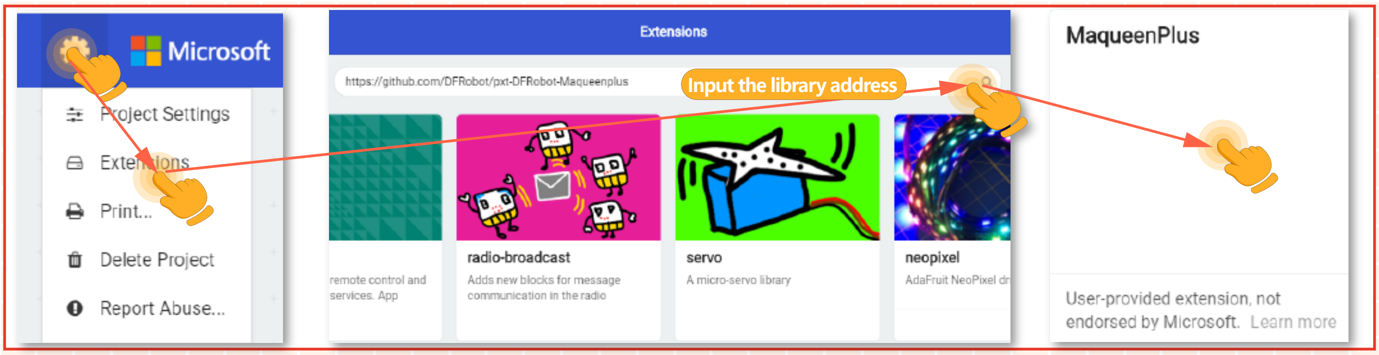


https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface



https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

remote control and services. App

Adds new blocks for message communication in the radio

A micro-servo library

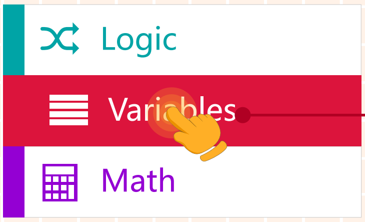
AdaFruit NeoPixel dr

User-provided extension, not endorsed by Microsoft. Learn more

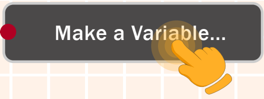
3.Add the extension library

## Step 2 Programming

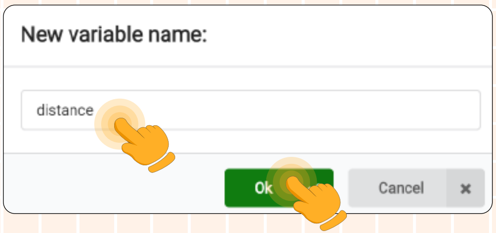
1.Create a variable and name it as “distance”.



1.Click “Variable” in the command block section



2.Click “Make a variable”

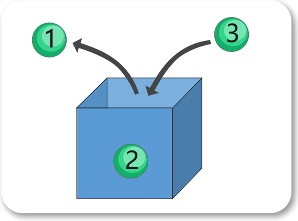


New variable name:

distance

3.Name the variable as “distance”
4.Click “OK”.

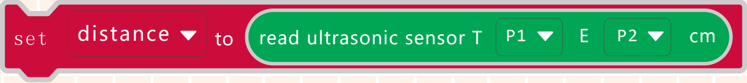
### Knowledge Expansion



What is a variable?

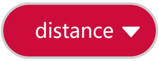
We may think of a variable as a container or box where we can store data that we need to use later, and each box can only hold one value (number, text and Boolean data) at a time. For example, use it to store an integer, after we put 1into it, we put 2 into it, then we can only get 2 from this box. The name of the box is the variable’s name and the value of the variable is placed inside the box.

2.In this project, we need to monitor the distance value the sensor detected in real-time, so we have to assign the value to the variable “distance”.



```
set distance ▾ to read ultrasonic sensor T P1 ▾ E P2 ▾ cm
```

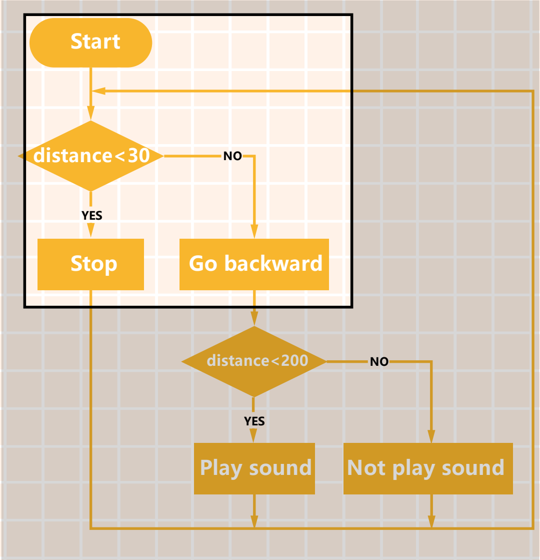
### Knowledge Expansion

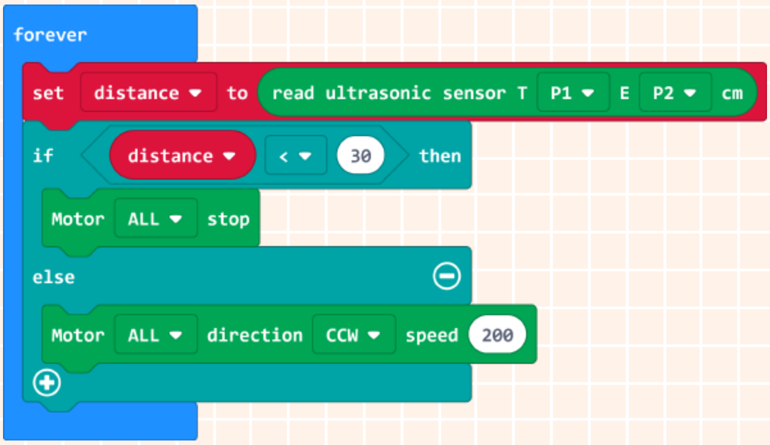


Then we can directly use the variable “distance” to call the distance value later.

3.Once the distance between the car and the obstacle is smaller than the preset value (distance <30cm in the example), the car stops. If the distance is long enough, the car goes backward, and the buzzer keeps beeping.

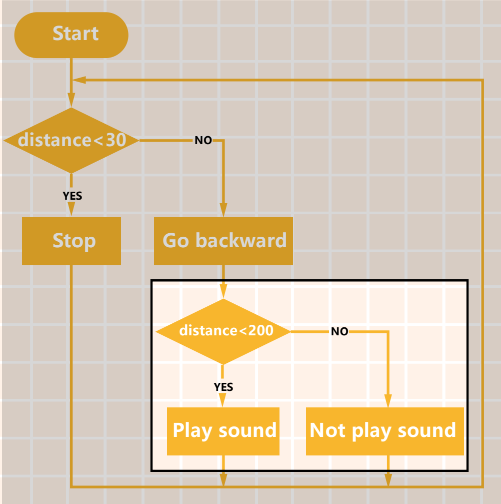
Drawing a flowchart is gonna help us understand the program.

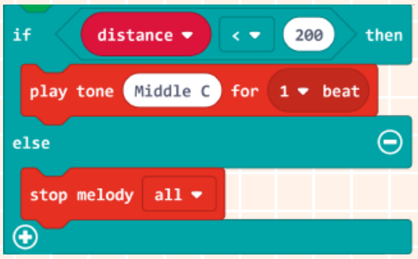




```
forever
    set distance to read ultrasonic sensor T P1 E P2 cm
    if distance < 30 then
        Motor ALL stop
    else
        Motor ALL direction CCW speed 200
```

4.Complete part of the program: car stops and goes backward.

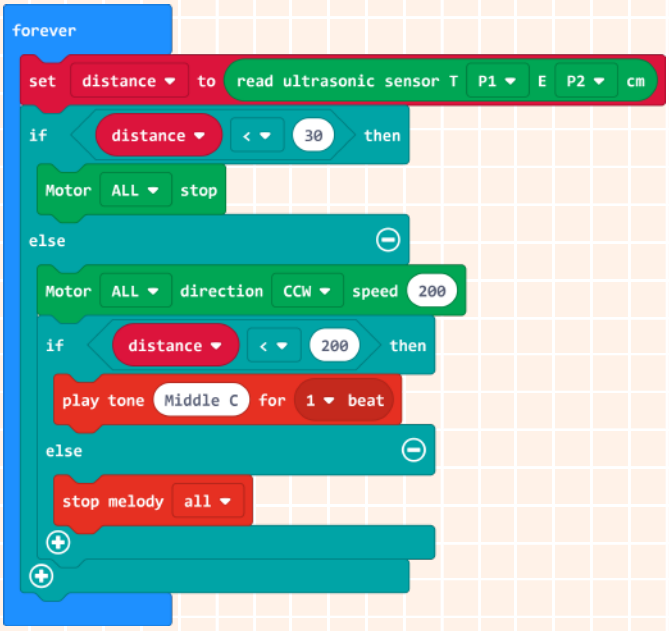




```
if distance < 200 then
    play tone Middle C for 1 beat
else
    stop melody all
```

Only when the car is going  backward will the buzzer start making sound, so the condition block “distance<200” should be put under the car backward block.

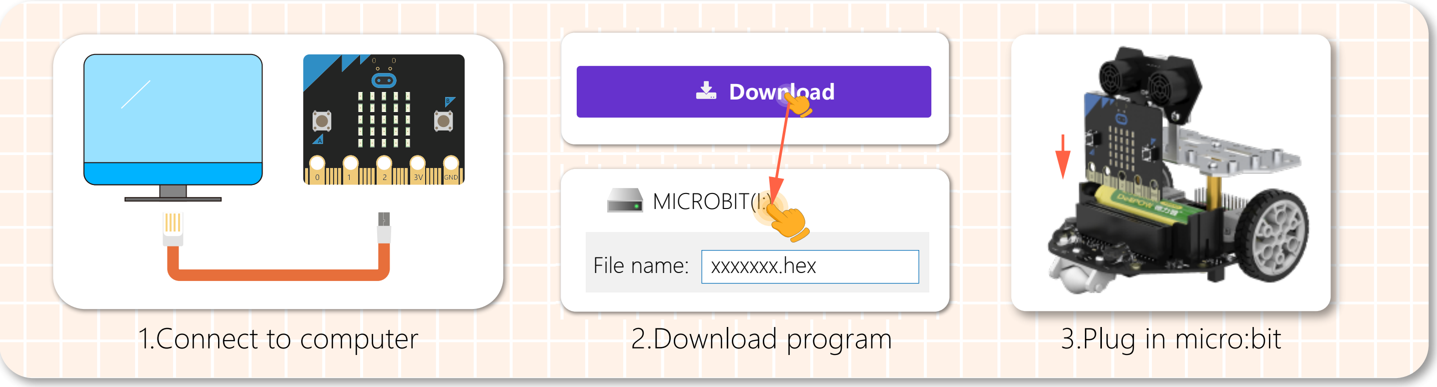
5.The complete program is shown below.



```
forever
    set distance to read ultrasonic sensor T P1 E P2 cm
    if distance < 30 then
        Motor ALL stop
    else
        Motor ALL direction CCW speed 200
        if distance < 200 then
            play tone Middle C for 1 beat
        else
            stop melody all
```

6.Name your project as “Car Reversing Helper” and save it.

## Step 3 Download Program



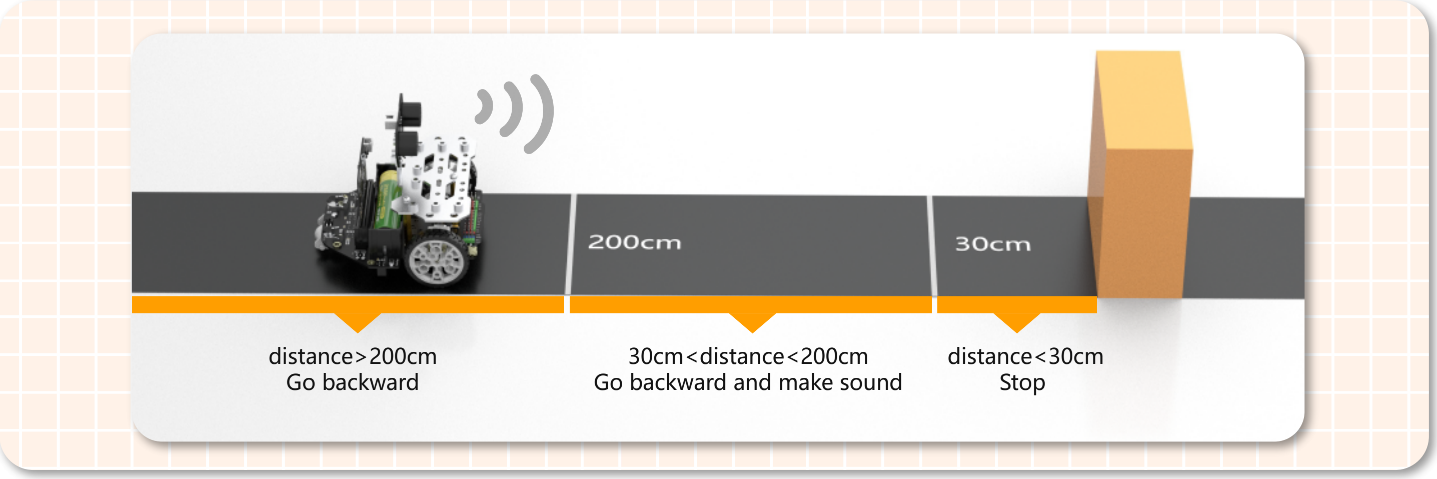
1.Connect to computer

2.Download program

3.Plug in micro:bit

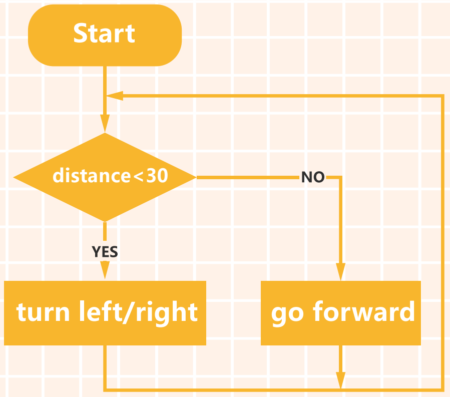
## Step 4 Effect Display

After we completed all the above steps, put an obstacle behind the Maqueen Plus car, turn on the power switch. When the detected distance is smaller than 200, the car starts going backward; when the distance is between 30 to 200, the buzzer keeps beeping; distance <30, the car stops.

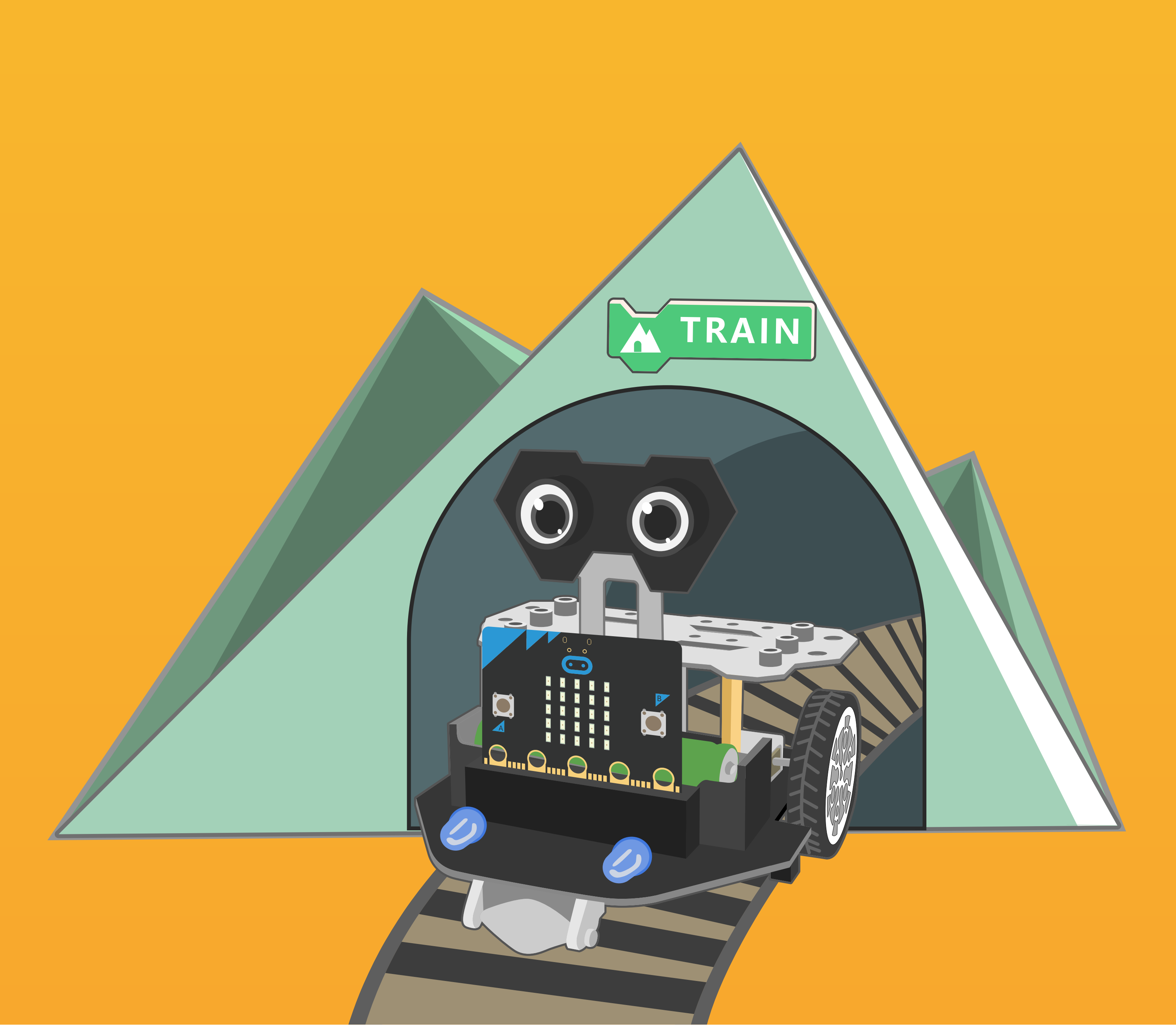


# Think & Explore

We have made a car reversing helper with Maqueen Plus, the car will stop when it is very near an obstacle, but can you let the car bypass it? Make an obstacle avoidance car according to the flowchart below.



Tip: the ultrasonic sensor should point to the direction the car moves toward.



# Chapter 9

## Line-tracking Robot

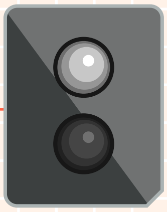
Our line-tracking robot is fond of exploring things with a map. No matter how complicated the road is, give him a long enough track, he will trace it to the end of the world. Let's step on an adventure with Maqueen Plus robot!

## Goal

1.Learn the principle of Grayscale sensor
2.Learn the logic “and”

## Electronic Component

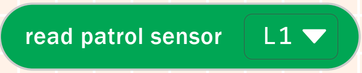
### Figure of the line-tracking sensor



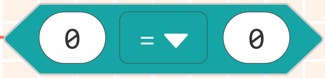
Detect the colors of objects and surfaces by aiming directly at close range, which can allow robots to track specified lines.

## Command Learning

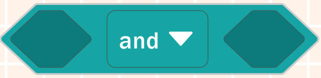
### Block Brief



Measure the intensity of light from black to white; help robot car move along a specified route.

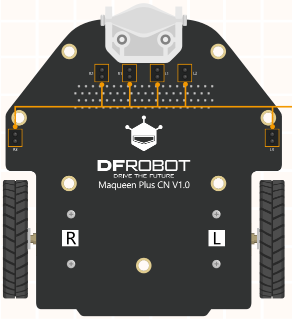


Return true if both inputs equal to each other.

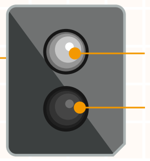


Return true if both inputs are true.

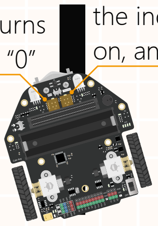
How does a grayscale sensor work?



There are 6 line-tracking sensors integrated on Maqueen Plus board. Each sensor includes an IR transmitter and receiver.



When a line-tracking sensor detected the black line on the map, the indicator turns on, and output “1”, otherwise, the indicator turns off, and output “0”.

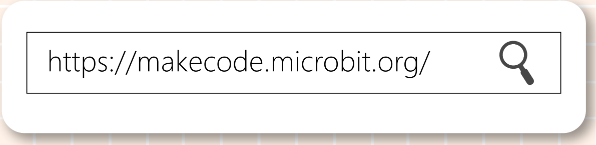

the indicator turns off, and output “0”
the indicator turns on, and output “1”

Note:
1.Since dark colors absorb light(including IR light), when the line-tracking sensor detected black, the IR light emitted by the transmitter cannot be reflected back to the receiver.
2.The output “0” or “1” does not refer to a high/low level, it’ s just a value obtained by processing the read grayscale.

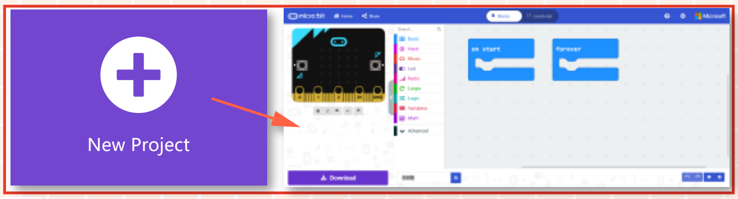
# Hands-on Practice

## Step 1 Create a New Project

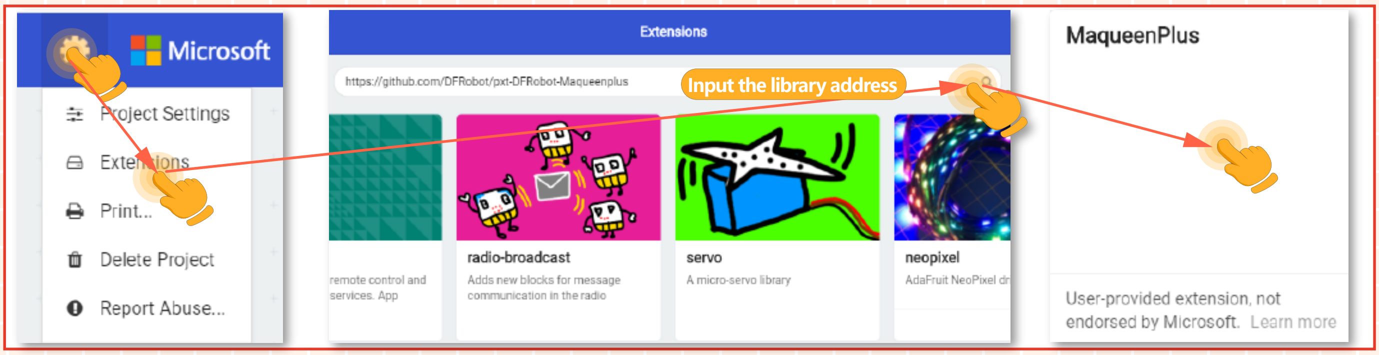
1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus


https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface



remote control and services. App

Adds new blocks for message communication in the radio

A micro-servo library

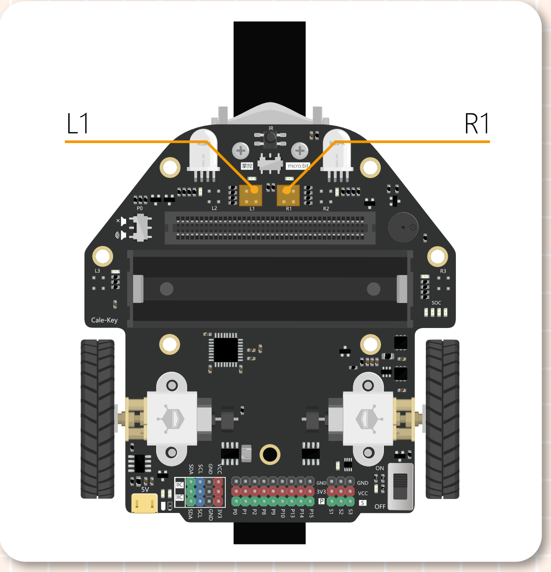
AdaFruit NeoPixel dr

User-provided extension, not endorsed by Microsoft. Learn more

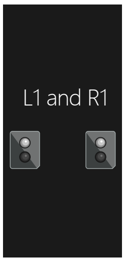
3.Add the extension library

## Step 2 Programming

Maqueen Plus moves along the black line on the map. If you don’t have a map, you can make one using adhesive tape. (Sensors R1 and L1 will be used in this project.)

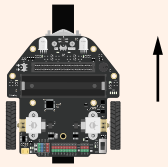
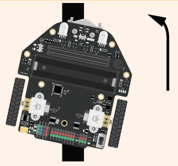
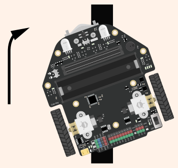


### Knowledge Expansion

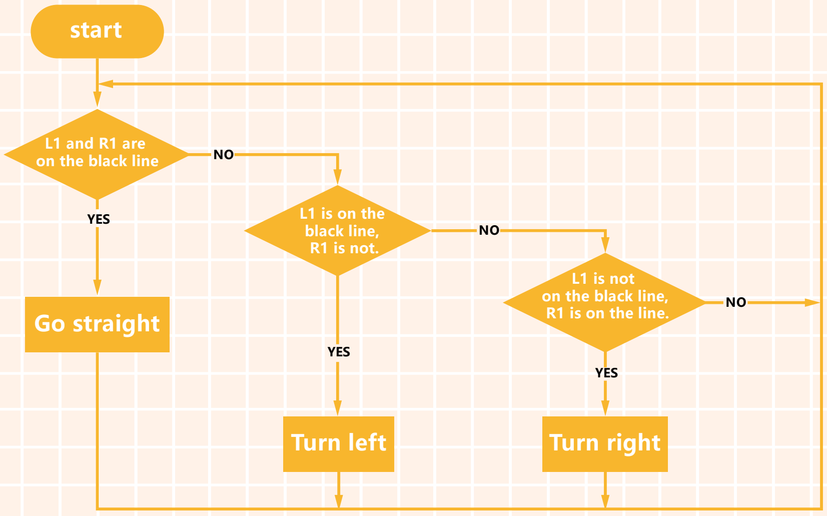


When you draw your own map, please make sure that both sensors L1 and R1 can be placed on the black line.

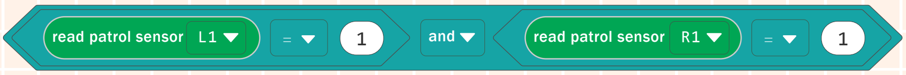
1.There are three possibilities when Maqueen Plus drives on the map.

| Status Image | Sensor Status | Detection | Output | Motor Movement |
|---|---|---|---|---|
|  | L1 and R1 are on the black line | Both the left and right sensors detected the black line. | L1 = 1<br>R1 = 1 | Go straight |
|  | L1 is on the black line, R1 is not. | Only the left sensor detected the black line. | L1 = 1<br>R1 = 0 | Turn left |
|  | L1 is not on the black line, R1 is on the line. | Only the right sensor detected the black line | L1 = 0<br>R1 = 1 | Turn right |

2.Draw the corresponding program flowchart.


start
L1 and R1 are on the black line
NO
YES
Go straight
L1 is on the black line, R1 is not.
NO
YES
Turn left
L1 is not on the black line, R1 is on the line.
NO
YES
Turn right

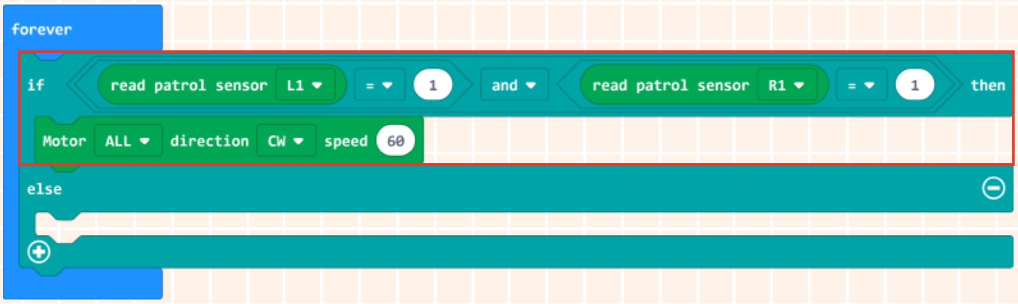
3.Since there are two conditions to judge, the outputs from sensor R1 and L1, we need to use a “and” block to combine them together.



```
read patrol sensor L1 = 1 and read patrol sensor R1 = 1
```

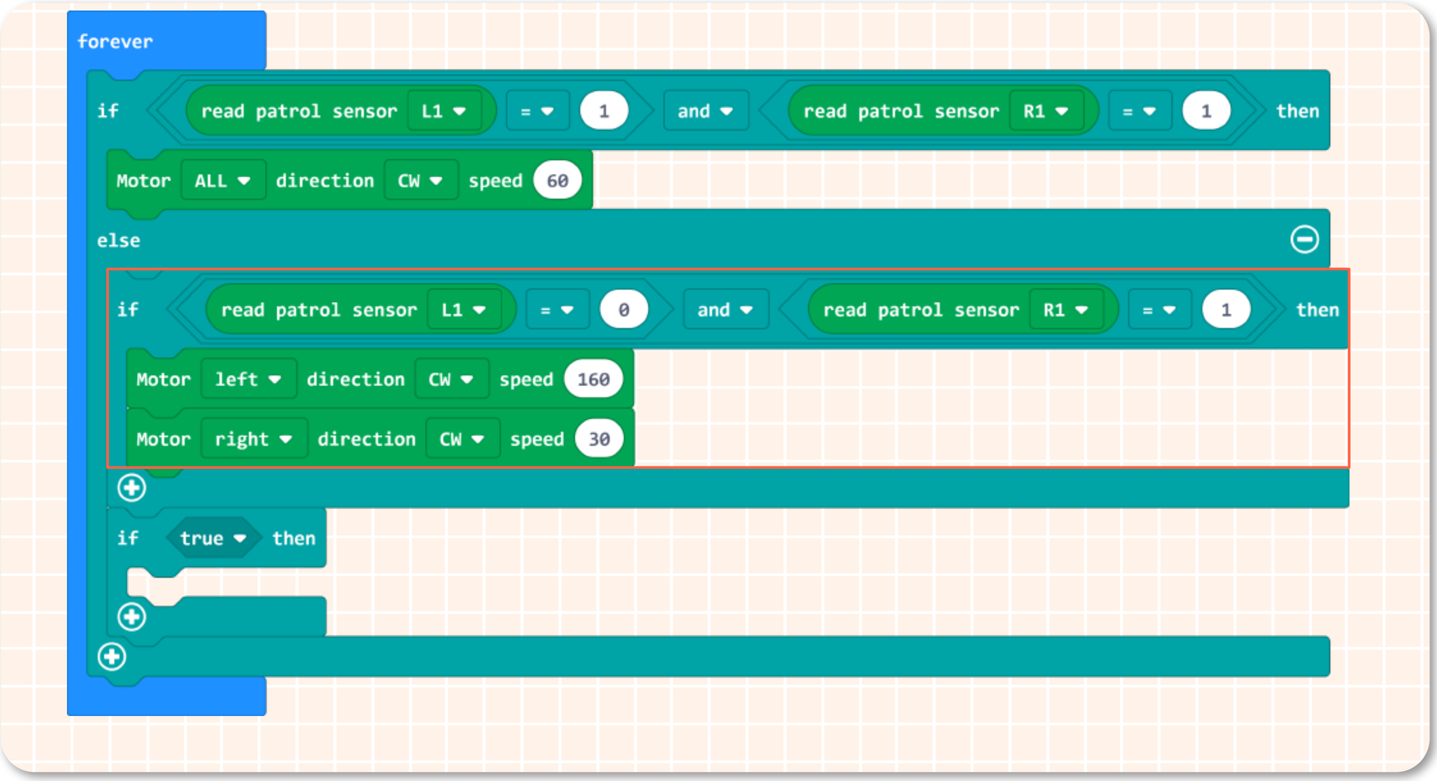
Note: the program means that both sensors (L1 and R1) detected the black line.

4.When the sensor L1 and R1 detected the black line, Maqueen Plus car moves forward.



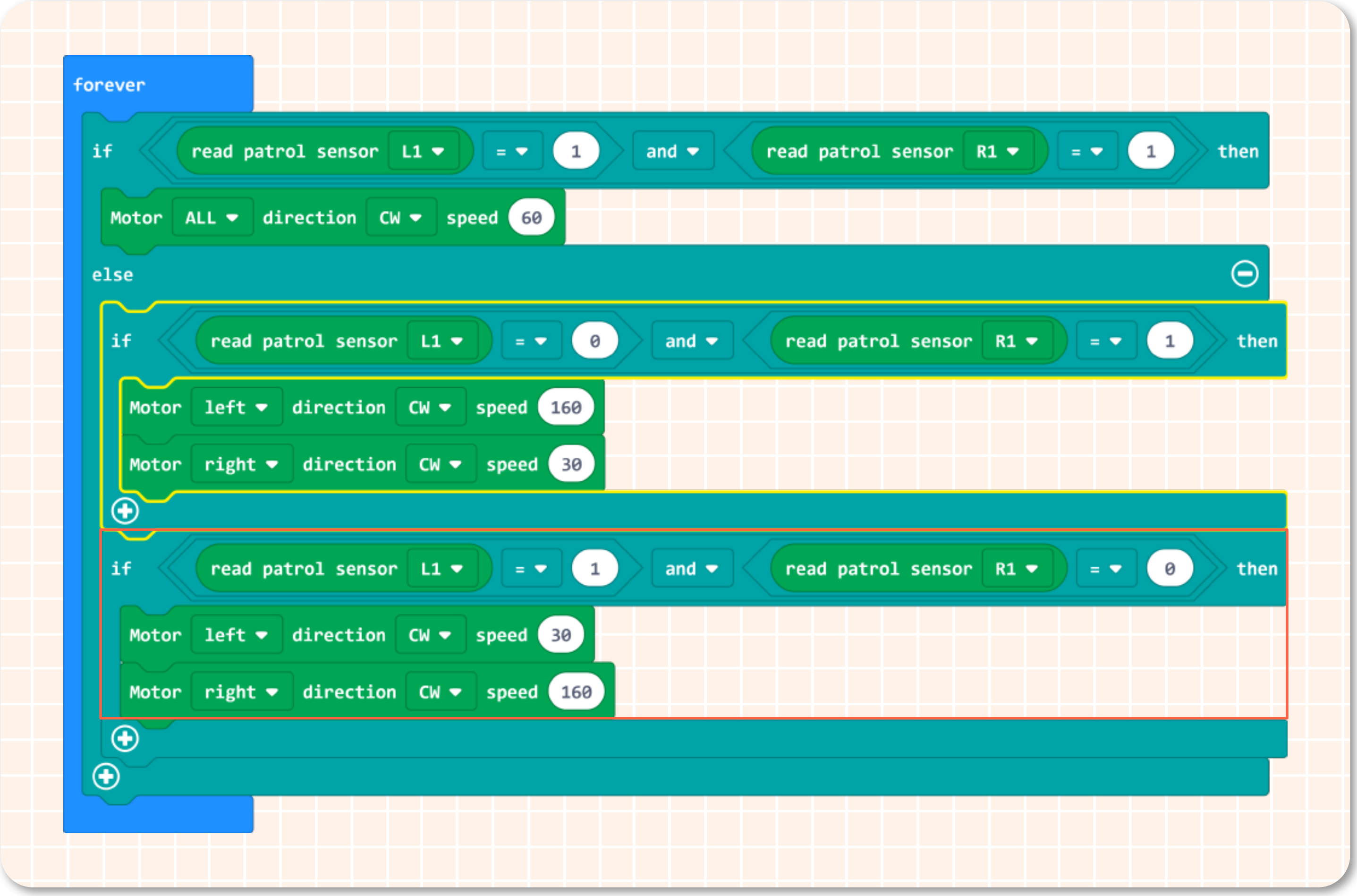
```
forever
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 1 then
    Motor ALL direction CW speed 60
  else
```

5.When only the right sensor R1 detected the black line, Maqueen Plus car turns right.



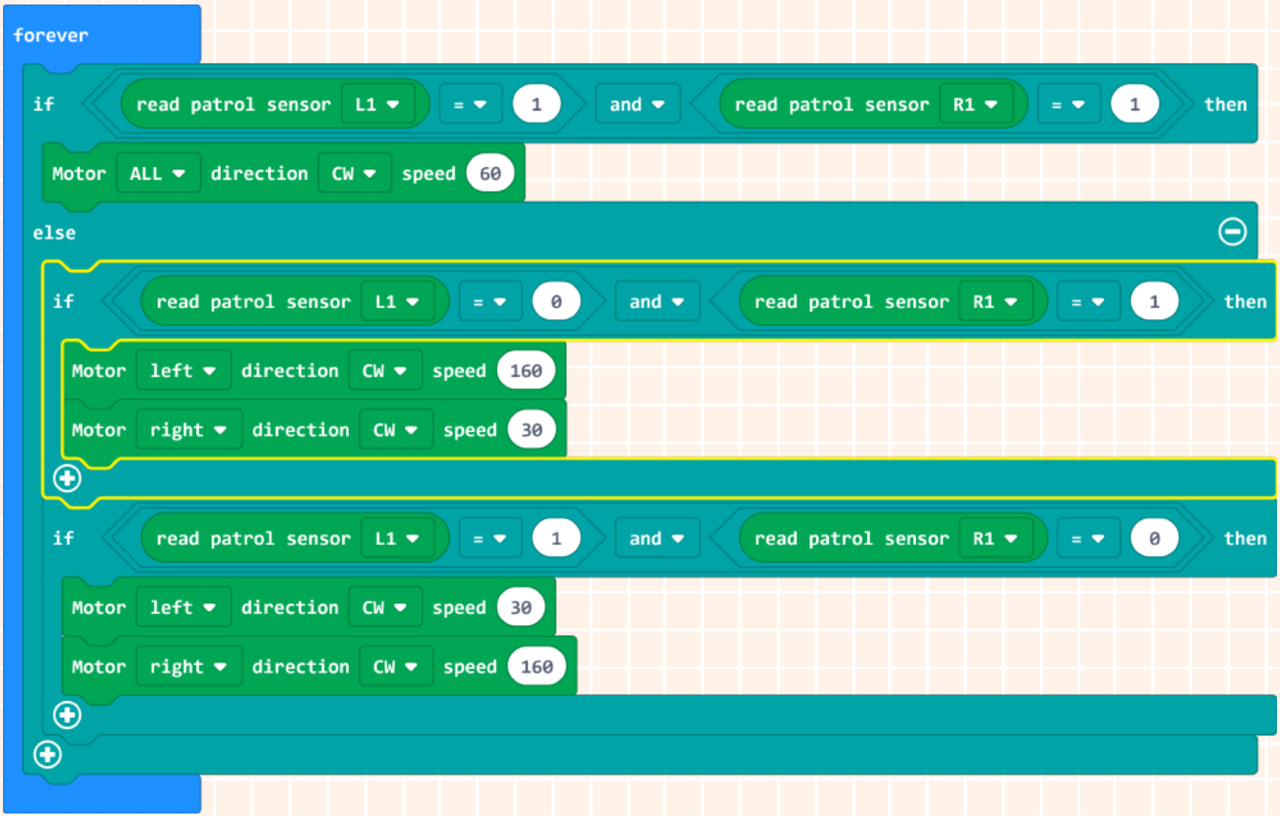
```
forever
  if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 1  then
    Motor ALL direction CW speed 60
  else
    if  read patrol sensor L1 = 0  and  read patrol sensor R1 = 1  then
      Motor left direction CW speed 160
      Motor right direction CW speed 30
    if  true  then
```

6.When only the left sensor (L1) detected the black line, Maqueen Plus car turns left.



```
forever
  if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 1  then
    Motor ALL direction CW speed 60
  else
    if  read patrol sensor L1 = 0  and  read patrol sensor R1 = 1  then
      Motor left direction CW speed 160
      Motor right direction CW speed 30
    if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 0  then
      Motor left direction CW speed 30
      Motor right direction CW speed 160
```

7.The complete program is shown below:

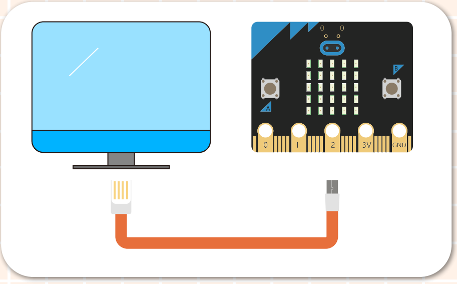


```
forever
    if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 1  then
        Motor ALL direction CW speed 60
    else
        if  read patrol sensor L1 = 0  and  read patrol sensor R1 = 1  then
            Motor left direction CW speed 160
            Motor right direction CW speed 30
        if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 0  then
            Motor left direction CW speed 30
            Motor right direction CW speed 160
```

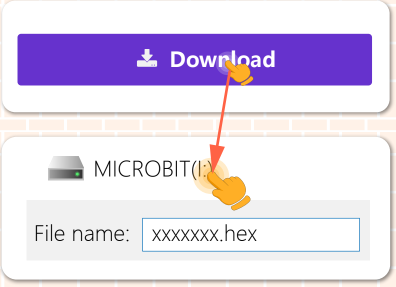
Note: if the Maqueen Plus car turns left or right too much, you can change its motor speed to adjust.

8.Name your project as “Line-tracking Robot ” and save it.

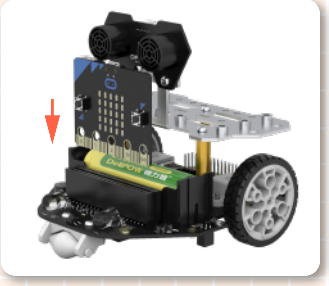
## Step 3 Download Program



1.Connect to computer



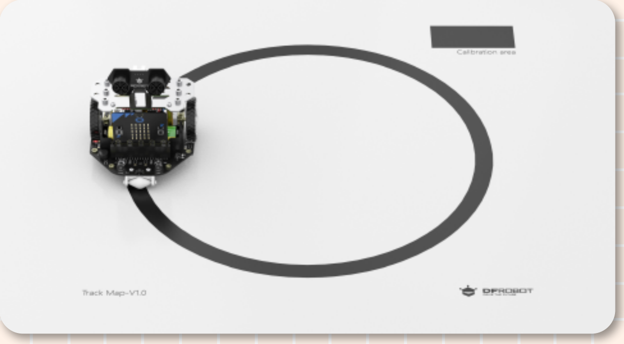
2.Download program



3.Plug in micro:bit

## Step 4 Effect Display

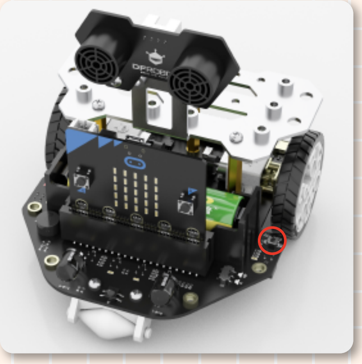
Turn on the power switch after all the above steps done, put Maqueen Plus car on the map, then it will automatically move along the black line, just like a track train!



## Extension-Sensor Calibration

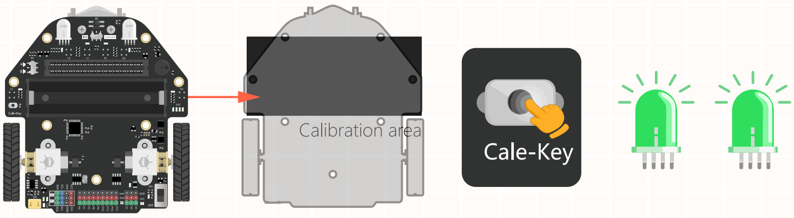
The grayscale sensors on Maqueen Plus can be directly used since they are factory calibrated. But if you find that your sensors cannot detect black line accurately, you can calibrate them as the way shown below:

The button circled in red is the calibrate button.



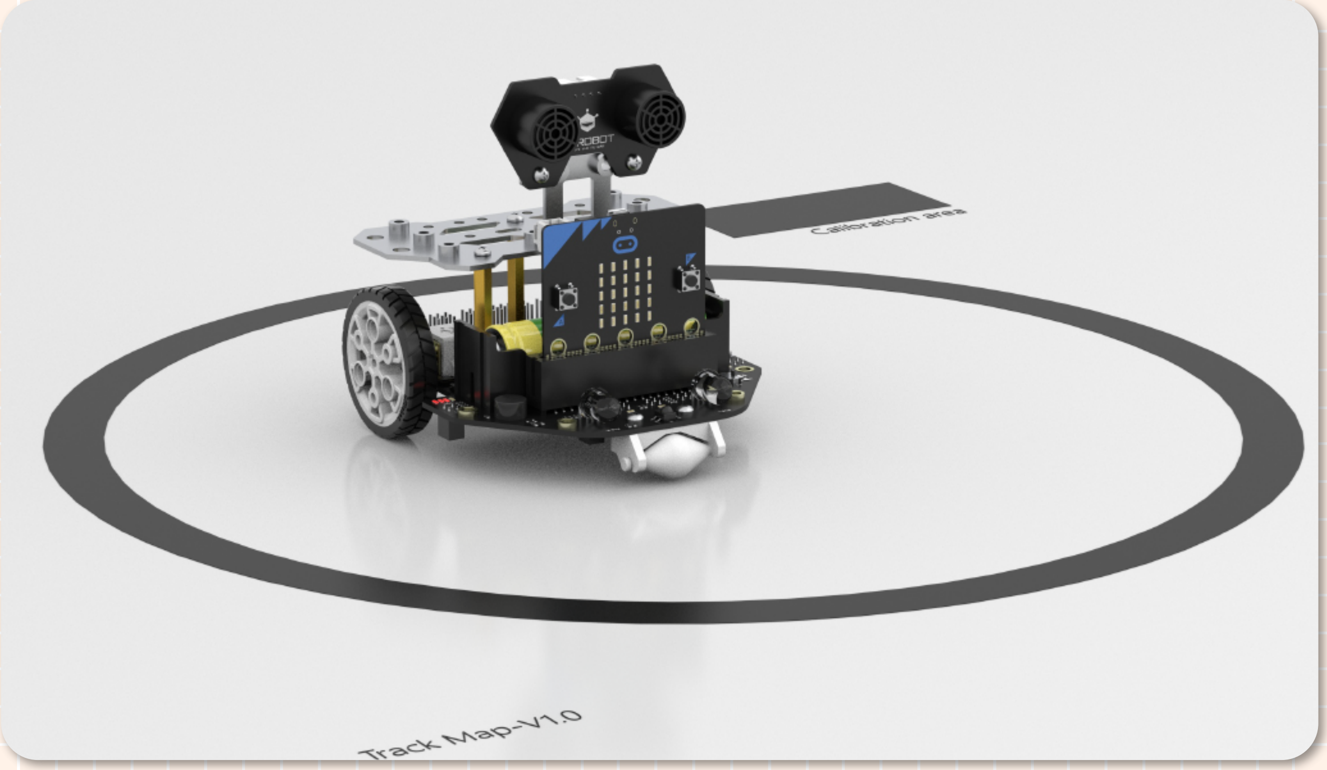
Place Maqueen Plus on the black calibrating area of the map, and make sure all the grayscale sensors are within that area. Press down the calibration button, when the two RGB LEDs flashes green light, release the button, then sensor calibration is done.

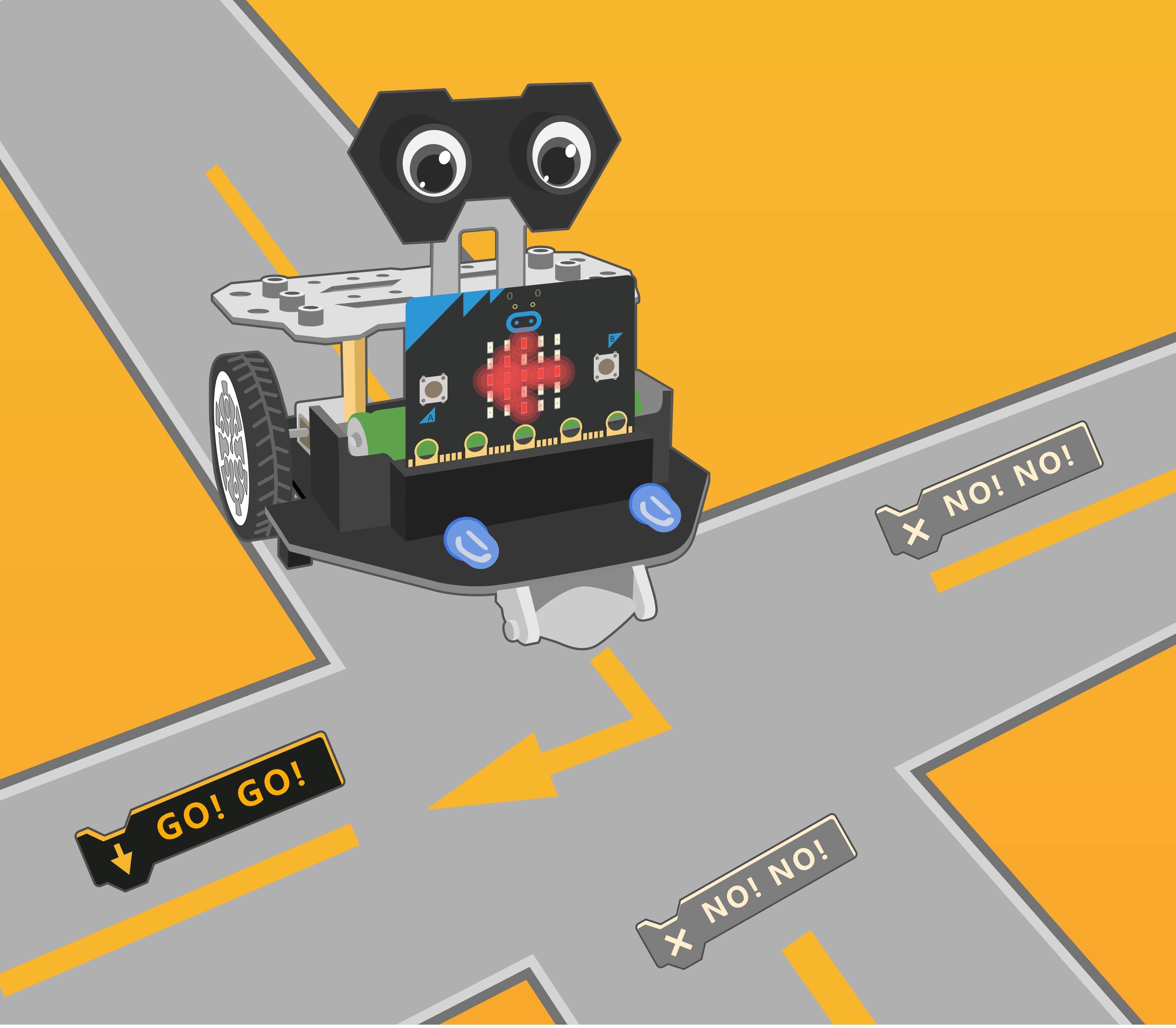
Note: the black line printed by printer cannot be accurately detected sometimes. You can use black tape or marker to make a map if necessary.



## Think & Explore

With the development of technology, sweeping robot is gradually becoming a part of our family life. Simply place it on the floor and turn it on, here it goes！ To prevent the robot from falling off the stairs, the bottom of the robot usually is surrounded by many sensors. Our Maqueen Plus has 6 grayscale sensors, so it can totally meet the requirements. Let's make a sweeping robot with Maqueen Plus. Take the black line on the map as the edge of the stairs, and the robot will be only allowed to move within that area.





# Chapter 10

## Tour of Crossroad

Standing at a crossroad, Maqueen Plus is wondering which way he should go. Every road is so unique. Well, why not try all the roads? That sounds a good idea, right! Let' s help Maqueen Plus to start his tour of crossroad!

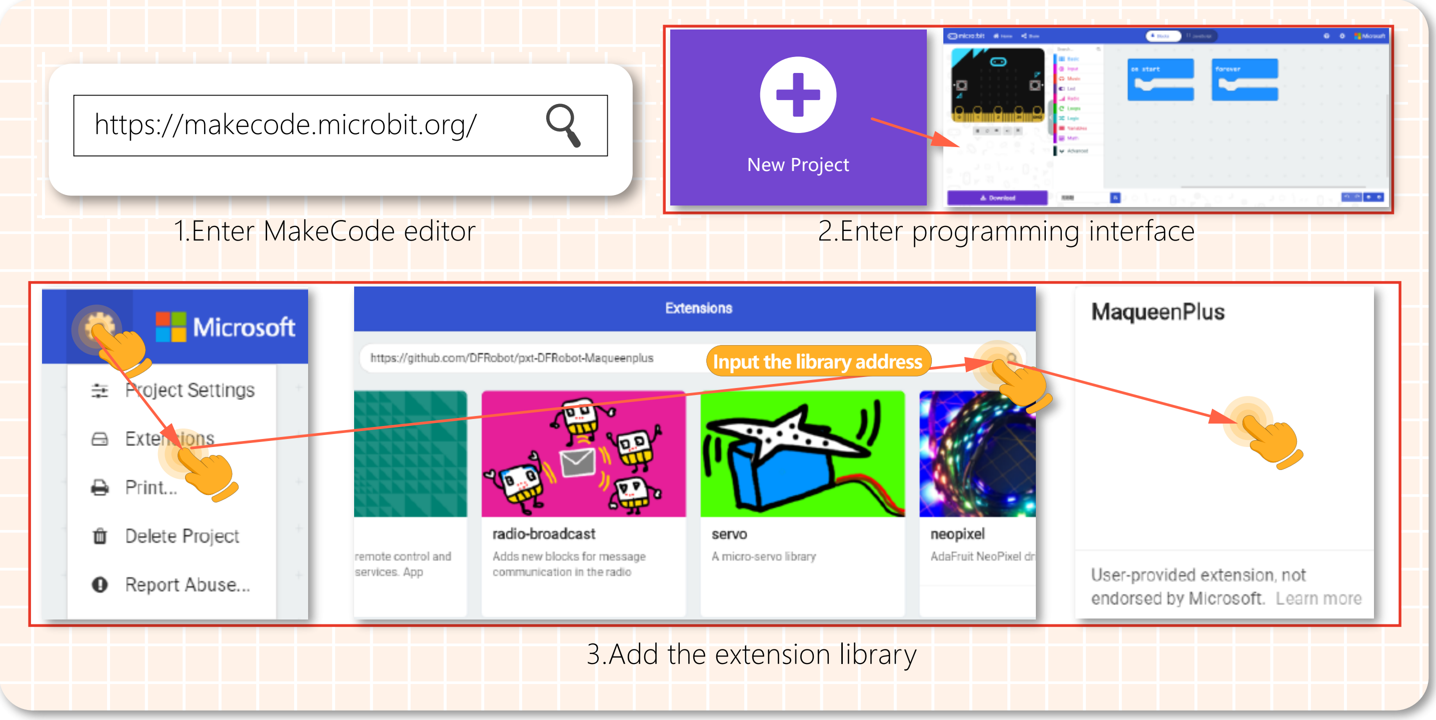
## Goal

Learn the use of multiple line-tracking sensors

## Hands-on Practice

### Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus


https://makecode.microbit.org/
New Project
Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
remote control and services. App
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

1.Enter MakeCode editor

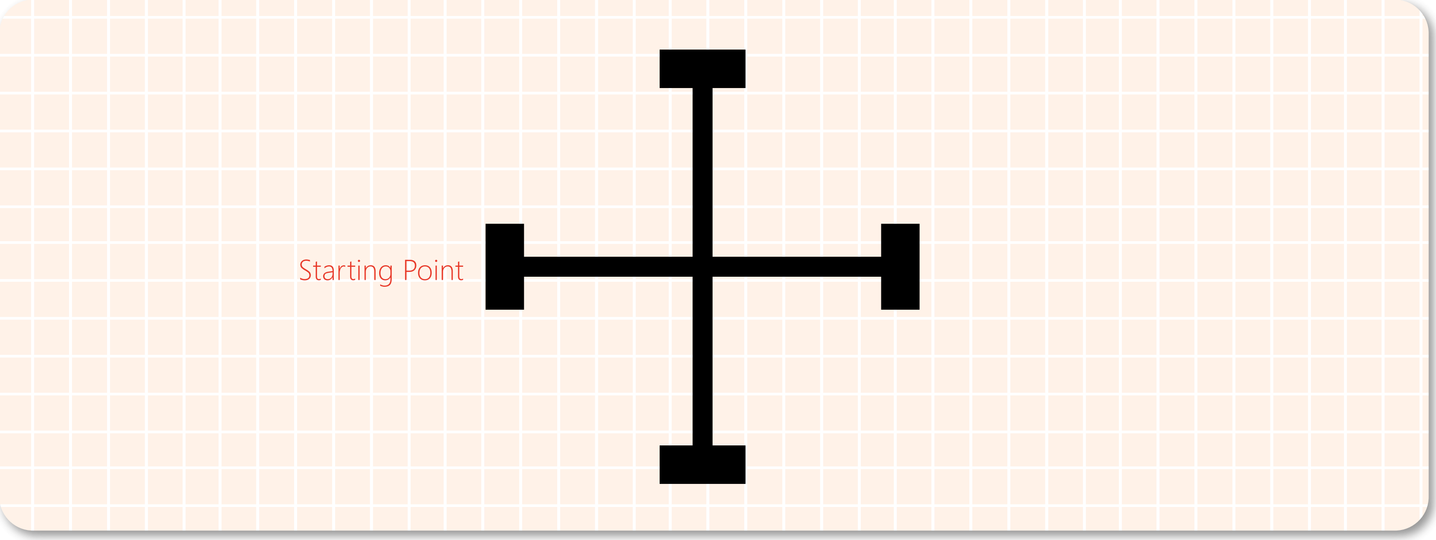
2.Enter programming interface

3.Add the extension library

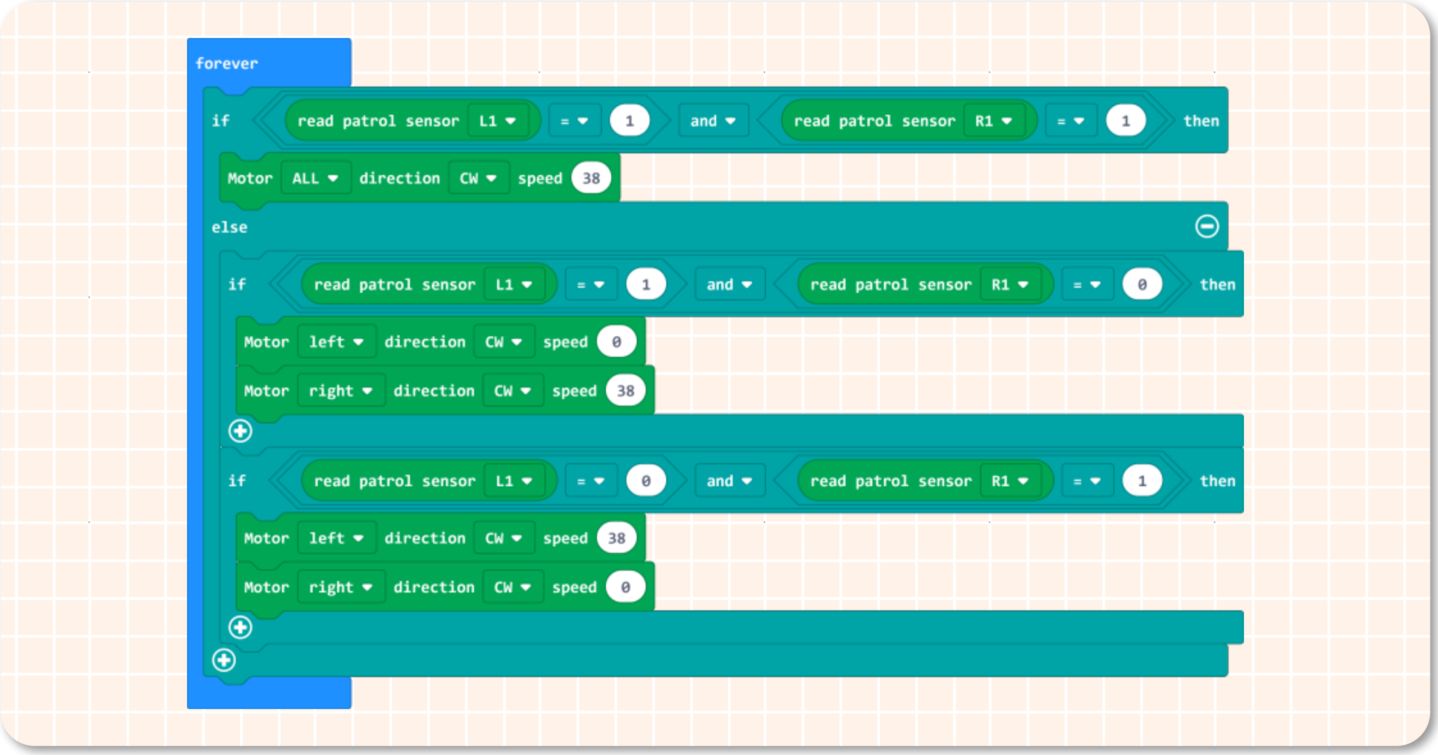
### Step 2 Programming

1.In this project, Maqueen Plus will try all roads of the crossroad, and then back to the starting point. How can we realize that by programming?

In the process of line-tracking, Maqueen Plus turns left/right at the intersection, and then turns around at the end of the road. Repeat this series of actions all the time

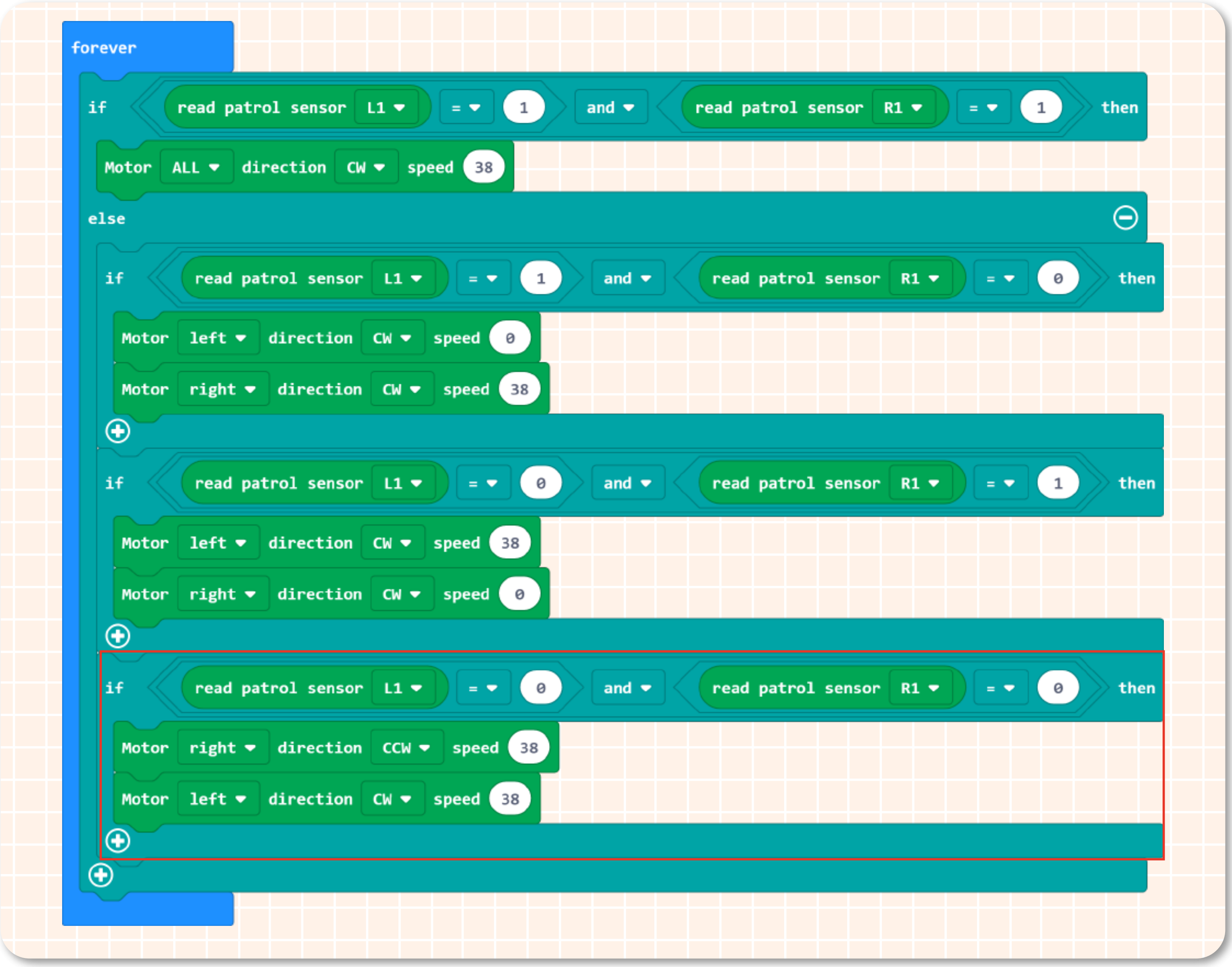


2.First,complete the program of line-tracking.



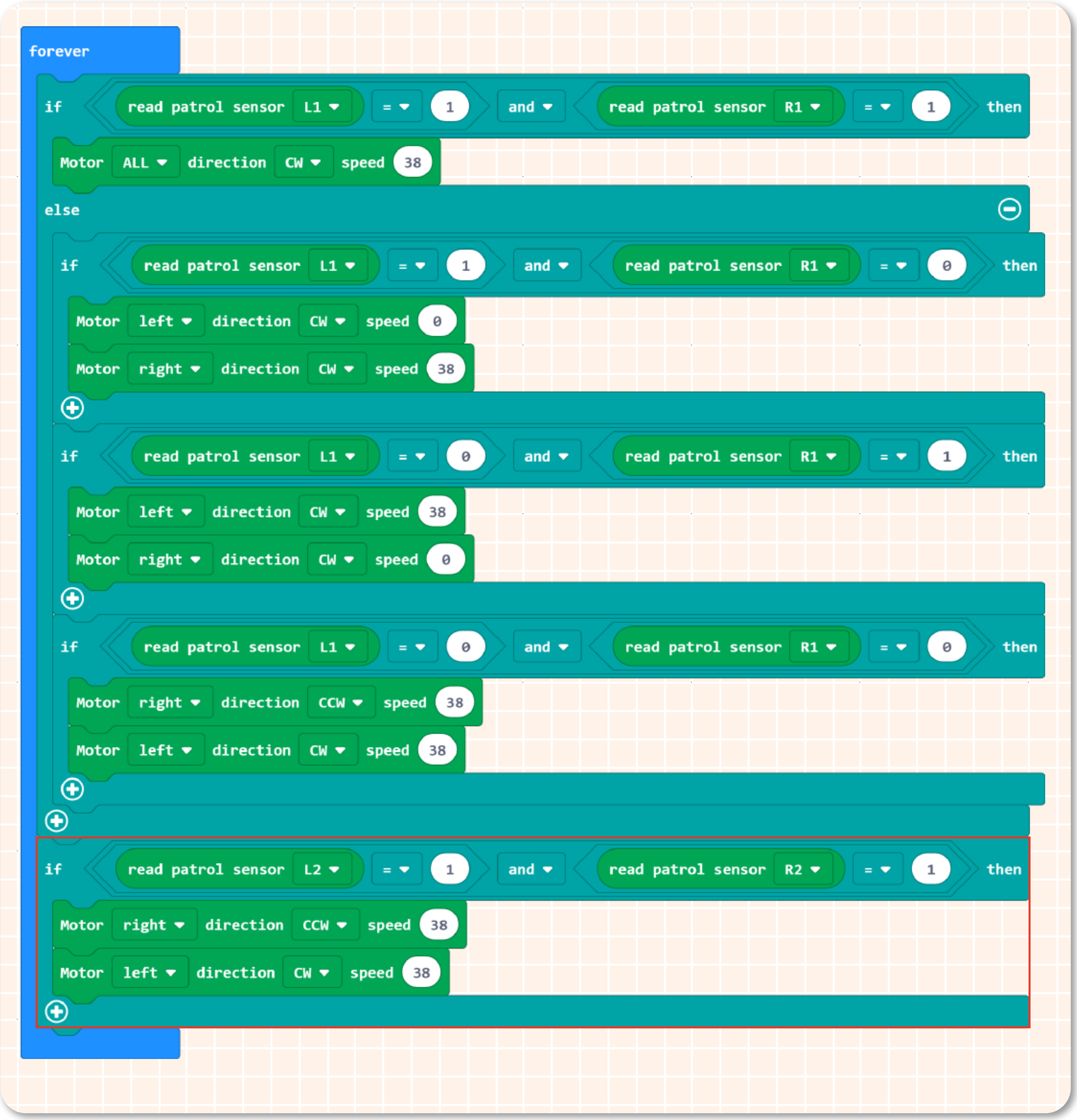
```
forever
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 1 then
    Motor ALL direction CW speed 38
  else
    if read patrol sensor L1 = 1 and read patrol sensor R1 = 0 then
      Motor left direction CW speed 0
      Motor right direction CW speed 38
    if read patrol sensor L1 = 0 and read patrol sensor R1 = 1 then
      Motor left direction CW speed 38
      Motor right direction CW speed 0
```

3.When no black line is detected by sensors L1 and R1, it means that the car drives out of the black track. Now let the Maqueen Plus spin around until the black track is found again.



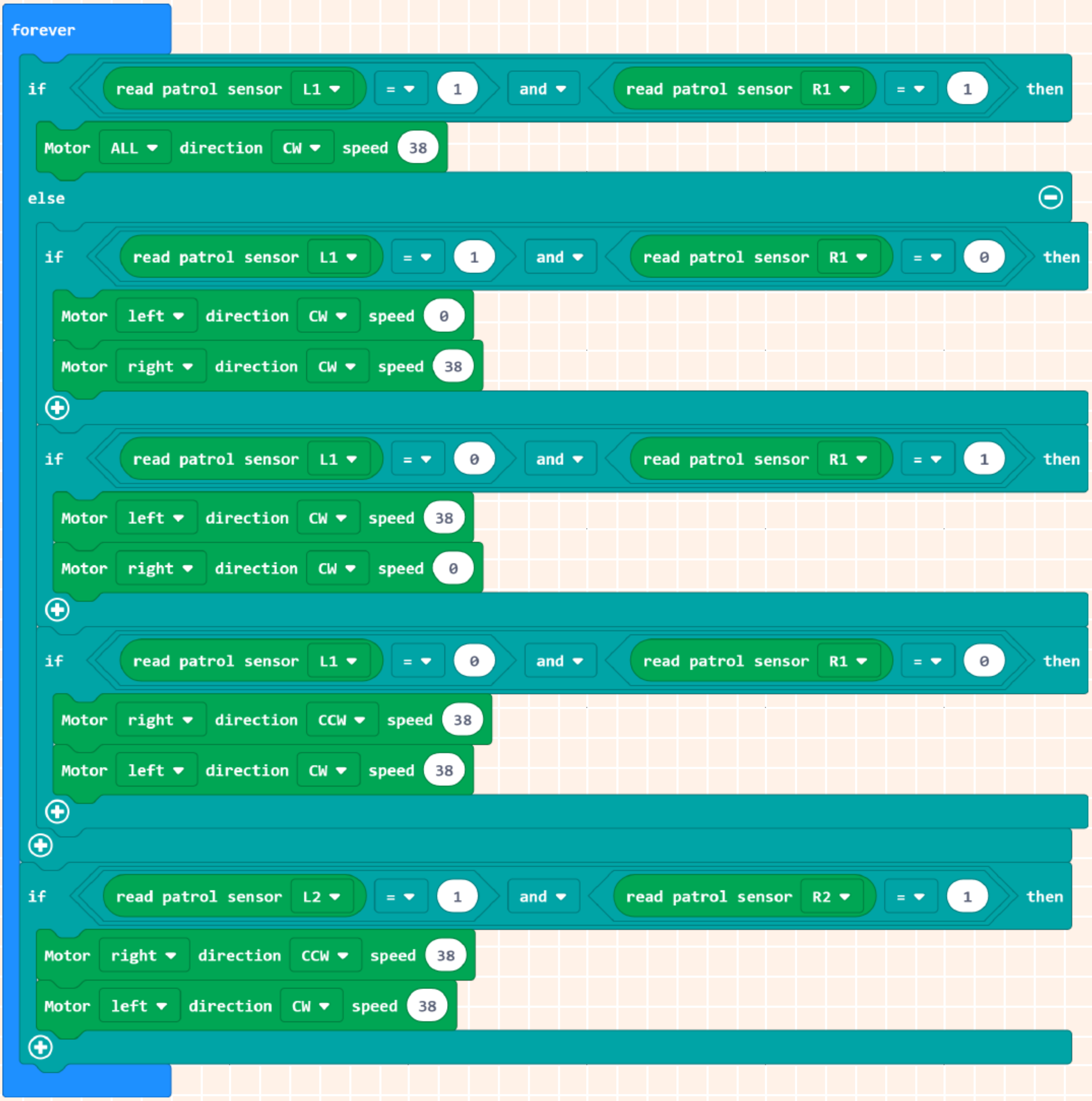
```
forever
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 1 then
    Motor ALL direction CW speed 38
  else
    if read patrol sensor L1 = 1 and read patrol sensor R1 = 0 then
      Motor left direction CW speed 0
      Motor right direction CW speed 38
    if read patrol sensor L1 = 0 and read patrol sensor R1 = 1 then
      Motor left direction CW speed 38
      Motor right direction CW speed 0
    if read patrol sensor L1 = 0 and read patrol sensor R1 = 0 then
      Motor right direction CCW speed 38
      Motor left direction CW speed 38
```

4.When the sensors L2 and R2 detected the black line, it means that the car has arrived the intersection or the end of the road, and the car needs to spin around to find the black track again.



```
forever
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 1 then
    Motor ALL direction CW speed 38
  else
    if read patrol sensor L1 = 1 and read patrol sensor R1 = 0 then
      Motor left direction CW speed 0
      Motor right direction CW speed 38

    if read patrol sensor L1 = 0 and read patrol sensor R1 = 1 then
      Motor left direction CW speed 38
      Motor right direction CW speed 0

    if read patrol sensor L1 = 0 and read patrol sensor R1 = 0 then
      Motor right direction CCW speed 38
      Motor left direction CW speed 38

  if read patrol sensor L2 = 1 and read patrol sensor R2 = 1 then
    Motor right direction CCW speed 38
    Motor left direction CW speed 38
```

5.The complete program is shown below.

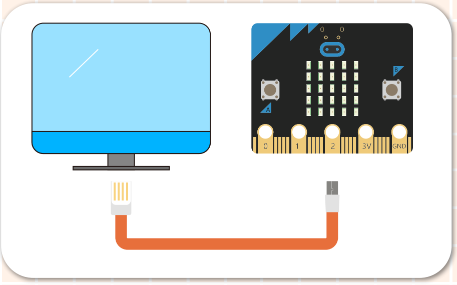


```
forever
  if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 1  then
    Motor ALL direction CW speed 38
  else
    if  read patrol sensor L1 = 1  and  read patrol sensor R1 = 0  then
      Motor left direction CW speed 0
      Motor right direction CW speed 38
    ⊕
    if  read patrol sensor L1 = 0  and  read patrol sensor R1 = 1  then
      Motor left direction CW speed 38
      Motor right direction CW speed 0
    ⊕
    if  read patrol sensor L1 = 0  and  read patrol sensor R1 = 0  then
      Motor right direction CCW speed 38
      Motor left direction CW speed 38
    ⊕
  ⊕
  if  read patrol sensor L2 = 1  and  read patrol sensor R2 = 1  then
    Motor right direction CCW speed 38
    Motor left direction CW speed 38
  ⊕
```

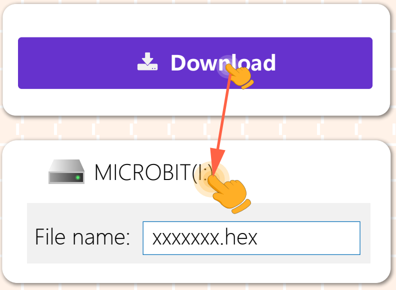
Note: if the Maqueen Plus car turns left or right too much, you can change its motor speed to adjust.

6.Name your project as "Tour of crossroad" and save it.

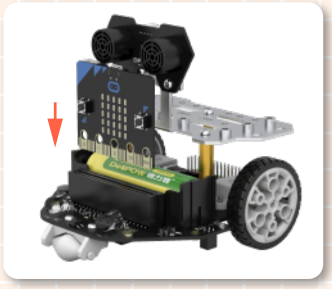
## Step 3 Download Program



1.Connect to computer



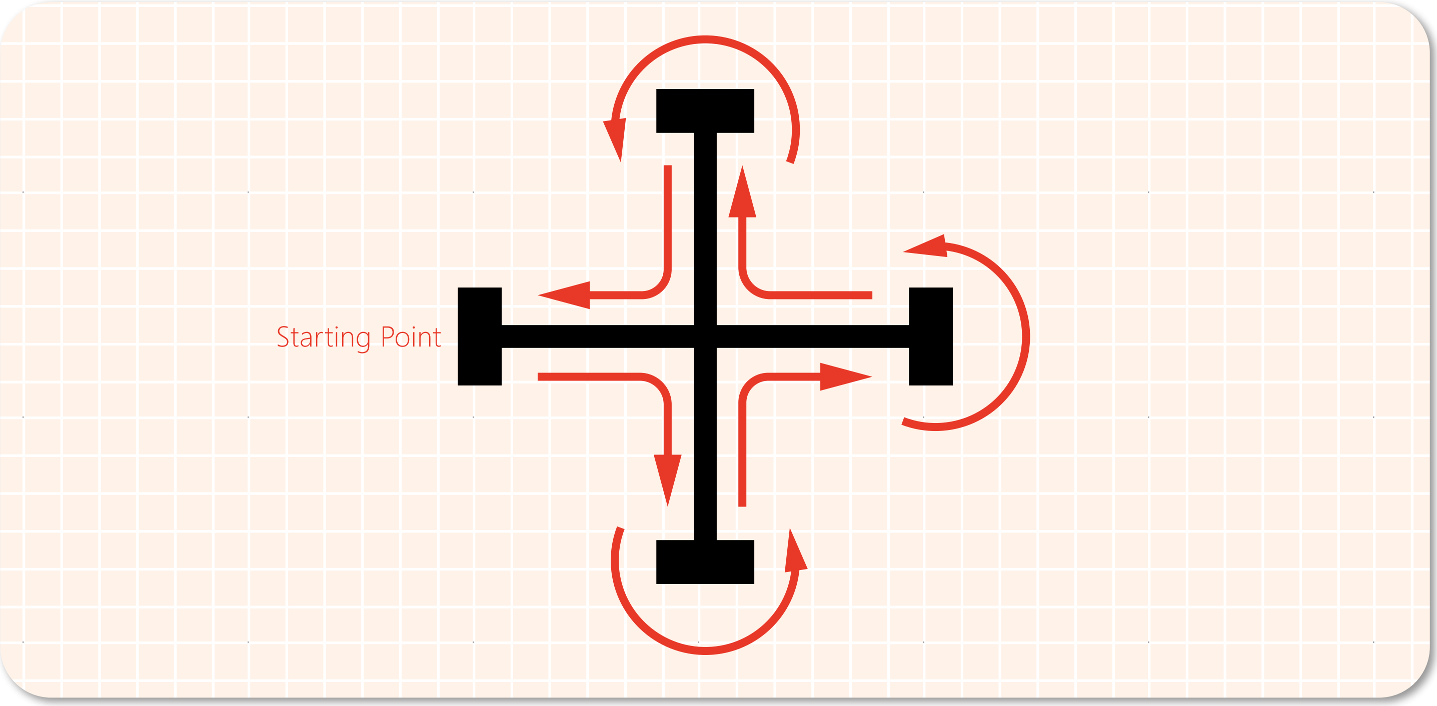
2.Download program



3.Plug in micro:bit

### Step 4 Effect Display

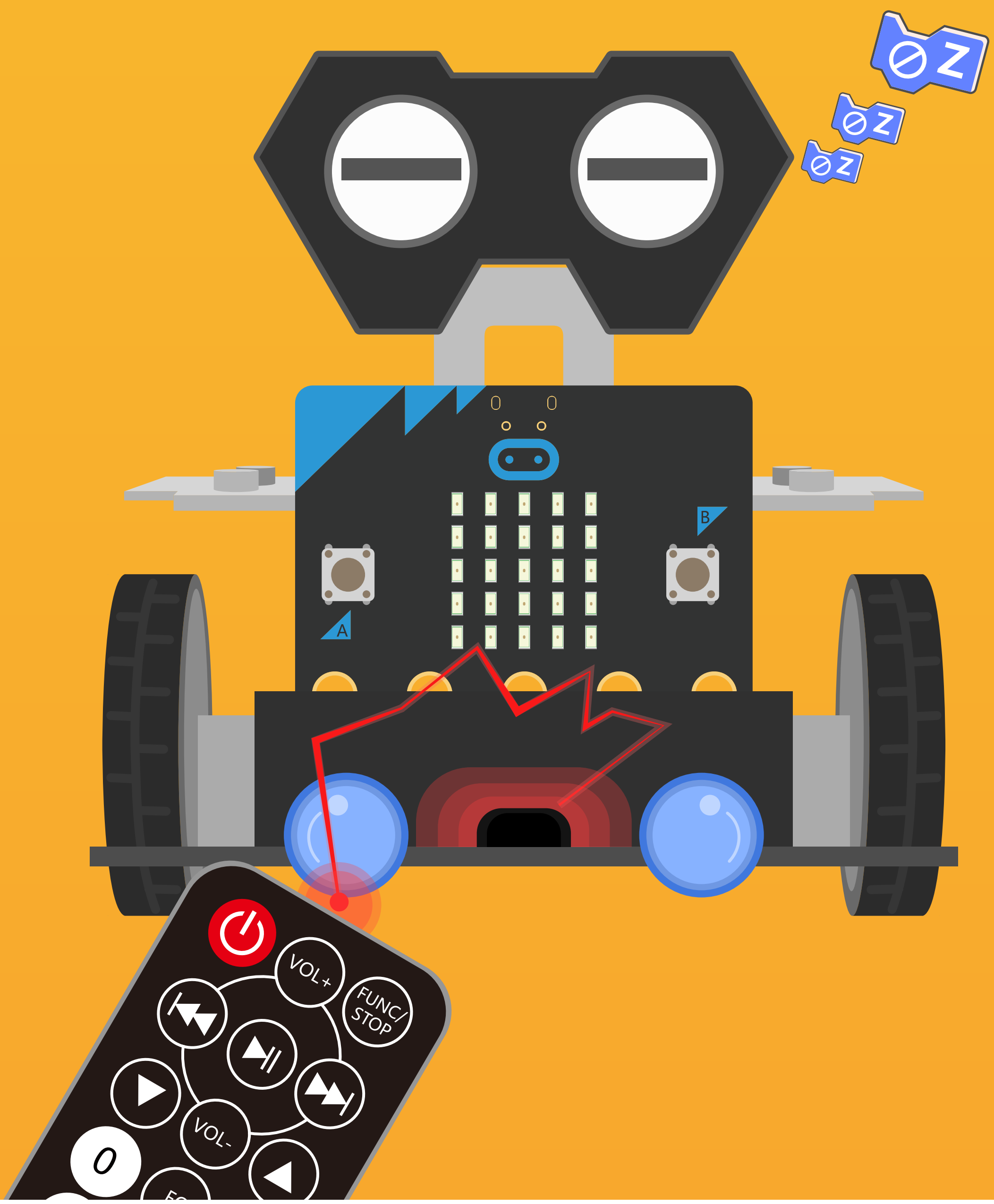
When completed all the steps, put Maqueen Plus on the crossroad map, turn on its power switch.



## Think & Explore

When Maqueen Plus arrives at the intersection, he will turn right, well, now let's make its right RGB LED flash while turning right, and then the both RGB LEDs flash when turning around.

Tip: add RGB blocks in the program above.



# Chapter 11

## IR-controlled Robot

The invention of remote controller allows people to operate devices from a certain distance, which brings a lot of convenience for our daily life. IR remote controller is the most commonly-used one. In this chapter, we will use an IR remote controller with our Maqueen Plus to make an "IR-controlled Robot".

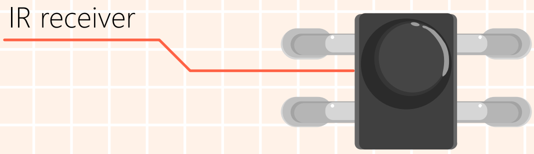
## Goal

1.Learn how to use function block

2.Learn how to use IR remote controller to control motor

## Electronic Component

### IR receiver

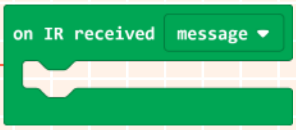


Receive a coded infrared signal and convert it into an electric signal.

## Command Learning

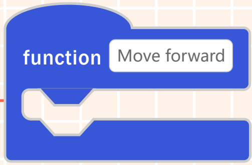
### Block Brief

IR receiver


on IR received message ▾

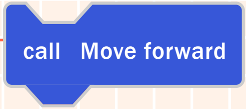
Receive and read the infrared value.

Create a function


function Move forward

When a continuous action needs to occur multiple times in the program, in order to make the program more clear, we need to define a sub-function.

Call a function


call Move forward

Drag the sub-function into the program to call it.

## Hands-on Practice

We will use an IR remote controller to operate our Maqueen Plus, so we have to get key value first. The decimal number of each key on the remote controller is shown below.

VOL+
FUNC/STOP
VOL-
0
EQ
ST/REPT
1
2
3
4
5
6
7
8
9

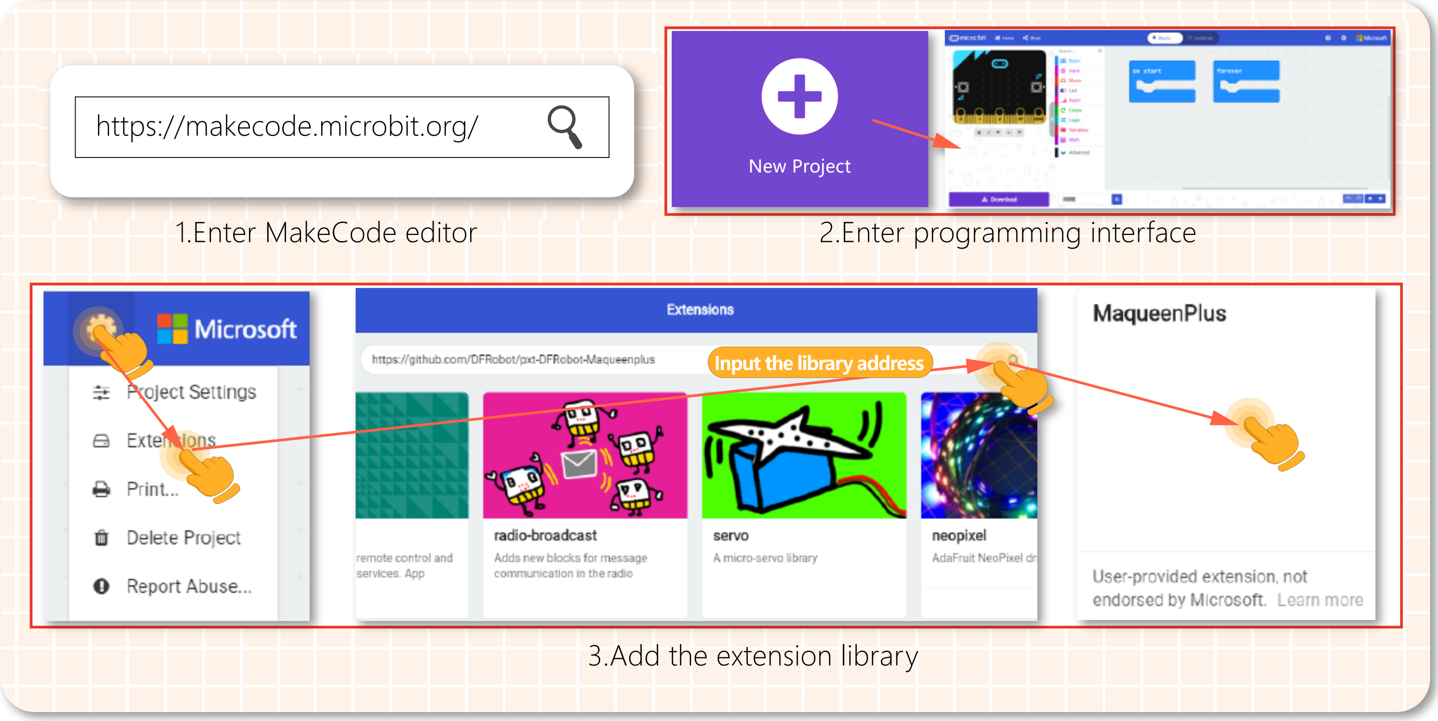
| Key | Value |
|---|---|
| Red key | 0 |
| VOL+ | 1 |
| FUNC/STOP | 2 |
| Left arrow | 4 |
| Pause | 5 |
| Right arrow | 6 |
| Down arrow | 8 |
| VOL- | 9 |

| Key | Value |
|---|---|
| Up arrow | 10 |
| 0 | 12 |
| EQ | 13 |
| ST/REST | 14 |
| 1 | 16 |
| 2 | 17 |
| 3 | 18 |
| 4 | 20 |

| Key | Value |
|---|---|
| 5 | 21 |
| 6 | 22 |
| 7 | 24 |
| 8 | 25 |
| 9 | 26 |

## Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus


https://makecode.microbit.org/
New Project
Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

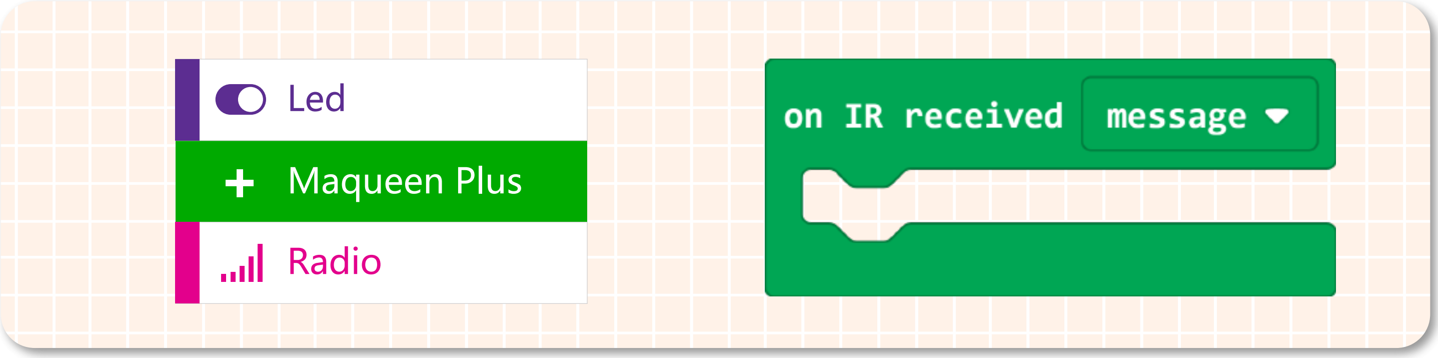
1.Enter MakeCode editor

2.Enter programming interface

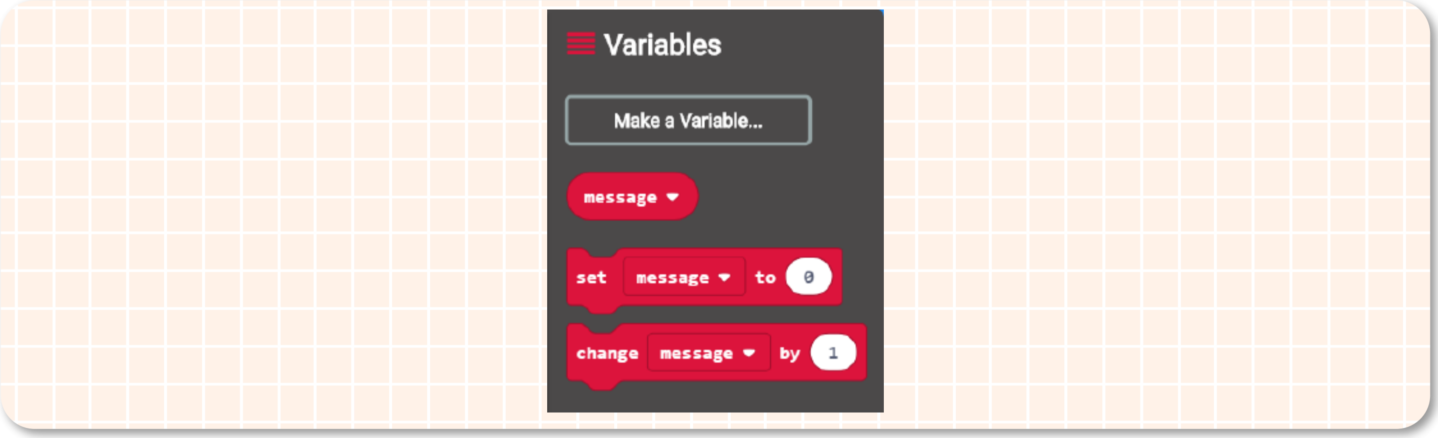
3.Add the extension library

## Step 2 Programming

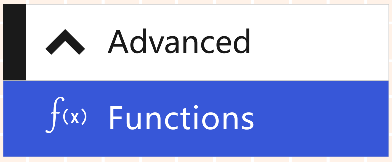
1.Drag the IR receive block to the editing section.



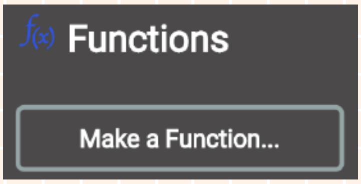
After we dragged the IR receive block to the editing section, there will be a variable named "message" appearing in the variable command section for storing key-value of IR remote controller.



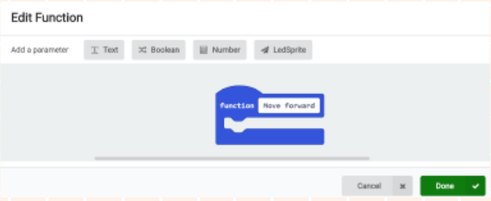
2.Create a function and name it as "Move forward".



① Click "Advanced"->"Function";



② Click "Make a function";



③ Name the function as "Move forward";

④ Click "OK".

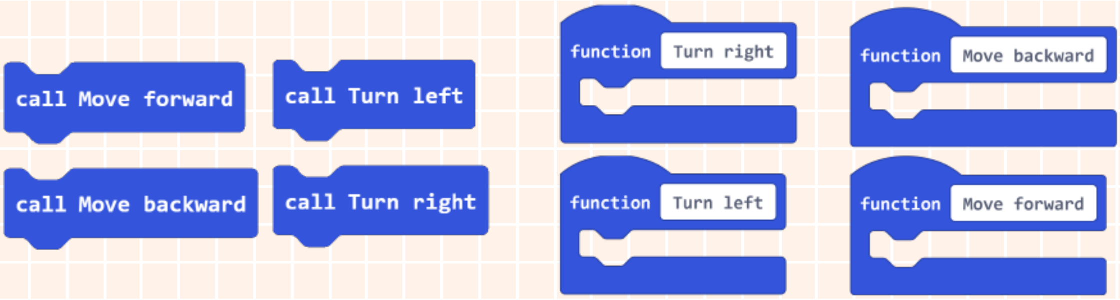
## Knowledge Expansion

### What is a function?

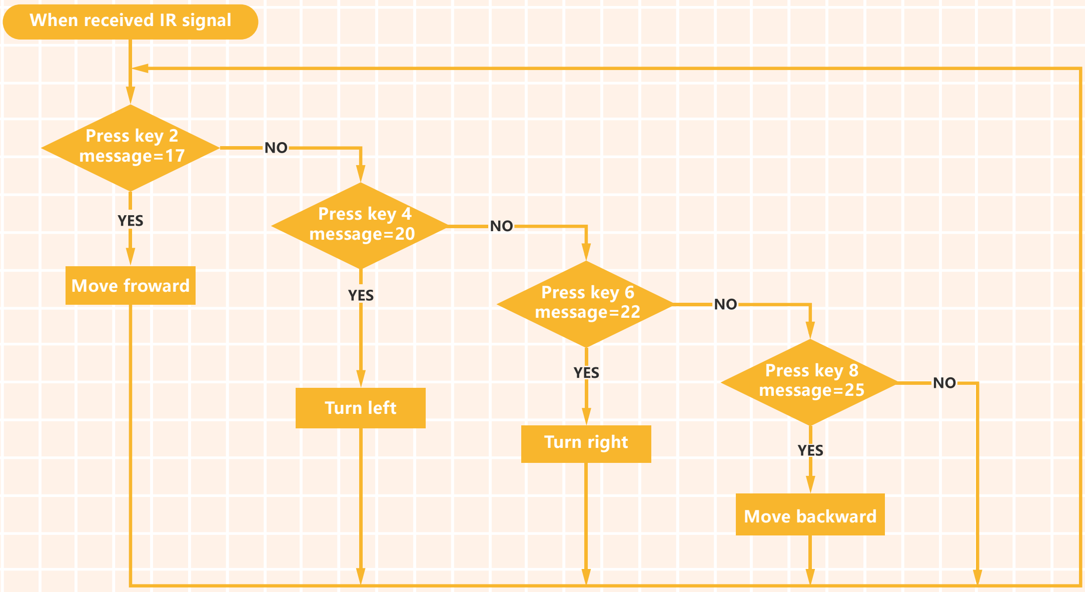
A function, also known as procedure or subroutine, can be defined as the organized block of reusable code which can be called whenever required.

Generally, a large program can be divided into many basic building blocks and each block can realize a specified function. A function can be called multiple times by other main functions and sub-functions, which not only reduces the workload of rewriting the program segment, but also improves the utilization of the program.

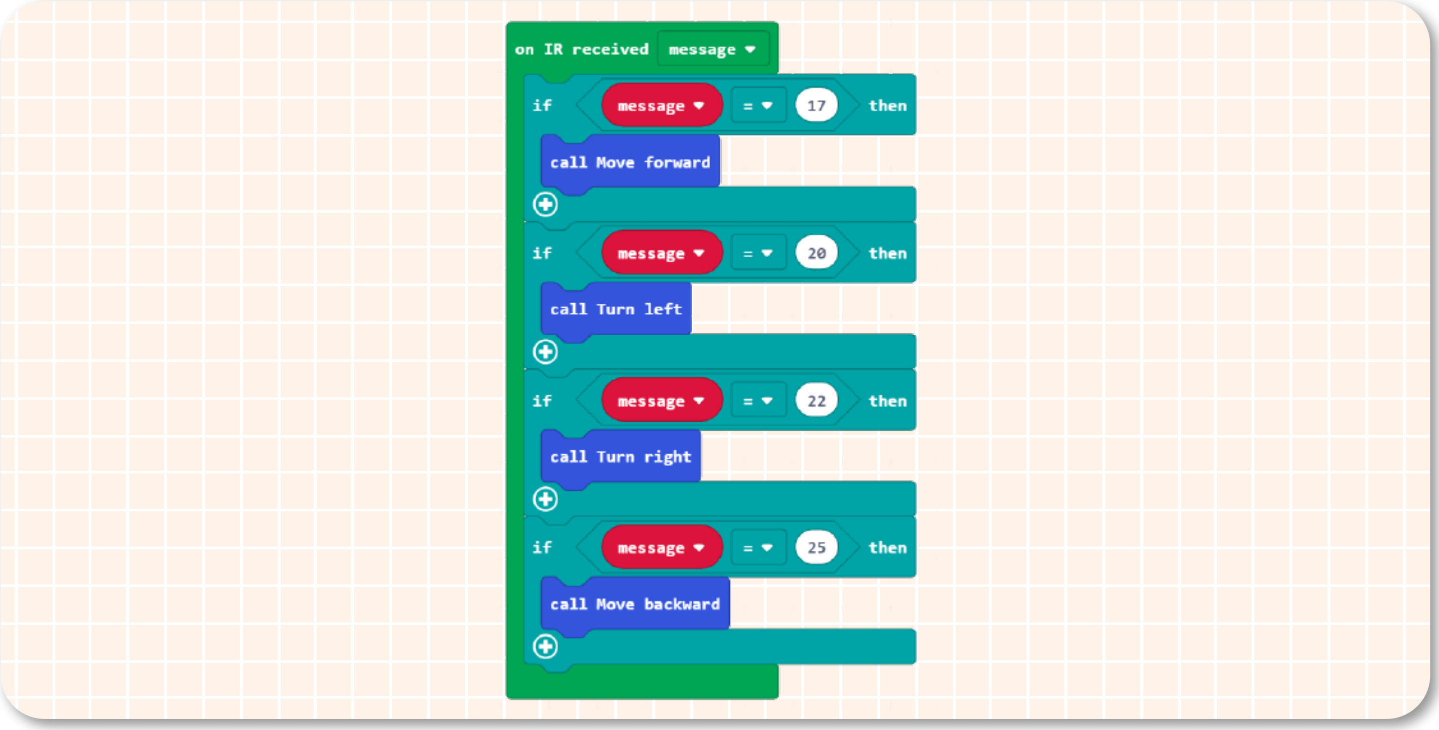
3.Create functions "Move backward", "Turn left" and "Turn right" in the same way above. The customized function will be shown in the editing section and the command block section.



4.How can we use the key 2, 4, 6, and 8 on the IR remote controller to operate our Maqueen Plus car? Let's draw a flowchart to analyze this question:

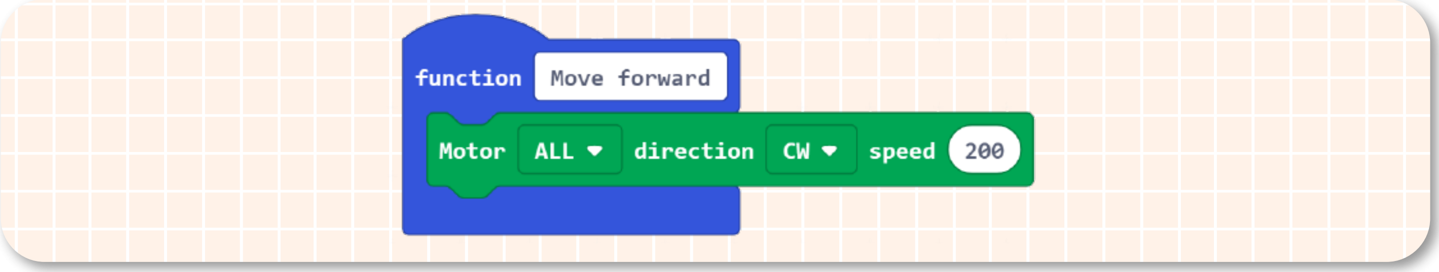


5.The flowchart above shows that we have to press the related key first, and then judge if the key value meets the condition. When the key value "message=17", call the function "Move forward", and so on.



```
on IR received message
    if message = 17 then
        call Move forward
    if message = 20 then
        call Turn left
    if message = 22 then
        call Turn right
    if message = 25 then
        call Move backward
```

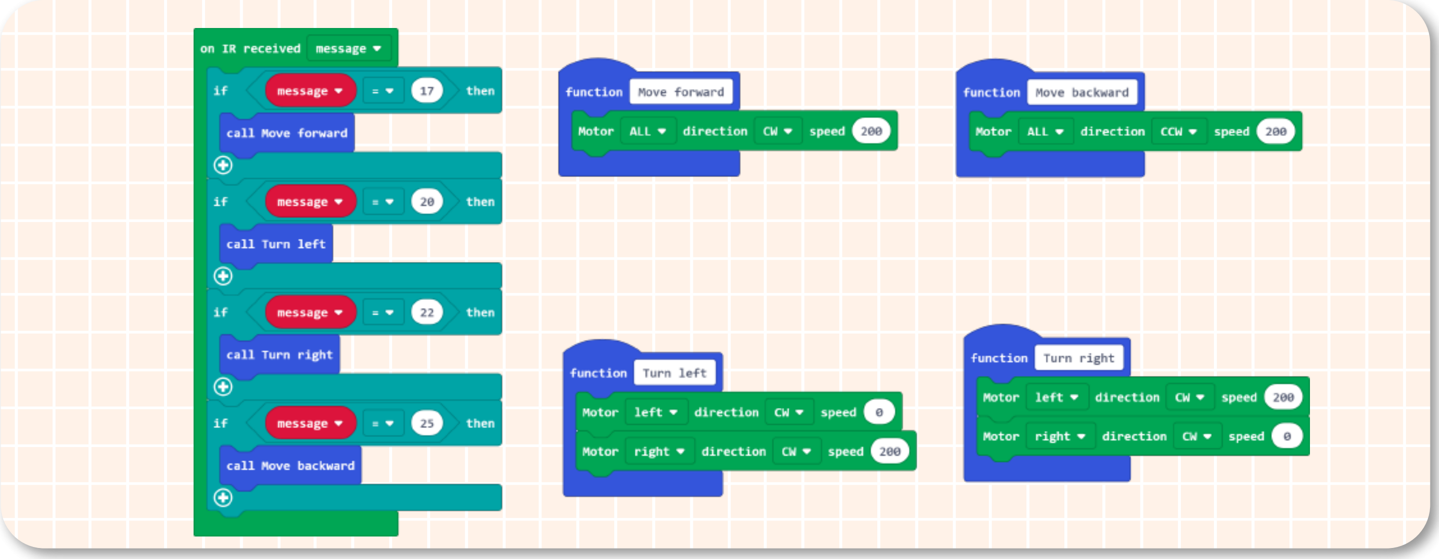
6.The above program is just a framework. The detailed operations need to be implemented in the functions. For example, press key 2, the car move forward. How to achieve that? Well, it's easy, just add a motor control block inside the move forward function.



```
function Move forward
    Motor ALL direction CW speed 200
```

As long as we have a clear logic, the realization of the program will be not so hard. The complete program is shown below:

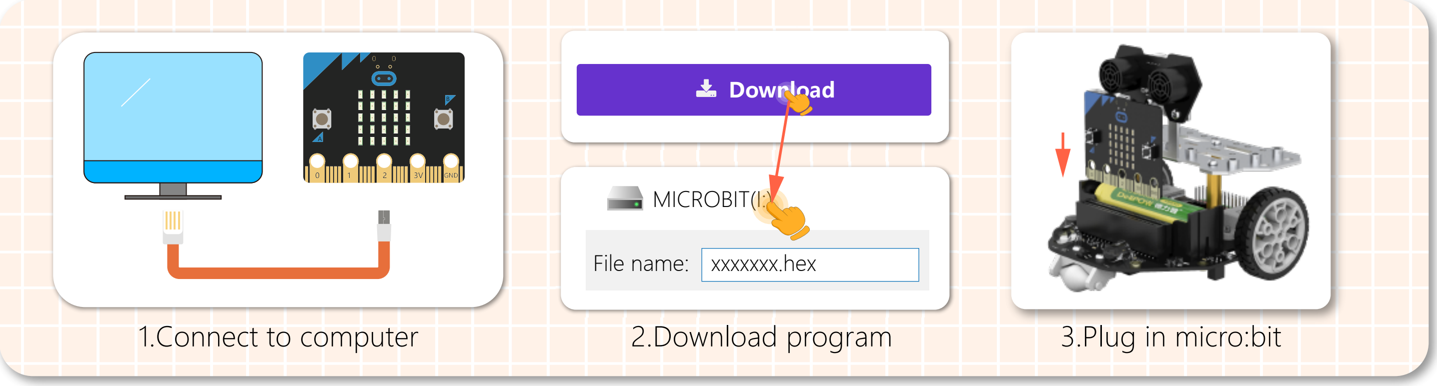
7.The complete program is shown below.



```
on IR received message
    if message = 17 then
        call Move forward
    if message = 20 then
        call Turn left
    if message = 22 then
        call Turn right
    if message = 25 then
        call Move backward

function Move forward
    Motor ALL direction CW speed 200

function Move backward
    Motor ALL direction CCW speed 200

function Turn left
    Motor left direction CW speed 0
    Motor right direction CW speed 200

function Turn right
    Motor left direction CW speed 200
    Motor right direction CW speed 0
```

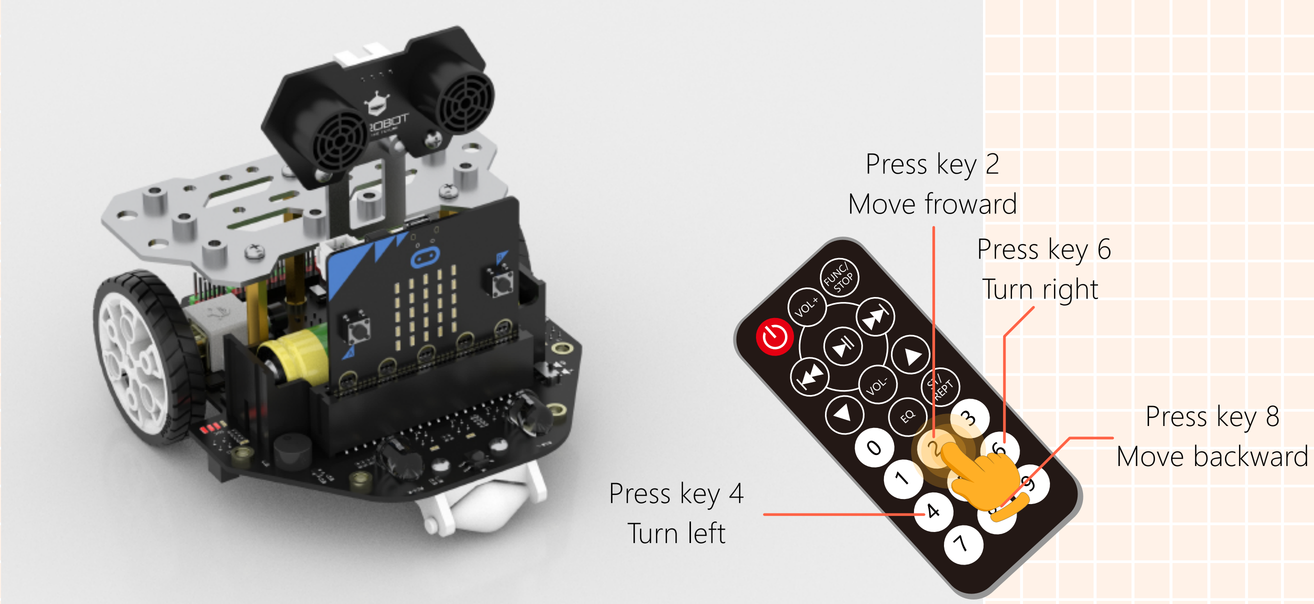
8.Name your project as "IR-controlled robot" and save it.

## Step 3 Download Program



## Step 4 Effect Display

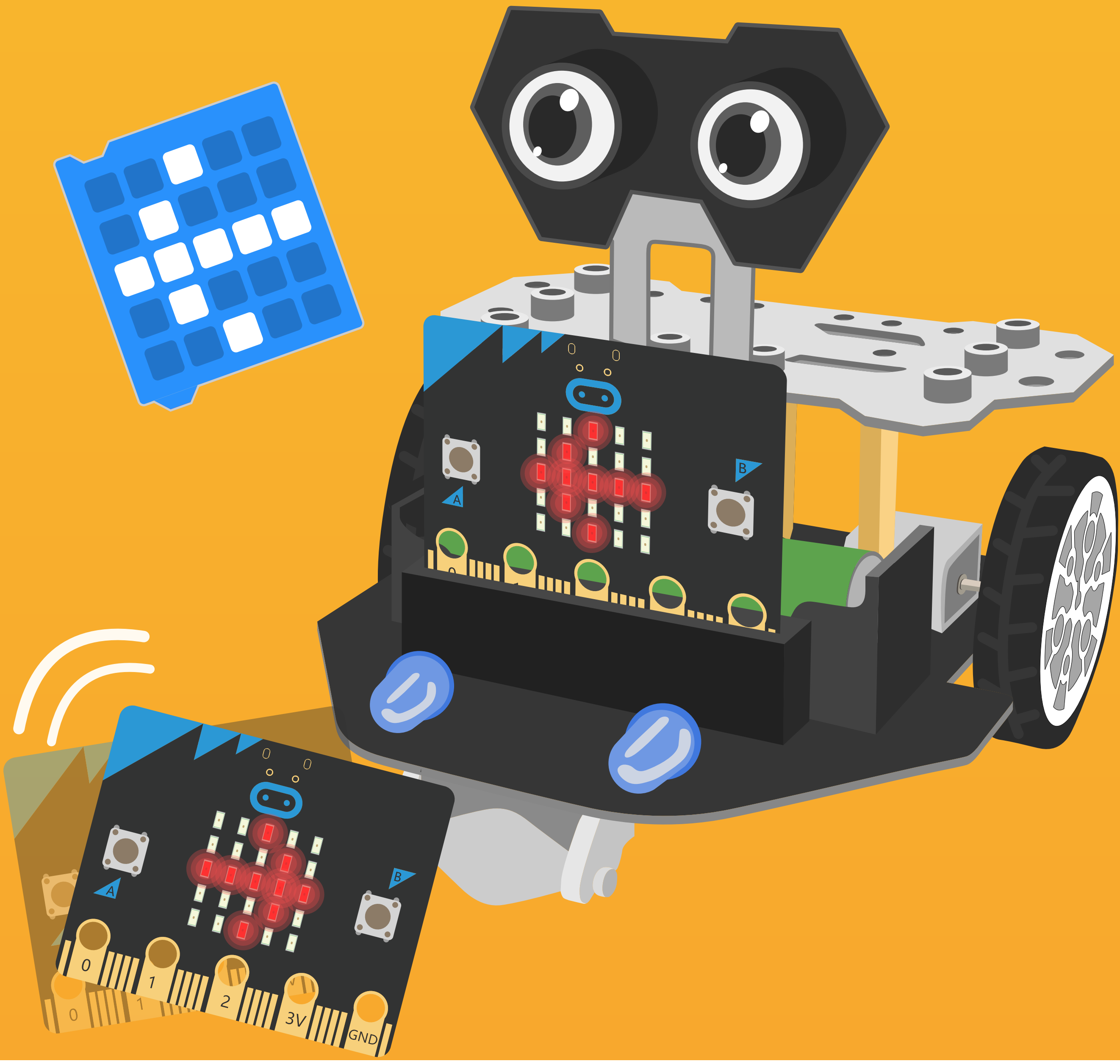
After completing all the above steps, use the IR remote controller to operate our Maqueen Plus.



## Think & Explore

We may find that once we enabled the Maqueen Plus to move, it won't stop until we turn off its power switch, which could be inconvenient for us to operate. Now here is a task for you: use the red button on the remote controller to make Maqueen Plus stop.

Tip: create a function to stop the motor!



# Chapter 12

## Motion Sensing Robot

Have you ever tried motion-sensing games, like the racing car? This kind of game requires us to control the movement of the car on the screen by changing the direction of the remote controller. It is extremely exciting! Our Maqueen Plus can also realize motion-sensing since the accelerometer on the micro:bit can detect the orientation of the board. With accelerometer and radio communication functions, we can make a similar motion-sensing game using Maqueen Plus.

## Goal

1.The basics of radio communication
2.Learn how to use accelerometer sensor

## Electronic Component

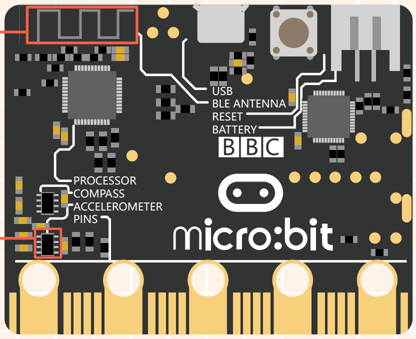
**Figure of the Bluetooth and the acceleration sensor Figure of the Bluetooth and the acceleration sensor**

Radio Communication

Allow two or more micro:bits communicate with each other, receive and send radio signals

Accelerometer

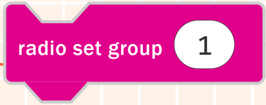
The accelerometer on the micro:bit detects the acceleration in 3 planes: x, y and z.



## Command Learning

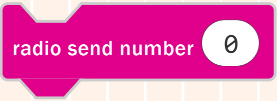
**Block Brief**

Radio set group


radio set group 1

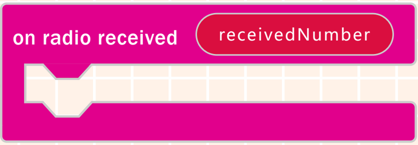
Sets the group id for radio communications. A micro:bit can only listen to one group ID at any time.

Radio send number


radio send number 0

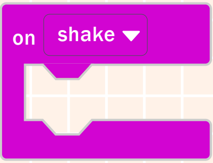
Broadcasts a number via  radio to any connected micro:bit in the group.

Radio received


on radio received receivedNumber

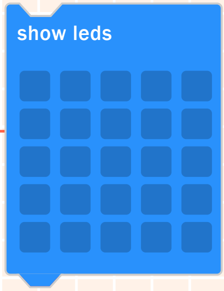
Register code to run when the radio receives a number.

Gesture


on shake ▾

Do something when a gesture is done (11 gestures)

Show LEDs


show leds

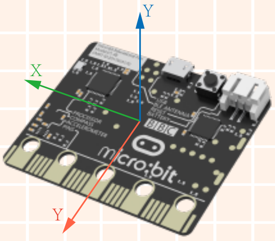
Draws an image on the LED screen.

## What is an accelerometer sensor?

An accelerometer is a device that can detect the change of an object's acceleration. The on-board accelerometer of micro:bit can sense motion of the micro:bit, including its movement, angle, gesture and so on.

### Introduction of the acceleration sensor

The micro: bit comes with a three-axis acceleration sensor that can detect gravity accelerations in three directions: x, y, and z. The measured value on each axis should be positive or negative. When the reading of one axis is 0, it indicates that the acceleration sensor is parallel to that axis. The different attitudes of the micro: bit is determined by calculating the vector sum of the three axes of the acceleration sensor x, y, and z. A vector is a quantity that has magnitude and direction, and vector sums refer to the sum of direction and magnitude.



# Hands-on Practice

## Motion Sensing Robot---Transmitting End

Before we start programming, let's analyze how do we use the accelerometer sensor in this project.

The movement of the robot car is controlled by the gesture of micro:bit. When micro:bit logo up, an "Up arrow"-shows on the LED screen, and then the car moves forward; When logo down, show "Down arrow" and the car moves backward; tile left, show "left arrow" and the car turns left; tile right, show "right arrow" and the car turns right.

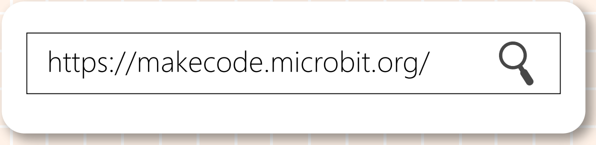
Tip: to realize radio communication, we need two micro:bits here. One for sending out signal (Transmitting end), one for receiving signal (Receiving end).

## Step 1 Create a New Project

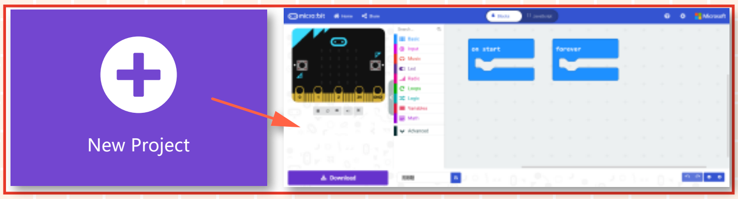
1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.

2.Click "new project" to enter MakeCode programming interface.

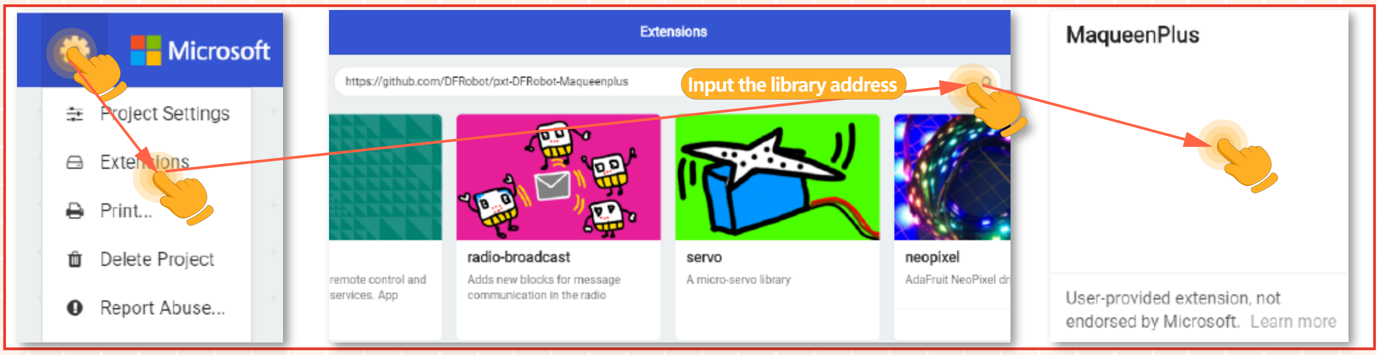
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus


https://makecode.microbit.org/

1.Enter MakeCode editor



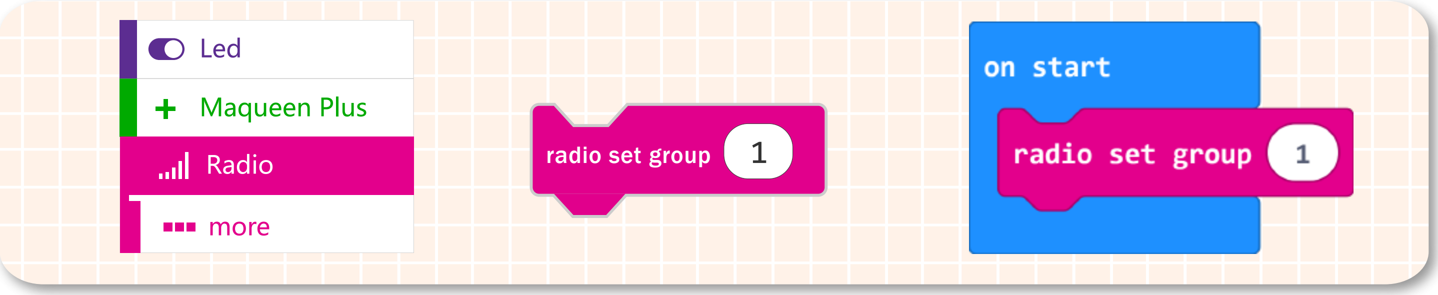
2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

3.Add the extension library

## Step 2 Programming

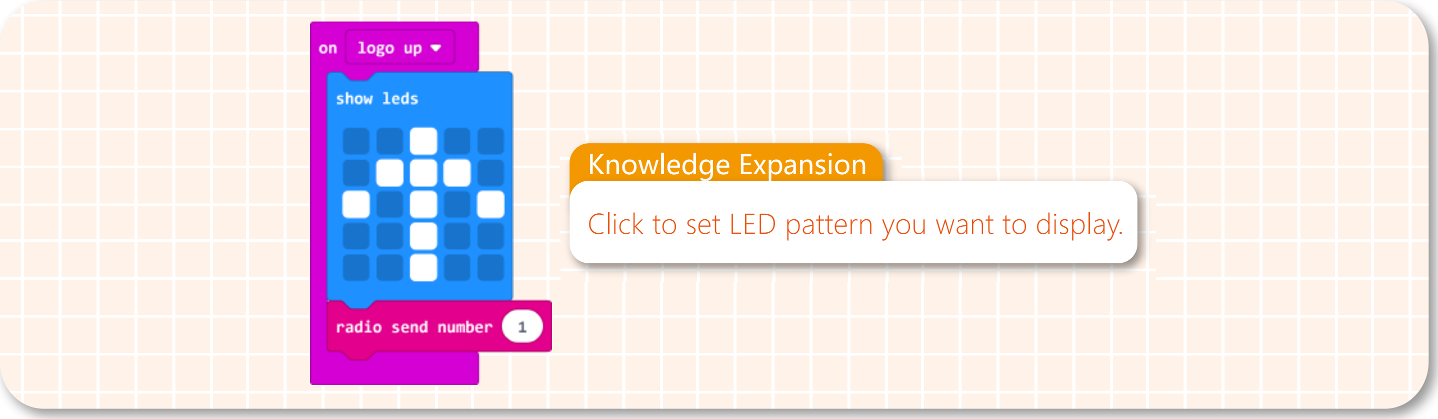
1.Enable the radio communication, set the radio group to 1.



```
radio set group 1
```

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

2.When the micro:bit log up, show an “Up arrow” in its LED screen, and send data 1 via radio to the Maqueen Plus.

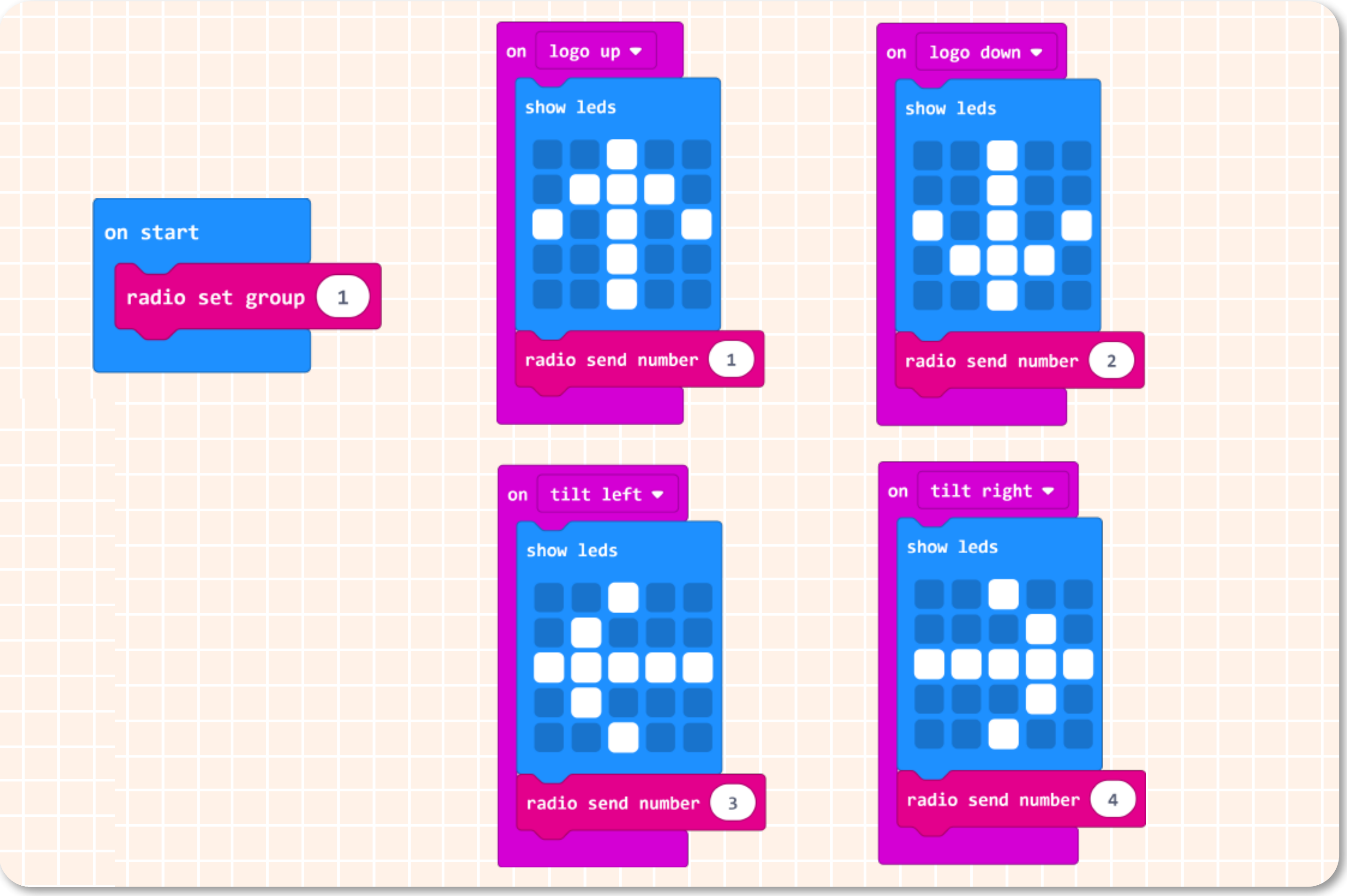


```
on logo up
    show leds
        . . # . .
        . # # # .
        # . # . #
        . . # . .
        . . # . .
    radio send number 1
```

Knowledge Expansion

Click to set LED pattern you want to display.

3.The programs for the rest actions of micro:bit are in much the same way. The whole program for the transmitting end is shown below:



```
on start
    radio set group 1

on logo up
    show leds
        . . # . .
        . # # # .
        # . # . #
        . . # . .
        . . # . .
    radio send number 1

on logo down
    show leds
        . . # . .
        . . # . .
        # . # . #
        . # # # .
        . . # . .
    radio send number 2

on tilt left
    show leds
        . . # . .
        . # . . .
        # # # # #
        . # . . .
        . . # . .
    radio send number 3

on tilt right
    show leds
        . . # . .
        . . . # .
        # # # # #
        . . . # .
        . . # . .
    radio send number 4
```

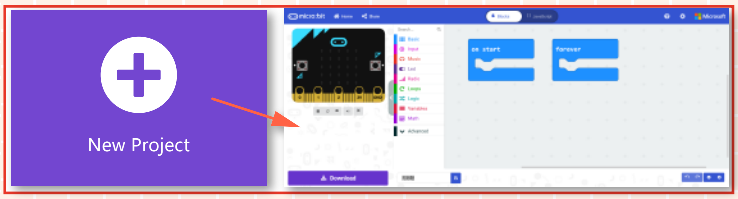
4.Name your project as “Motion sensing robot-Transmitting end” and download it into the micro:bit of transmitting end.

## Motion Sensing---Receiving End:

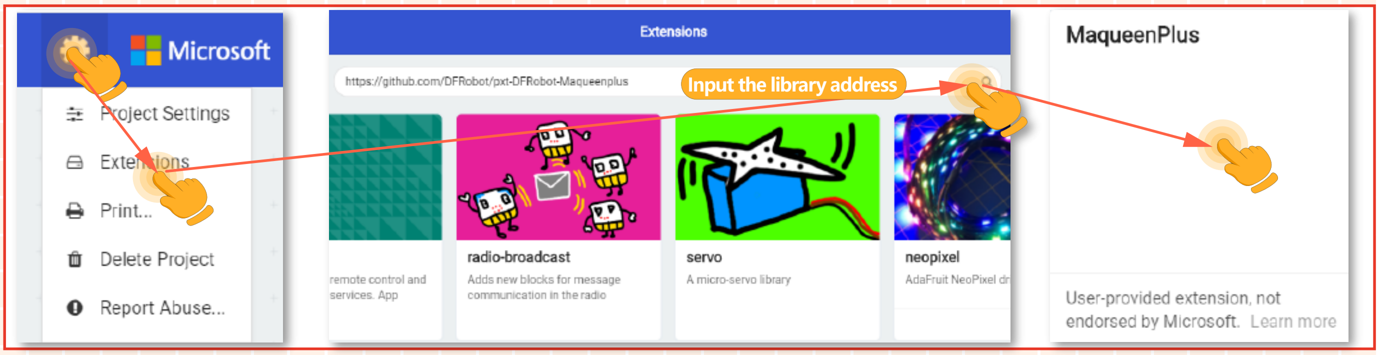
### Step 1 Create a new project

https://makecode.microbit.org/

1.Enter MakeCode editor



2.Enter programming interface



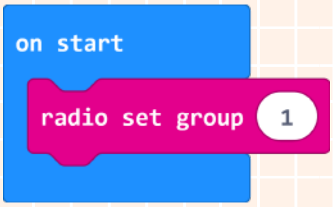
Input the library address

https://github.com/DFRobot/pxt-DFRobot-Maqueenplus

3.Add the extension library

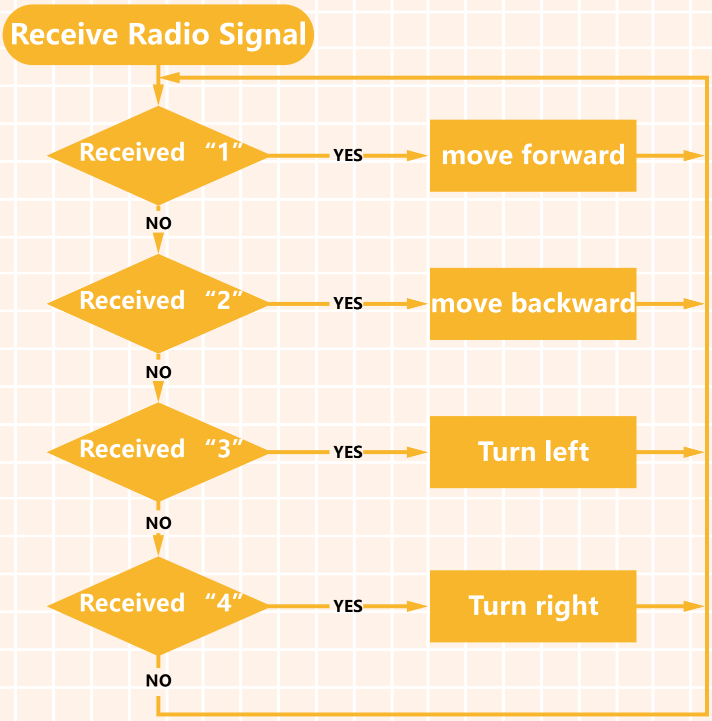
### Step 2 Programming

1.Enable radio communication function, set the radio group to 1. (The radio group of transmitting end should be in the same with that of the receiving end.)

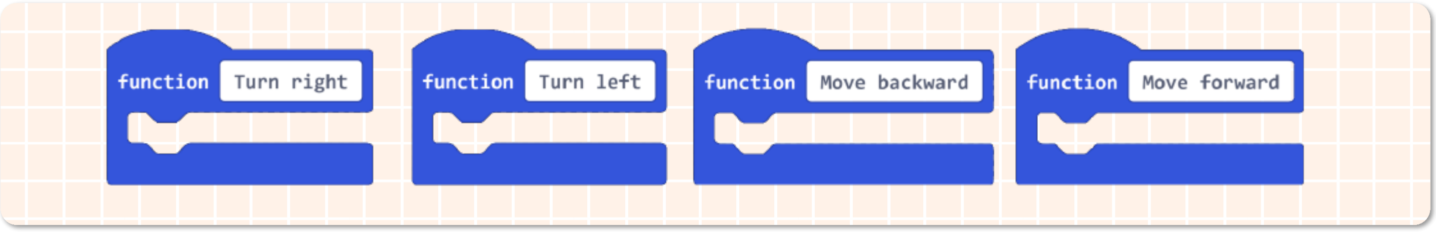


```
on start
  radio set group 1
```

2.Program Maqueen Plus to react according to the received signal.

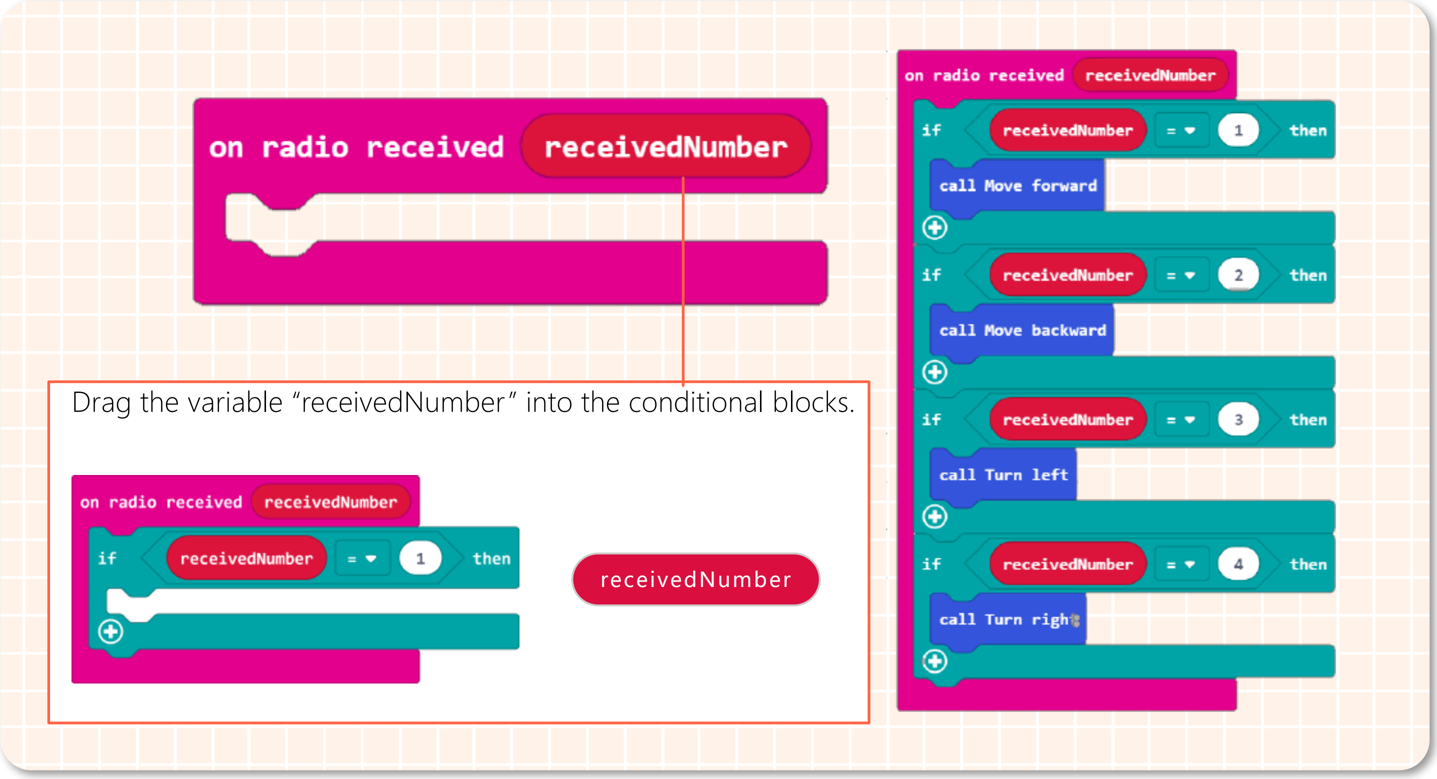


3.Create four functions “move forward”, “move backward”, “Turn left”, “Turn right”.



```
function Turn right

function Turn left

function Move backward

function Move forward
```

4.We have to use radio receive block to control the Maqueen Plus do different reactions according to the received data.



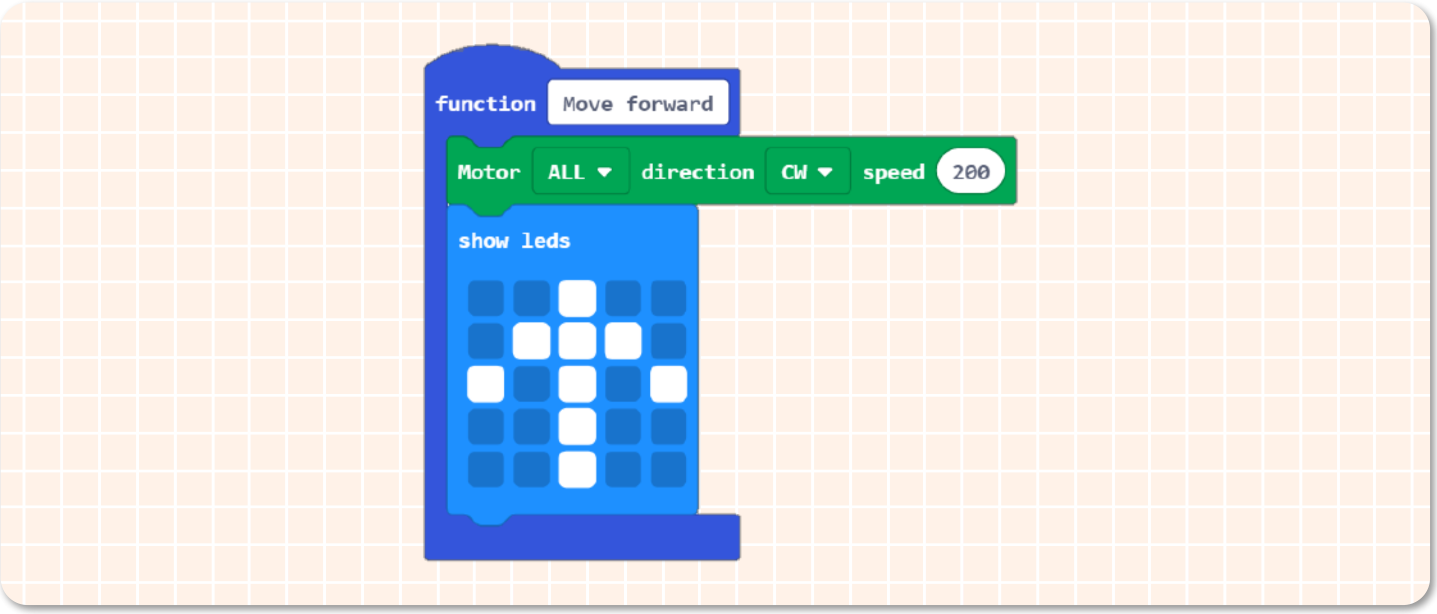
```
on radio received receivedNumber
```

Drag the variable “receivedNumber” into the conditional blocks.

```
on radio received receivedNumber
    if receivedNumber = 1 then
```

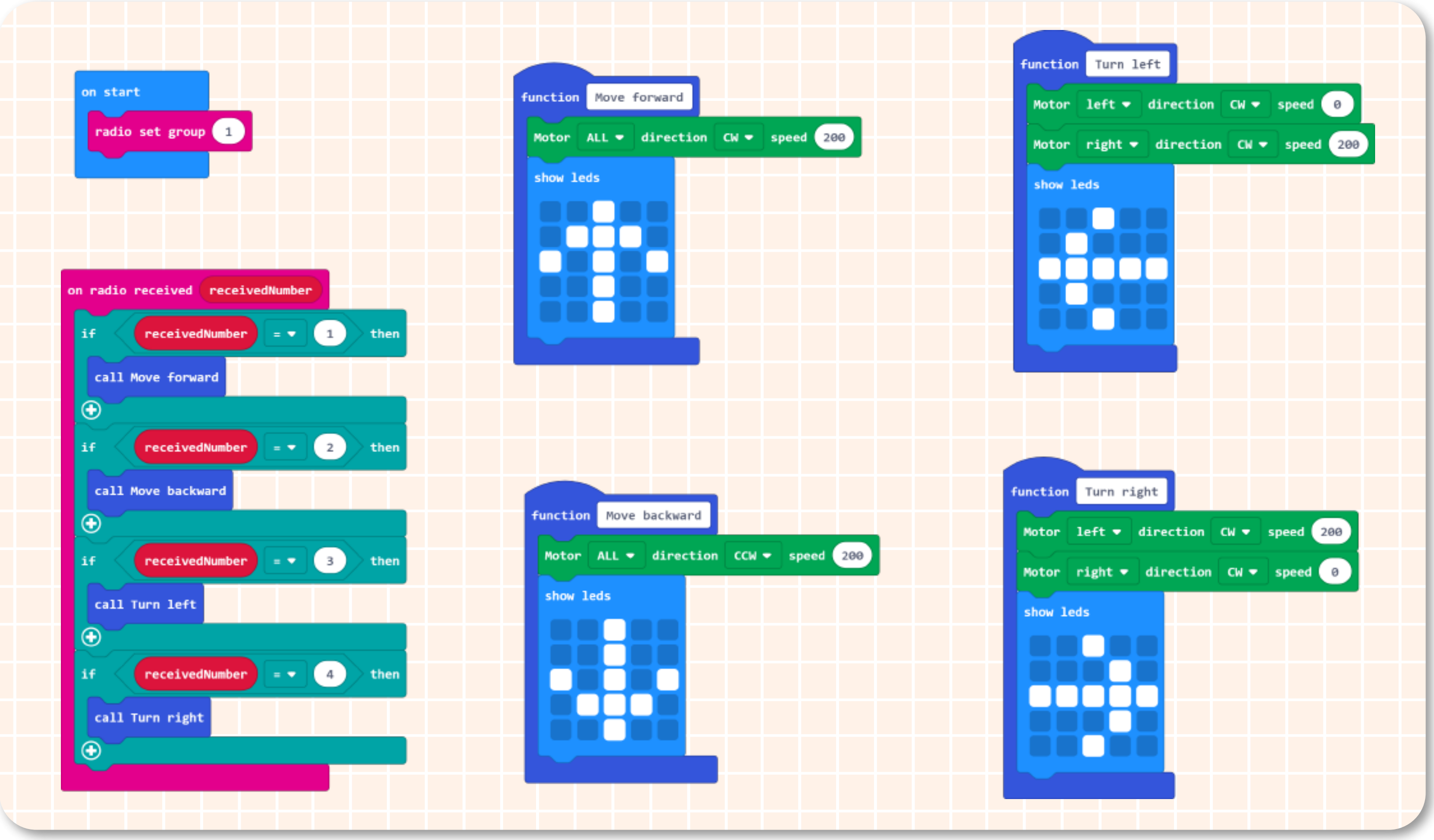
```
on radio received receivedNumber
    if receivedNumber = 1 then
        call Move forward
    if receivedNumber = 2 then
        call Move backward
    if receivedNumber = 3 then
        call Turn left
    if receivedNumber = 4 then
        call Turn right
```

5.Take the function “move forward” as an example. When the Maqueen Plus car moves forward, we let its LED screen display an “Up arrow”.



```
function Move forward
    Motor ALL direction CW speed 200
    show leds
```

Please complete the rest functions by yourself.
6.The program of the receiving end is shown below:

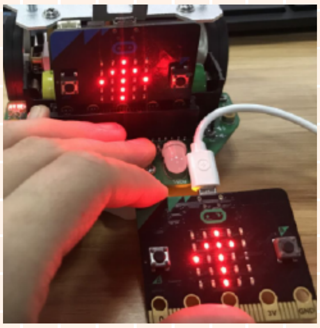


```
on start
    radio set group 1

on radio received receivedNumber
    if receivedNumber = 1 then
        call Move forward
    if receivedNumber = 2 then
        call Move backward
    if receivedNumber = 3 then
        call Turn left
    if receivedNumber = 4 then
        call Turn right

function Move forward
    Motor ALL direction CW speed 200
    show leds
        . . # . .
        . # # # .
        # . # . #
        . . # . .
        . . # . .

function Move backward
    Motor ALL direction CCW speed 200
    show leds
        . . # . .
        . . # . .
        # . # . #
        . # # # .
        . . # . .

function Turn left
    Motor left direction CW speed 0
    Motor right direction CW speed 200
    show leds
        . . # . .
        . # . . .
        # # # # #
        . # . . .
        . . # . .

function Turn right
    Motor left direction CW speed 200
    Motor right direction CW speed 0
    show leds
        . . # . .
        . . . # .
        # # # # #
        . . . # .
        . . # . .
```

7.Name your project as "Motion sensing robot-Receiving end" and then download the program to the micro:bit in Maqueen Plus.

## Effect Display

When completed all the above steps, turn Maqueen Plus's power switch on, then we can use the micro:bit of the transmitting end to control our car. Give it a go!



Move forward

Note: only when powered on can the micro:bit board work, so the micro:bit of transmitting end should be always connected to a computer during operation.

## Think & Explore

When playing a motion-sensing game, the larger angle we turn, the larger angle the object moves on the screen. How do we achieve this on our Maqueen Plus?

# Chapter 13

## Crazy Racing

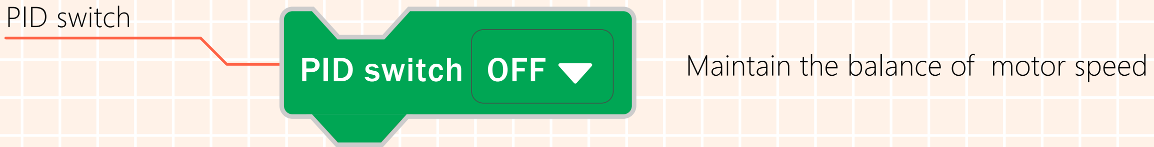
The road conditions of the car racing could be very complicated, winding mountain road, dirt road in a dark forest and so on. Can our Maqueen Plus run on such kind of racetrack, give it a go, let's turn Maqueen Plus into a racing robot car!

## Goal

What is PID algorithm?

## Command Learning

### Block Brief



PID switch

```
PID switch OFF ▾
```

Maintain the balance of  motor speed

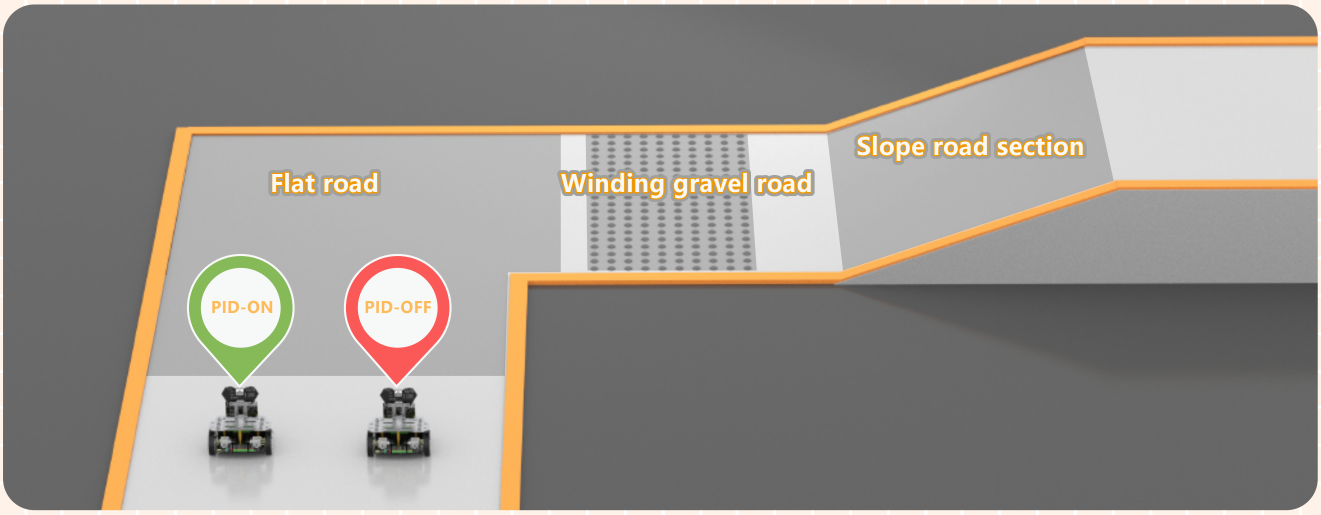
### What is PID algorithm?

PID is the most commonly-used and classical algorithm in control theory. We may not be familiar with its concept, but it can be said surely that we use it in some applications every day, such as, quadcopter, balancing car, cruise control in car, temperature control in 3D printer, thermostatic water heater and so on. When we need to keep a physical quantity(temperature, speed, balance, etc) in a stable state, PID must be used. Well, now we know what PID is, let's check the rules of the car racing.

## Hands-on Practice

### Competition Rules:

The speed of the competition is set at 30. The track consists of three different courses: ordinary road; winding gravel road; slope road section. The one who reaches the finish line first wins. The picture below is the route of the competition.



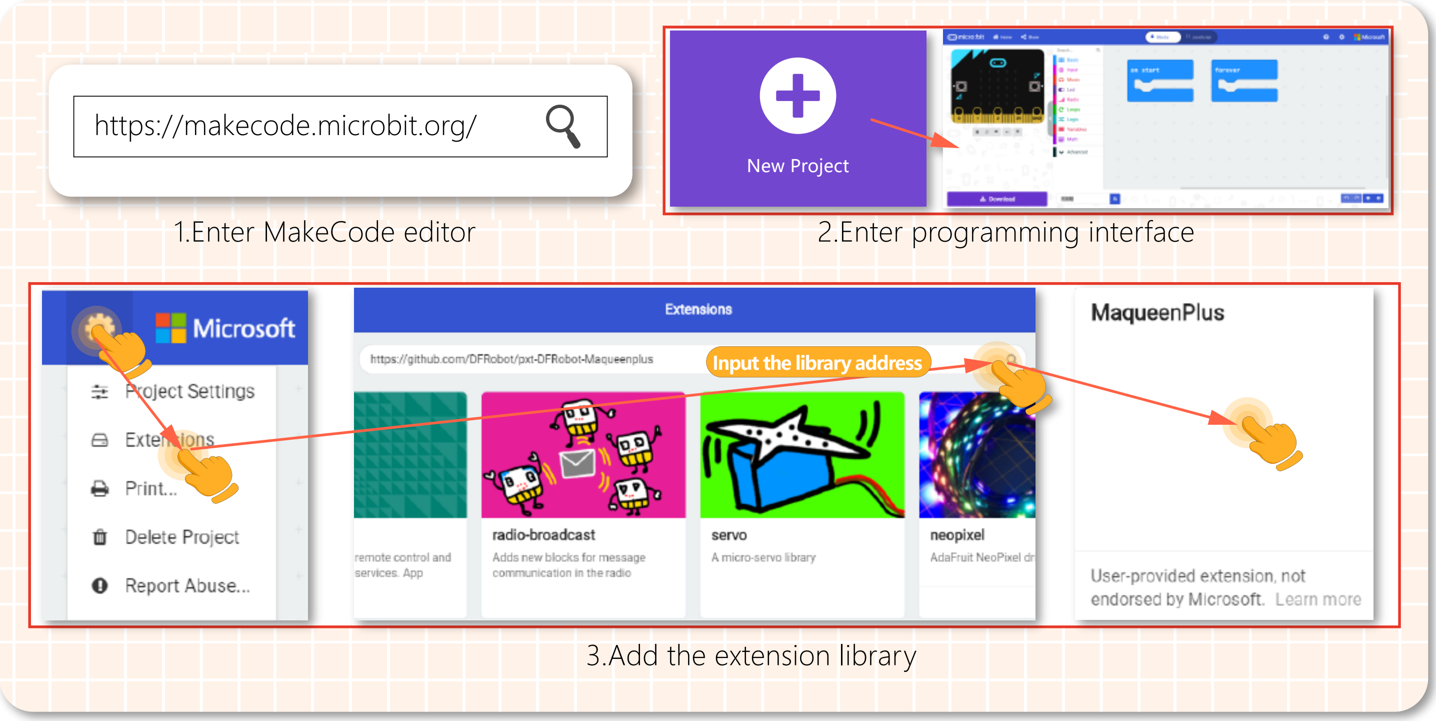
Note:
1.The race road needs to be built by yourself.
2.Two Maqueen Plus cars are necessary for this project.

Then, we are going to learn how PID works in this game, and see the differences when PID is enabled and disabled.

# Maqueen Plus 1-PID Enabled

## Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus


https://makecode.microbit.org/
New Project
Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

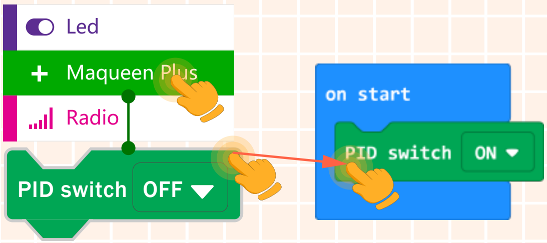
1.Enter MakeCode editor

2.Enter programming interface

3.Add the extension library

## Step 2 Programming

1.Turn on the PID switch when the program starts.



Knowledge Expansion

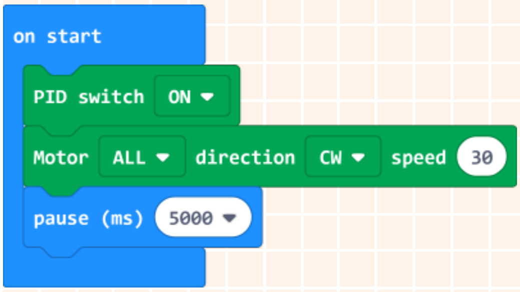
What does PID stand for?

P: proportional, adjust the output proportionally to the error.

D: derivative, responsible for compensating for future error.

I: integral, remove any steady-state error in the system because it accumulates that error over time and compensates for it

2.In the first course of the race, Maqueen Plus car runs at the speed of 30 for 5s.

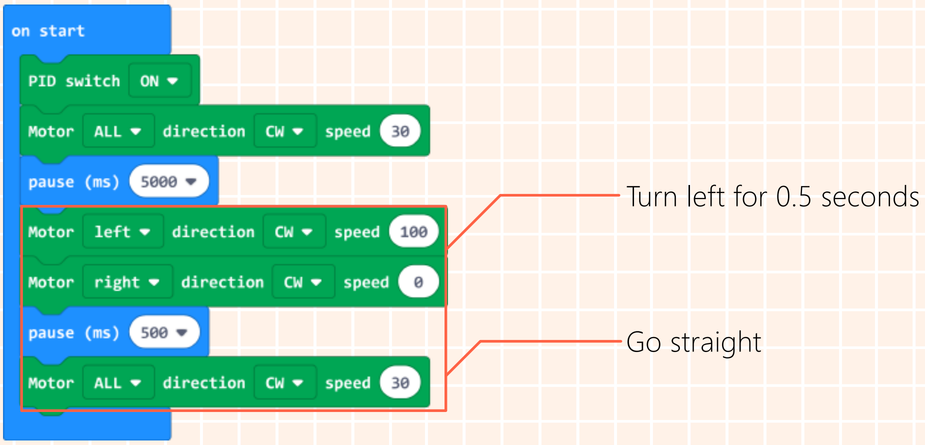


```
on start
    PID switch ON
    Motor ALL direction CW speed 30
    pause (ms) 5000
```

Knowledge Expansion

Note: set motor speed and pause time

3.Before the second course, Maqueen Plus1 needs to turn right and then go straight to the ending point.

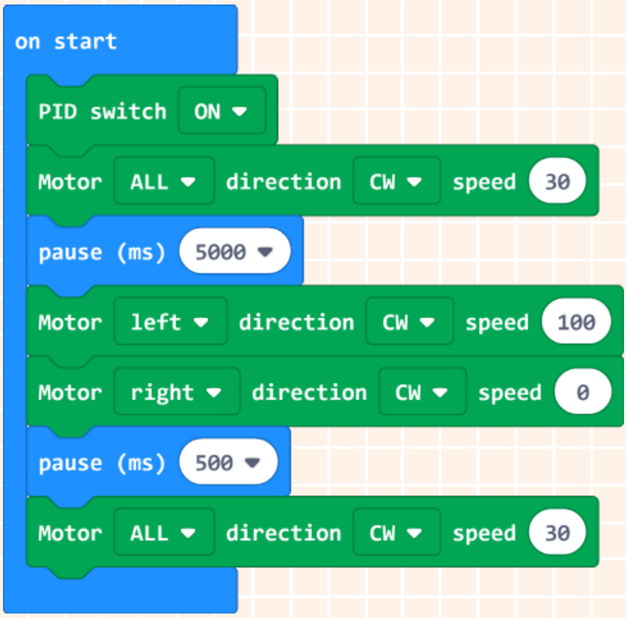


```
on start
    PID switch ON
    Motor ALL direction CW speed 30
    pause (ms) 5000
    Motor left direction CW speed 100
    Motor right direction CW speed 0
    pause (ms) 500
    Motor ALL direction CW speed 30
```

Turn left for 0.5 seconds

Go straight

4.The complete program is shown below:



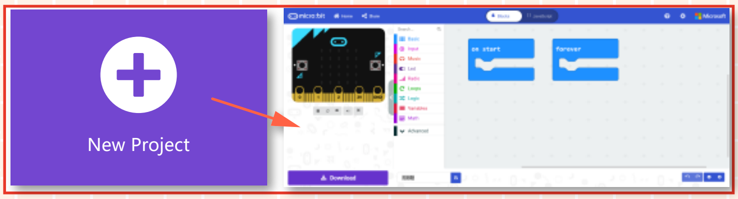
```
on start
    PID switch ON
    Motor ALL direction CW speed 30
    pause (ms) 5000
    Motor left direction CW speed 100
    Motor right direction CW speed 0
    pause (ms) 500
    Motor ALL direction CW speed 30
```

5.Name your project as “Crazy Racing 1-PID enabled”, and download the program to Maqueen Plus1.

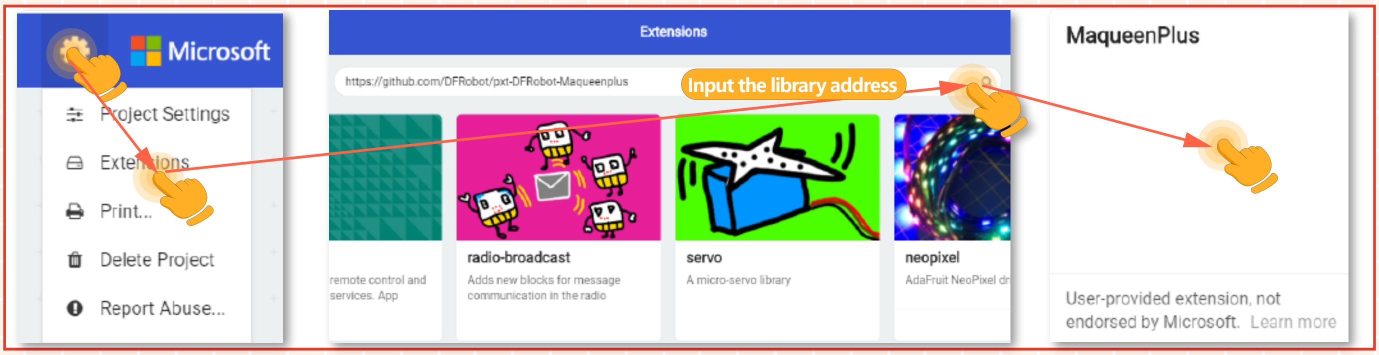
## Maqueen Plus-PID Disabled

https://makecode.microbit.org/

1.Enter MakeCode editor



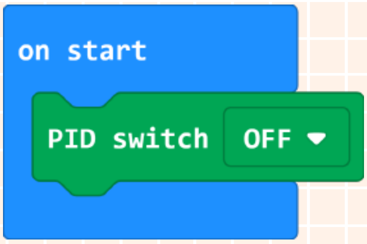
2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more

3.Add the extension library

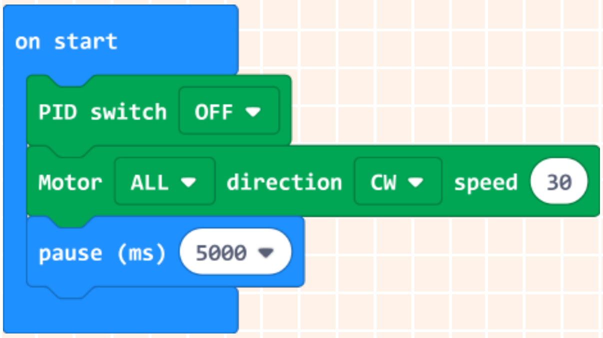
## Step 2 Programming

1.Turn off the PID switch when the program starts.



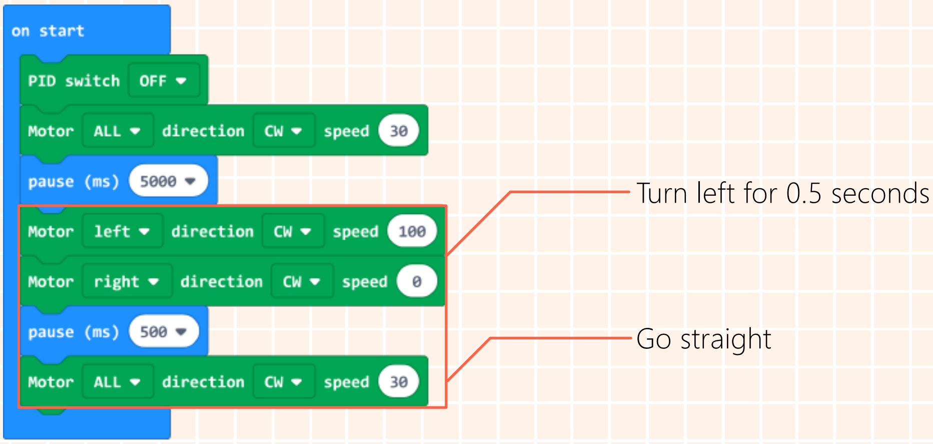
```
on start
    PID switch OFF
```

2.Maqueen Plus runs at the speed of 30 for 5s in the first course.



```
on start
    PID switch OFF
    Motor ALL direction CW speed 30
    pause (ms) 5000
```

3.Before the second course, Maqueen Plus 2 needs to turn right and then go straight to the ending point.

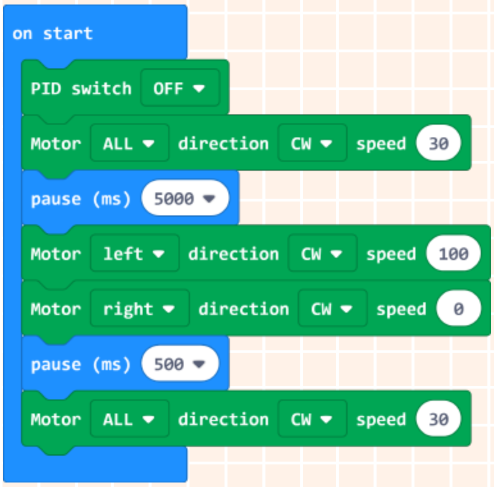


```
on start
    PID switch OFF
    Motor ALL direction CW speed 30
    pause (ms) 5000
    Motor left direction CW speed 100
    Motor right direction CW speed 0
    pause (ms) 500
    Motor ALL direction CW speed 30
```

Turn left for 0.5 seconds

Go straight

4.The entire program is shown below:



```
on start
  PID switch OFF
  Motor ALL direction CW speed 30
  pause (ms) 5000
  Motor left direction CW speed 100
  Motor right direction CW speed 0
  pause (ms) 500
  Motor ALL direction CW speed 30
```

5 .Name your project as “Crazy Racing 2-PID disabled”, and download the program to Maqueen Plus 2.

## Effect Display

After the above steps, put the car on our track and turn on the switches of the two Maqueen Plus cars at the same time. Then we can see:

At the first racecourse: both cars move forward at a slow and same speed.

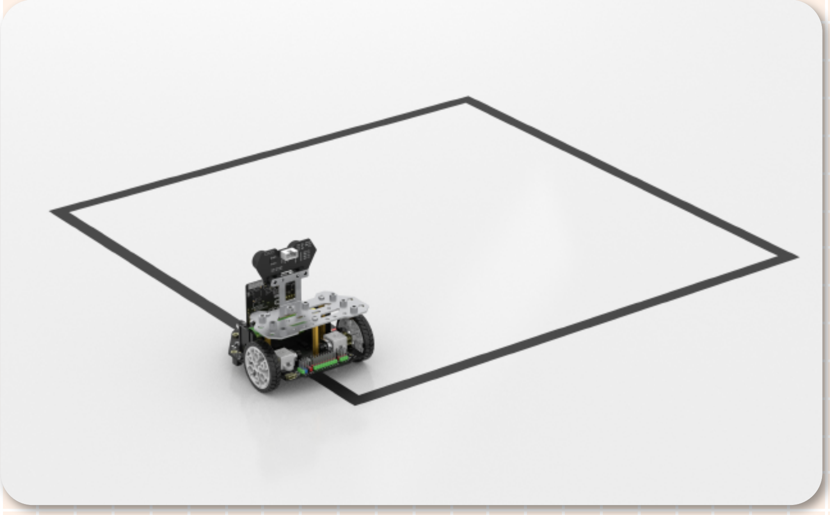
At the second racecourse: after both cars turn right and come to the track with sand and stone obstacles, the car 1 (PID enabled) is still moving; the race car 2(PID disabled) stopped when encountering obstacles.

At the 3rd racecourse: in the slope, race car 1 can continue to drive; the race car 2 still stay where it met the obstacle.

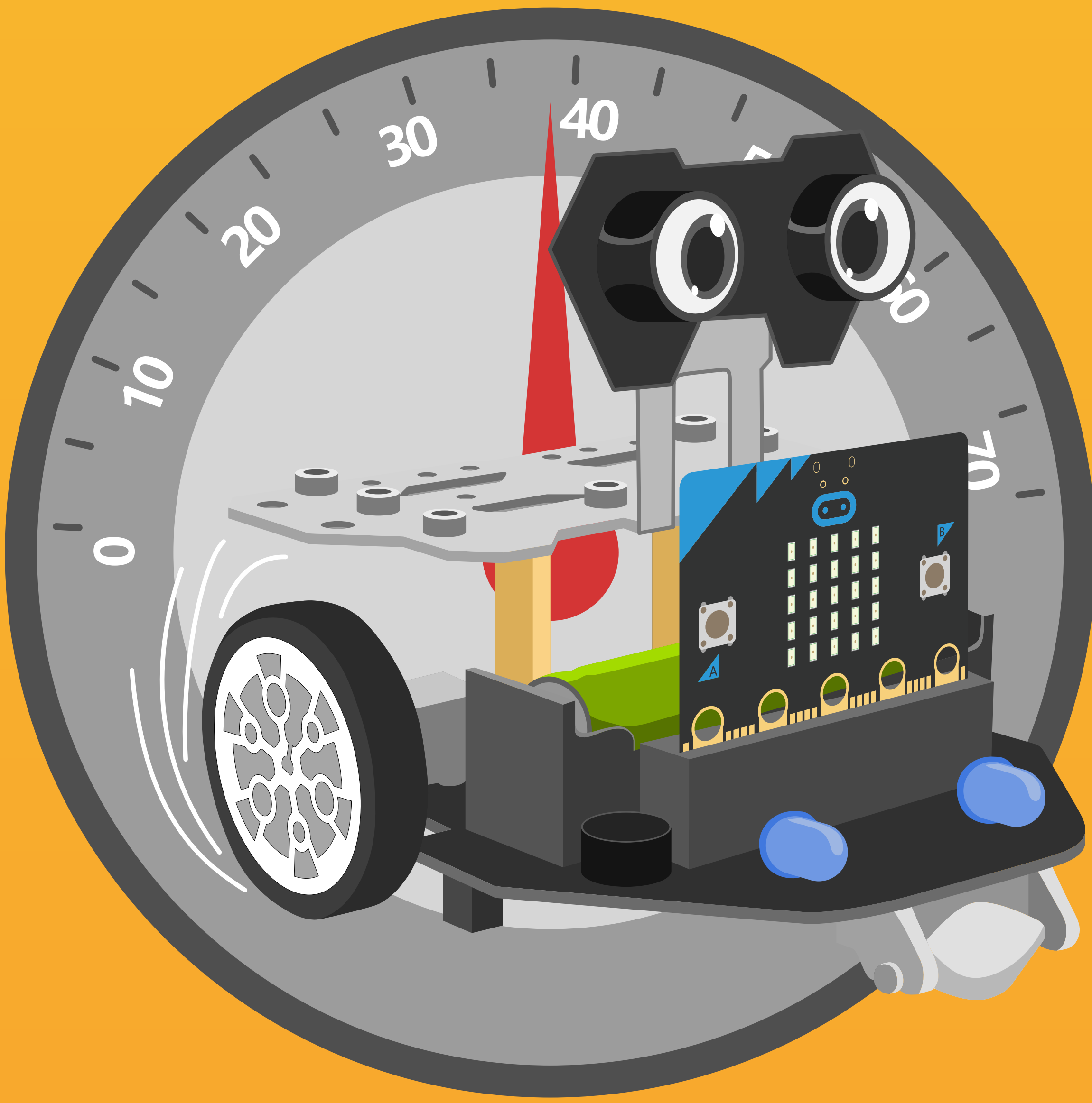
Eventually, the race car 1(PID enabled) wins.

## Think & Explore

Since we have knew the advantage of PID: maintain the motor balance of the car and make the car move stably. Next, think about how to help the car to race on a square track.



Tips: two parts for the racecourse: go straight and turn 90 degrees. Enable PID to allow the Maqueen Plus car to move steadily on the track.



# Chapter 14

## Speed Recorder

The car speed recorder can accurately record the driving state, such as, direction turning, speed increasing and reducing, and so on. The collected data can be displayed in real-time for the driver to master the vehicle condition. How can we visually see the speed information of Maqueen Plus? An OLED display helps here. Maqueen Plus has 3 IIC ports for connecting Gravity modules with IIC communication. Let's start to make a speed recorder for Maqueen Plus.

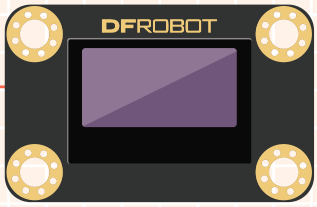
# Goal

Learn how to use OLED display

# Electronic Component

**OLED**

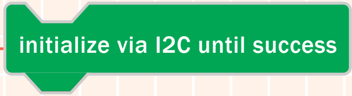
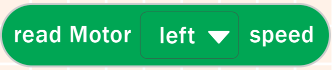
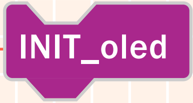
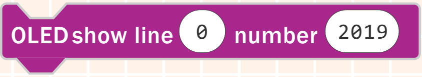
OLED-2864 Display



OLED-2864 display module can work without backlight, support IIC communication, high transmission rate and 60Hz refresh frequency.

# Command Learning

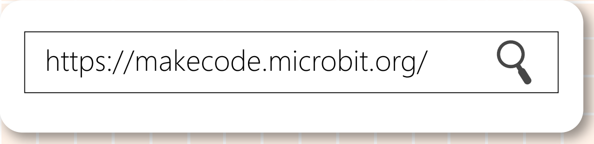
**Block Brief**

| | | |
|---|---|---|
| I2C Initiate |  initialize via I2C until success | Initiate IIC to detect if the communication between micro:bit and Maqueen Plus is successfully built. |
| Read motor speed |  read Motor left speed | Read the current speed of the motor |
| Initiate OLED |  INIT_oled | Initiate OLED display module |
| OLED display number |  OLED show line 0 number 2019 | Display number at the n line of OLED screen |

# Hands-on Practice

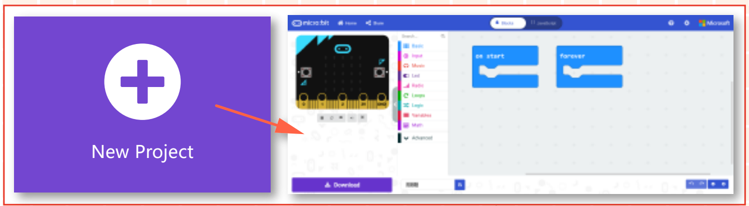
## Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click “new project” to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
4.Click “setting”->”Extension”, input the following address into the search bar, and select “OLEDV1”.
https://github.com/DFRobot/pxt-OLEDV1
5.OLEDV1-related blocks will appear at the command block section when the OLED library is added successfully.

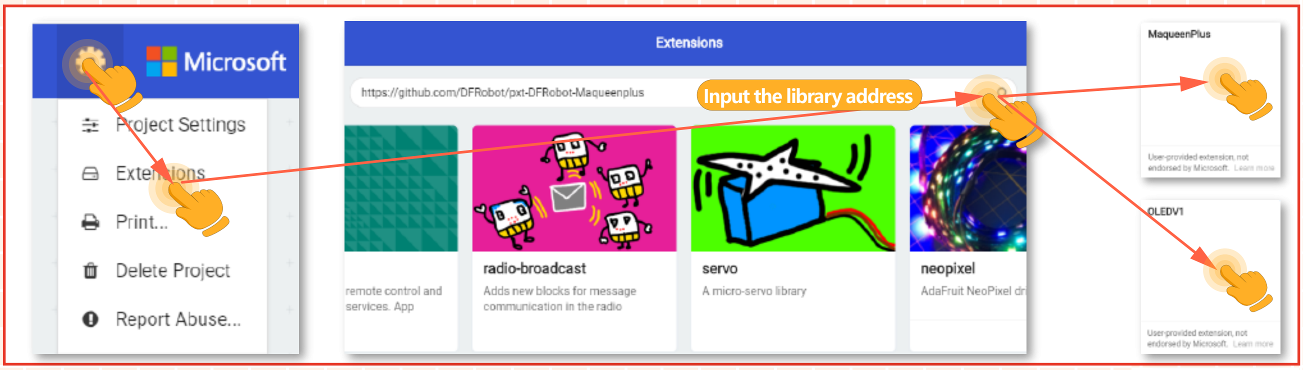


https://makecode.microbit.org/

1.Enter MakeCode editor

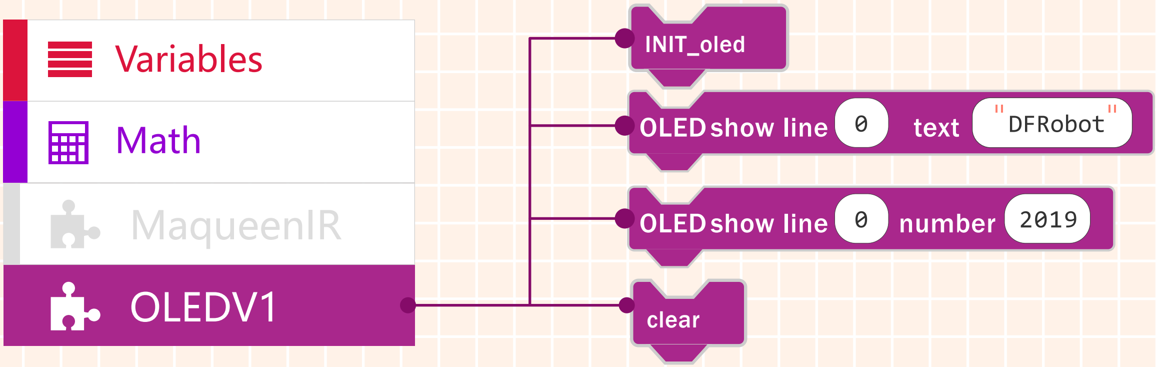


2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more
OLEDV1
User-provided extension, not endorsed by Microsoft. Learn more

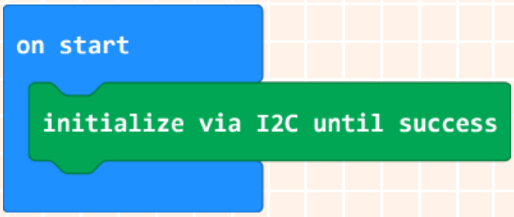
3.Add the extension library

## OLED program module



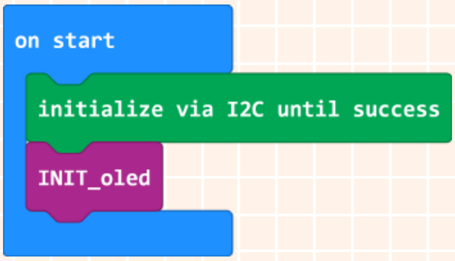
## Step 2 Programming

1.Initiate the IIC address when the program starts.



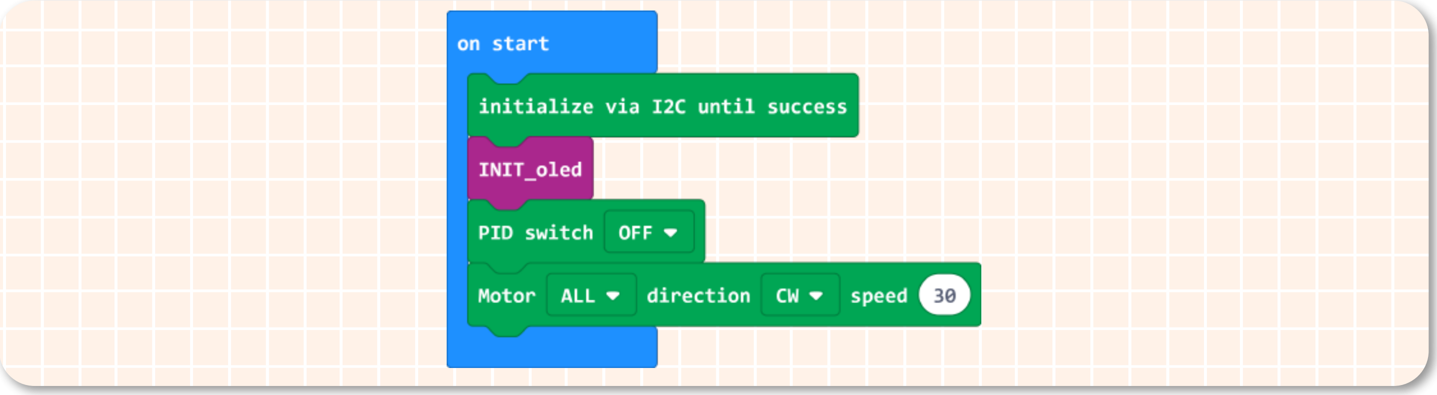
```
on start
    initialize via I2C until success
```

1.Initiate the IIC address when the program starts.



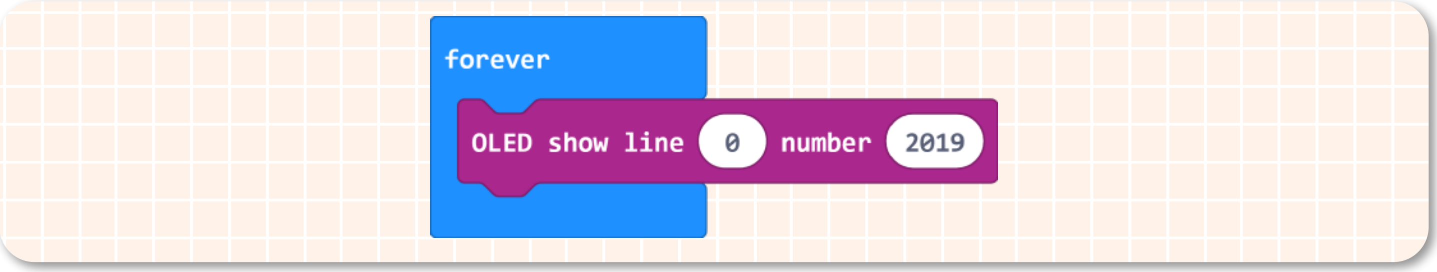
```
on start
    initialize via I2C until success
    INIT_oled
```

3.Disable PID, and let the Maqueen Plus car move forward at the speed of 30.



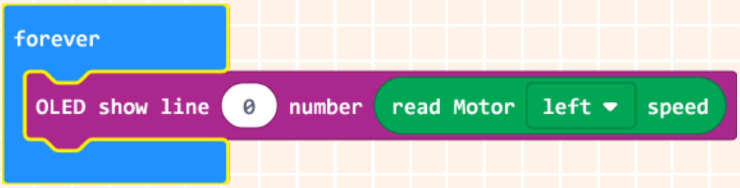
```
on start
    initialize via I2C until success
    INIT_oled
    PID switch OFF
    Motor ALL direction CW speed 30
```

4.Place the OLED number display block inside the “forever” block for showing data in real-time.



```
forever
    OLED show line 0 number 2019
```

5.Drag the “read motor speed” block into the “OLED display number” block to display the left motor speed on the OLED display.



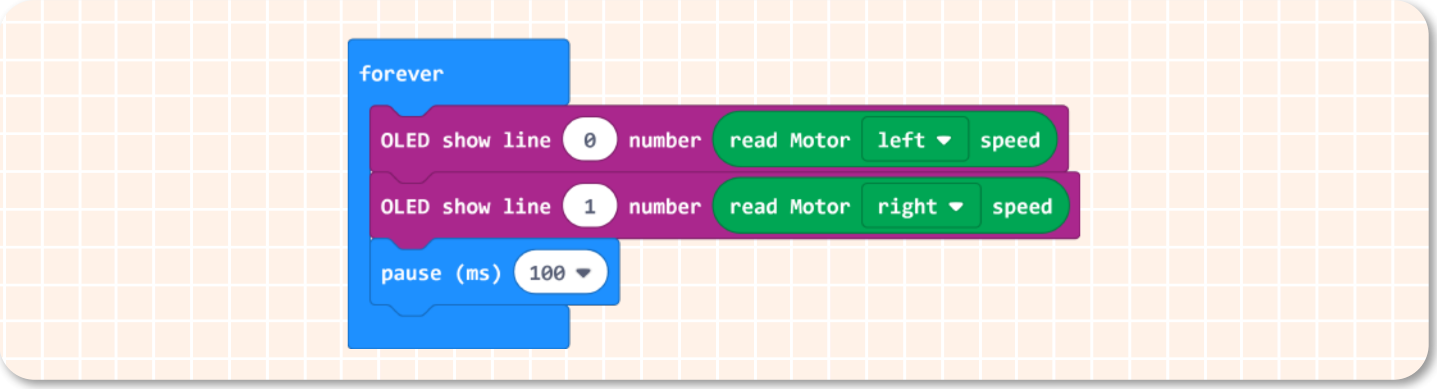
```
forever
    OLED show line 0 number read Motor left speed
```

Knowledge Expansion

You may have a doubt about the program “ OLED show line 0 number read Motor left speed ”, how can we display the number at the line 0 of the OLED screen?

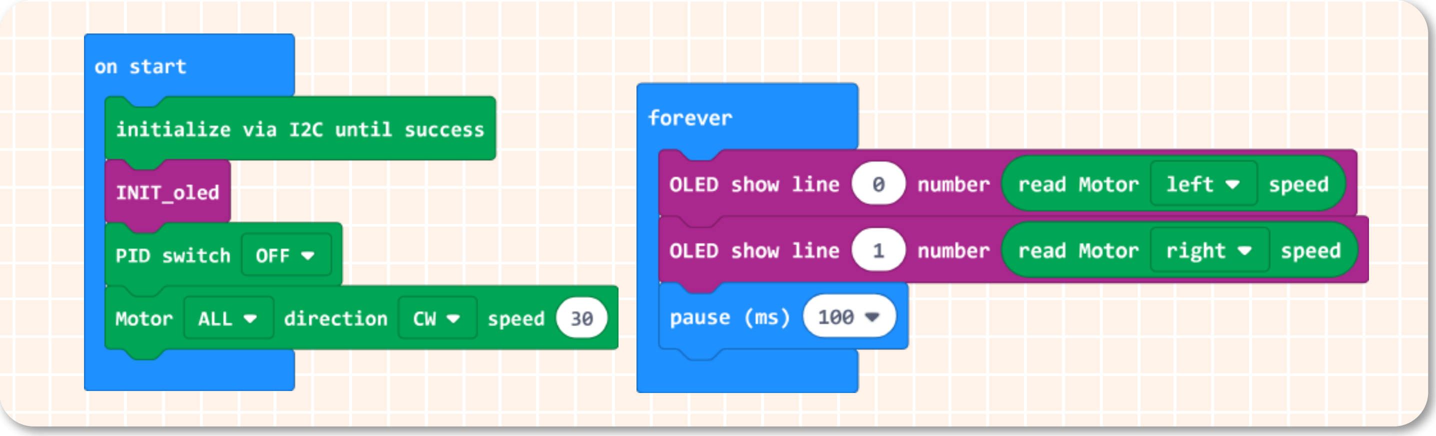
In programming, it often counts from 0, but in fact, line 0 in the OLED screen corresponds to the first line of the screen.

6.Program to display the right motor speed at the second line of the OLED screen by the same way, and refresh the data every 0.1 second. The complete program is shown below:



```
forever
    OLED show line 0 number read Motor left speed
    OLED show line 1 number read Motor right speed
    pause (ms) 100
```

7.The complete program is shown below:

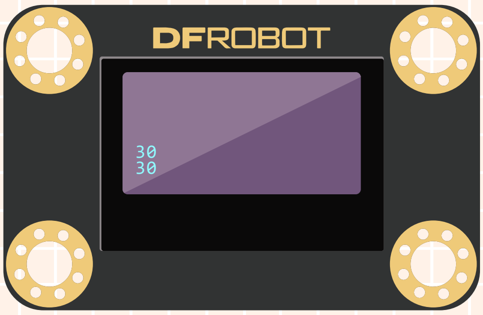


```
on start
    initialize via I2C until success
    INIT_oled
    PID switch OFF
    Motor ALL direction CW speed 30

forever
    OLED show line 0 number read Motor left speed
    OLED show line 1 number read Motor right speed
    pause (ms) 100
```

8.Name your project as “Speed Recorder” and save it.

## Step 5 Effect Display

Turn on the power switch, then the left motor speed will be displayed at the first line of the screen, and the right motor speed will be shown at the second line of the screen. The data will be refreshed constantly.

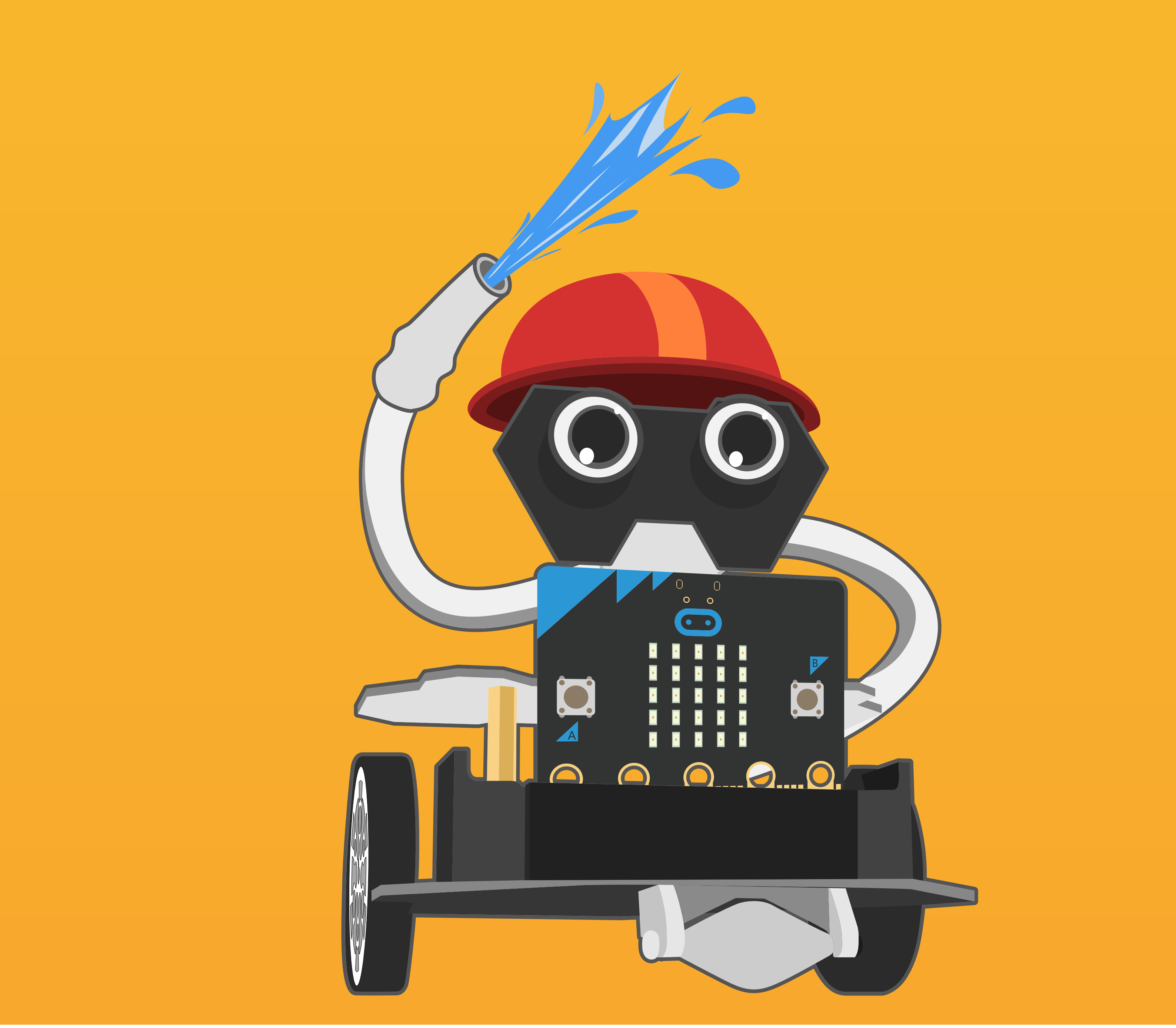


## Think & Explore

1.We may find that though we set the speed of both motors to 30, the data on the OLED still changes continuously. Why is that? How can we make the speed data less changeable?

Tips: turn on the PID switch to maintain the speed balance of the motors.

2.There are several projects involving LED matrix in the previous chapters, like light sensing robot, little ranging expert, etc. Now replace the LED display with OLED screen.



# Chapter 15

## Firefighting Robot

Firefighting is a highly dangerous occupation. When an emergency response call comes, firefighters must get themselves dressed in the appropriate gear and onto the emergency vehicles quickly to arrive at the fire scene, and then extinguish the fires. Firefighters are risking their lives to defend our cites and country. So everybody, can we use what we learned to help to reduce the risk? How about a firefighting robot? Let' s make a firefighting robot with Maqueen, and let it complete three actions: Call out, Firefighting, Mission done. Flame sensor and servo will be used in this project. Maqueen Plus has 8 GPIO ports and 3 servo ports for connecting Gravity module and servos. Everything is ready, let's start!

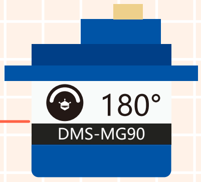
## Goal

Learn how to drive a servo
Learn how to use a flame sensor

## Electronic Component

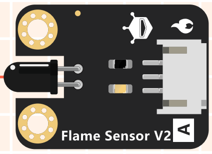
### Figure of the servo and the Flame sensor

Servo



0°~ 180°Micro Servo

Flame sensor

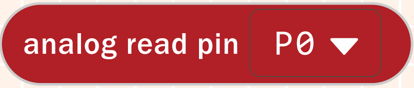


Flame sensor can detect fire and light with 760~1100nm.

## Command Learning

### Block Brief

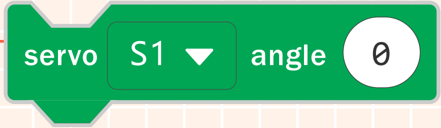
Analog read



analog read pin P0

Read analog value of the specified pin

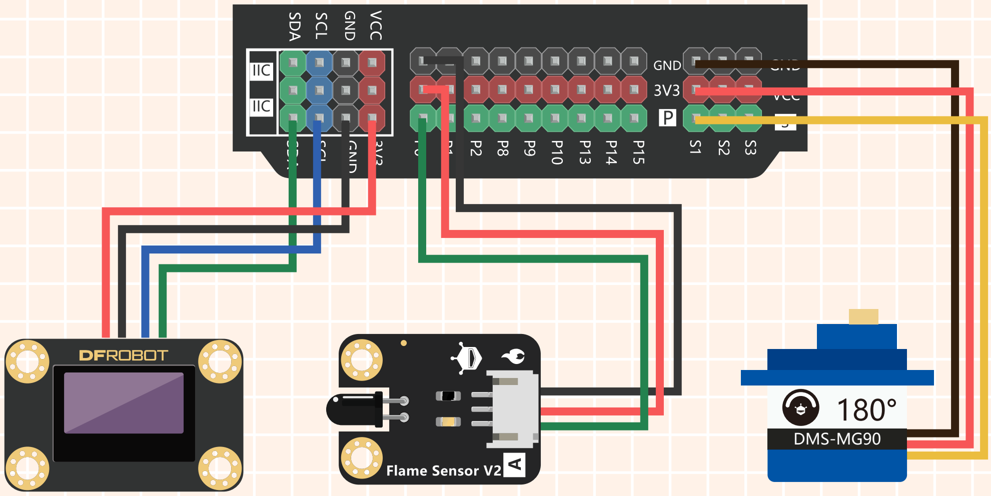
Servo



servo S1 angle 0

Set the angle of servo

### Firefighting Robot Hardware Connection



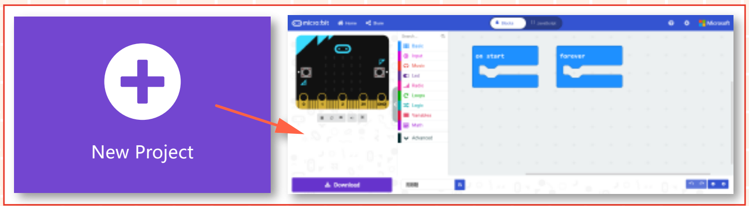
Flame sensor to P0;
Servo to the S1 port;
OLED screen to I2C

# Hands-on Practice

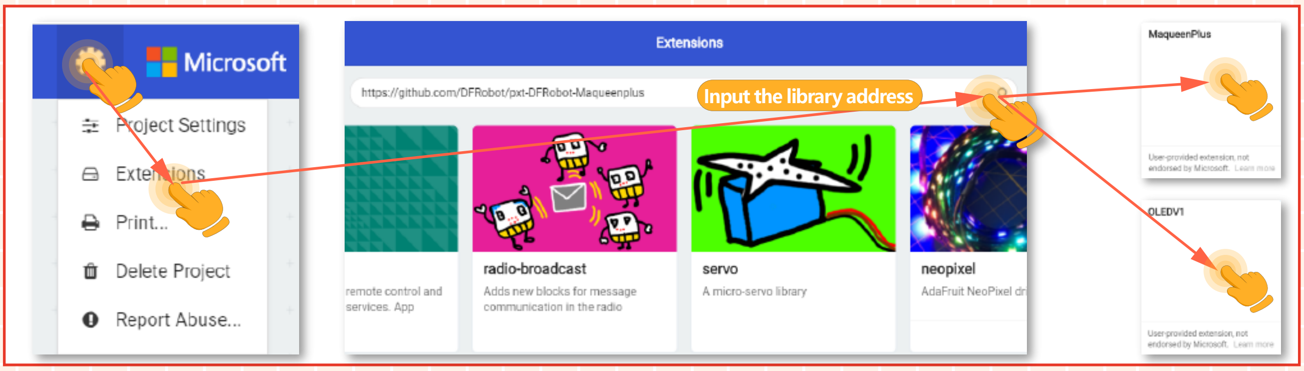
## Step 1 Create a New Project

1.Input https://makecode.microbit.org/ into your browser to enter MakeCode editor.
2.Click "new project" to enter MakeCode programming interface.
3.Add the Maqueen Plus library:https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
4.Click "Setting"->"Extension", input the following address and click the result OLED1.
https://github.com/DFRobot/pxt-OLEDV1

1.Enter MakeCode editor



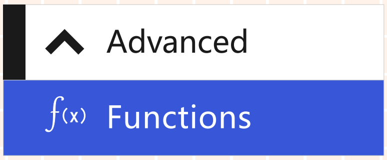
2.Enter programming interface


Microsoft
Project Settings
Extensions
Print...
Delete Project
Report Abuse...
Extensions
https://github.com/DFRobot/pxt-DFRobot-Maqueenplus
Input the library address
remote control and services. App
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel dr
MaqueenPlus
User-provided extension, not endorsed by Microsoft. Learn more
OLEDV1
User-provided extension, not endorsed by Microsoft. Learn more

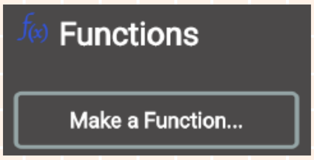
3.Add the extension library

## Step 2 Programming

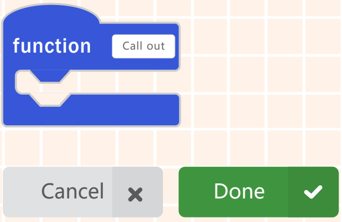
1.Generally, there are three parts to the firefighting process.
① Received the emergency call, go to the fire scene.
② Arrived at the scene, extinguish the fire.
③ Mission completed.
So, we will create three functions: "Call out", "Firefighting", "Mission done". Take the first one as an example.



1)Click "Advanced"->"Functions";

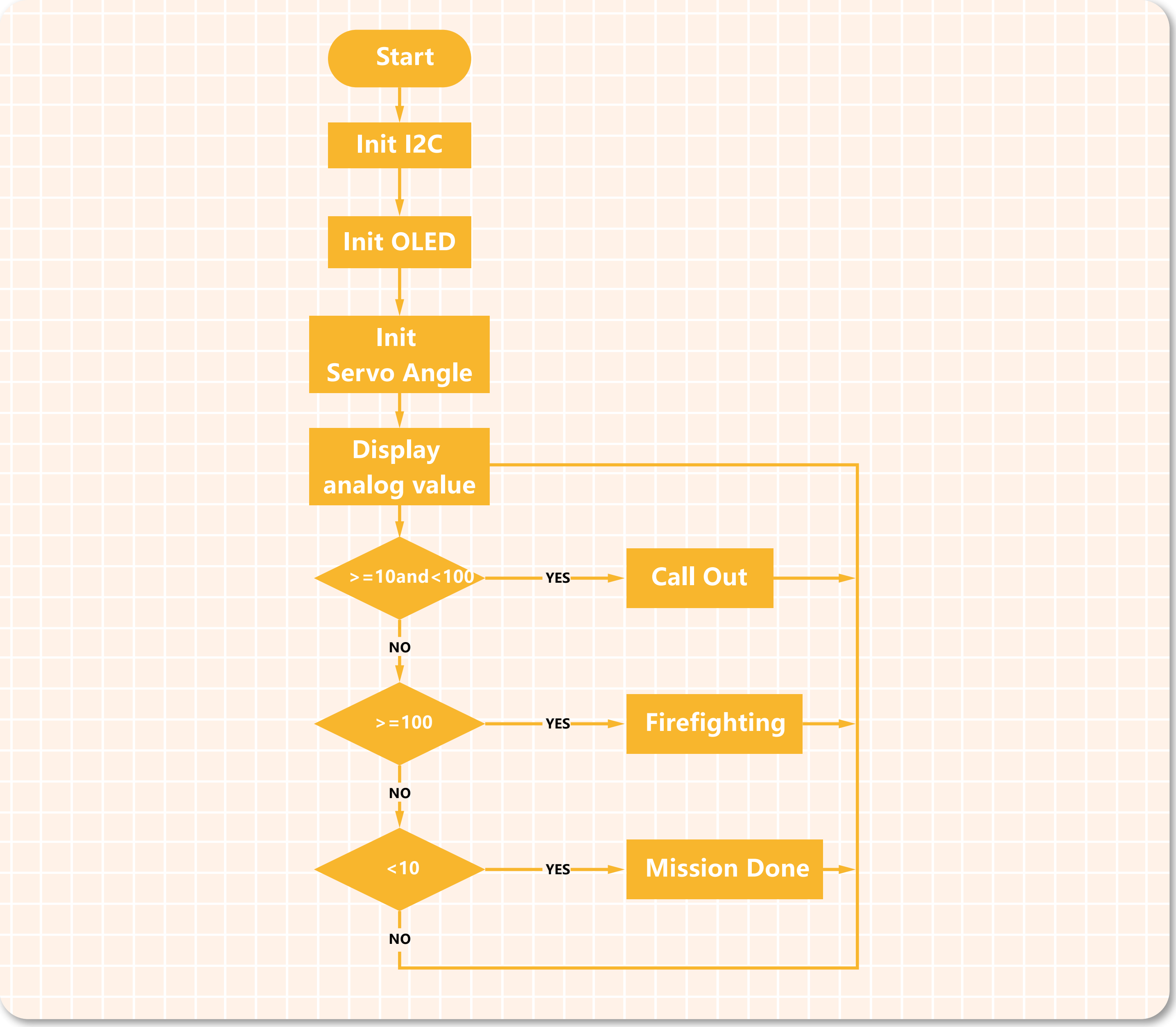


2)Click "Make a Function";

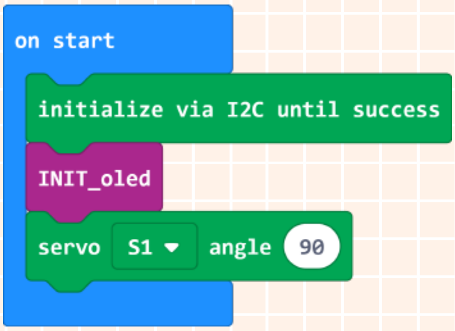


3)Name the function as "Call Out";
4)Click "OK".

2.Create another two functions in the same way above. Then do condition judgment as the flowchart shown below:



3.Initialize the I2C, OLED, Servo angle when the program starts.

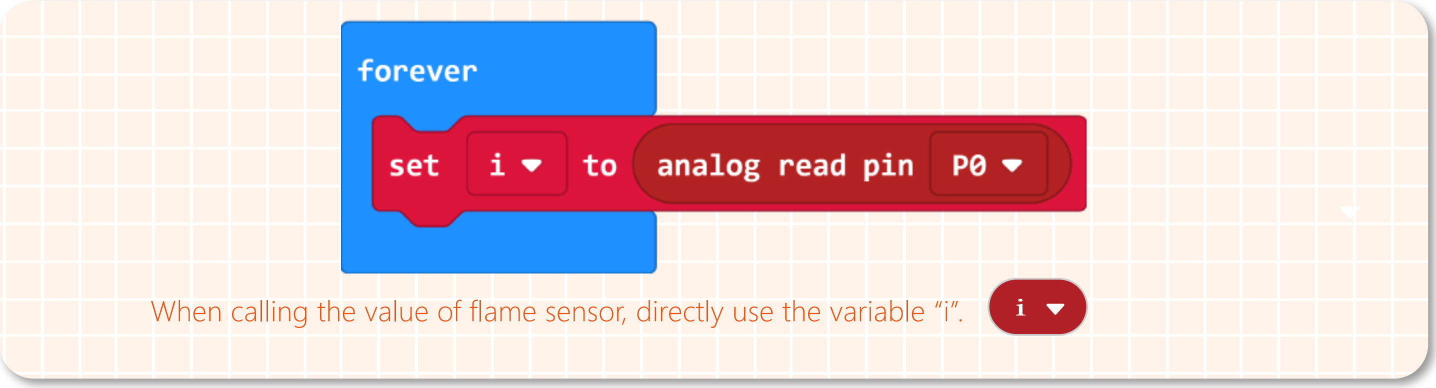


```
on start
  initialize via I2C until success
  INIT_oled
  servo S1 ▾ angle 90
```

Knowledge Expansion

Initialize the servo angle to represent that the fire hose is in off state.

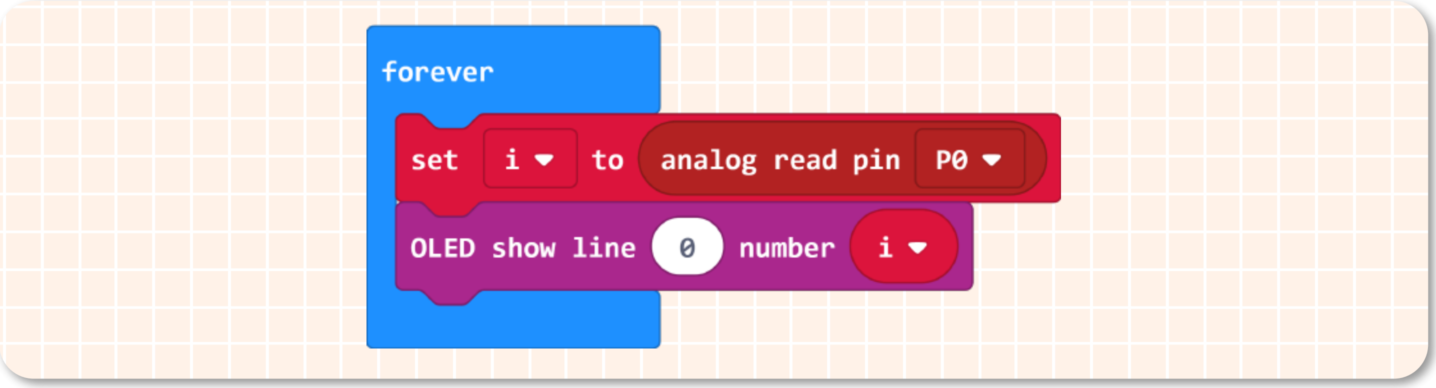
4.Create a variable “i” to store the analog value read from the flame sensor.
Make a variable “i”.



```
forever
    set i ▾ to analog read pin P0 ▾
```

When calling the value of flame sensor, directly use the variable “i”. i ▾

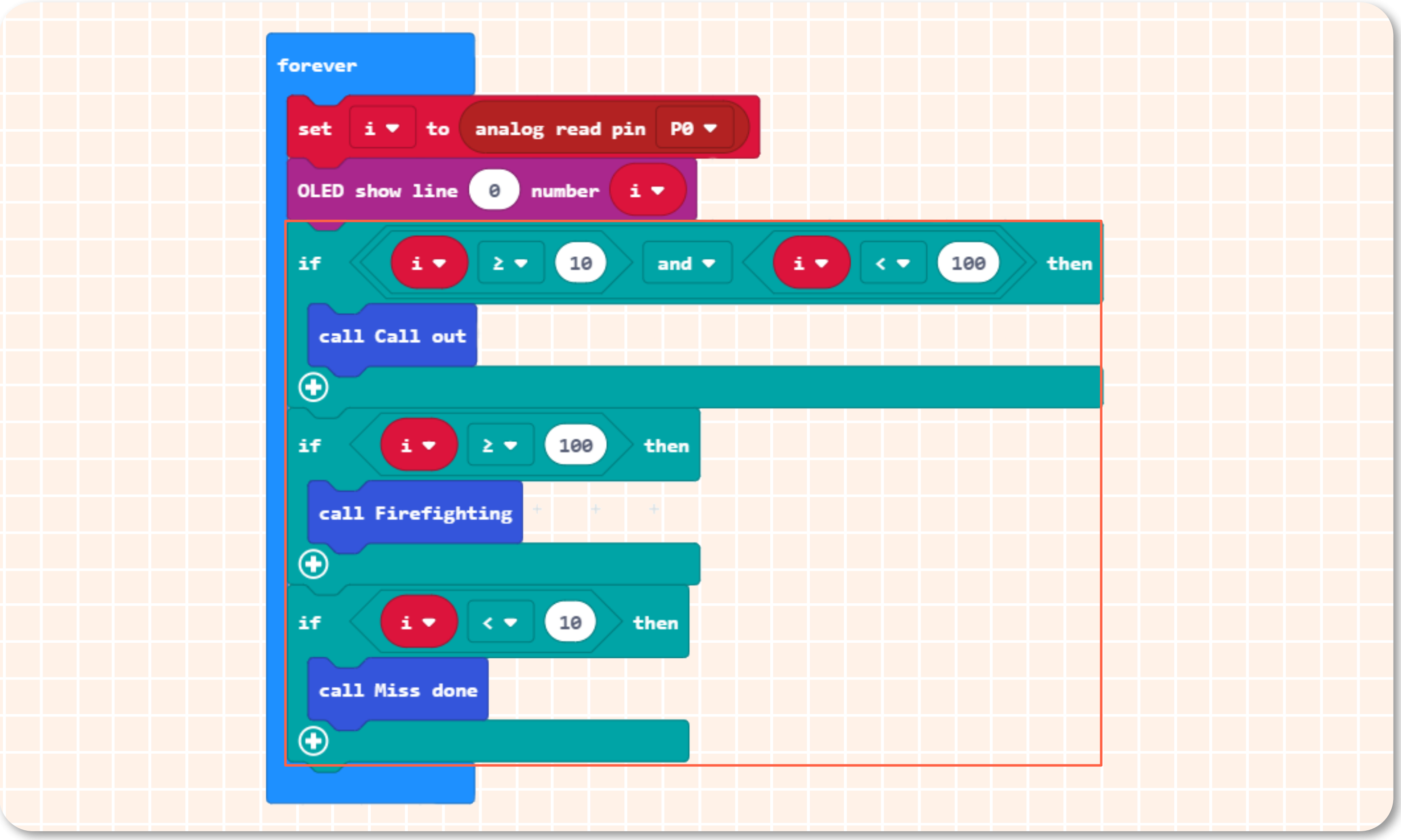
5.Display the analog value of the flame sensor on the first line of the OLED screen.



```
forever
    set i ▾ to analog read pin P0 ▾
    OLED show line 0 number i ▾
```

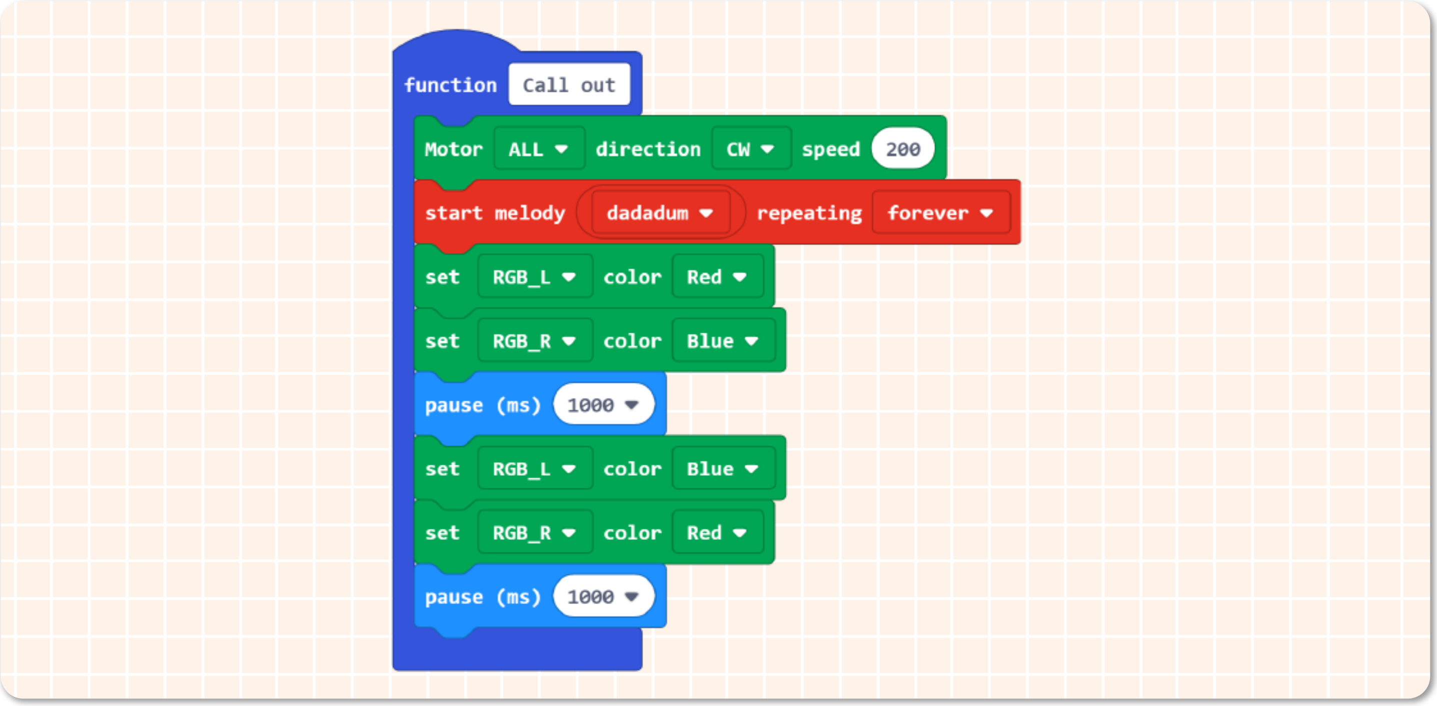
6.Call the related functions according to the program flowchart.

When the analog value detected by the flame sensor is between 10~100, it means that there is a certain distance between the firefighting robot and the fire scene, and the robot has to go forward to the scene; if the value is larger than 100, the robot has arrived at the fire scene, and starts to extinguish the fire; when it is less than 10, the firefighting mission is done.



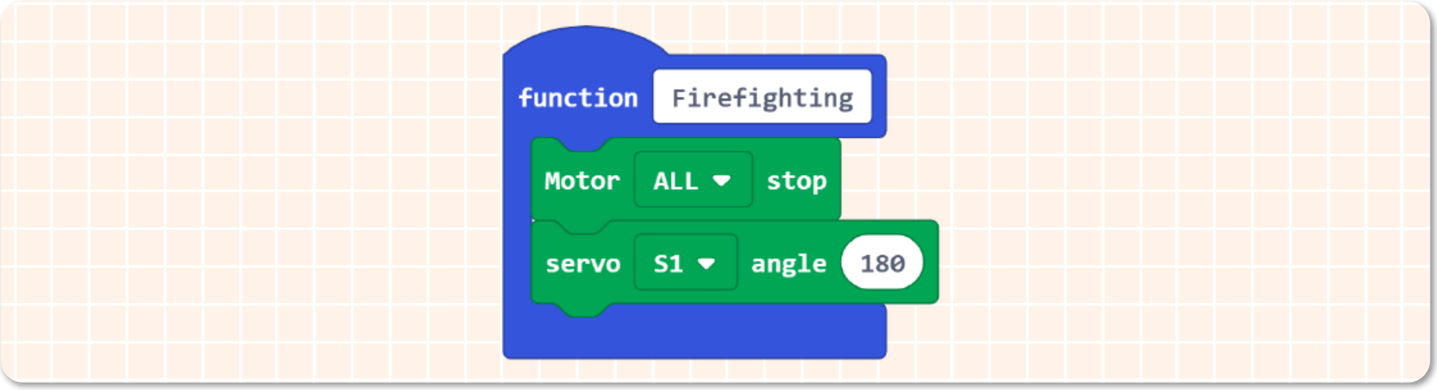
```
forever
    set i ▾ to analog read pin P0 ▾
    OLED show line 0 number i ▾
    if i ▾ ≥ ▾ 10 and ▾ i ▾ < ▾ 100 then
        call Call out
    ⊕
    if i ▾ ≥ ▾ 100 then
        call Firefighting
    ⊕
    if i ▾ < ▾ 10 then
        call Miss done
    ⊕
```

7.The realization of the“Call out” function: when the firefighting robot drives to the fire scene, the two LEDs flash red and blue alternatively, meanwhile, the keep the siren blaring (use the sound “dadadum” to simulate the siren).



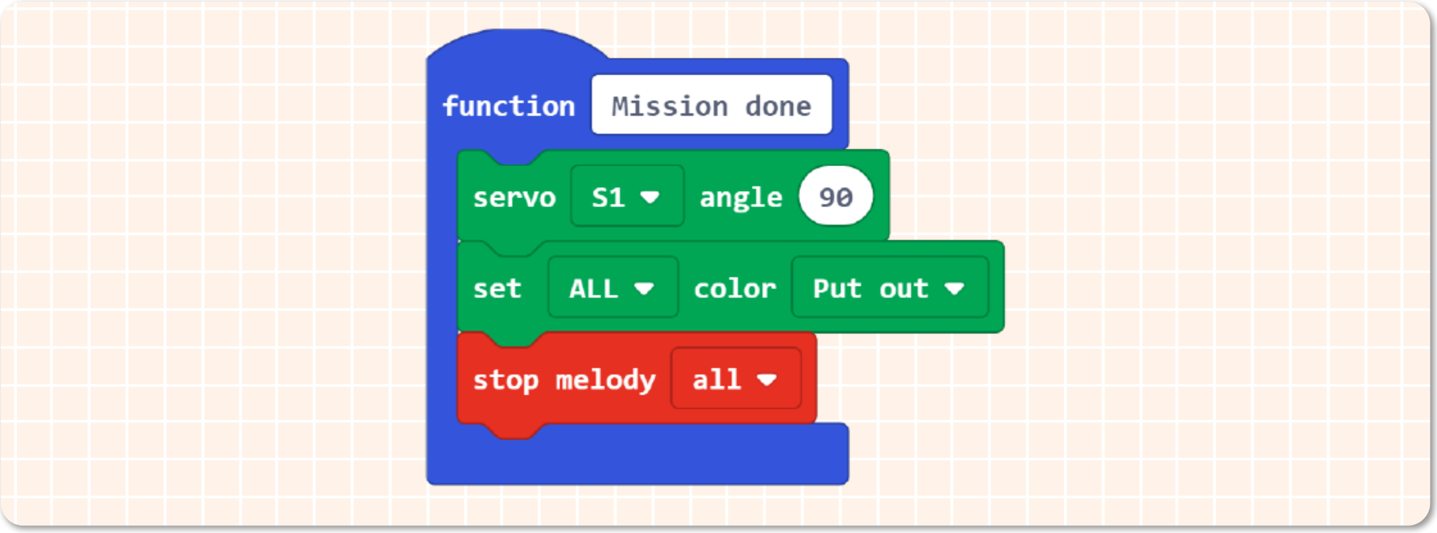
```
function Call out
    Motor ALL ▾ direction CW ▾ speed 200
    start melody dadadum ▾ repeating forever ▾
    set RGB_L ▾ color Red ▾
    set RGB_R ▾ color Blue ▾
    pause (ms) 1000 ▾
    set RGB_L ▾ color Blue ▾
    set RGB_R ▾ color Red ▾
    pause (ms) 1000 ▾
```

8.The realization of the function “Firefighting”: when the firefighting robot arrived at the scene, turn on its fire hose to put out the fire (change the angle of the servo to simulate this process).



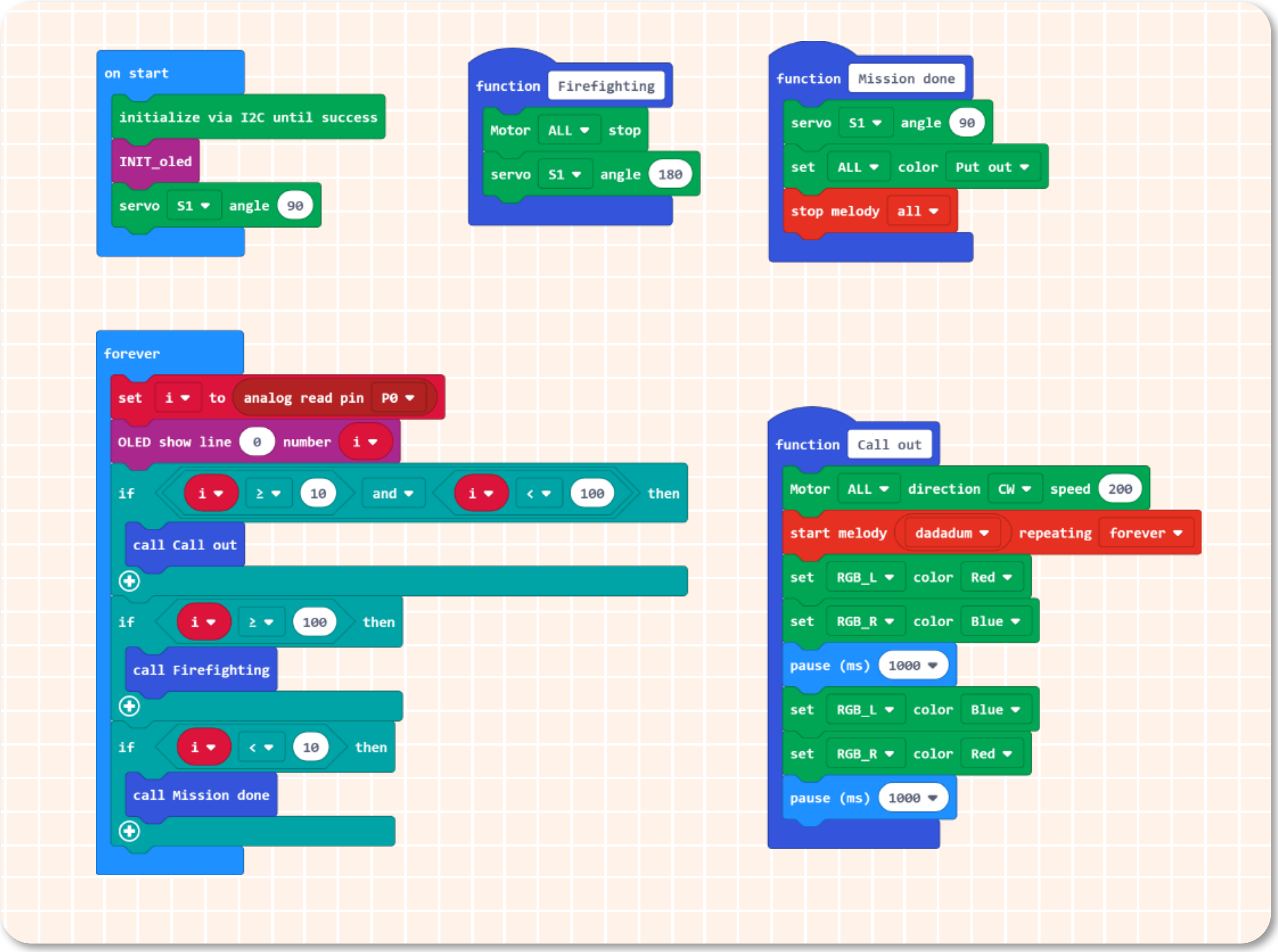
```
function Firefighting
    Motor ALL ▾ stop
    servo S1 ▾ angle 180
```

9.The function “Mission done”: turn off the fire hose, LEDs, and siren.



```
function Mission done
    servo S1 ▾ angle 90
    set ALL ▾ color Put out ▾
    stop melody all ▾
```

10.The complete program is shown below.



```
on start
    initialize via I2C until success
    INIT_oled
    servo S1 angle 90

function Firefighting
    Motor ALL stop
    servo S1 angle 180

function Mission done
    servo S1 angle 90
    set ALL color Put out
    stop melody all

forever
    set i to analog read pin P0
    OLED show line 0 number i
    if i ≥ 10 and i < 100 then
        call Call out
    if i ≥ 100 then
        call Firefighting
    if i < 10 then
        call Mission done

function Call out
    Motor ALL direction CW speed 200
    start melody dadadum repeating forever
    set RGB_L color Red
    set RGB_R color Blue
    pause (ms) 1000
    set RGB_L color Blue
    set RGB_R color Red
    pause (ms) 1000
```

11.Name your project as “Firefighting robot”, and download it into Maqueen Plus.

## Step 3 Effect Display

Turn on the power switch when completed all the steps above. Then the analog value the flame sensor detects will be constantly displayed on the OLED screen. When the value is in 10~100, the firefighting robot moves forward at the speed of 200 with its light flashing and siren blaring; when it is more than 100, the robot car stops, and its servo rotates 180 degrees; when less than 10, the servo back to 90 degrees, stop playing sound and turn off the RGB LEDs.

Note: we can use a lighter to imitate the fire scene. Although the flame sensor can detect fire, it is not fireproof. Please make sure the sensor always keeps a certain distance from the fire.

This project involves dangerous action, please complete this chapter with the assistance of guardians or teachers.

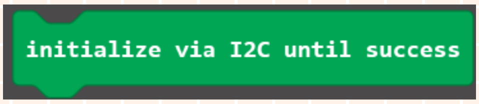
## Think & Explore

We all know that smoking is not only harmful to our own health but effecting others. Therefore, smoking is forbidden in some public places. Can we make an "Anti-smoking robot" to monitor smoking in real-time?

Tip: add a gas sensor based on this project.

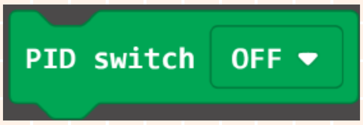
# Appendix 1. Maqueen Plus Block Description And Basic Tutorial

Technically, Maqueen Plus is a device and block is the tool to operate it, just like a TV set and remote controller. So we have to use the following blocks to make our Maqueen Plus "come alive". Let's get to know how these blocks work.



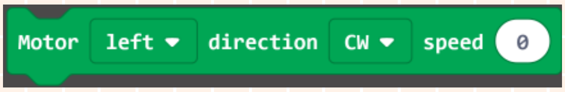
```
initialize via I2C until success
```

Initialize I2C to detect if the communication between micro:bit and Maqueen Plus is successfully established.
This block only needs to run once at the start of main program.
Init setting is essential when using I2C communication.
I2C Address: 0x10



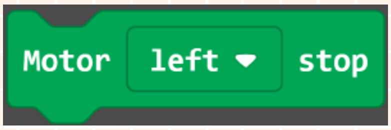
```
PID switch OFF ▾
```

PID algorithm is able to maintain the speed balance of motor.
Enable PID when using motor.
Two state: off/on.



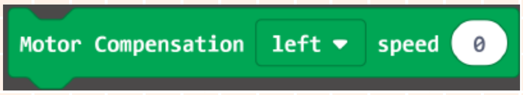
```
Motor left ▾ direction CW ▾ speed 0
```

Set the direction and speed for "left/right/all"motor.
Direction: forward, backward
Speed: 0~255



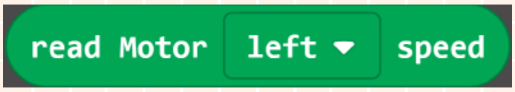
```
Motor left ▾ stop
```

Stop the "left/right/all"motor.



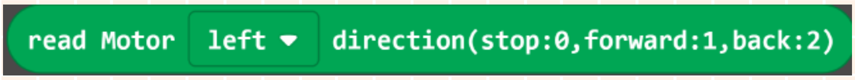
```
Motor Compensation left ▾ speed 0
```

When there is a speed difference between the two motors, set speed compensation for "left/right" motor to adjust speed difference.
Adjustable range: 0~10



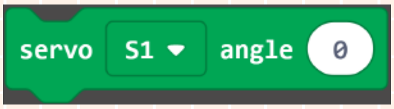
```
read Motor left ▾ speed
```

Read the current speed of "left/right"motor.



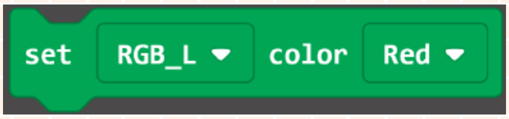
```
read Motor left ▾ direction(stop:0,forward:1,back:2)
```

Read the direction of "left/right" motor.
0: stop;
1: rotate forward;
2: rotate backward.



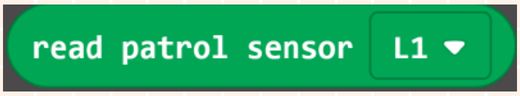
```
servo S1 ▾ angle 0
```

Set the angle of servo "S1-S3".
Range: 0° ~ 180°



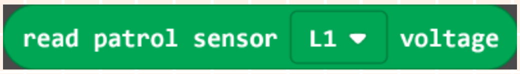
```
set RGB_L ▾ color Red ▾
```

Set the display color of "left/right/all"RGB LED. Seven colors to choose from.
Turn off the RGB LED.



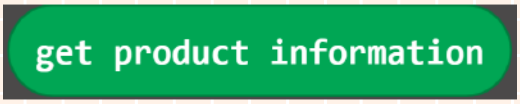
```
read patrol sensor L1 ▾
```

Read the value returned by line-tracking sensor"L1, L2, L3, R1, R2, R3".
Return value 1: detected black line.
Return value 0: no black line detected.



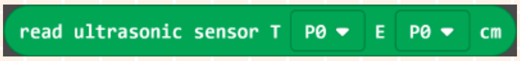
```
read patrol sensor L1 ▾ voltage
```

Read the grayscale value detected by line-tracking sensor "L1, L2, L3, R1, R2, R3".



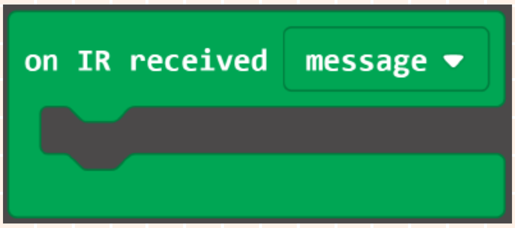
```
get product information
```

Get the current version of Maqueen Plus.



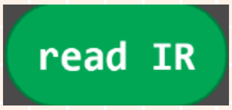
```
read ultrasonic sensor T P0 ▾ E P0 ▾ cm
```

Detect the distance between the ultrasonic sensor and the obstacle ahead. T: transmitting end; E: receiving end.
Set T and E to the corresponding GPIO ports according to hardware connection.



```
on IR received message ▾
```

The codes inside this block will run when IR receiver module received an external IR signal.



```
read IR
```

Read the received IR signal.
Display the code value in decimal.

### Program link for Maqueen Plus Basic Tutorial

Chapter 2-Let's move, Maqueen!: https://makecode.microbit.org/_Y4aajqg5Ac9F
Chapter 3-Walking Emoji: https://makecode.microbit.org/_UxCMVr3ttMm2
Chapter 4- City Defender-A Police Car: https://makecode.microbit.org/_D7UKTPVDiCrF
Chapter 5-Light Sensing Robot: https://makecode.microbit.org/_hq1HTt56dez0
Chapter 6-Moth Robot: https://makecode.microbit.org/_Ho4f769stFa6
Chapter 7-Little Ranging Expert: https://makecode.microbit.org/_hHpCq1AKcgwF
Chapter 8-Car Reversing Helper: https://makecode.microbit.org/_HDt2v5dz3XTi
Chapter 9-Line-tracking Robot: https://makecode.microbit.org/_4tx3s0K90W6m

Chapter 9-Line-tracking Robot： https://makecode.microbit.org/_4tx3s0K90W6m

Chapter 10-Tour of Crossroad： https://makecode.microbit.org/_0MThCUM8rXFT

Chapter 11-IR-controlled Robot： https://makecode.microbit.org/_LEwhMxMEYYF0

Chapter 12- Motion Sensing Robot---Transmitting End： https://makecode.microbit.org/_FhT3wChe89v7

Chapter 12- Motion Sensing Robot--- Receiving End： https://makecode.microbit.org/_XaF3XHXpWWCJ

Chapter 13-Crazy Racing -PID Enabled： https://makecode.microbit.org/_0YjYi7VprYEr

Chapter 13-Crazy Racing -PID Disabled： https://makecode.microbit.org/_TtCYvoKhMKL5

Chapter 14-Speed Recorder： https://makecode.microbit.org/_2Am8KaJhFERs

Chapter 15-Firefighting Robot： https://makecode.microbit.org/_H7V7Uk2gR1tv