



FREENOVE

FREE YOUR INNOVATION

Freenove is an open-source electronics platform.

www.freenove.com

Warning

When you purchase or use Freenove Micro Rover, please note the following:

- This product contains small parts. Swallowing or improper operation can cause serious infections and death. Seek immediate medical attention when the accident happened.
- Do not allow children under 3 years old to play with or near this product. Please place this product in where children under 3 years of age cannot reach.
- Do not allow children lack of ability of safe to use this product alone without parental care.
- Never use this product and its parts near any AC electrical outlet or other circuits to avoid the potential risk of electric shock.
- Never use this product near any liquid and fire.
- Keep conductive materials away from this product.
- Never store or use this product in any extreme environments such as extreme hot or cold, high humidity and etc.
- Remember to turn off circuits when not in use this product or when left.
- Do not touch any moving and rotating parts of this product while they are operating.
- Some parts of this product may become warm to touch when used in certain circuit designs. This is normal. Improper operation may cause excessively overheating.
- Using this product not in accordance with the specification may cause damage to the product.

About

Freenove is an open-source electronics platform. Freenove is committed to helping customer quickly realize the creative idea and product prototypes, making it easy to get started for enthusiasts of programing and electronics and launching innovative open source products. Our services include:

- Electronic components and modules
- Learning kits for Arduino
- Learning kits for Raspberry Pi
- Learning kits for micro:bit
- Learning kits for Technology
- Product customization service
- Robot kits
- Auxiliary tools for creations

Our code and circuit are open source. You can obtain the details and the latest information through visiting the following web sites:

<http://www.freenove.com>

<https://github.com/freenove>

Your comments and suggestions are warmly welcomed, and please send them to the following email address:
support@freenove.com

If you have any business matters, please feel free to contact us:

sale@freenove.com

 support@freenove.com

References

This product is named Freenove Micro Rover.

You can download the sketches and references used in this product in the following websites:

<http://www.freenove.com>

https://github.com/Freenove/Freenove_Micro_Rover

If you have any difficulties, you can send email to technical support for help.

Support

Freenove provides free and quick technical support, including but not limited to:

- Quality problems of products
- Problems in using products
- Questions for learning and technology
- Opinions and suggestions
- Ideas and thoughts

Please send email to:

support@freenove.com

On working day, we usually reply to you within 24 hours.

Copyright

Freenove reserves all rights to this book. No copies or plagiarizations are allowed for the purpose of commercial use.

The code and circuit involved in this product are released as Creative Commons Attribution ShareAlike 3.0.

This means you can use them on your own derived works, in part or completely, as long as you also adopt the same license. Freenove brand and Freenove logo are copyright of Freenove Creative Technology Co., Ltd and cannot be used without formal permission.

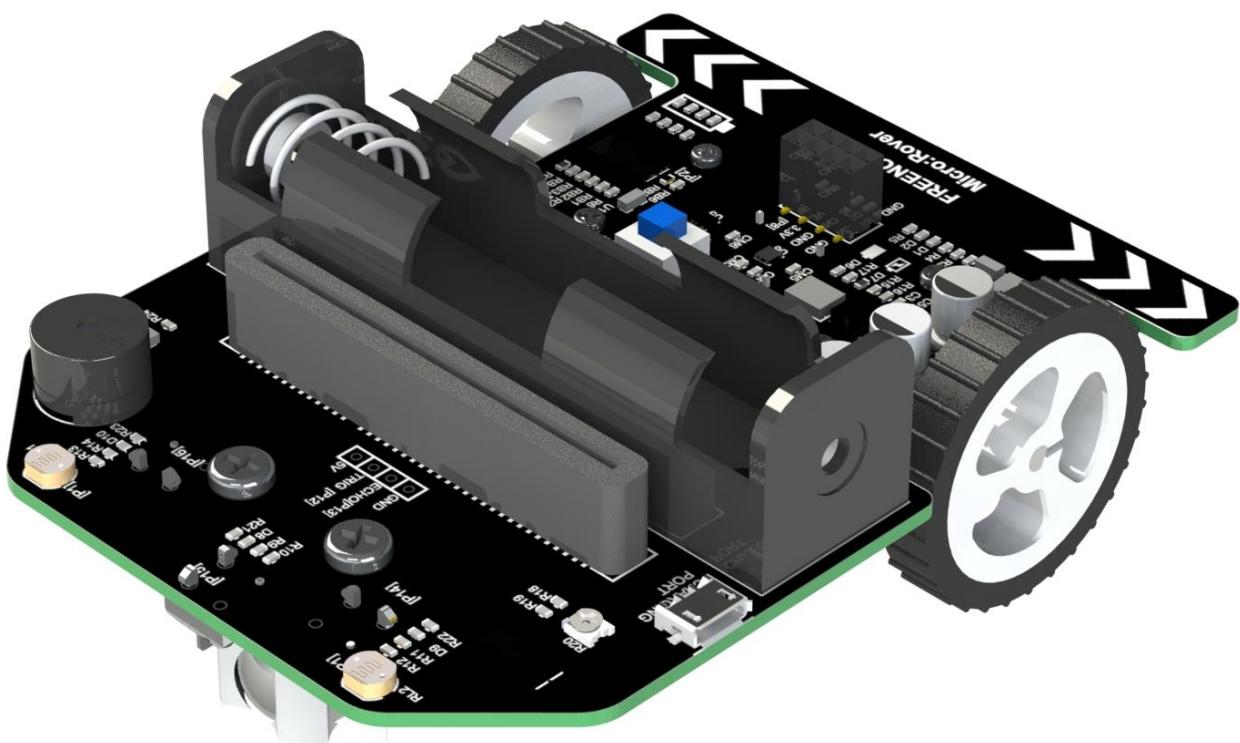
Contents

Contents.....	1
List.....	1
Preface.....	4
micro:bit.....	5
Meet micro:bit.....	5
Features.....	6
Hardware	7
Micro:Rover.....	8
Meet micro:rover.....	9
Features.....	11
Battery & Charging.....	12
Indicator	13
Assemble.....	14
Code & Programming.....	17
Quick Start.....	17
Makecode.....	21
Quick download	22
Extension for Makecode	24
Resource and code.....	29
Import Code.....	30
Android app.....	32
Install Freenove app.....	32
Chapter 1 Music	35
Preparation.....	35
Play a note.....	36
Play a melody.....	38
Play custom melody.....	39
Chapter 2 RGB LED.....	40
Preparation.....	40
Show one color.....	41
Show different colors.....	42
Show random color.....	43

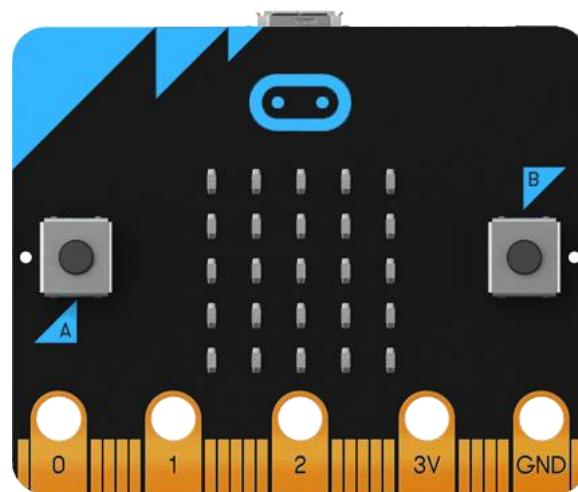
Show soft colors	44
Chapter 3 Ultrasonic Ranging.....	45
Preparation.....	45
Obtain value of ultrasonic ranging	46
Rover-Obstacle avoidance mode -1.....	47
Rover-Obstacle avoidance mode -2.....	49
Chapter 4 Light tracing	50
Preparation.....	50
Get value of light intensity sensor	51
Rover-light tracing mode.....	52
Chapter 5 Line tracking.....	54
Preparation.....	54
Get value of LineTrackingSensor	55
Rover-line tracking mode	56
Chapter 6 Bluetooth.....	58
Preparation.....	58
Getting Bluetooth Data	59
Bluetooth pairing	64
Rover-Remote control mode.....	67
Chapter 7 Android App	70
Preparation.....	70
Rover	71
Next	73
Appendix.....	74
LEDs	74
Motors	75
Sensors	75
Commands.....	76
What's next?	77

List

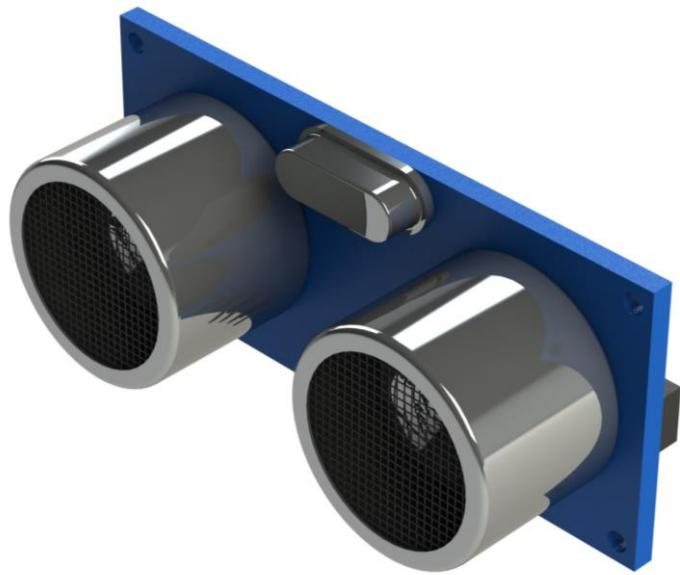
Micro:Rover



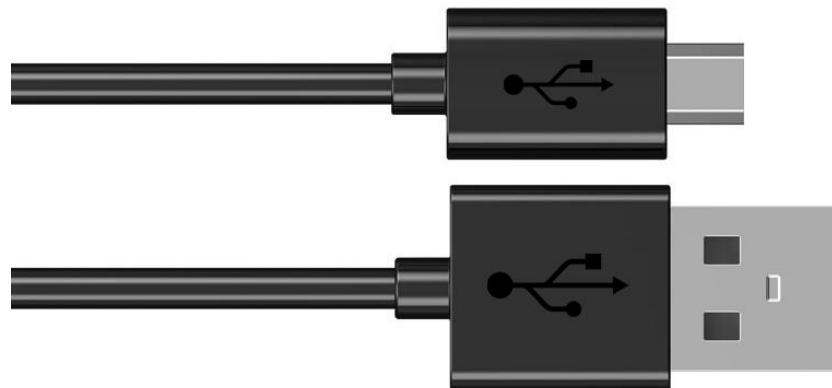
BBC micro:bit



Ultrasonic ranging module



Micro USB cable



Line-Tracking Map





Preface

Do you want to learn programming or get your own programming robot?

Currently, Program is developed into the lower age group, and everyone programming is a trend. From Arduino and Raspberry Pi to micro:bit, simple graphical programming makes child programming a possibility. Maybe you haven't heard of them, it doesn't matter. With this product and the tutorial, you can easily complete a multi-functional programming car and experience the fun as a Maker.

micro:bit is a powerful and simple development board. Even if you've never programmed before, simple graphical programming interface can make you easy to learn. It doesn't need any professional programming software. You just need a browser to program it. So, whether your computer system is Windows, Linux or Mac. They can all do it. And you can also program it with Python.

It has a lot of fans in the world. They are keen to exploration, innovation and DIY and they contributed a great number of high-quality open source code, circuit and rich knowledge base. So we can realize our own creativity more efficiently by using these free resource. Of course, you can also contribute your own strength to the resource.

Freenove Micro:Rover is one of them. Rover is a programmable car developed by Freenove and it is based on BBC micro:bit. It adopts integrated design, so that you don't need additional wiring, and it is easy to use. It has ultrasonic, infrared, photosensitive sensors. And it also has buzzer, RGB LED and other peripherals. Rich hardware resources will make you master more knowledge and skills. You can use your imagination to create more ways to play.

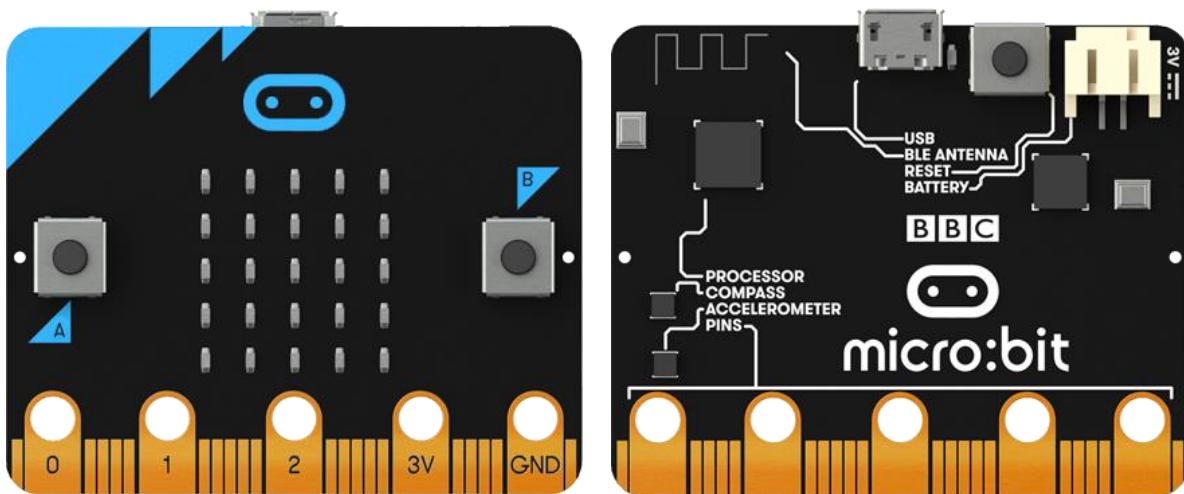
In each learning chapter of this tutorial, we provide program source code with detailed program explanations and burnable binaries. So you can really understand the meaning of each section of program.

Additionally, if you have any difficulties or questions about this tutorial and the kit, you can always ask us for quick and free technical support.

micro:bit

Hello, welcome to the world of electronic and programming. Let's start micro:bit and Micro:Rover tour from here.

Meet micro:bit

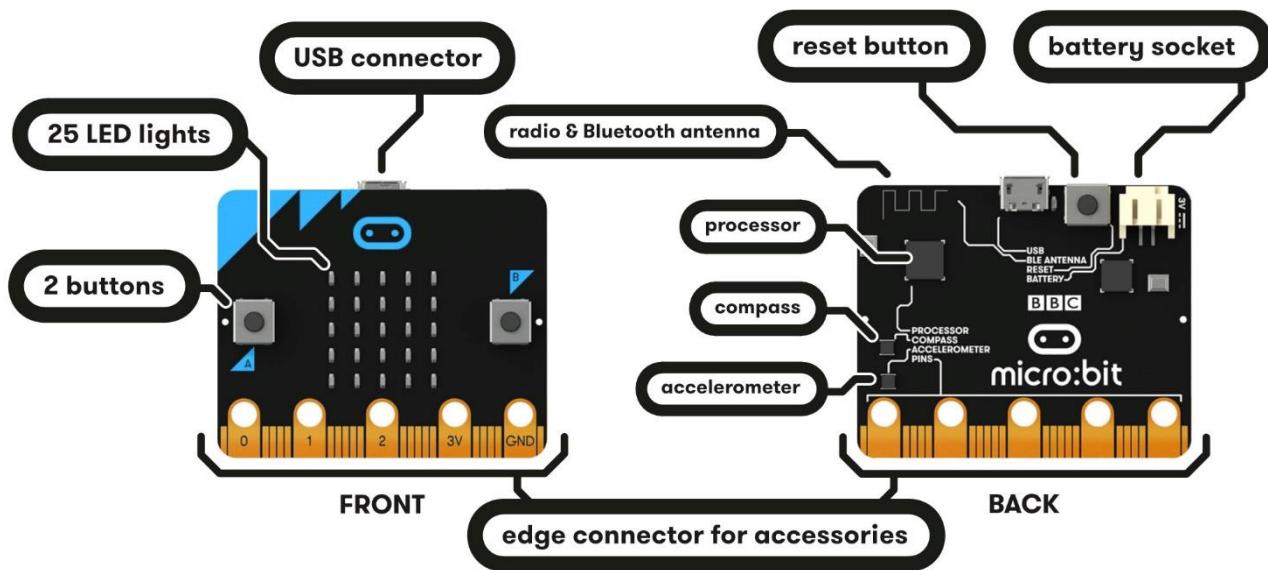


The BBC micro:bit is a handheld, programmable micro-computer that can be used for all sorts of cool creations, from robots to musical instruments. The possibilities are endless.

For more contents, please refer to:

<https://microbit.org/guide/>

Features



Your micro:bit has the following physical features:

- 25 individually-programmable LEDs
- 2 programmable buttons
- Physical connection pins
- Light and temperature sensors
- Motion sensors (accelerometer and compass)
- Wireless Communication, via Radio and Bluetooth
- USB interface

For more contents, please refer to:

<https://microbit.org/guide/features/>

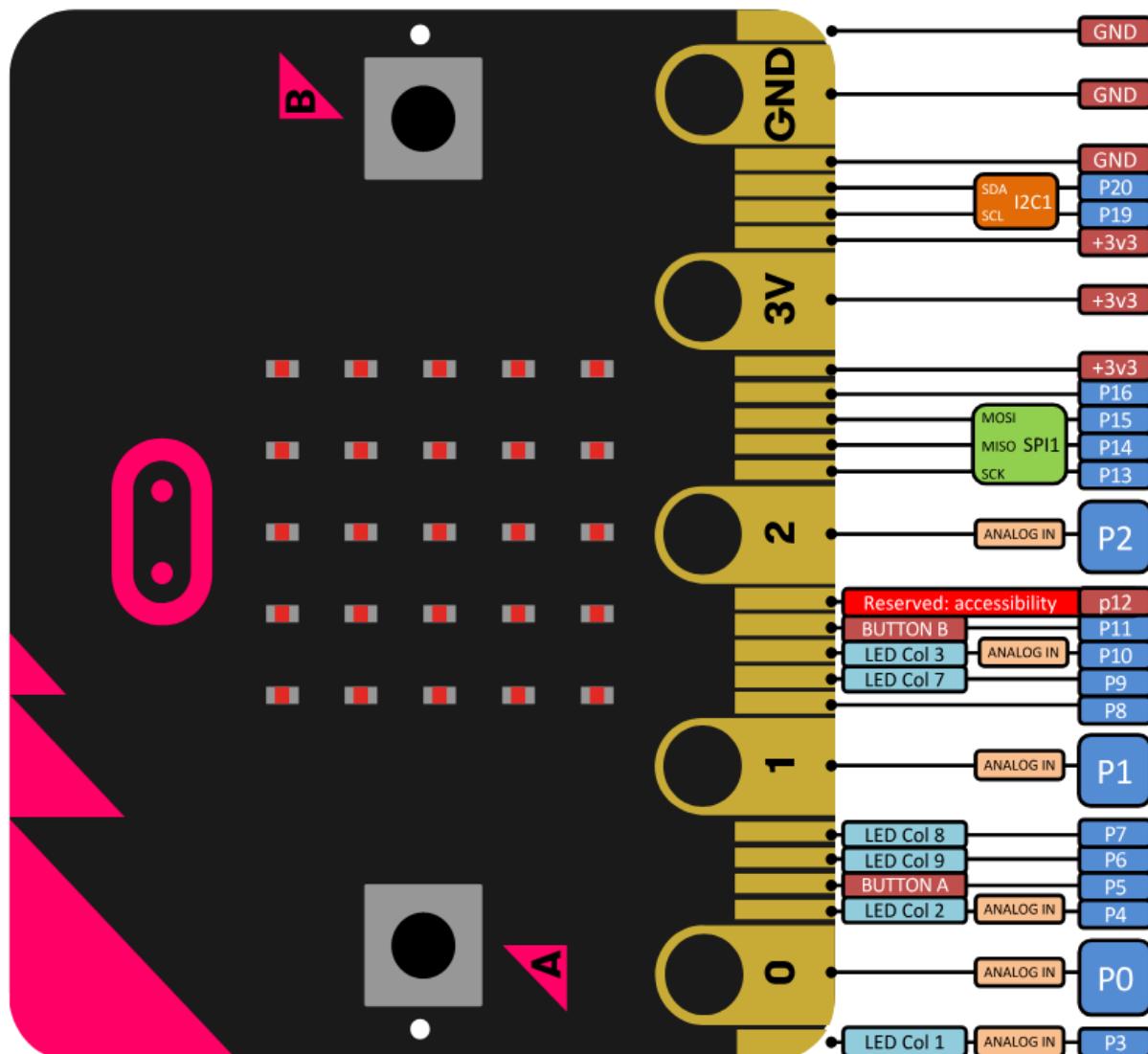
Hardware

For beginners, this part of information is not necessary. Just have a brief understanding. But if you want to be a developer, hardware information will be very helpful. Detailed hardware information about micro:bit can be found here: <https://tech.microbit.org/hardware/>.

To complete Rover, we just need a brief understanding of GPIO.

GPIO

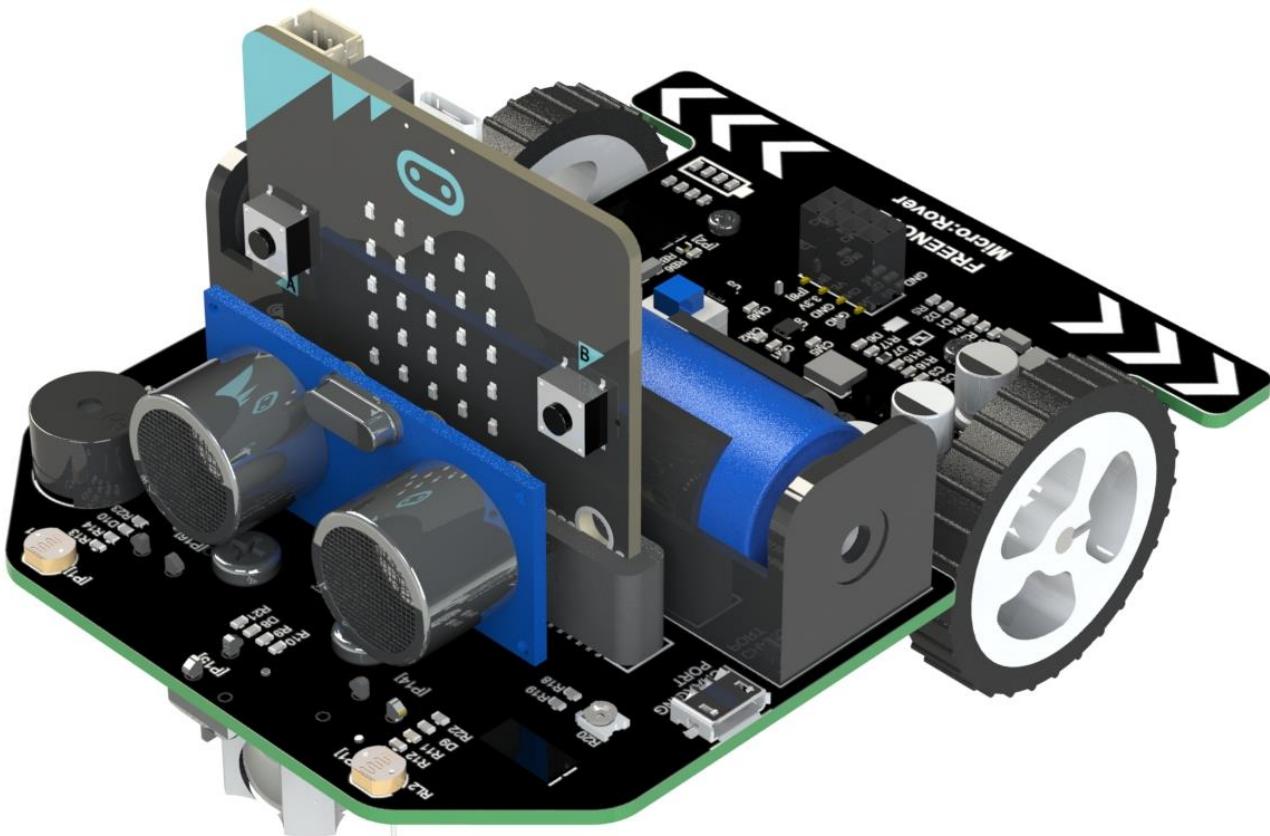
GPIO, namely General Purpose Input/output Pins, is a part of micro:bit. It is an important part for connecting external devices. All sensors and devices on Rover communicate with each other through micro:bit GPIO. The following is the GPIO serial number and function diagram of micro:bit:



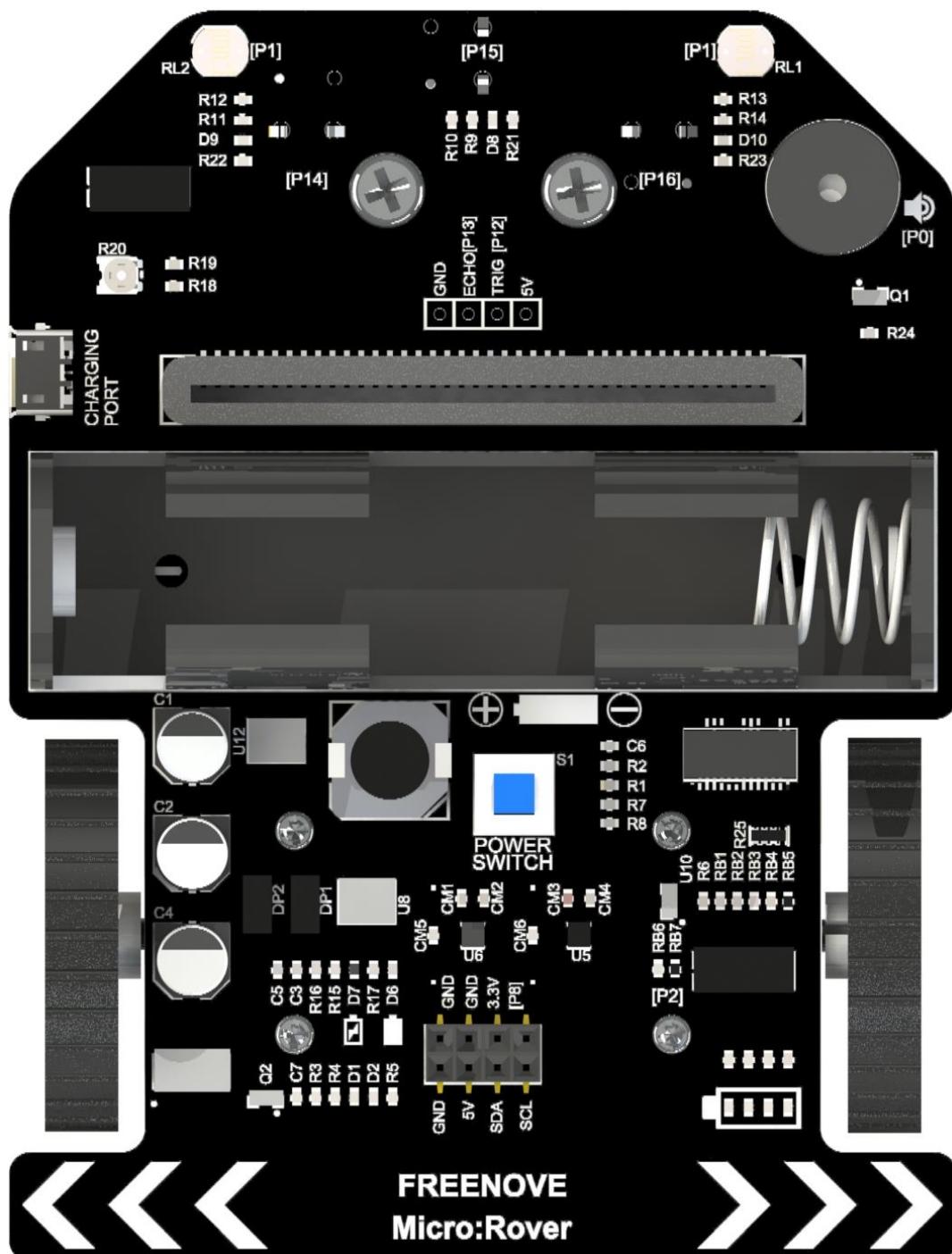
Micro:Rover

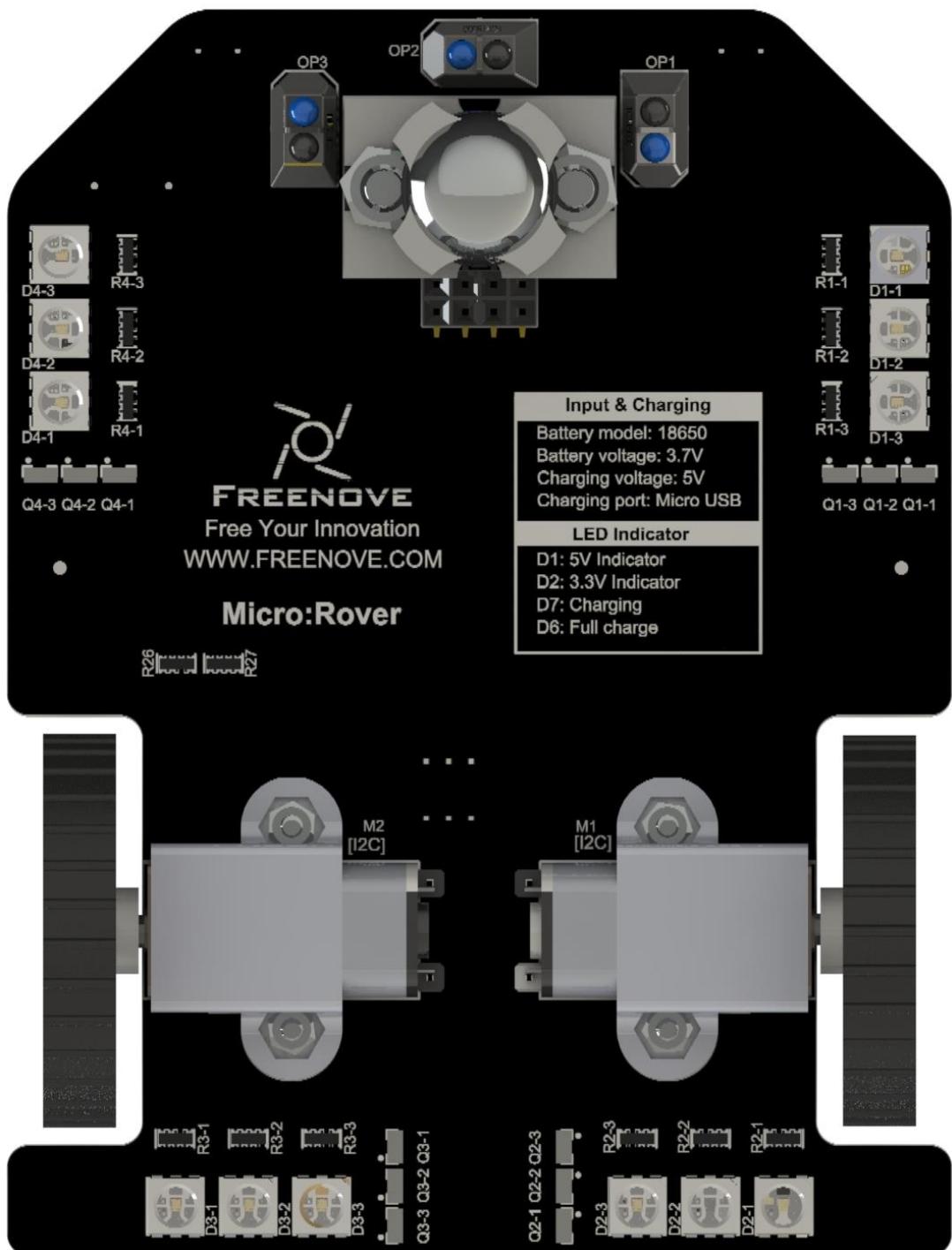
This chapter will introduce the functions and futures of Micro:Rover.

Freenove Micro:Rover is a programmable car and it is based on BBC micro:bit. It adopts integrated design, so that you don't need additional wiring, and it is easy to use. It has ultrasonic, infrared, photosensitive sensors. And it also has buzzer, RGB LED and other peripherals. Rich hardware resources will make you master more knowledge and skills. You can use your imagination to create more ways to play.



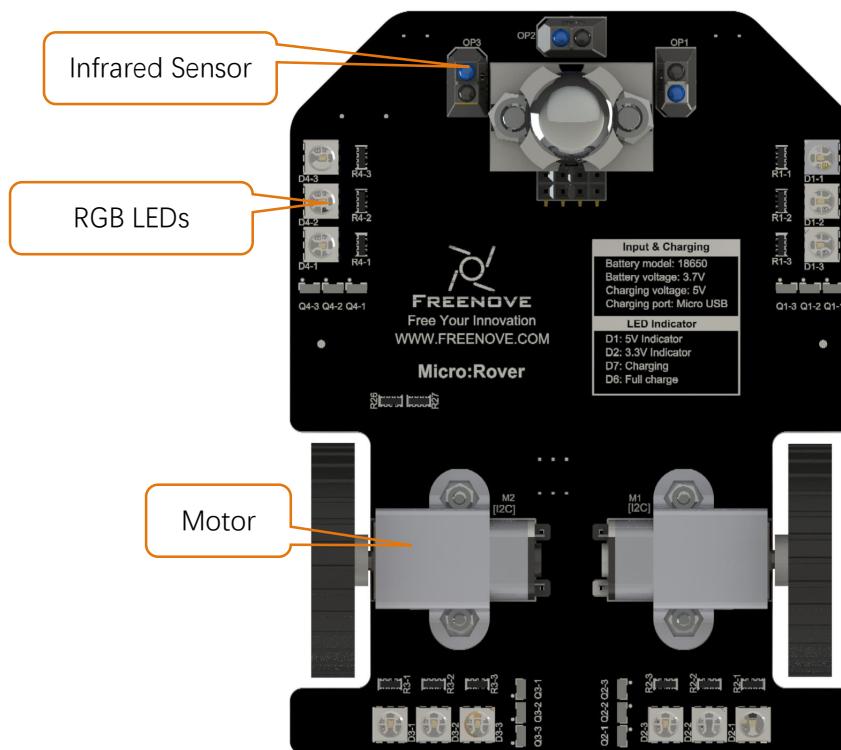
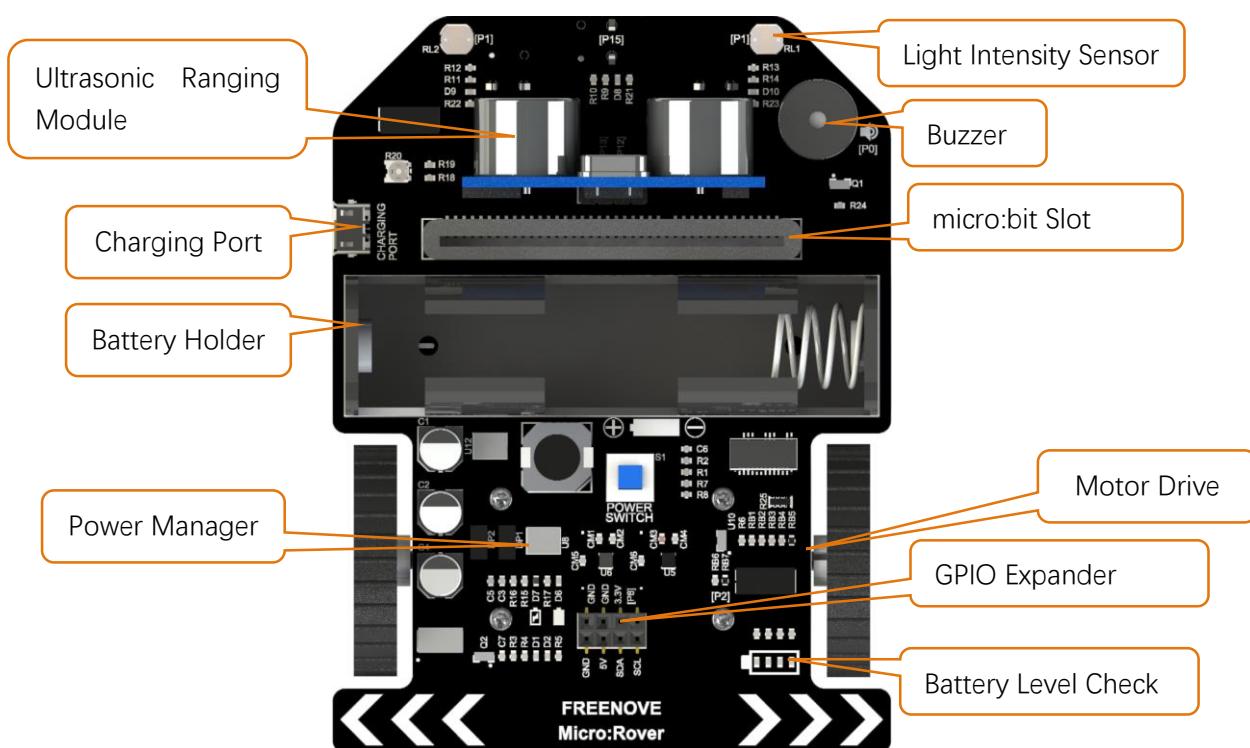
Meet micro:rover





The Freenove Micro:Rover is a multi-functional car, based on BBC micro:bit. It has rich sensors and peripherals to help you further learn how to use micro:bit and knowledge about electronics.

Features





Features of Micro:Rover:

- 4 Sets of controllable RGB LEDs (3 LEDs in each set)
- 2*Deceleration motors
- 1*Passive buzzer
- 1*Ultrasonic ranging module
- 2*Light intensity sensors
- 3*Infrared line-tracking sensors
- 1*micro:bit GPIO [P8]
- 1 Set of micro:bit I2C interface
- Battery power indication
- Battery charging and discharging management

Connection between Micro:Rover and micro:bit GPIO:

micro:bit GPIO	Micro:Rover
P0	Buzzer
P1	Light intensity sensor
P2	Battery voltage acquisition
P8	Expansion port
P12	Ultrasonic ranging module trig pin
P13	Ultrasonic echo pin
P14	Left Infrared line-tracing sensor
P15	Middle Infrared line-tracing sensor
P16	Right Infrared line-tracing sensor
P19-I2C SCL	Motors, RGBLEDs, I2C Extension port
P20-I2C SDA	

Battery & Charging

Freenove Micro:Rover is powered by one 18650 battery. The extended battery holder can be compatible with any type of 18650 batteries. Different types of batteries, such as pointed/flat/with or without protective plate, can be used. The discharge current of the battery should be at least 2A.

You can charge the battery with a USB port or a common USB charger. The maximum charging current is 700 mA.

Please note: this product does not contain battery.

Indicator

Some LEDs are integrated on Rover. Each LED state indicates the corresponding content. You can know better about working state of Rover through them.

LED	Indicating content	LED On	LED Off
D1	5V circuit	5V circuit is working.	5V circuit is not working.
D2	3.3V circuit	3.3V circuit is working.	3.3V circuit is not working.
D8	Middle line-tracking sensor	Black object is detected or no object is detected.	White object is detected
D9	Left line-tracking sensor		
D10	Right line-tracking sensor		

LED D6 and D7 are a set of LEDs, which indicate battery charging and discharging state. Under normal working conditions, their states and meanings are shown below.

D7 State	D6 State	Indicating meaning
On	Off	Charging.
Off	On	Charging is complete and the battery is full.
Blink	On	The charger did not detect the battery.
Off	Off	The boost circuit is working.

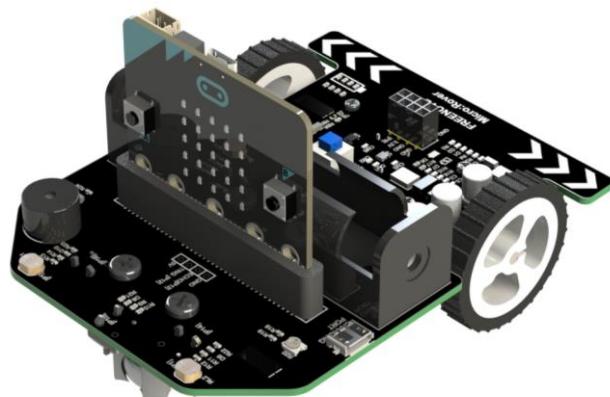
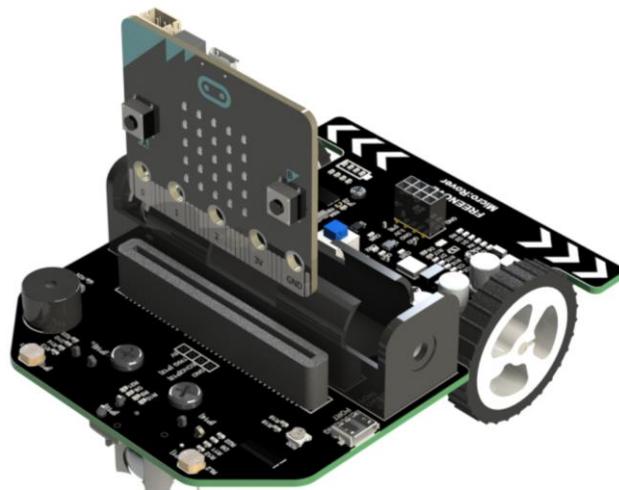
On bottom right corner of Rover, four LEDs are used to indicate battery level, as shown blow.



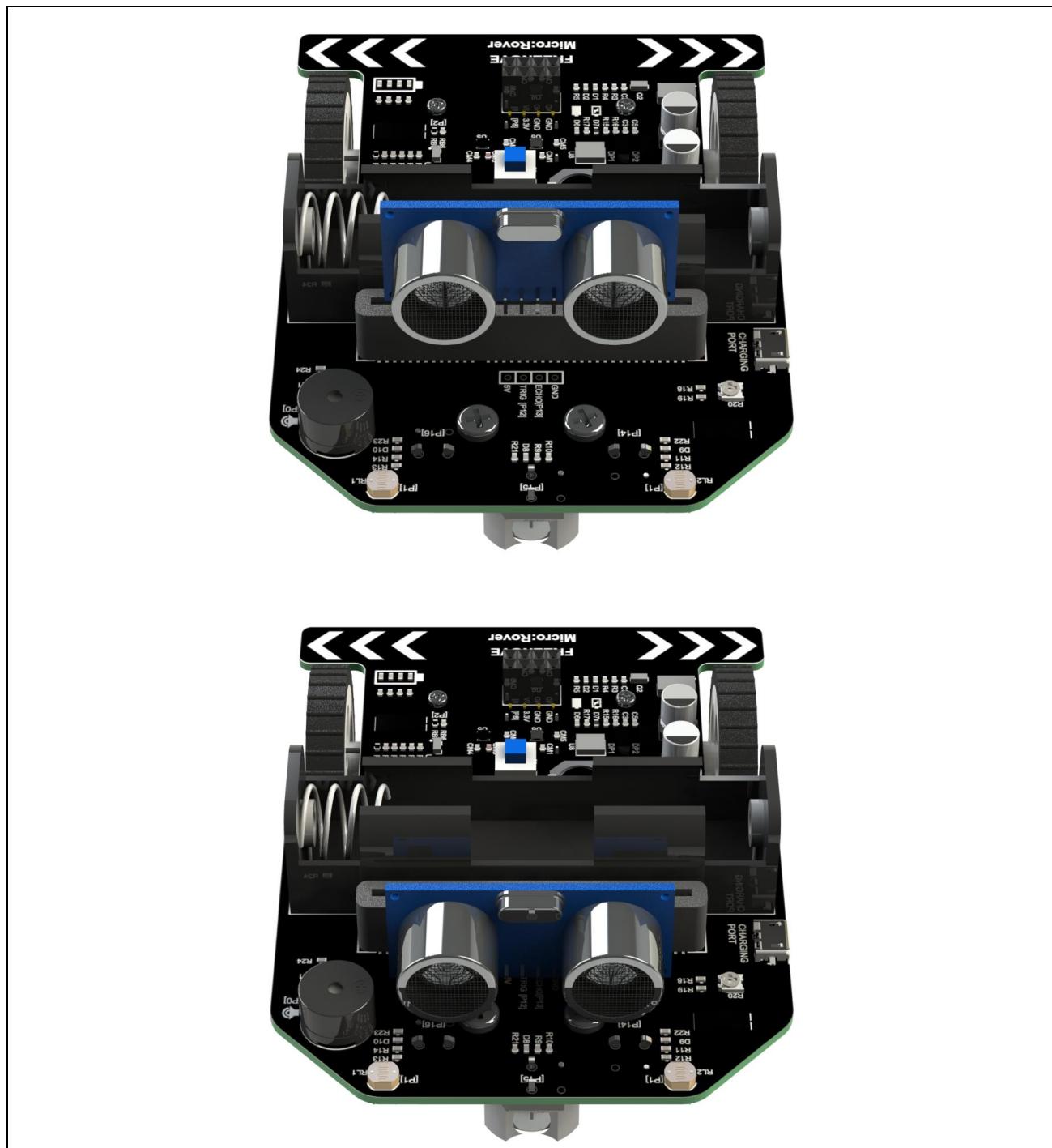


Assemble

Install micro:bit

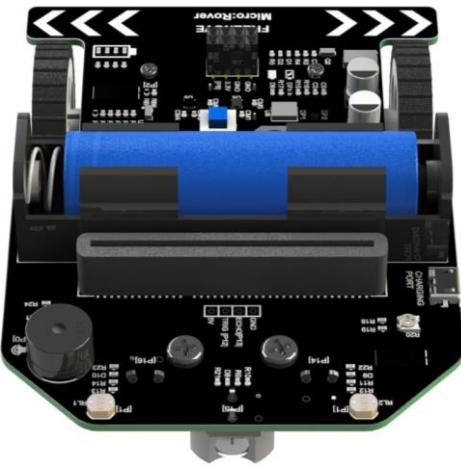


Install ultrasonic ranging module





Install battery



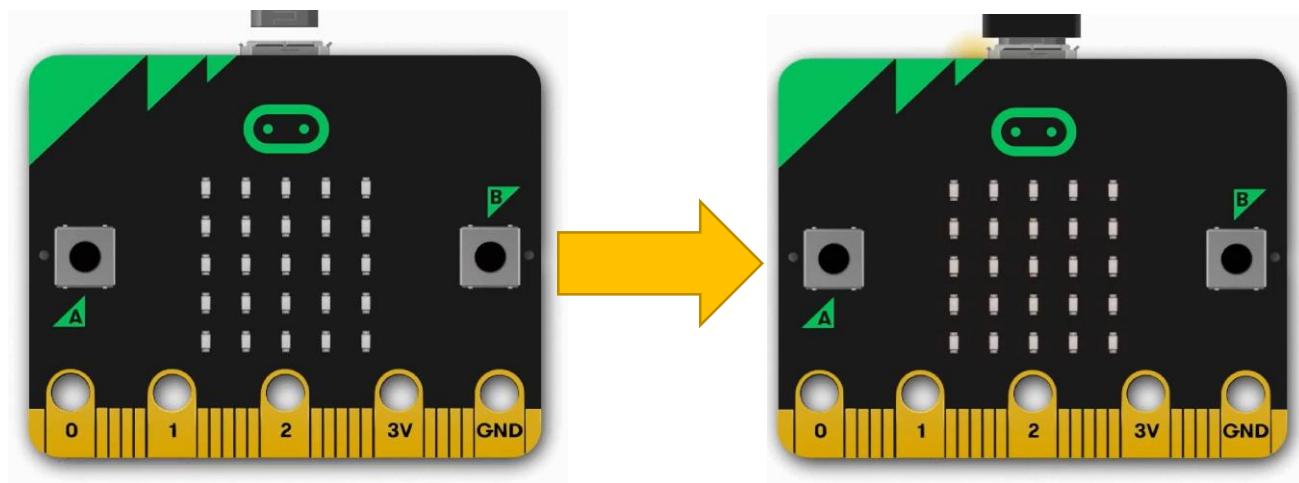
Code & Programming

Quick Start

This section describes how to write program for micro:bit and how to download them to micro:bit. There are very detailed tutorials on the official website. You can refer to: <https://microbit.org/guide/quick/>.

Step 1: Connect It

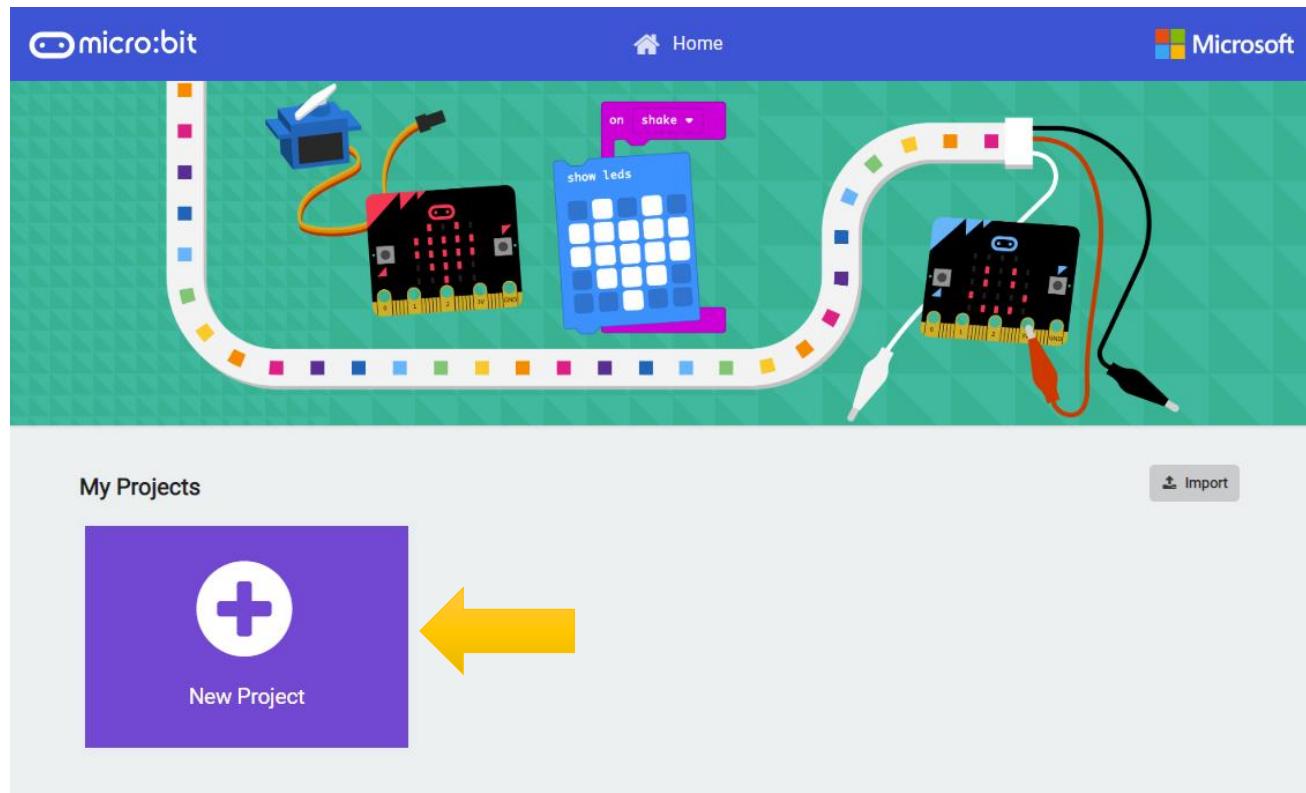
Connect the micro:bit to your computer via a micro USB cable. Macs, PCs, Chromebooks and Linux systems (including Raspberry Pi) are all supported. It comes with a fun application, give it a try!



Step 2: Program It

Visit <https://makecode.microbit.org/>. Then click "New Project" and start programming.

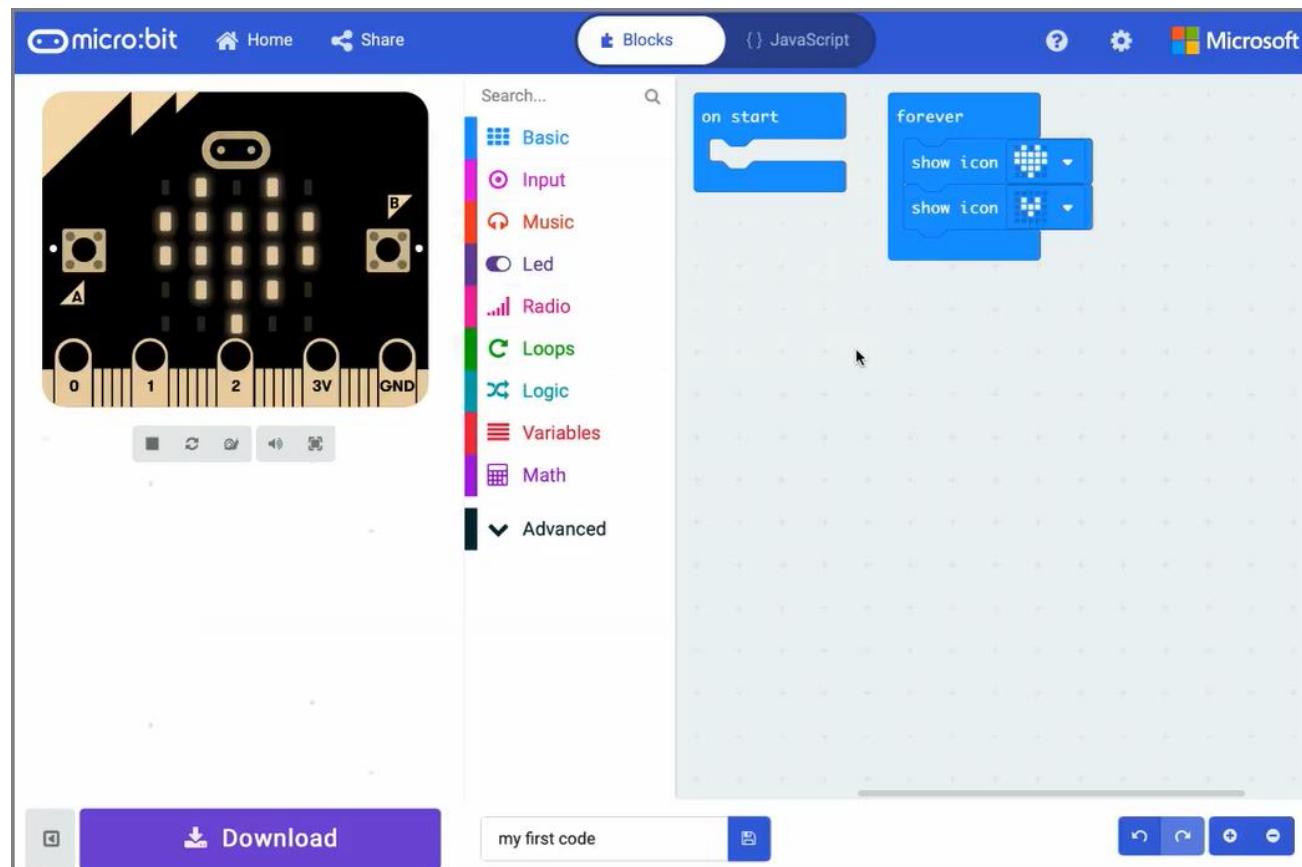
If your computer has Windows 10 operating system, you can also use Windows 10 App for programming, which is exactly the same as programming on browsers. [Get windows 10 App\(Click\)](#).



Write your first micro:bit code. For example, drag and drop some blocks and try your program on the Simulator in the MakeCode Editor, like in the image below that shows how to program a Flashing Heart.

Here is a demo video: <https://microbit.org/images/quickstart/makecode-heart.mp4>

Makecode will be further introduced in next section.



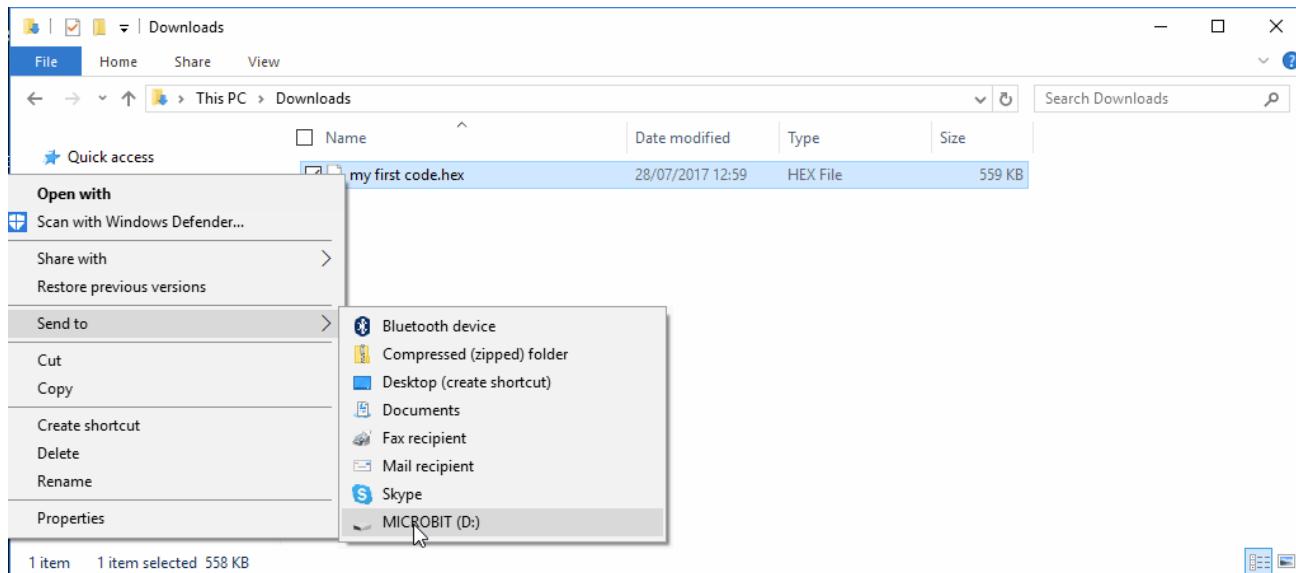


Step 3: Download It

If you write program by using Windows 10 App, you just need click the "Download" button, then the program will be downloaded directly to micro:bit without any other actions.

If you write program by using browser, please follow steps below:

Click the Download button in the editor. This will download a 'hex' file, which is a compact format of your program that your micro:bit can read. Once the hex file has downloaded, copy it to your micro:bit just like copying a file to a USB drive. On Windows you can right click and choose "Send to→MICROBIT."



Step 4: Play It

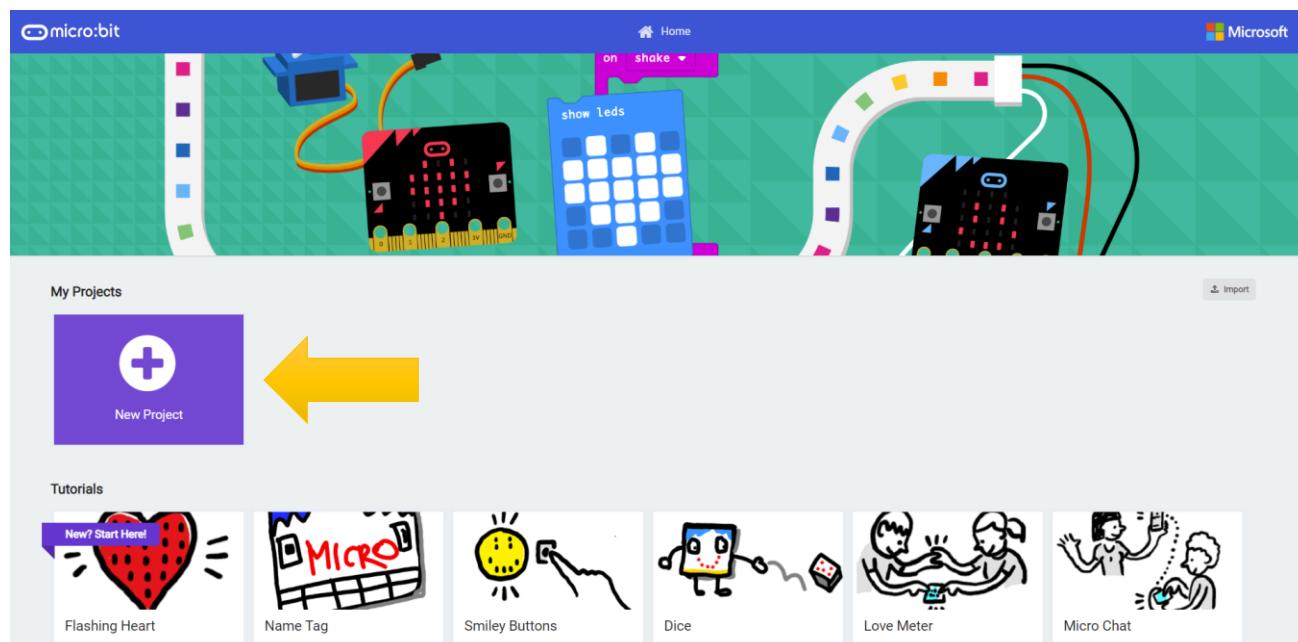
The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is programmed. Once that's finished the code will run automatically!

Waring

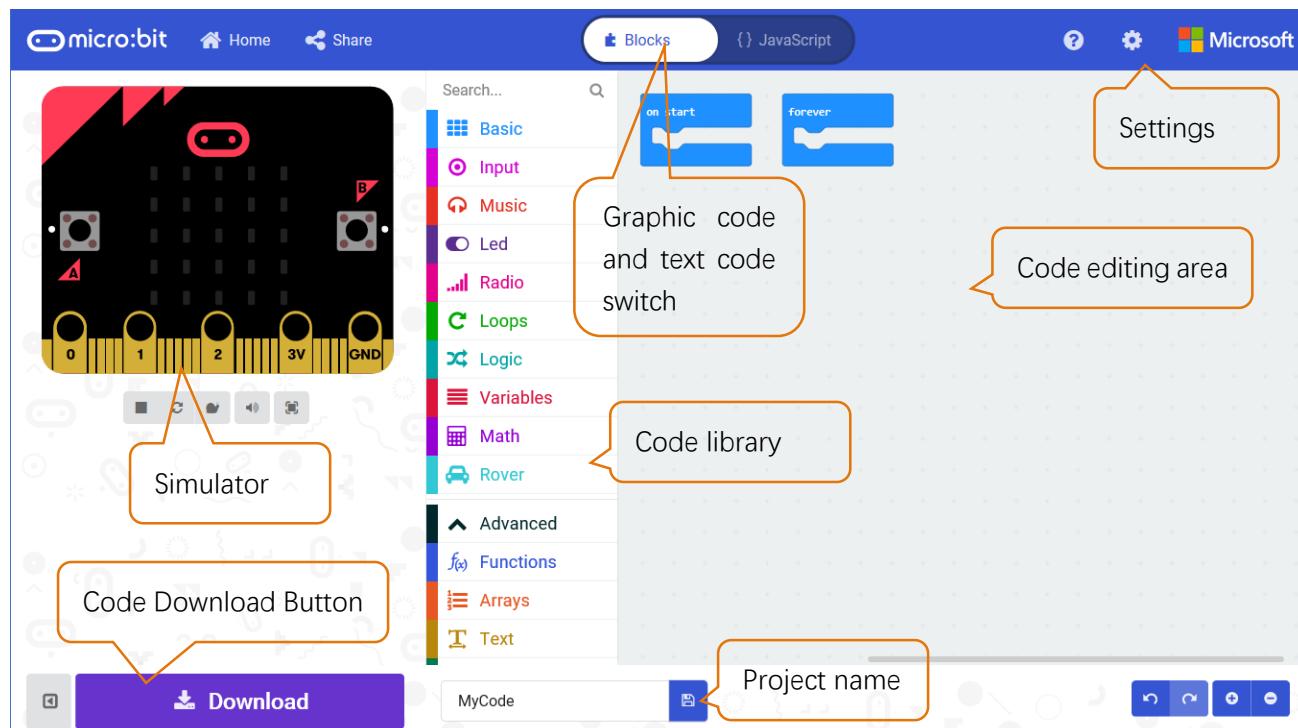
The MICROBIT drive will automatically eject and come back each time you program it, but your hex file will be gone. The micro:bit can only receive hex files and won't store anything else!

Makecode

Open web version of [makecode](#) or windows 10 app version of makecode.
<https://makecode.microbit.org/>



Click “New Project”, Makecode editor is as below:



In the code area, there are two fixed blocks “on start” and “forever”.

The code in the “on start” block will be executed only once after power-on or reset. And the code in “forever” block will be executed cyclically.



Quick download

As mentioned earlier, if you use Windows 10 App of makecode, you can quickly download the code to micro:bit by clicking the download button.

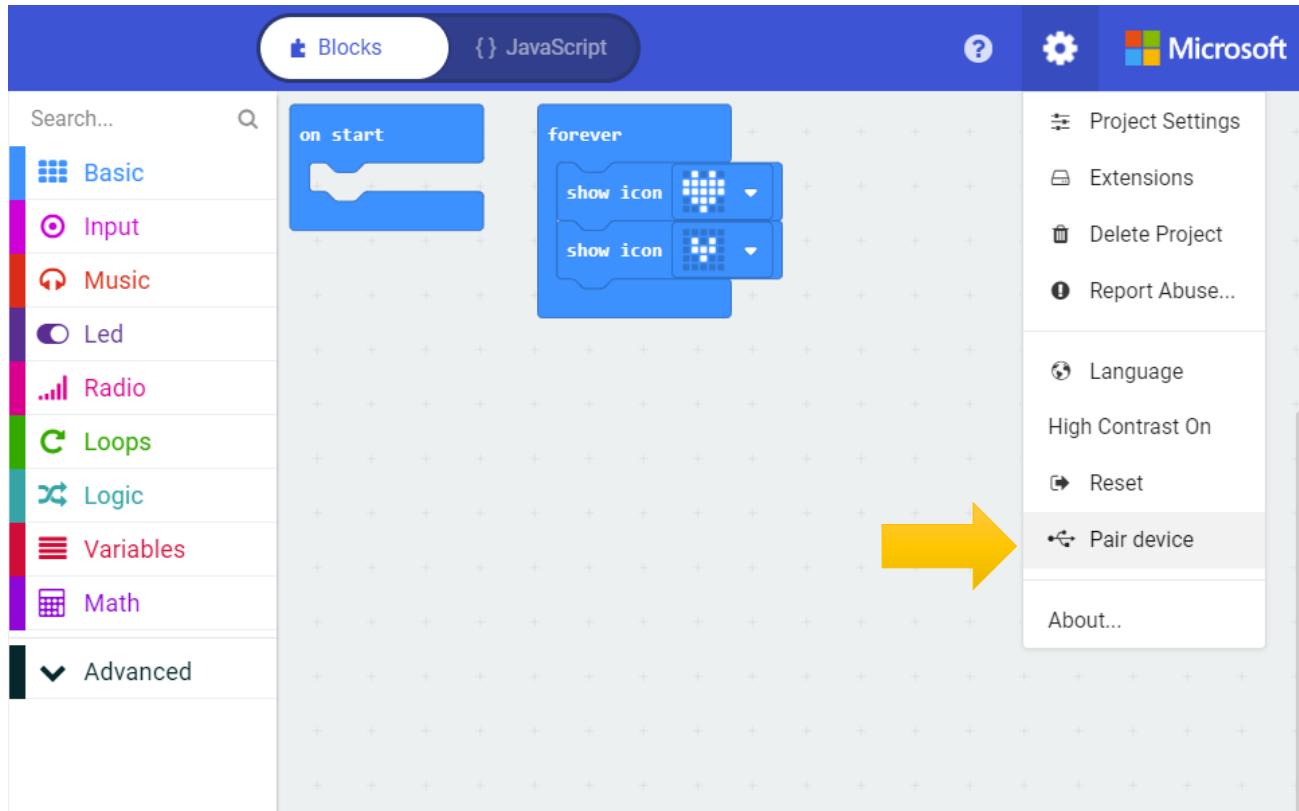
Using browser version of makecode may require more steps. But if you use **Google Chrome 65+ for platform Android, Chrome OS, Linux, macOS and Windows 10**, you can also realize the function of quick download.

Here we use webUSB feature of Chrome, which allows web pages to access your USB hardware devices. We will complete the connection and pairing of the micro:bit device with the webpage in following steps.

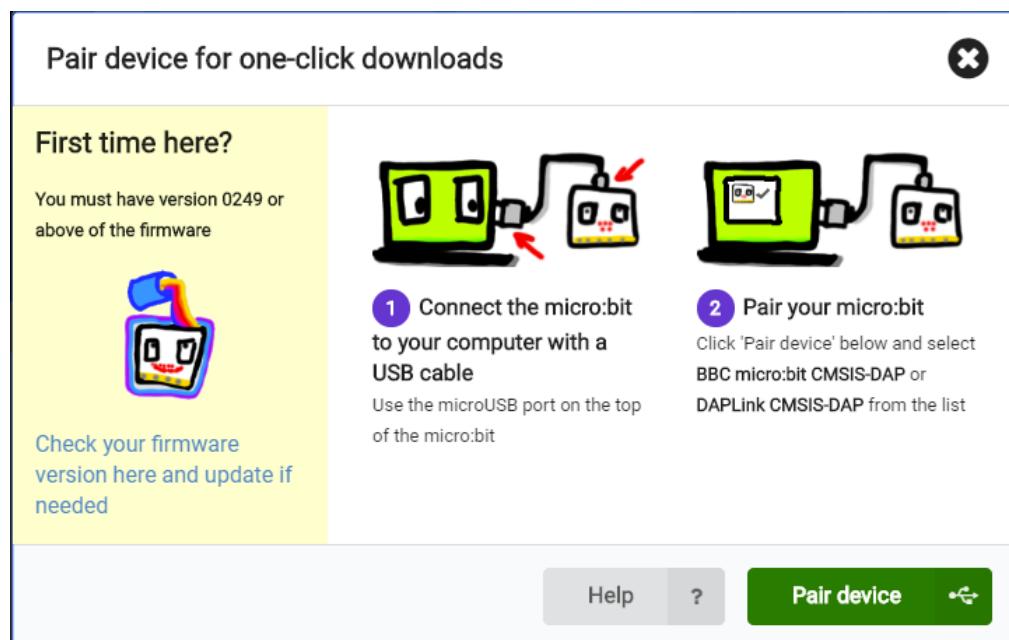
Pair device

Connect the computer and micro:bit with a USB cable.

Click the gear menu in the top right corner and then click on "Pair device".



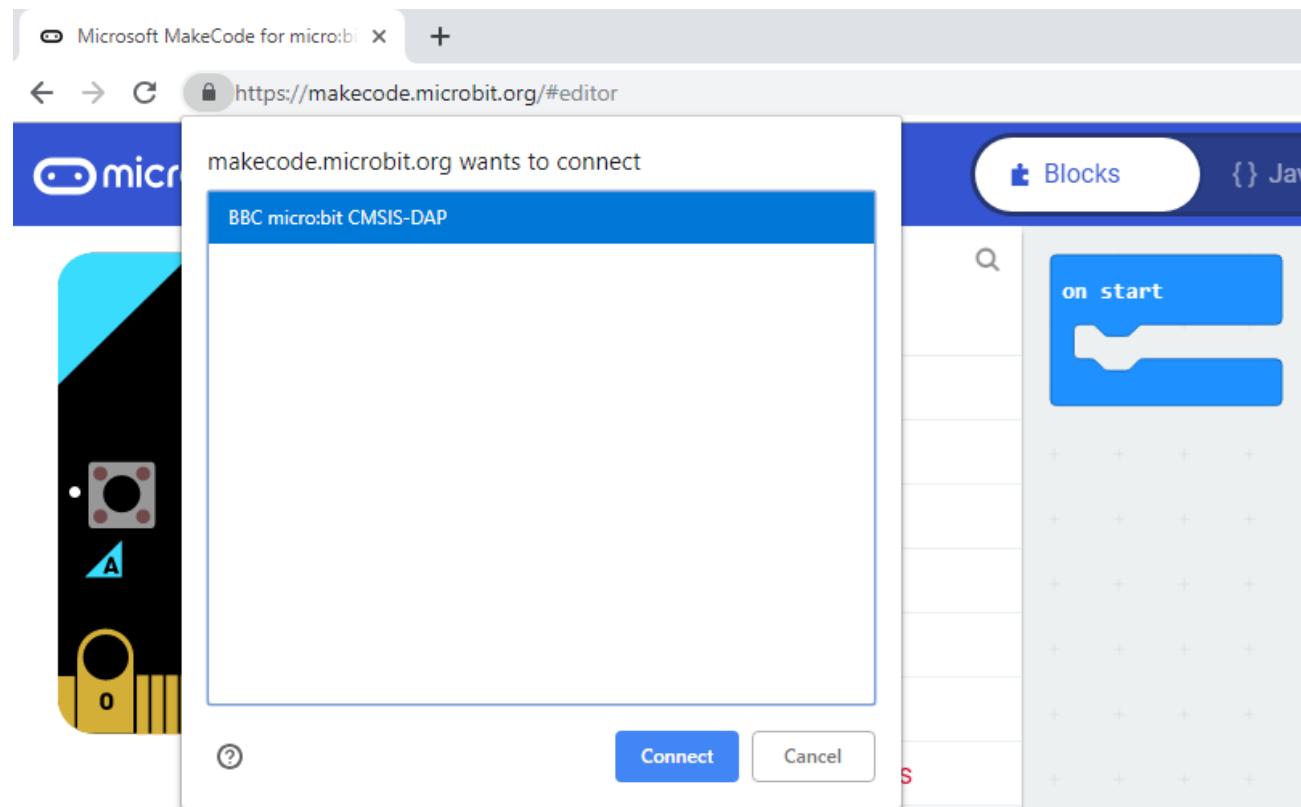
Then continue to click "Pair device" button.



Select device in popup window and click "Connect" button. If there are no devices in the pop window, please refer to following content: <https://makecode.microbit.org/device/usb/webusb/troubleshoot>

And we save the page as a file "Troubleshooting downloads with WebUSB - Microsoft MakeCode.pdf". You can read it directly in the folder of this tutorial.

And the file "Firmware microbit.pdf" introduces how to update firmware of micro:bit. Its content come from: <https://microbit.org/guide/firmware/>



After the connection succeeds, click the Download button and the program will be downloaded directly to Micro: bit.



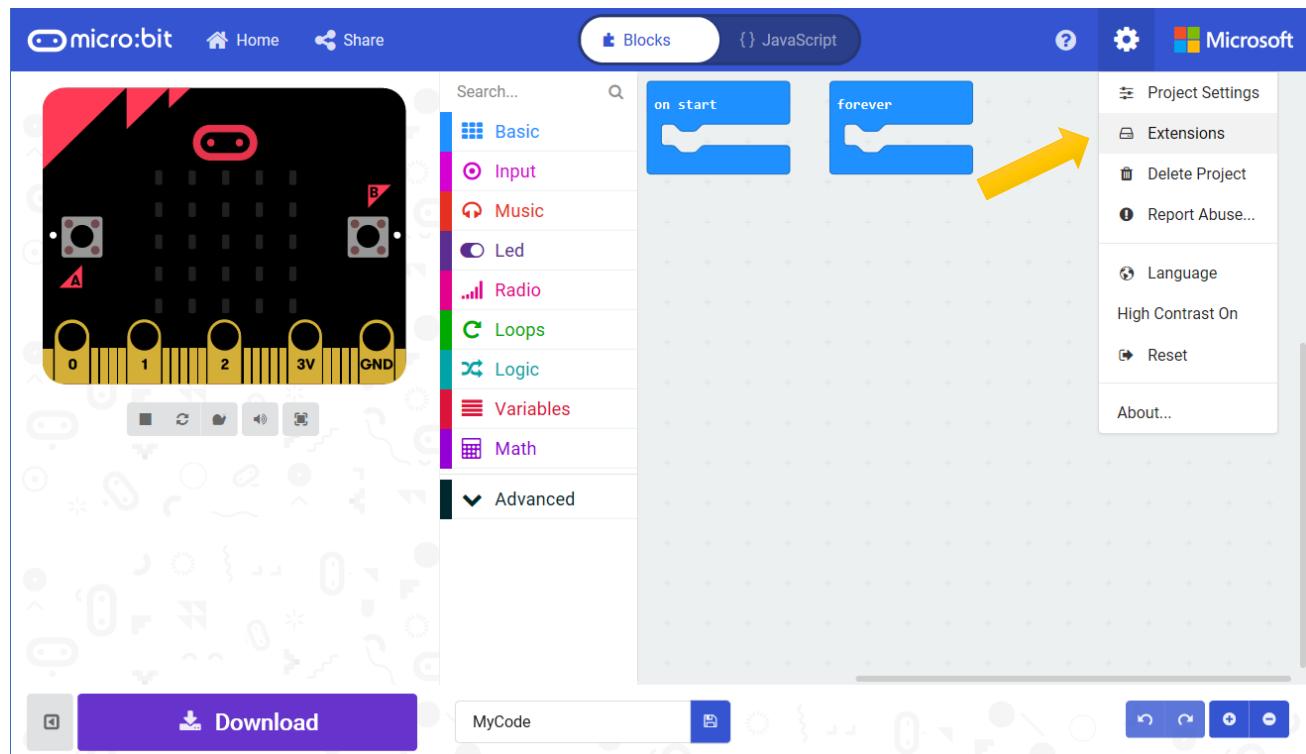
Extension for Makecode

In order to use Rover easily, we make a makecode Extension for Rover.

Add Rover Extension

You can add Rover Extension through following method.

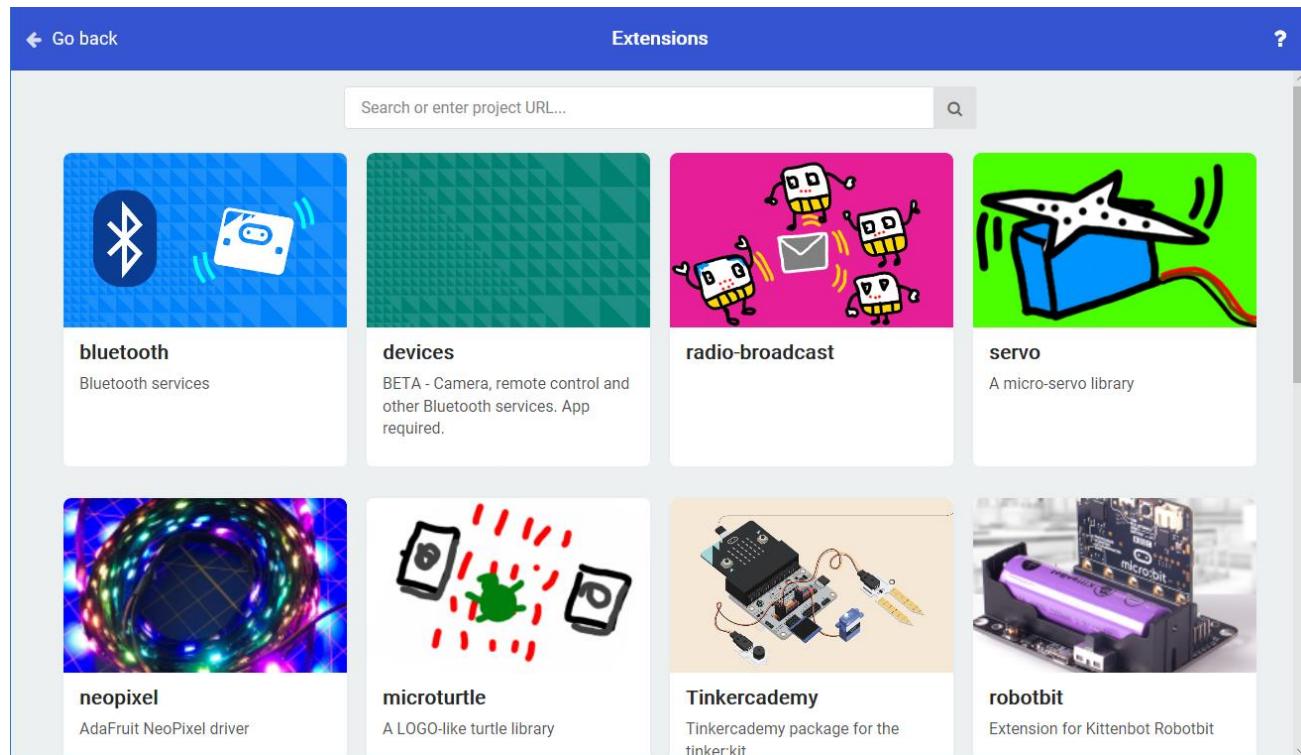
Open makecode, under any project, click gear icon(settings) → Extensions in turn.



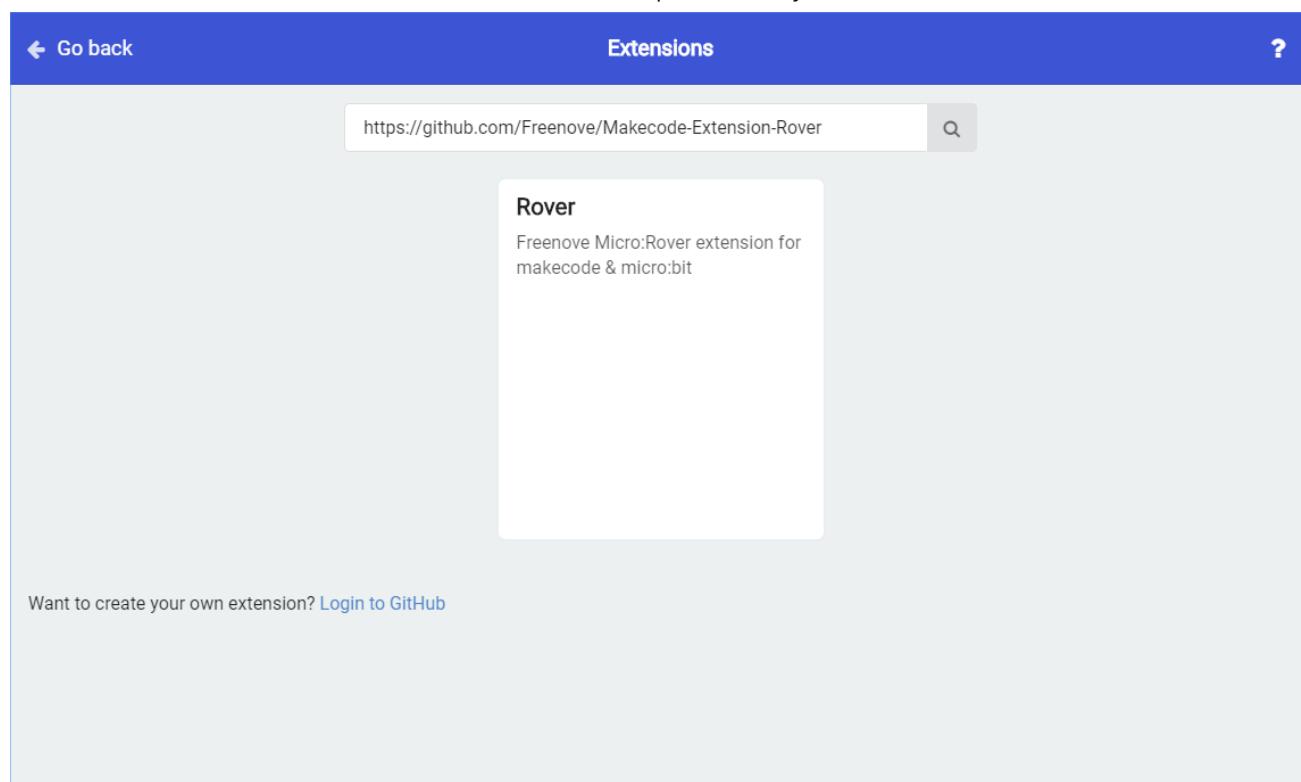
Enter this link in the search box:

<https://github.com/Freenove/Makecode-Extension-Rover>

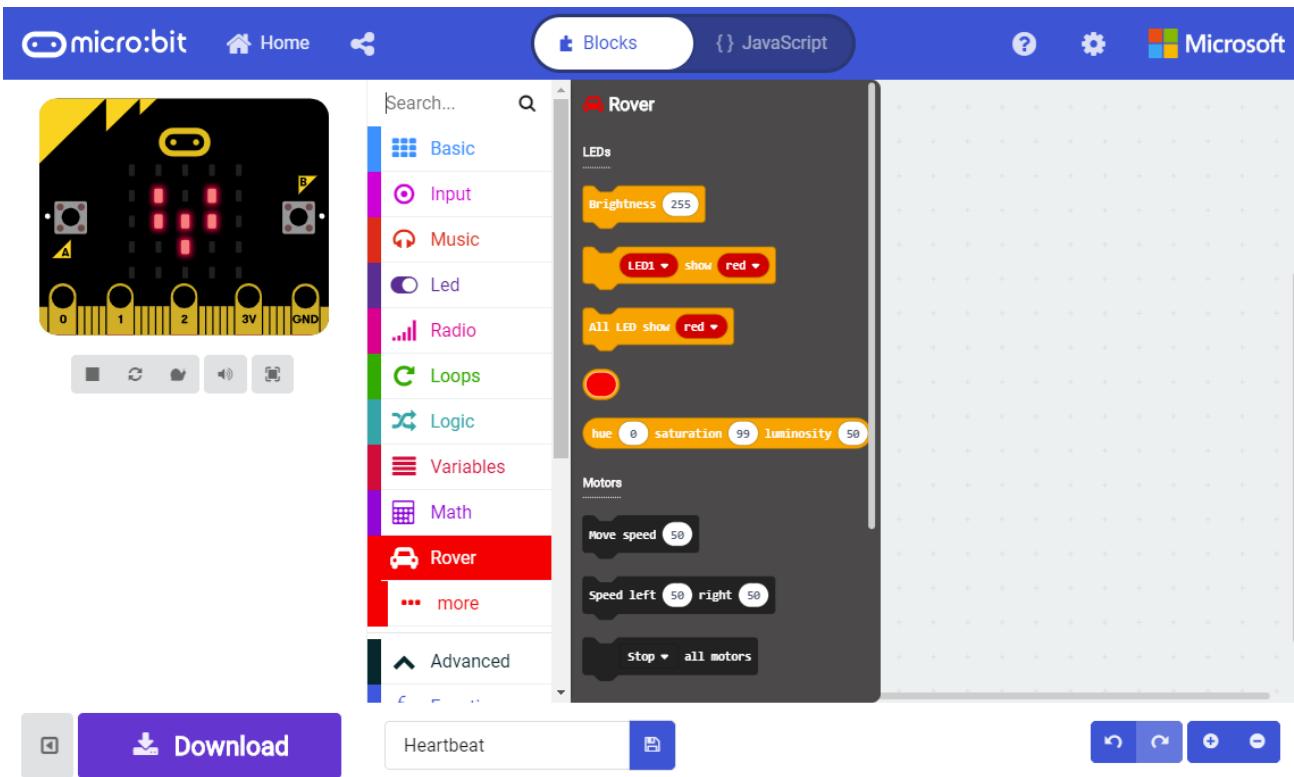
Then click search. Among them, Bluetooth, Servo, neopixel, etc. extension can be added without inputting link, and directly through clicking on the icon of corresponding extension library on the extended homepage.



Click search result Rover to download and install. The process may need a few seconds.



After the installation is completed, you can find Rover extension library on the left.



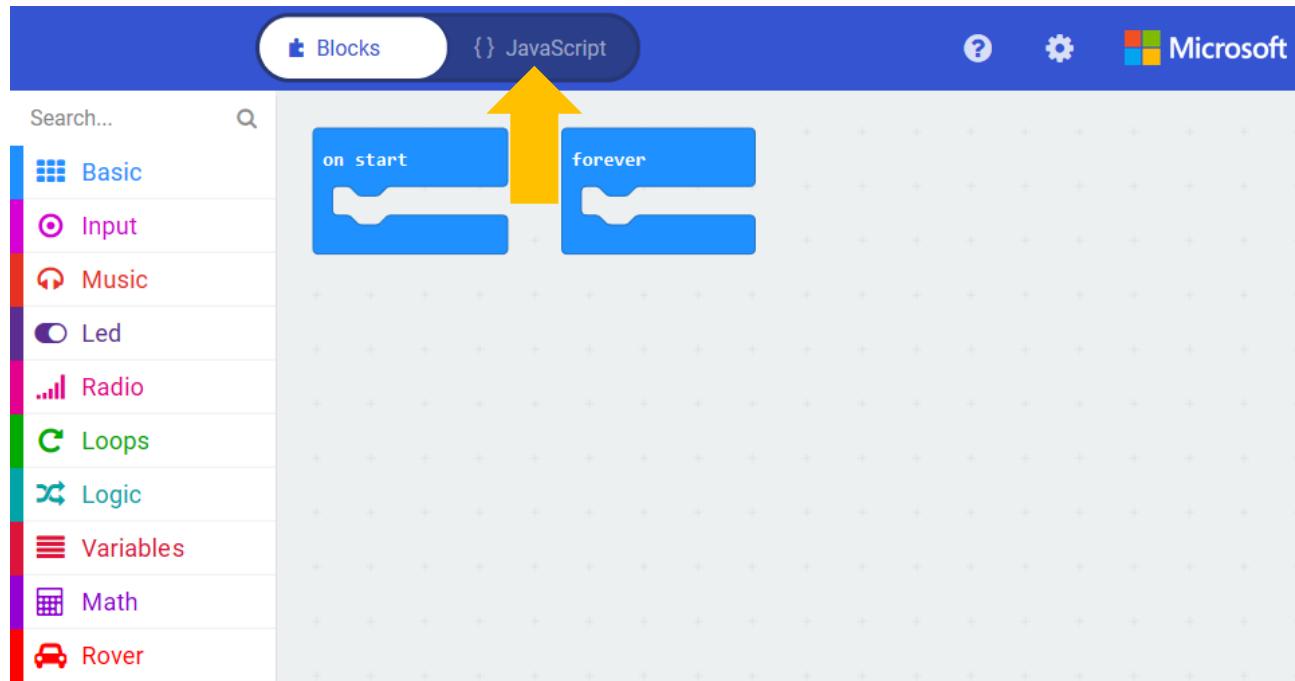
Note:

Extension library added to a project is only valid for the project, and will not appear in other projects. So when you create a new project of Rover, you need add Rover extension again.

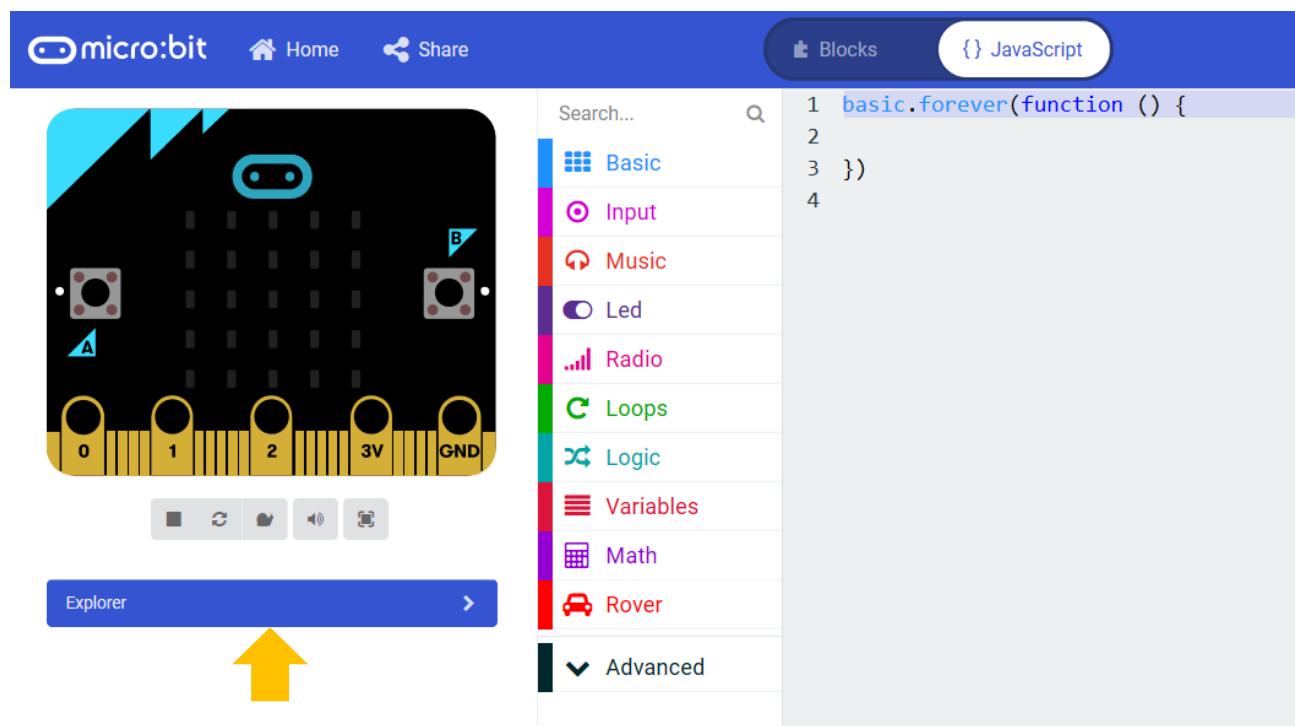
Update or Delete Rover Extension

If you need to update or delete Rover extension, please follow the instructions below.

Click the "JavaScript" button to switch to text code.



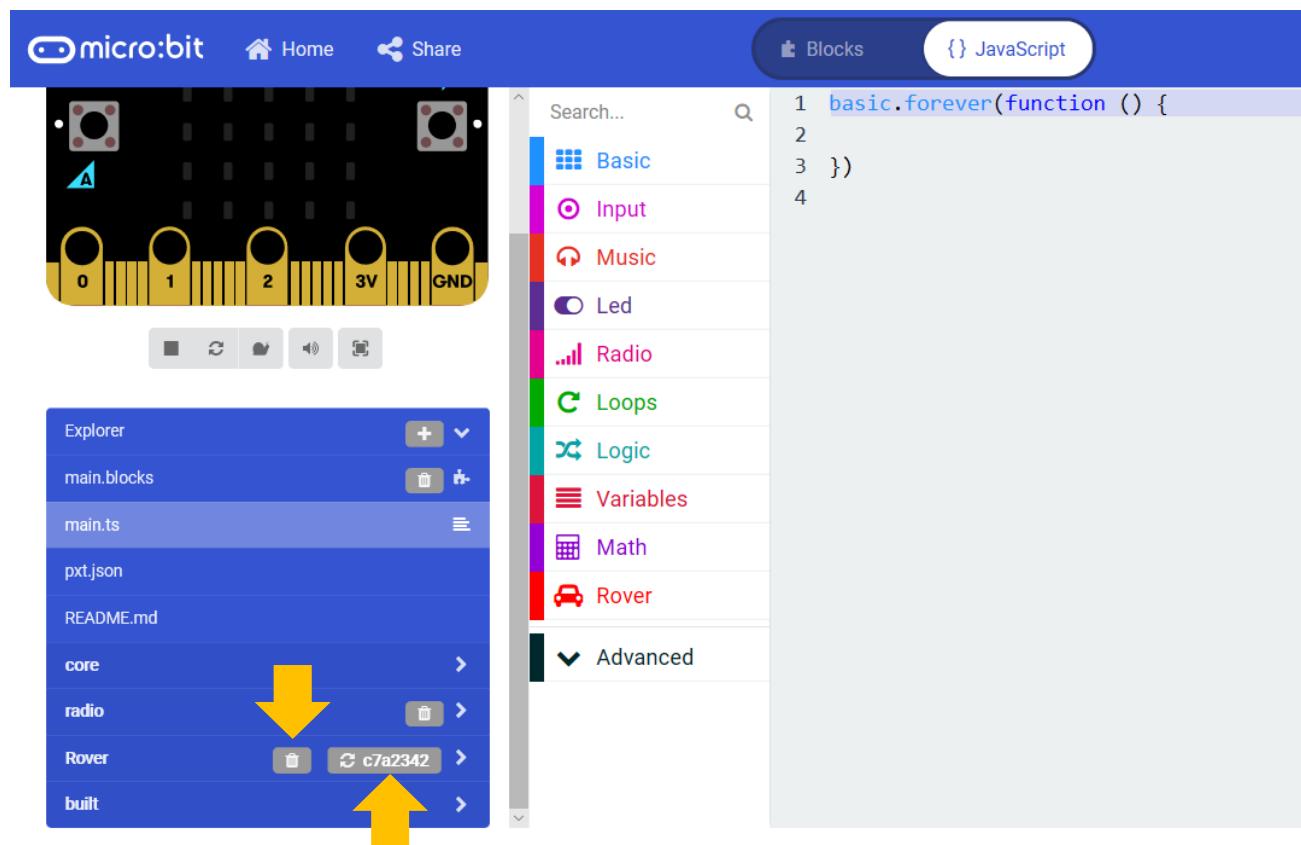
Then click on **Explorer** on the left.



Find Rover in the extended list.

Click on the **trash can icon** to delete Rover extension.

Click on the **refresh icon** to update Rover extension.



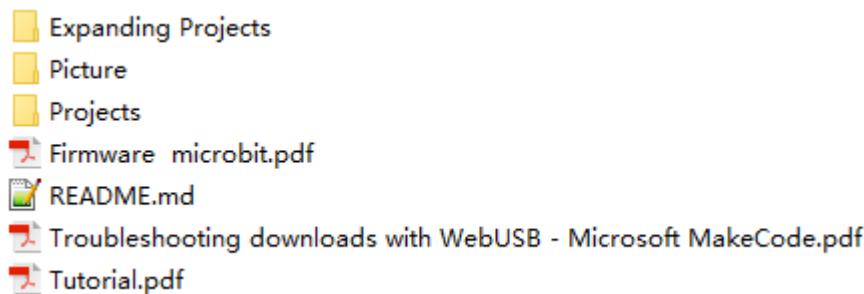
Resource and code

Resource and code of this kit can be viewed and downloaded below:

View: https://github.com/Freenove/Freenove_Micro_Rover

Download: https://github.com/Freenove/Freenove_Micro_Rover/archive/master.zip

After downloading and unzipping, a file, named Freenove_Micro_Rover or Freenove_Micro_Rover-master, is generated. You can place it anywhere on your computer disk. Enter the folder, then the following contents will be displayed.



This directory is called “root directory”, which is represented by “..”. And all the relative paths of the later resources in this tutorial start with this directory.

In most project directories, there are two files, xxx.hex, xxx.mp4. Their contents are shown in the following table:

File name	Content
xxx.hex	Project files for code, containing code and extension Libraries
xxx.mp4	Demo Video of Code Running on Rover



Import Code

We provide hex file (project files) for each project. Hex file contains all the contents of the project and can be imported directly. You can also complete the code of project manually. If you choose to complete the code by dragging code block, you may need add necessary extensions. ([How to add extension?](#))

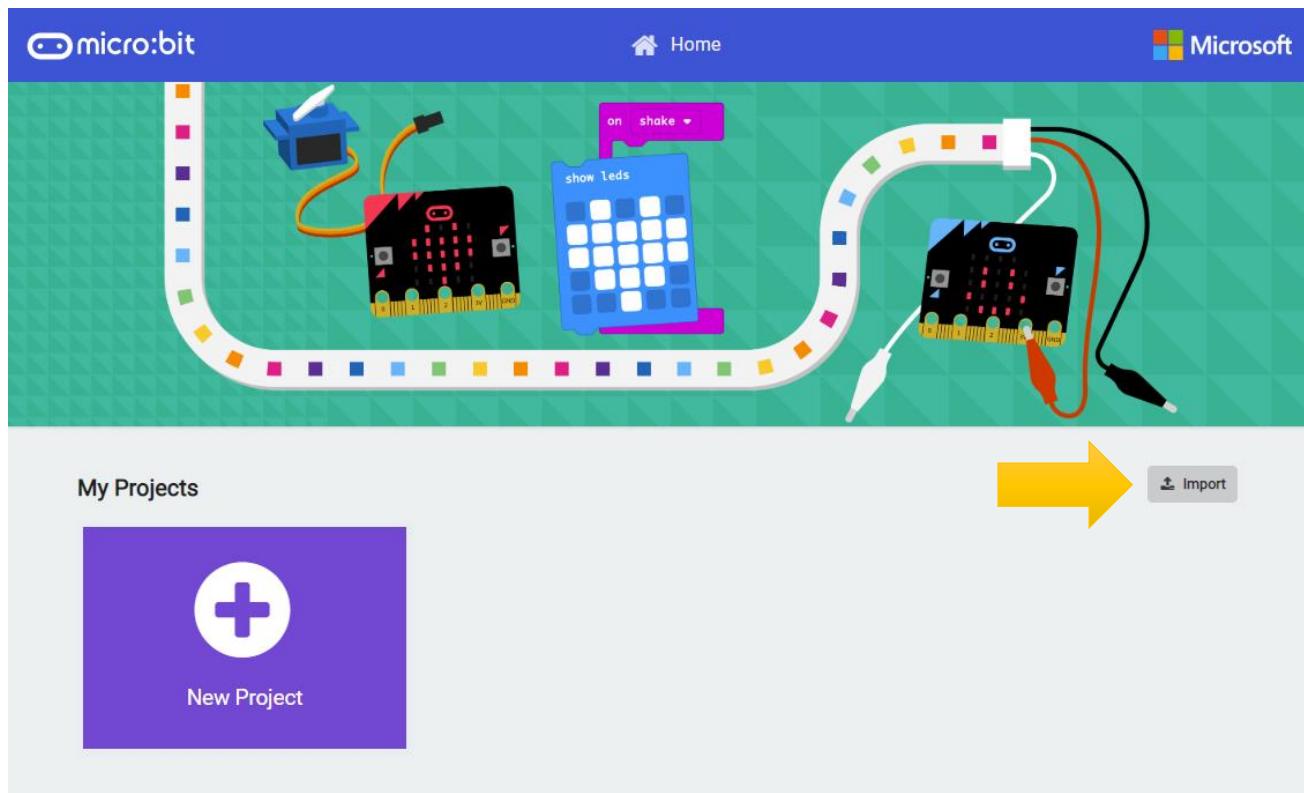
As for simple projects, it is recommended to complete the project by dragging code block.

As for complicated projects, it is recommended to complete the project by importing Hex code file.

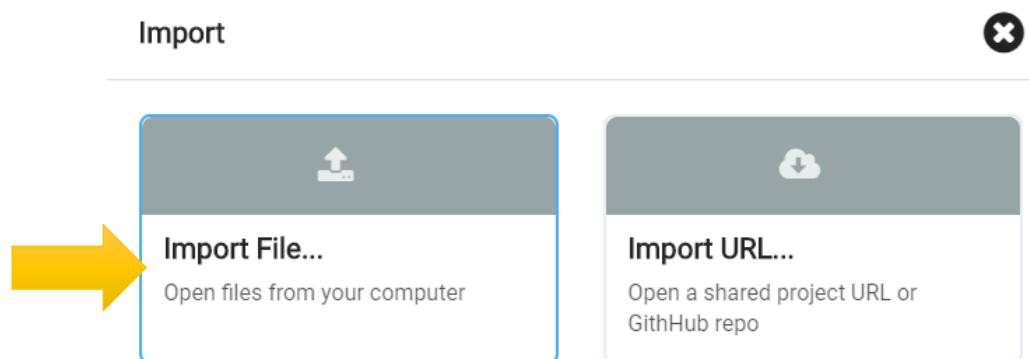
Next, we will take “Heatbeat” project as an example to introduce how to load code.

Open web version of [makecode](#) or windows 10 app version of makecode.

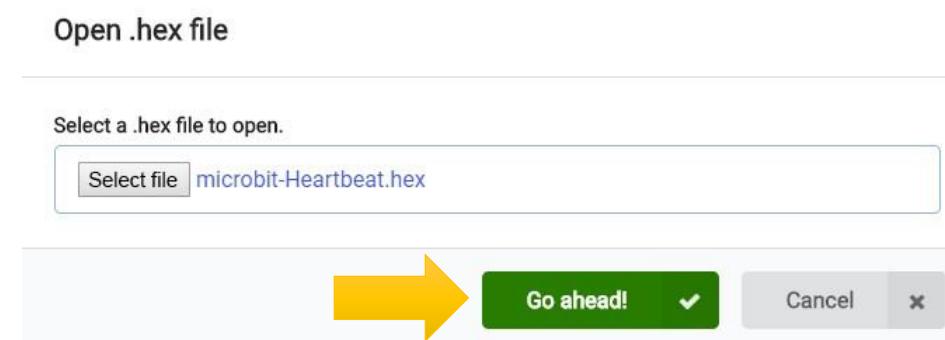
Click “Import” button on the right side of HOME page.



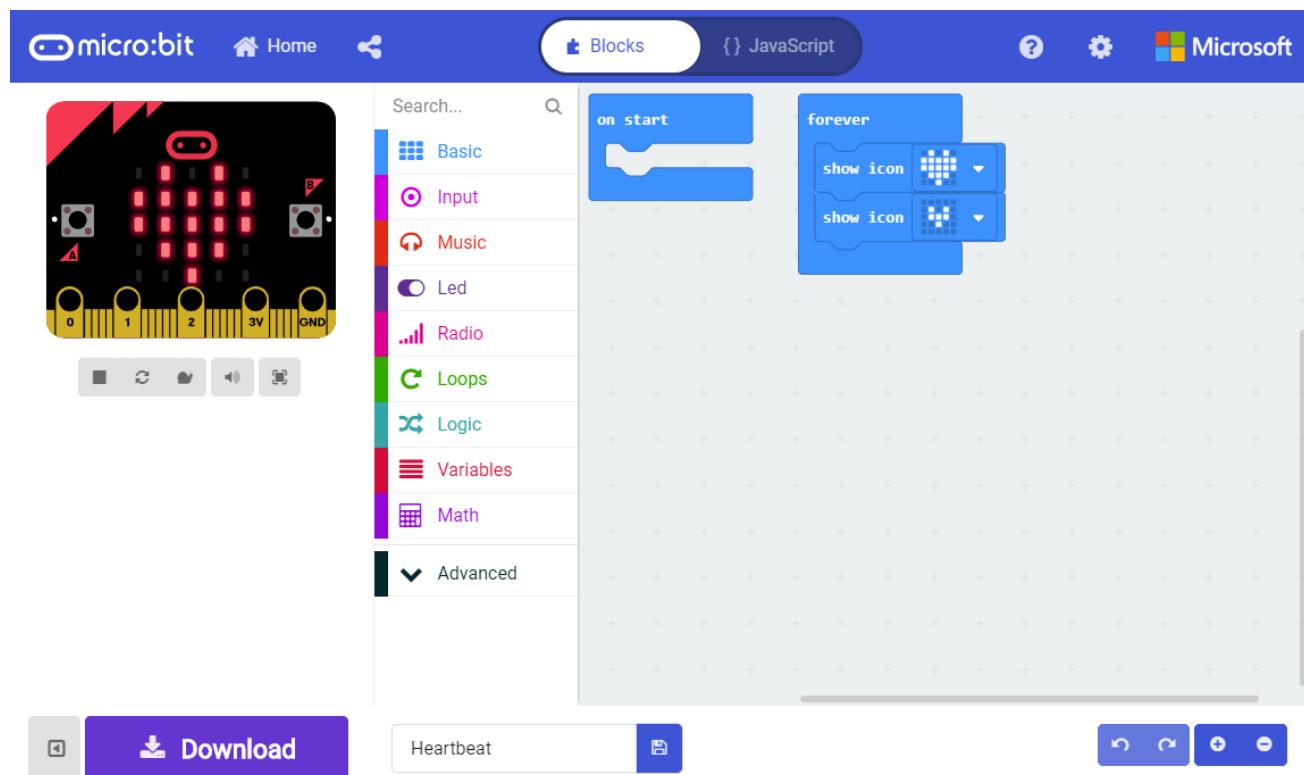
In the pop-up dialog box, click "Import File".



Select file "../Projects/00.1_Heartbeat/microbit-Heartbeat.hex". Then click "Go ahead!"



A few seconds later, the project is loaded successfully.





Android app

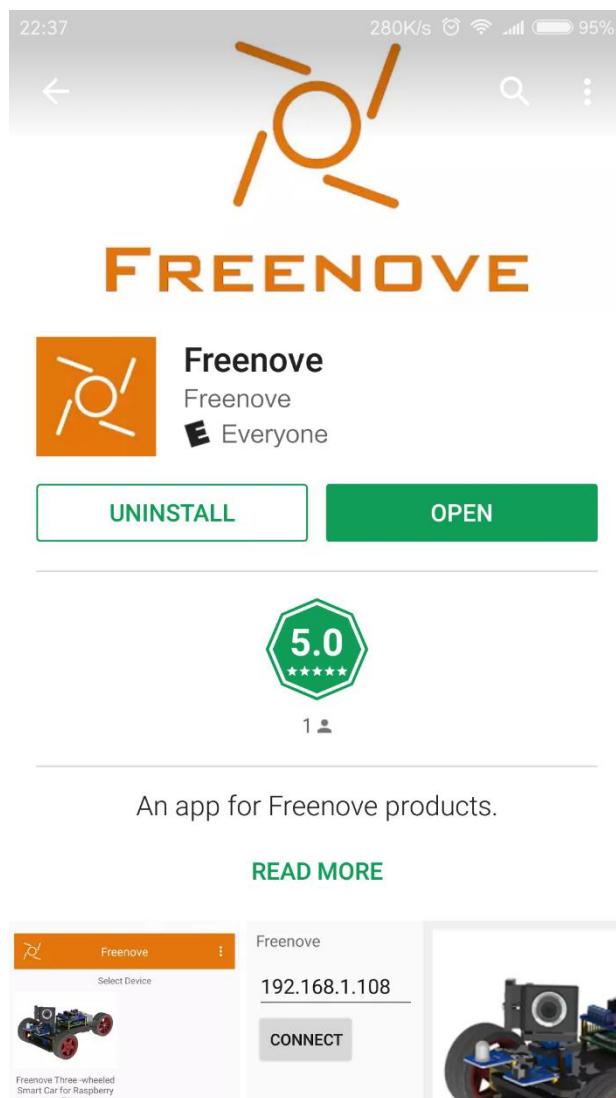
We design an Android app for Rover. Please install it as below. You will use it in later projects.

Install Freenove app

We have provided three ways to install app, you can choose any one.

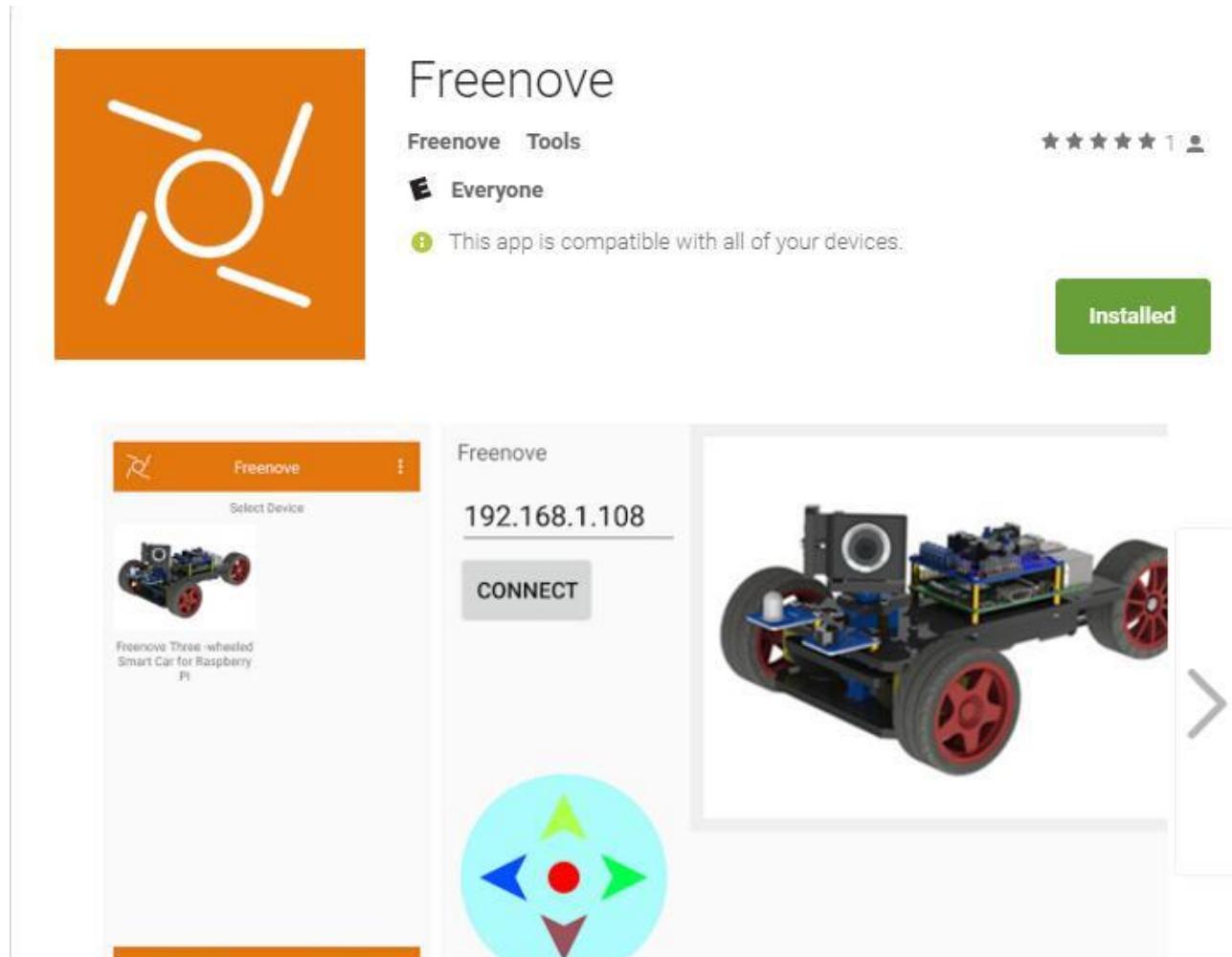
Method 1

Use Google play to search “freenove”, download and install.



Method 2

Visit <https://play.google.com/store/apps/details?id=com.freenove.suhayl.Freenove>, and click install.



Method 3

Visit https://github.com/Freenove/Freenove_app_for_Android, download the files in this library, and install freenove.apk to your Android phone manually.

The screenshot shows the GitHub repository page for 'Freenove / Freenove_app_for_Android'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. A green button labeled 'Clone or download' is highlighted with an orange callout bubble pointing to it. The file 'freenove.apk' is listed as the latest commit.

File	Commit	Time
Readme.txt	SuhaylZhao First Publish. ...	3 minutes ago
freenove.apk	First Publish.	3 minutes ago

Click here to download.

Chapter 1 Music

The buzzer integrated on Rover can be used to play music. You can create and play music with Rover through blocks of Music library in Makecode.

Preparation

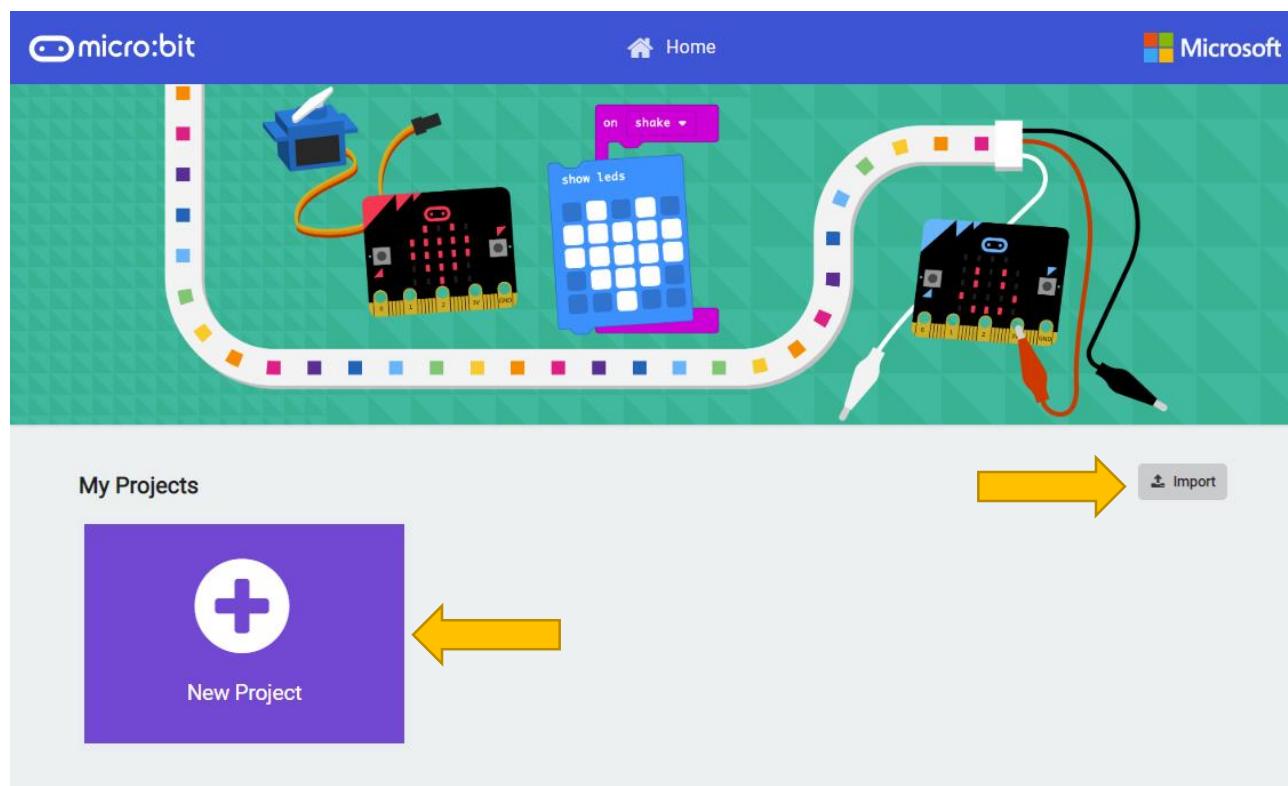
1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.

Open web version of or windows 10 app version of makecode.

If you want to load the whole project directly, click “Import”.

[\(How to import?\)](#)

If you want to drag the code block one by one, click “new project”.

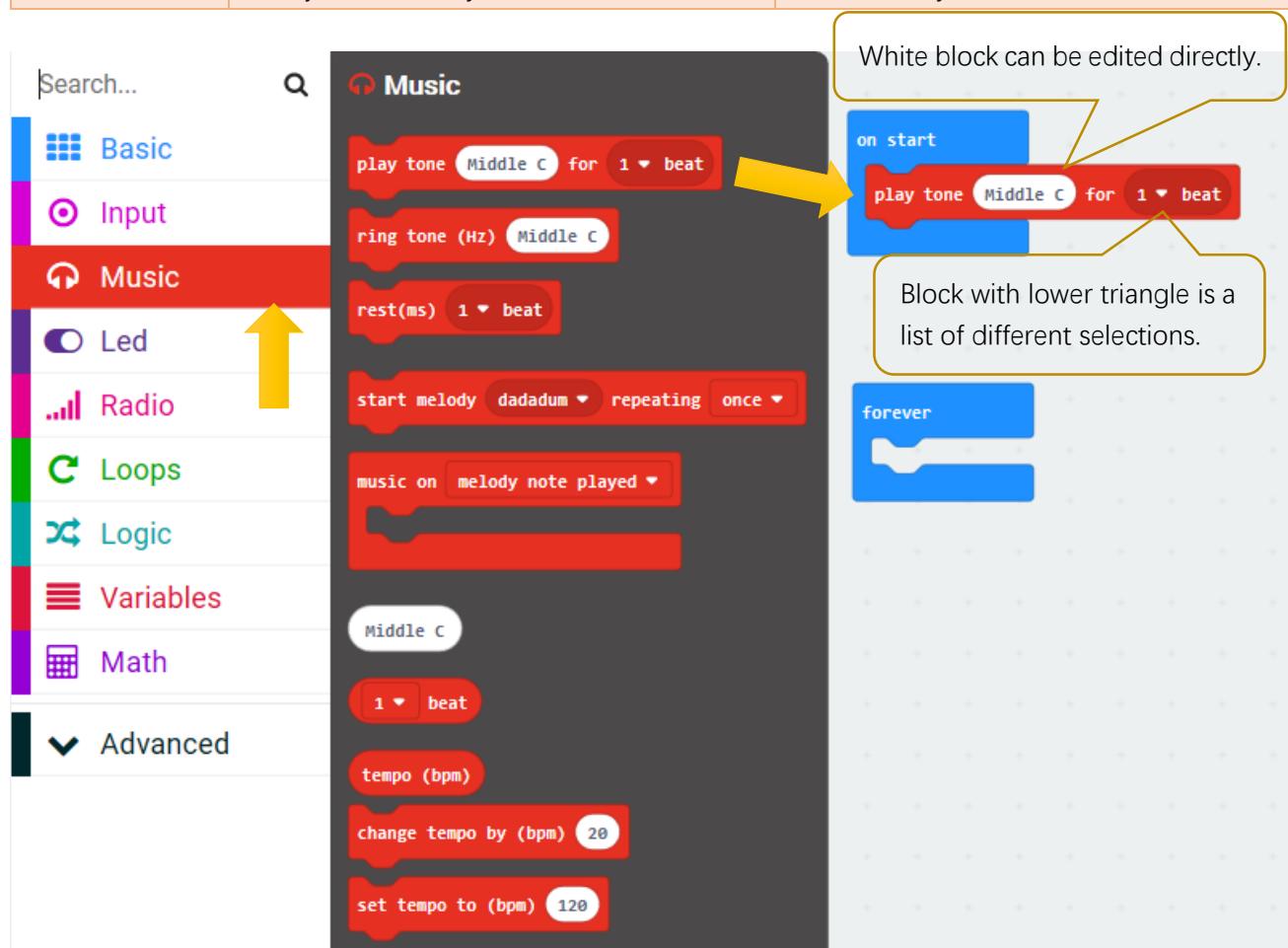


Play a note

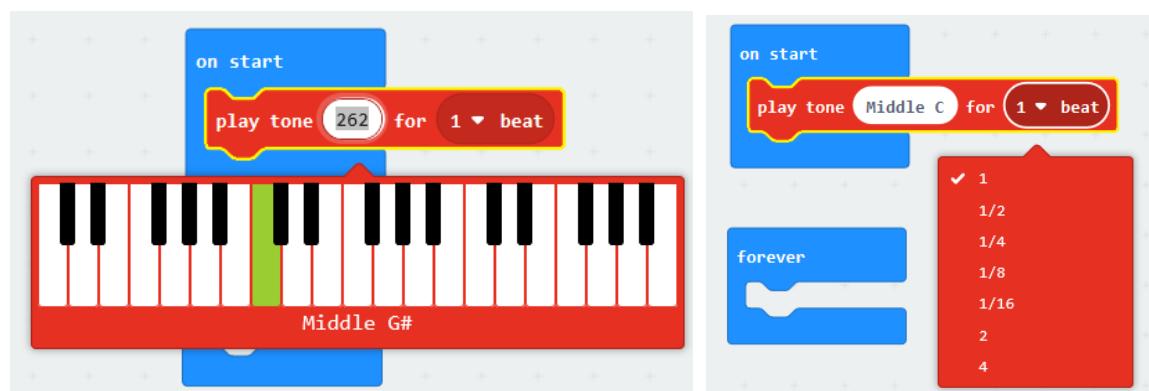
Let Rover play a note first.

Load code according to the table below (How to load?) or drag the code block as picture below:

File type	Path	File name
Hex file	./Projects/01.1_Play-a-note	microbit-Play-a-note.hex



The mean of this code is: playing note Middle C lasting for one beat. You can change the note and beat by clicking elliptical area in code block.

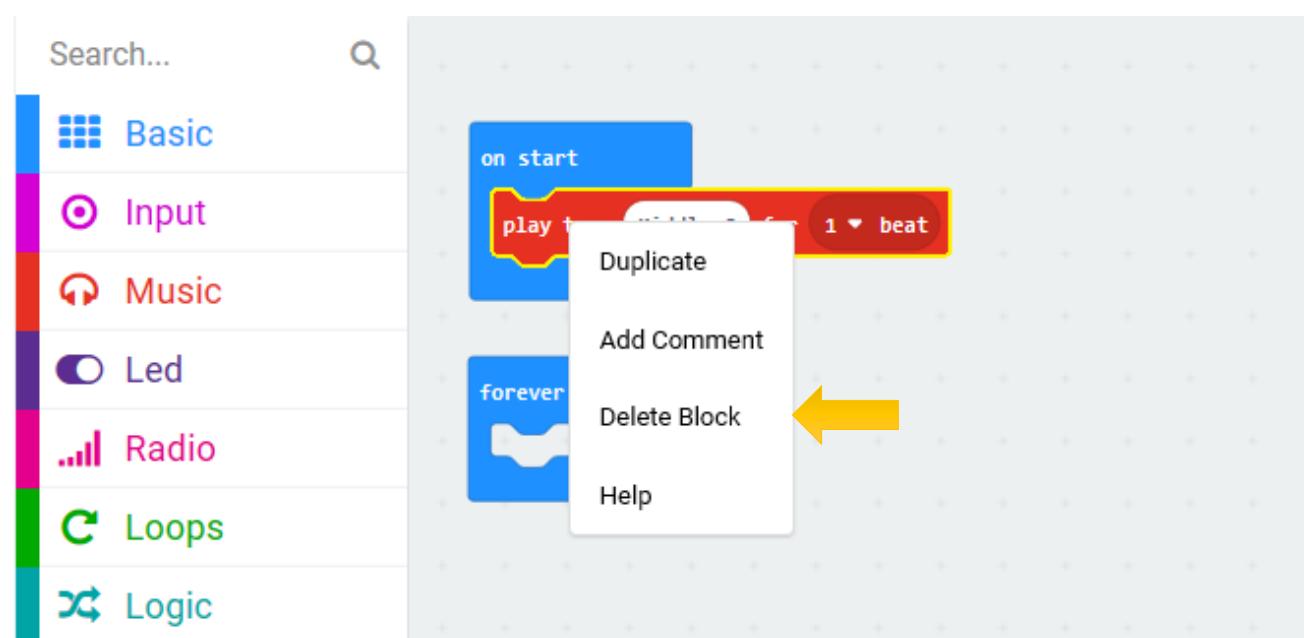


If you want to delete a code block, you can drag the block to the left extension area.



Or select the block in the code area and press “Delete” key.

Or right click on the block and select “Delete Block”.



Download the code to micro:bit in previous way, and Rover will play the notes and rhythms in your code.

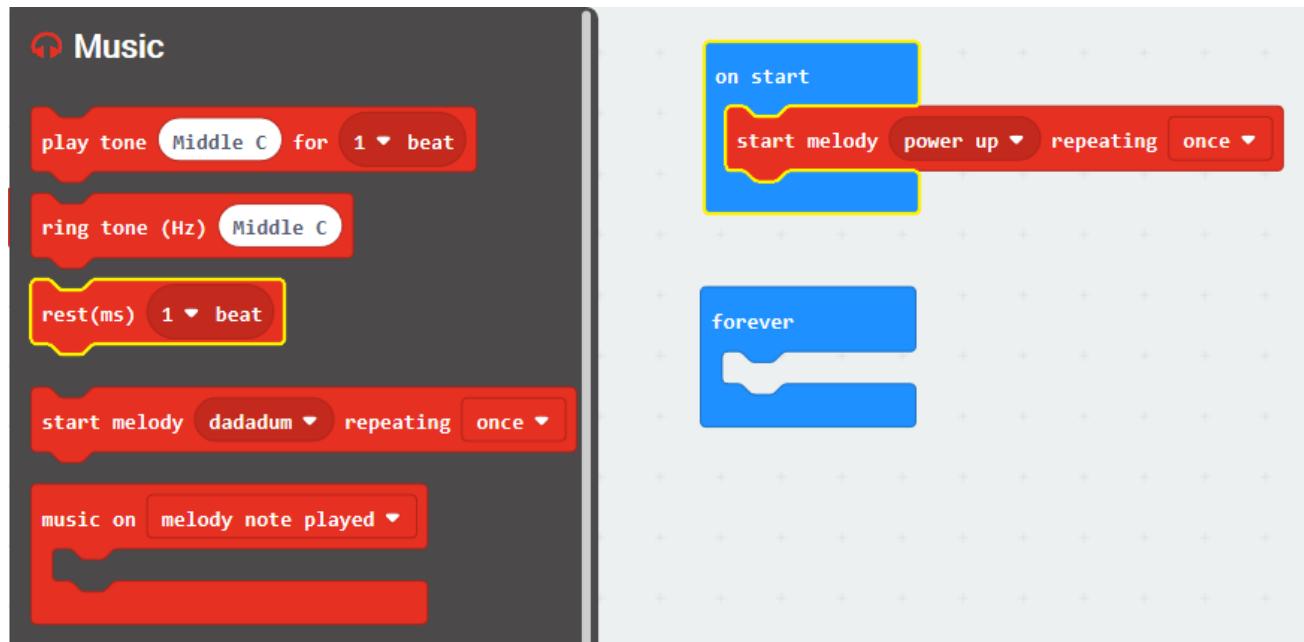
([How to download?](#) [How to quick download?](#))

If there is a download problem, disconnect usb and micro: bit, then reconnect them and reopen Makecode to try downloading again.

Play a melody

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	../Projects/01.2_Play-a-melody	microbit-Play-a-melody.hex

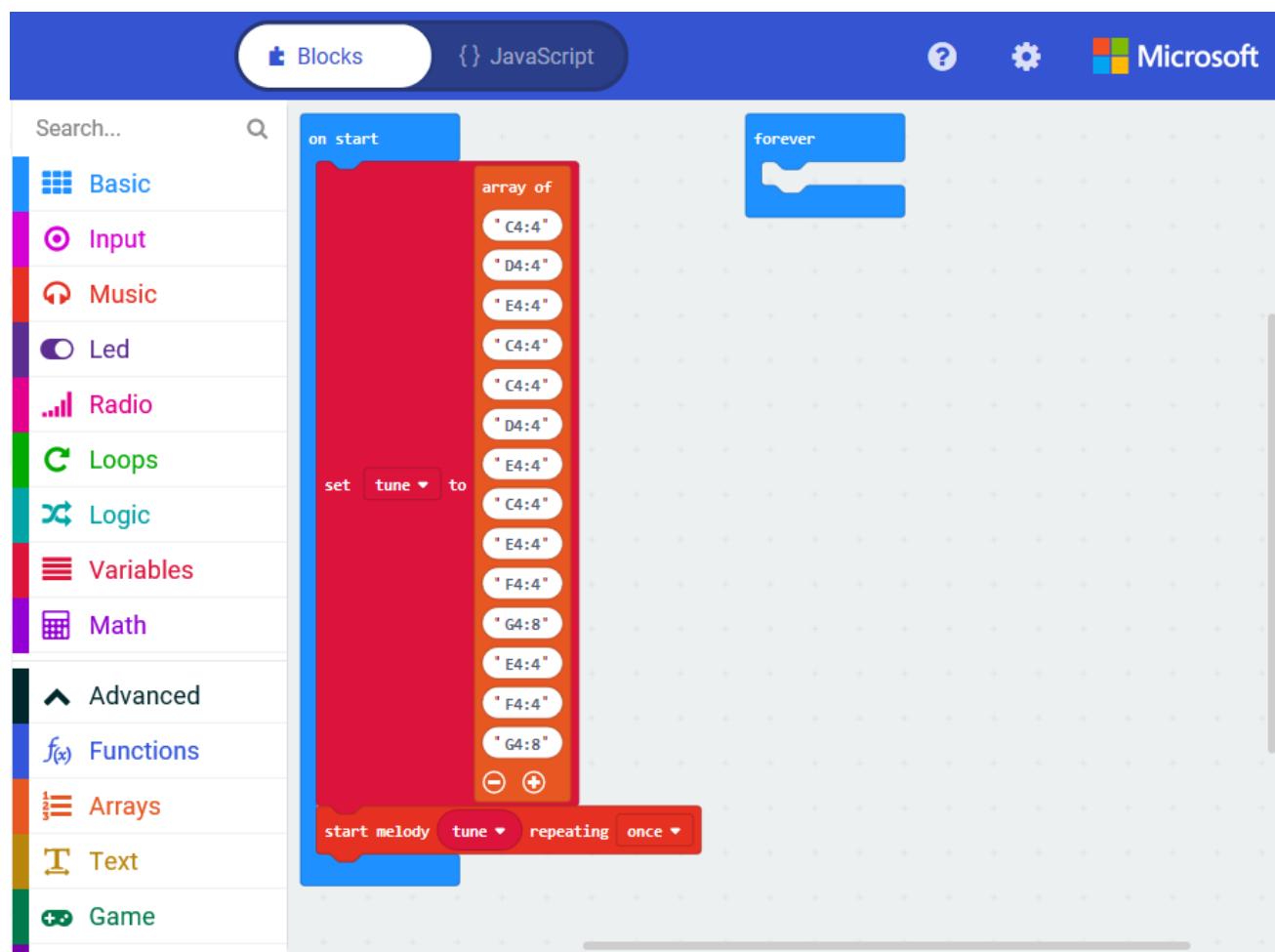


In this drop-down list of this code block, there are dozens of melodies to choose. You can choose any one. Then download the code to micro:bit, Rover will play melody of the code.

Play custom melody

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	./Projects/01.3_Custom-Melody	microbit-Custom-Melody.hex



In this code, an array is used to define a melody. Each data represents a note and a beat. For example, "C4:4" means that the note is C4 and the beat is 4. "G4:8" means that the note is G4 and the beat is 8. So you can create your own music.

Download the code to micro:bit. Rover will play melody of the code.



Chapter 2 RGB LED

There are four sets of RGBLEDs integrated on Rover. Each group has three RGBLEDs. Each set can be controlled independently. You can control the brightness and color through the blocks in the Rover library to show 16 million colors. Now create your own RGB light.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.
5. Rover White paper or Rover tracking map. (Optional)

This chapter will introduce RGB LED of Rover.

Put Rover on white paper, which will make the color displayed more clear.

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

If you choose to drag code manually, you first need to add Rover extensions.

[\(How to add Rover extension?\)](#)

Show one color

First let all RGB LEDs of Rover show same color and brightness.

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	./Projects/02.1_LED-OneColor	microbit-LED-OneColor.hex



In this code, set brightness of all RGBLED to 255 and set the color for all to red.

You can modify brightness in the rang of 0~255 and choose different color.

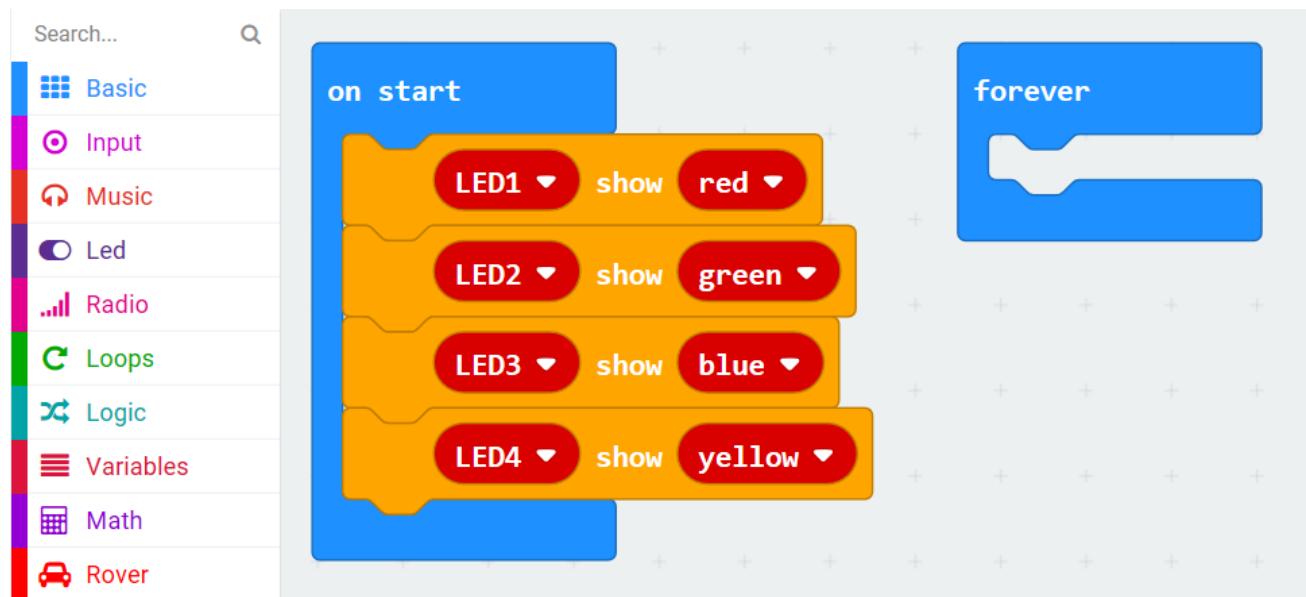
Download the code to micro:bit. All RGBLED on Rover will show red.



Show different colors

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	../Projects/02.2_LED-DifferentColors	microbit-LED-DifferentColors.hex



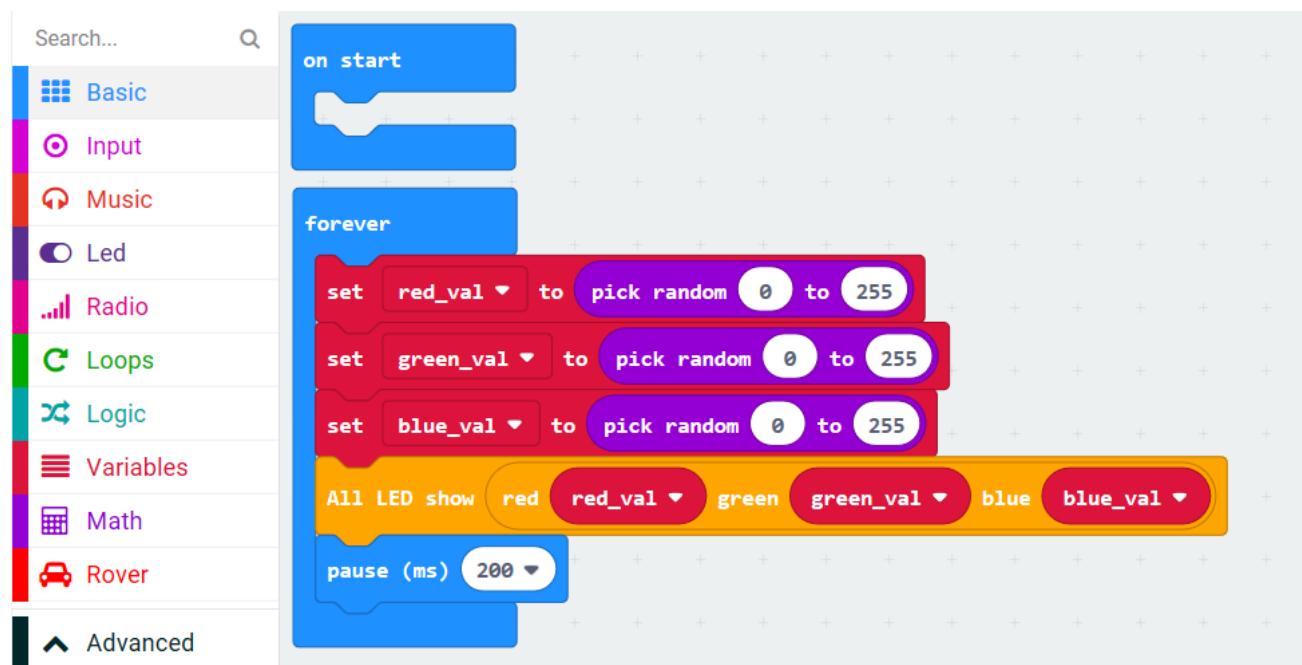
In this code, set color of RGBLED1,2,3,4 to red, green, blue, and yellow respectively.

Download the code to micro:bit. All RGBLED on Rover will show red.

Show random color

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	../Projects/02.3_LED-RandomColors	microbit-LED-RandomColors.hex



In this code, three variables(`red_val`, `green_val` and `blue_val`) are defined. Three Random values between 0 and 255 are assigned to them respectively. These three values are used as parameters of RGB LED. Then all RGBLED will show random color.

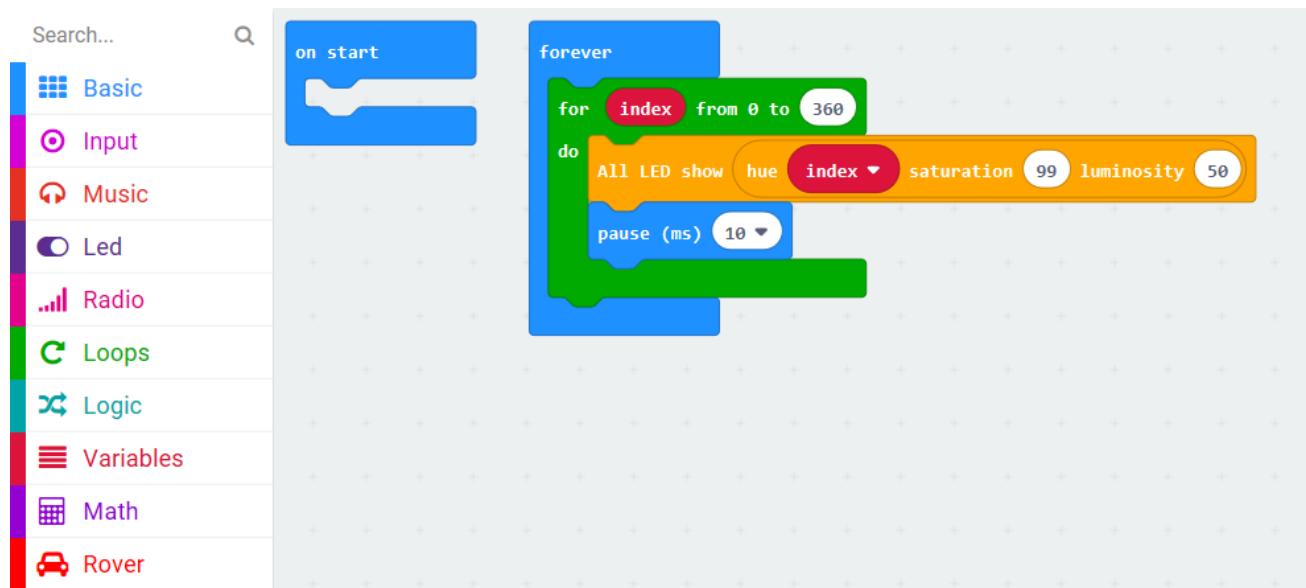
Download the code to micro:bit. Then observe the color showed by RGBLED on Rover.



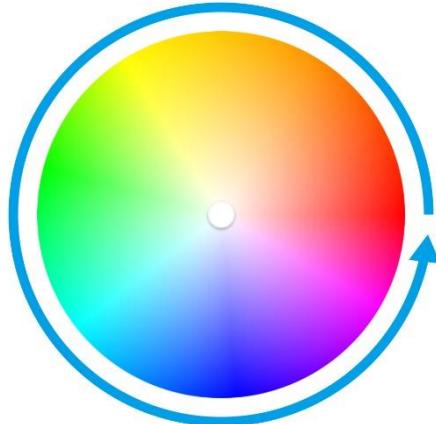
Show soft colors

Load code according to the table below or drag the code block picture below:

File type	Path	File name
Hex file	./Projects/02.4_LED-SoftColors	microbit-LED-SoftColors.hex



In this code, HSL color model is used, as shown below.



The hue range is 0-360. The saturation range is 0-99 and the Intensity(Lightness) range is 0-99.

Download the code to micro:bit. Then observe the color showed by RGBLED on Rover.

Chapter 3 Ultrasonic Ranging

The ultrasonic module on Rover can measure the distance between Rover and the front obstacle. So we can realize obstacle avoidance action for Rover through this sensor. When Rover is close to the obstacle in front, let it turn to avoid the obstacle.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install ultrasonic ranging module on Rover.
3. Install battery into Rover.
4. Turn on Rover power.
5. Connect micro:bit and computer through USB cable.

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

If you choose to drag code manually, you first need to add Rover extensions.

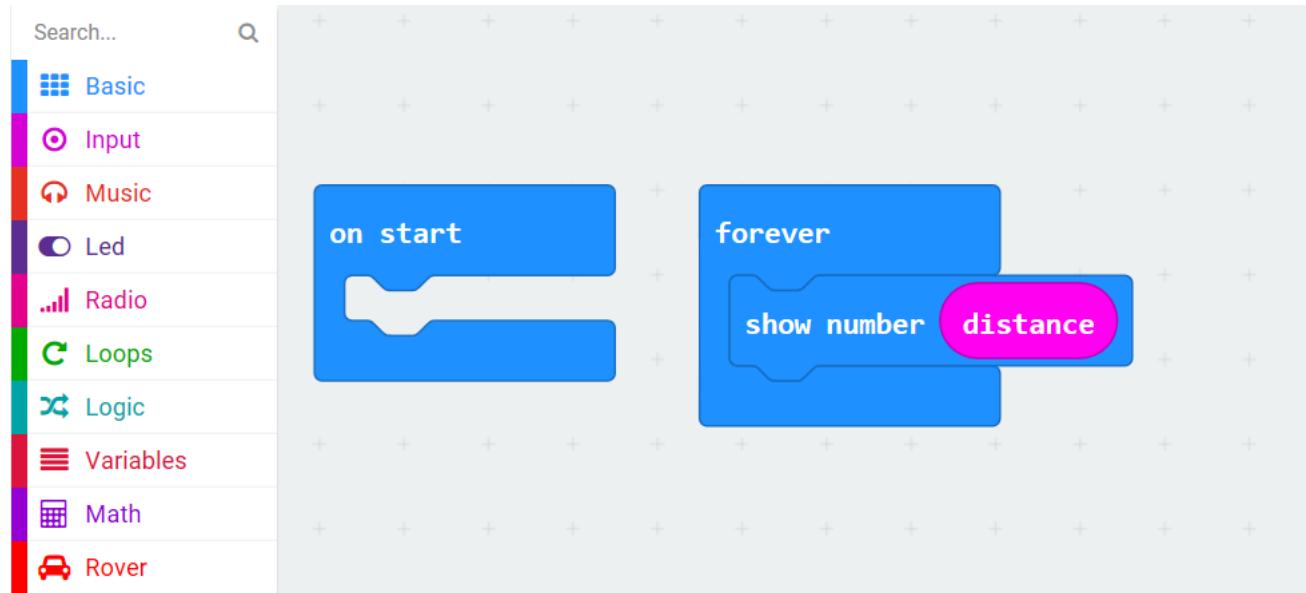
[\(How to add Rover extension?\)](#)



Obtain value of ultrasonic ranging

Load code according to the table below or drag the code block picture below:

File type	Path	File name
Hex file	../Projects/03.1_Ranging	microbit-Ranging.hex



This code is used to make LED matrix of micro:bit show the value of ultrasonic ranging.

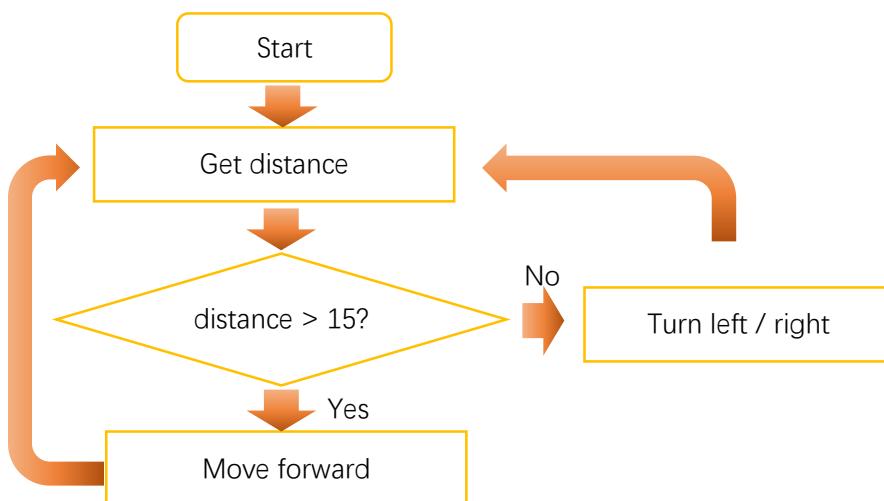
Place an obstacle in front of ultrasonic ranging module. Download the code to micro:bit. Then change the distance between the two objects, and observe the value of showed on LED matrix of micro:bit.

Rover-Obstacle avoidance mode -1

This project will realize the obstacle avoidance mode of Rover.

Flow chart

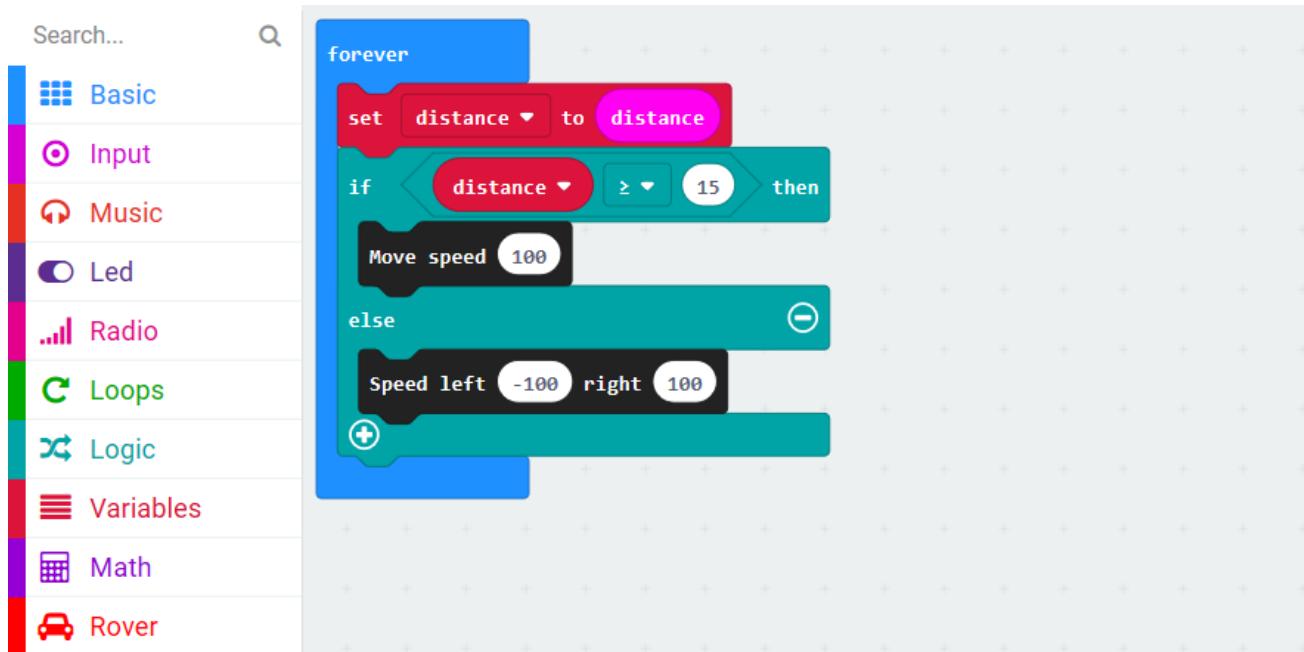
The program code is written according to flow chart, as shown below.



Code

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	./Projects/03.2_Obstacle-Avoidance-1	microbit-Obstacle-Avoidance-1.hex



In this code, code block is used to set speed of both motors to a certain value. Positive value means forward, and negative value means backward.

Code block is used to set speed of left motor and right motor. When the value is different, Rover will turn.

Download the code to micro:bit. Then observe the motion of Rover.

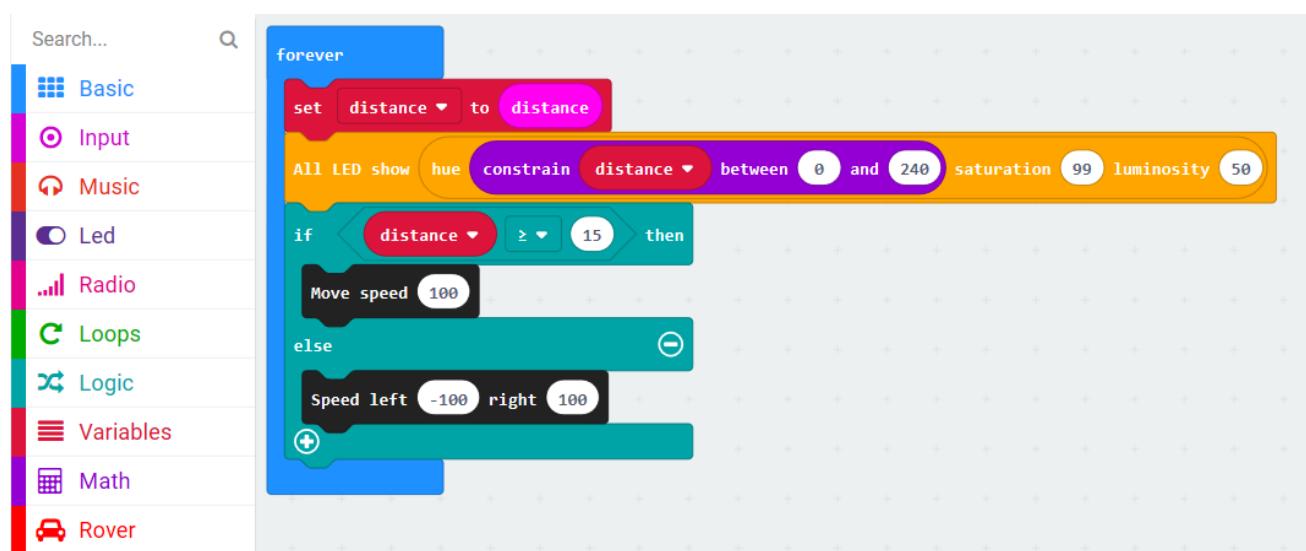
Rover-Obstacle avoidance mode -2

This project will combine RGBLED and LED matrix to provide some auxiliary instructions for obstacle avoidance mode.

Code

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	../Projects/03.2_Obstacle-Avoidance-2	microbit-Obstacle-Avoidance-2.hex



In this code, the logical is exactly the same as previous section. The difference is that when measured distances are different, RGBLEDs will show different colors. The color will change from red to green when the distance changes from near to far.

Download the code to micro:bit. Then observe motion of Rover and the show of RGBLED and LED matrix.

Chapter 4 Light tracing

There are two light intensity sensors on Rover. So Rover can get the difference between the values of two sensors. Then we can know which sensor gets stronger light. Therefore, Rover can realize tracking or avoiding light action.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

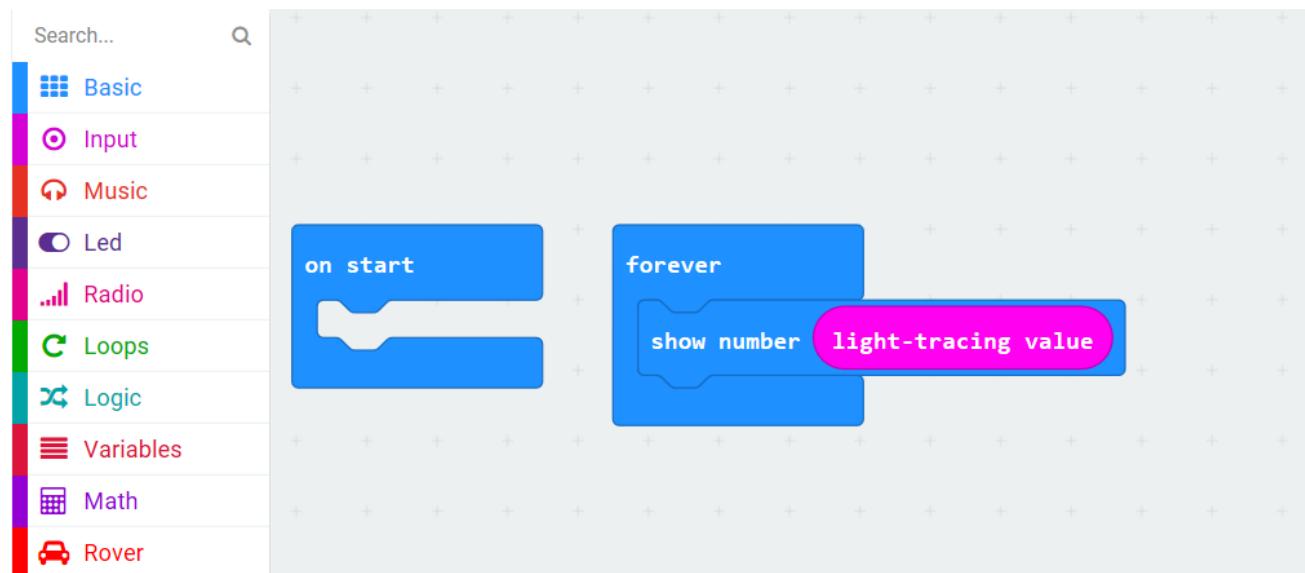
If you choose to drag code manually, you first need to add Rover extensions.

[\(How to add Rover extension?\)](#)

Get value of light intensity sensor

Load code according to the table below or drag the code block picture below:

File type	Path	File name
Hex file	../Projects/04.1_Light	microbit-Light.hex



In this code, the block light-tracing value represents the acquired value (β) by the light intensity sensor, which is jointly generated by the two sensors. Its range is 0~1023.

Theoretically,

If $\beta = 512$, it indicates that light intensities sensed by two sensors are equal.

If $\beta > 512$, it indicates that light intensity sensed by right sensor is stronger than left sensor.

If $\beta < 512$, it indicates that light intensity sensed by left sensor is stronger than right sensor.

But in fact, it is influenced by the difference of the sensors and the working voltage. When the two light sensors detect the same sensitivity, β (the center value) may be not 512, but a number near 512. So when we use this sensor, we should calibrate it first.

Download the code to Micro::bit. And use a light source to illuminate one light intensity sensor, or use your finger to cover one light intensity sensor, then observe the values showed by micro:bit LED matrix.

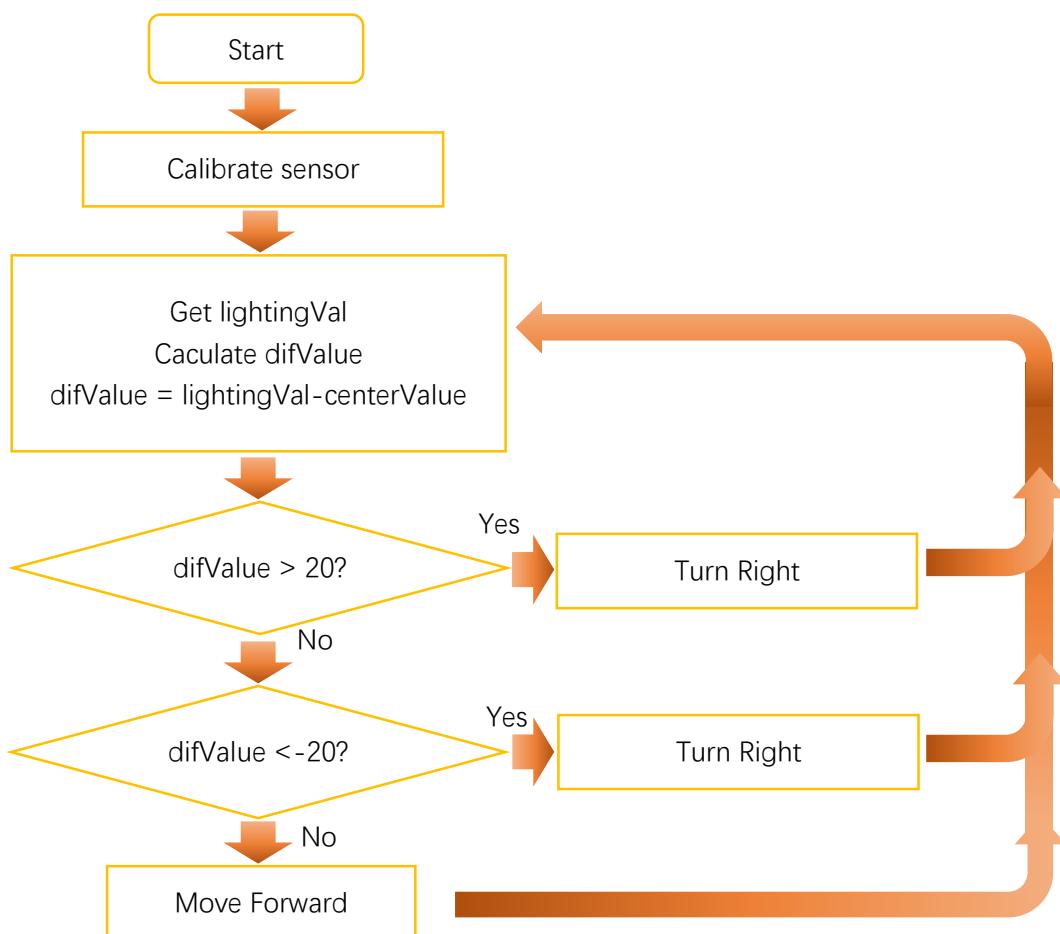


Rover-light tracing mode

This project will realize light tracing mode of Rover.

Flow chart

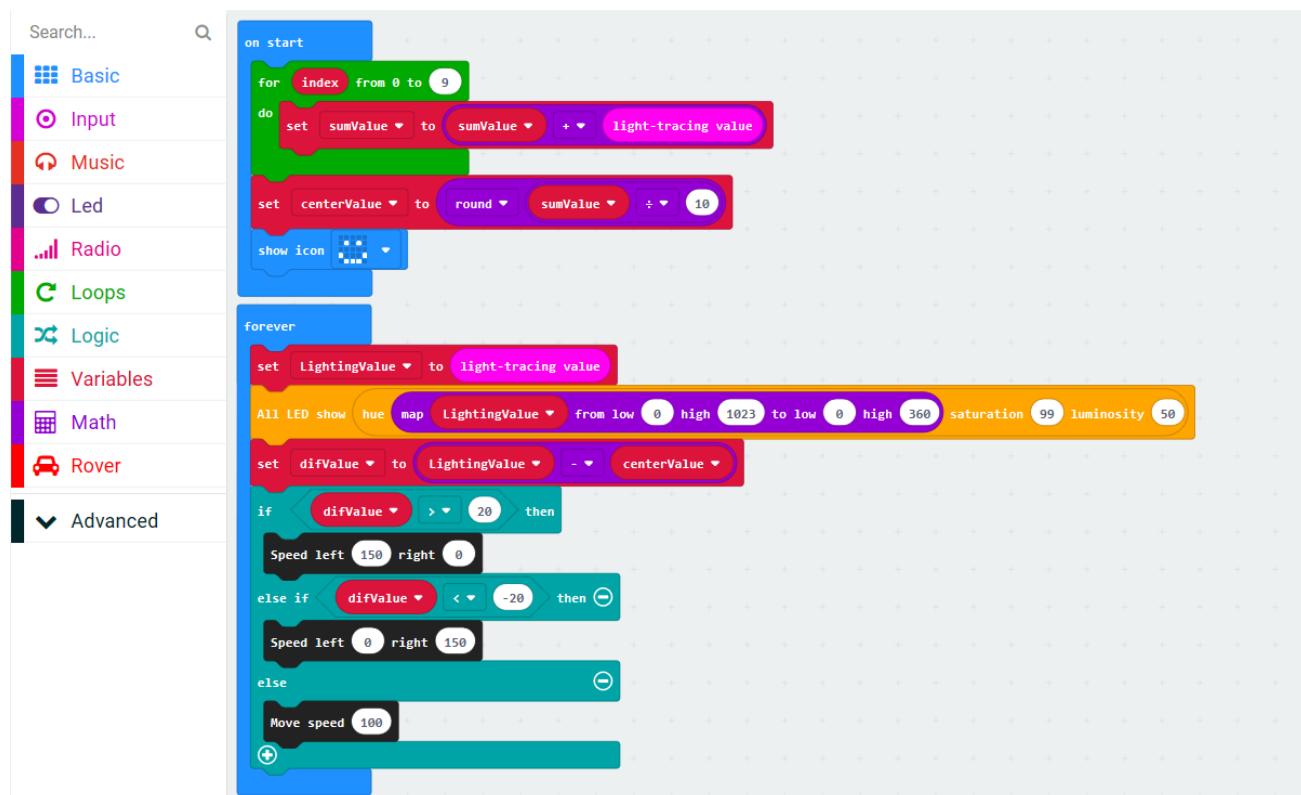
The program code is written according to flow chart, as shown below.



Code

Load code according to the table below:

File type	Path	File name
Hex file	./Projects/04.2_LightTracing	microbit-LightTracing.hex



As we mentioned before, when using light sensors, we need calibrate them first. Therefore, in the code, the sensor values are read 10 times after booting up, and the sum is made, then the average is taken as the center value after calibration.

After the sensor value is got, it will be compared with the centerValue to get the difference (difValue). And Rover will do different actions according to the difference.

The RGB LED show is added in the code. Different sensor values will be mapped to different colors.

Download the code to micro:bit. In order to improve accuracy of the calibration, put the two light sensors in the same light intensity environment when booting. Wait the micro:bit LED Matrix to show a smile, which indicates that the calibration is completed.

Then use a flashlight or other light source to illuminate the light intensity sensor and observe motion of Rover.

Chapter 5 Line tracking

There are three Reflective Optical Sensors on Rover. When the infrared light emitted by infrared diode shines on the surface of different objects, the sensor will receive light with different intensities after reflection. As we know, black objects absorb light better. So when black lines are drawn on the white plane, the sensor can detect the difference. So we can realize Line tracking mode for Rover. The sensor can also be called Line Tracking Sensor.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

If you choose to drag code manually, you first need to add Rover extensions.

[\(How to add Rover extension?\)](#)

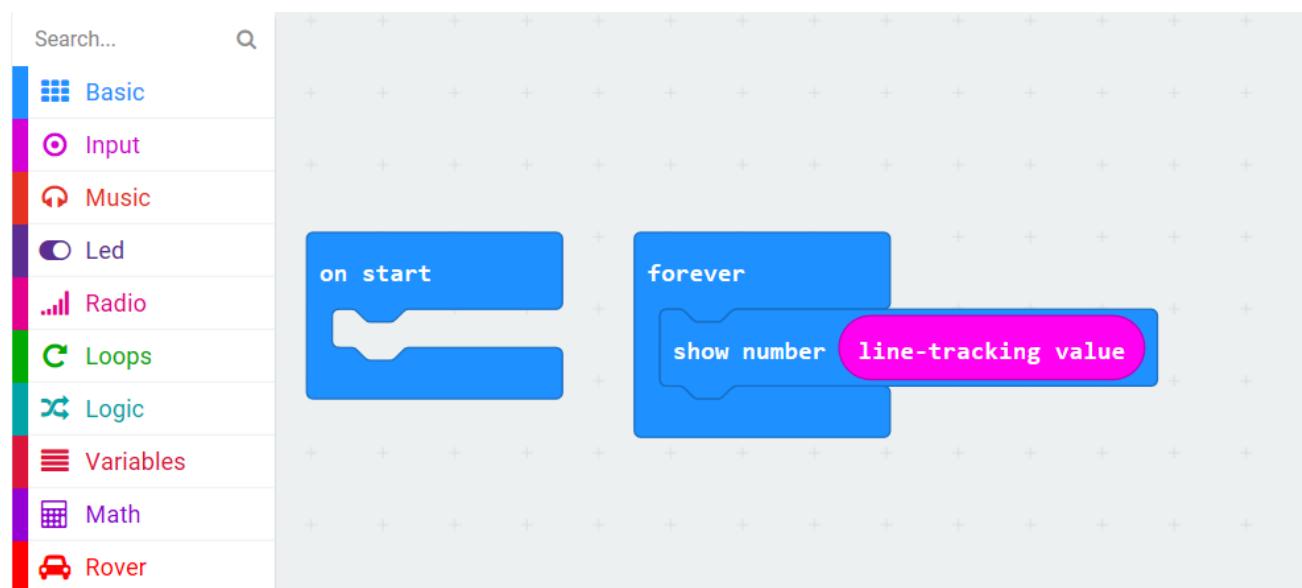
Warning

Reflective Optical Sensor(including Line Tracking Sensor) should be avoided using in environment with infrared interference, like sunlight. Sunlight contains a lot of invisible light such as infrared and ultraviolet. Under environment with intense sunlight, Reflective Optical Sensor cannot work normally.

Get value of LineTrackingSensor

Load code according to the table below or drag the code block as picture below:

File type	Path	File name
Hex file	../Projects/05.1_GetTrackingValue	microbit-GetTrackingValue.hex



Download the code to micro:bit. Close the Tracking Sensor with black and white objects, respectively, and observe the values showed by the indicator lamp and the Micro: bit led matrix.

The following table shows the values of all cases when three Tracking Sensors detect objects of different colors. Among them, black objects or no objects were detected to represent 1, and white objects were detected to represent 0.

Left	Middle	Right	Value(binary)	Value(decimal)
0	0	0	000	0
0	0	1	001	1
0	1	0	010	2
0	1	1	011	3
1	0	0	100	4
1	0	1	101	5
1	1	0	110	6
1	1	1	111	7

You can verify the running result of Rover with this table.



Rover-line tracking mode

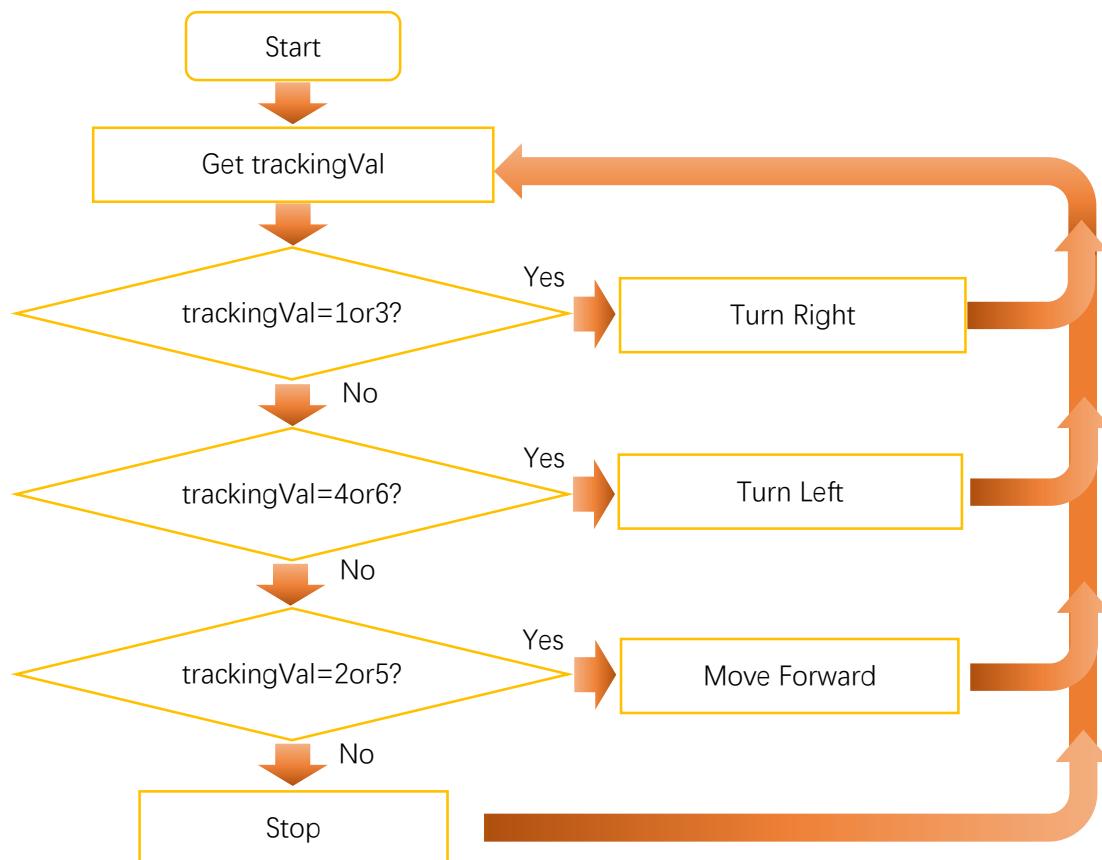
This project will realize line tracking mode of Rover.

Flow chart

Rover will make different actions according to the value transmitted by the line tracking sensor.

Left	Middle	Right	Value(binary)	Value(decimal)	Rover Action
0	0	0	000	0	Stop
0	0	1	001	1	Turn Right
0	1	0	010	2	Move Forward
0	1	1	011	3	Turn Right
1	0	0	100	4	Turn Left
1	0	1	101	5	Move Forward
1	1	0	110	6	Turn Left
1	1	1	111	7	Stop

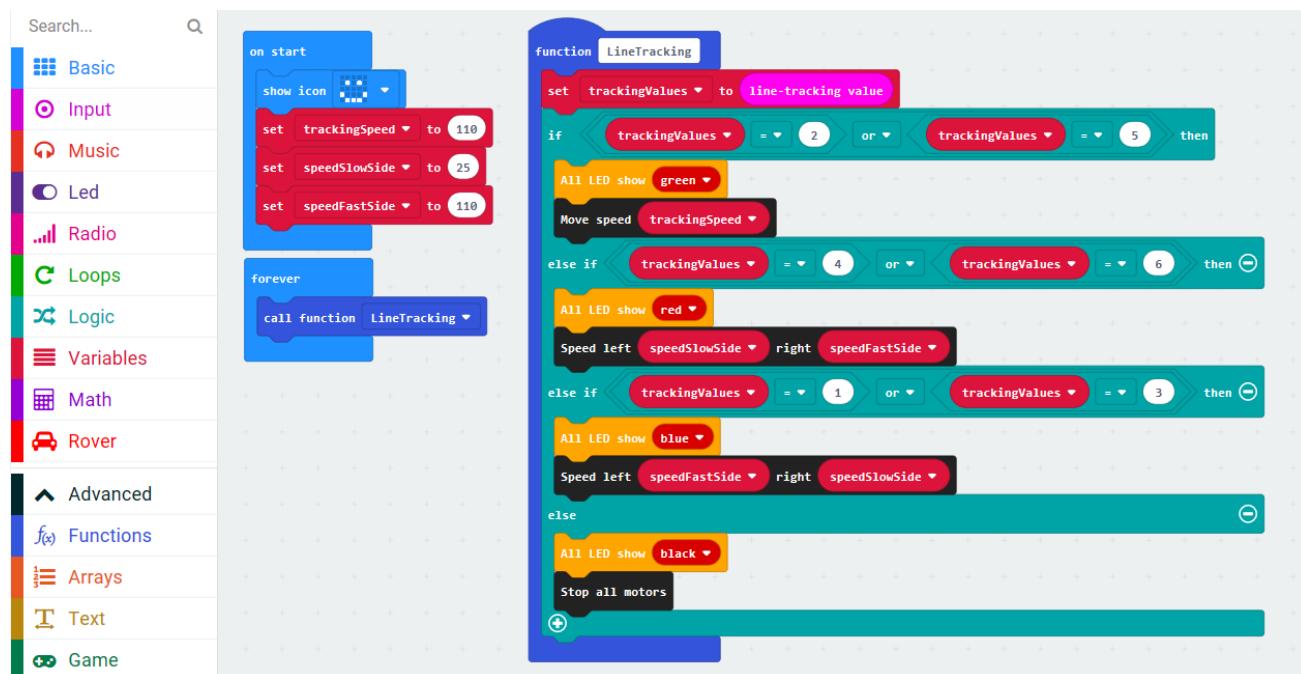
Flow chart is as below:



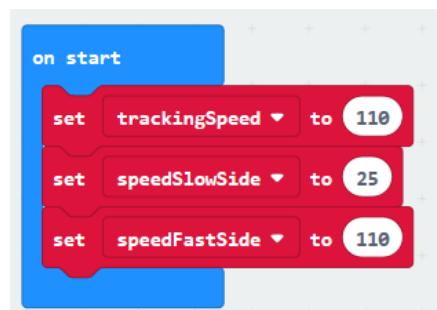
Code

Load code according to the table below:

File type	Path	File name
Hex file	./Projects/05.2_LineTracking	microbit-LineTracking.hex



This code uses three additional variables to set speeds and one subfunction.



Among them, `trackingSpeed` is used to set moving forward speed of Rover. It is not recommended to be too large, in order to avoid running out of the runway. Variable `speedSlowSide` and `speedFastSide` are used to set the turning speed. The smaller the speed difference, the smoother the turning, but the harder to pass through sharp turn. The larger the speed difference, the less smooth the turning, but the easier to pass through the sharp turn. For the specific runway we provide, the default speed in the program is enough. For a custom runway, you need to debug the most appropriate speed.



"Call function `LineTracking`" is equivalent to putting everything in the subfunction `LineTracking` in forever. And we will see many this kind of use later.

Download the code to micro:bit. And put Rover on map attached in this kit. Then observe motion of Rover.

Chapter 6 Bluetooth

micro:bit integrates a BLE(Bluetooth Low Energy) device with low power consumption, which can be connected and paired with smartphones through Bluetooth. Accordingly, we can create communicate between micro:bit and mobile phone, and use app to control Rover's action to realize the function of wireless remote control.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install battery into Rover.
3. Turn on Rover power.
4. Connect micro:bit and computer through USB cable.
5. Android mobile phone
6. Android app Freenove.([How to get android app?](#))

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

If you choose to drag code manually, you first need to add Rover and Bluetooth extension libraries as before. ([How to add Rover extension?](#)) Due to the limitation of micro:bit hardware, Bluetooth and Radio cannot work at the same time, so their extension library is not incompatible with each other. When installing Bluetooth extension library, you will be prompted to delete Radio extensions. Just confirm the removing.

Some extensions will be removed

Extension radio is incompatible with bluetooth. Remove radio and add bluetooth?

Remove extension(s) and add bluetooth



Cancel



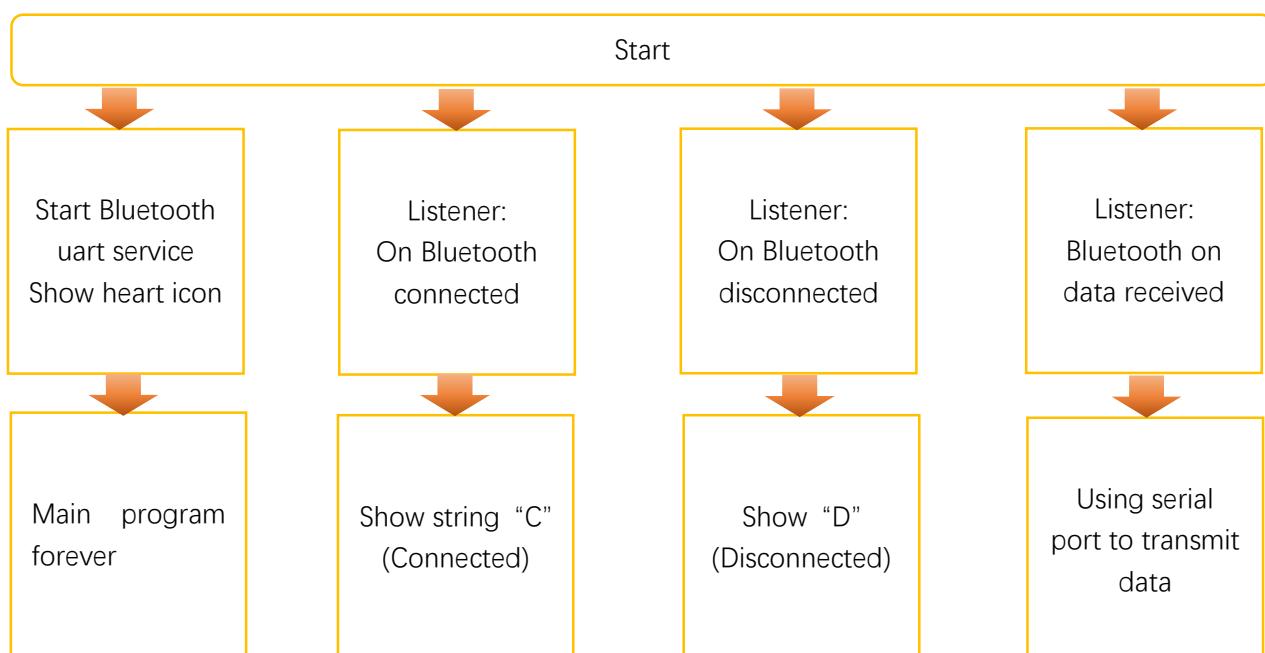
Getting Bluetooth Data

Content in this section is quite theoretical, which is for analyzing the Bluetooth data. If you really have no conditions to complete, you can jump to next section. It has no influence to later projects.

Get Bluetooth data and print the original Bluetooth data in makecode through uart.

Since the code in this section uses the uart printing function, the execution of this code must use the Windows 10 App, or Google Chrome with webUSB feature (how to use?) to see the same result. If you understand uart, you can also use a third-party serial tool.

Flow chart



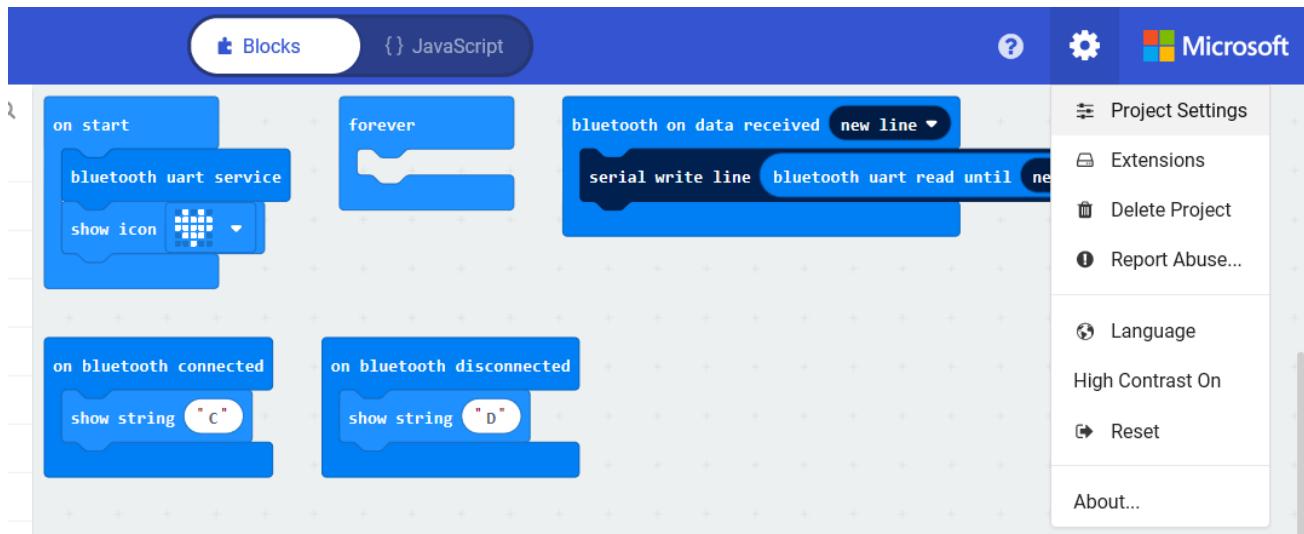
Code

Load code according to the table below or drag the code block as picture below:

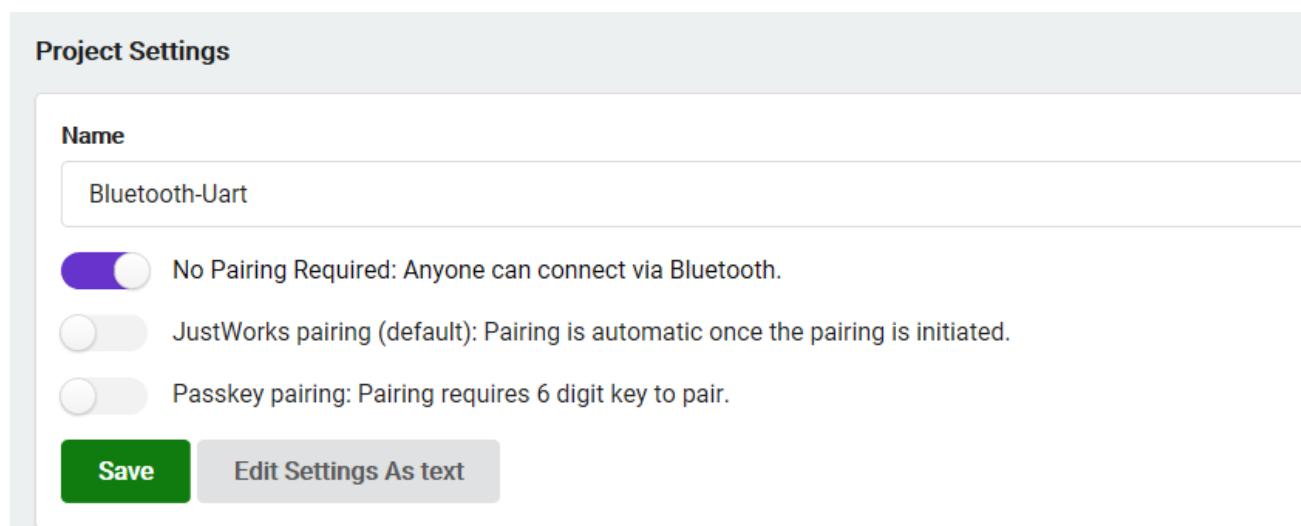
File type	Path	File name
Hex file	./Projects/06.1_Bluetooth-Uart	microbit-Bluetooth-Uart.hex



Then click gear icon(settings)→Project Settings.



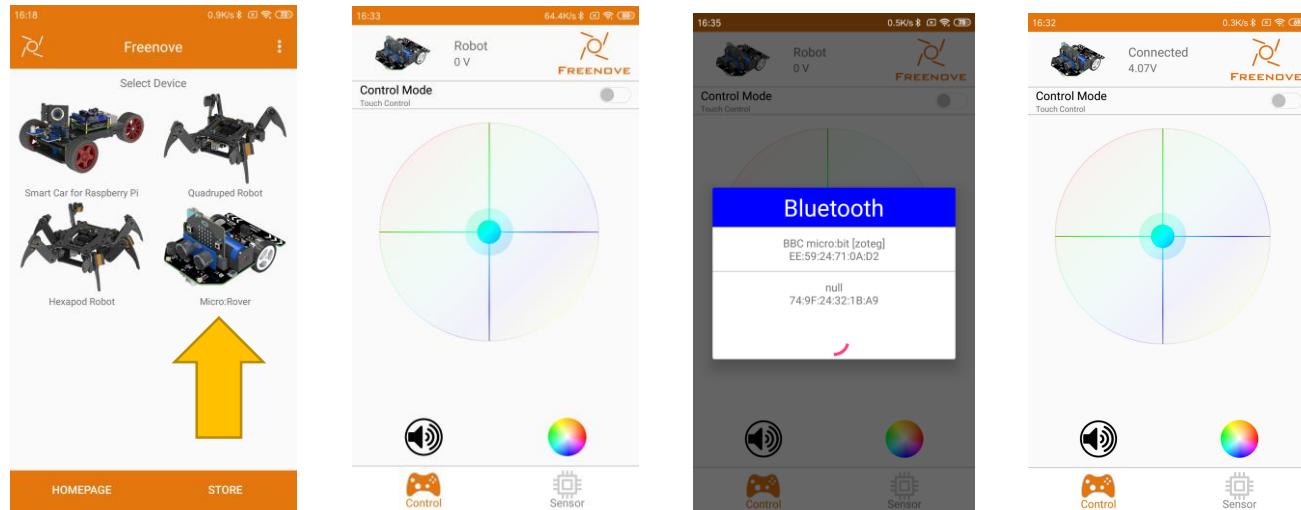
Then following page appears. And set "No Pairing Required" to open state as below.



Click Save button.

Download the code to micro:bit. Later, the LED matrix of micro:bit will show heart pattern.

Open android app Freenove. **Click on Micro:Rover**. Then **click Rover icon** in the upper left corner to search Bluetooth device. Then **click your micro:bit in the search list** to connect. After the connection is successful, "Connected" is showed in the status bar above, indicating that the connection is successful. In this process, there may be a pop-up window asking for permission to obtain location. Click OK, otherwise Bluetooth will not be searched.

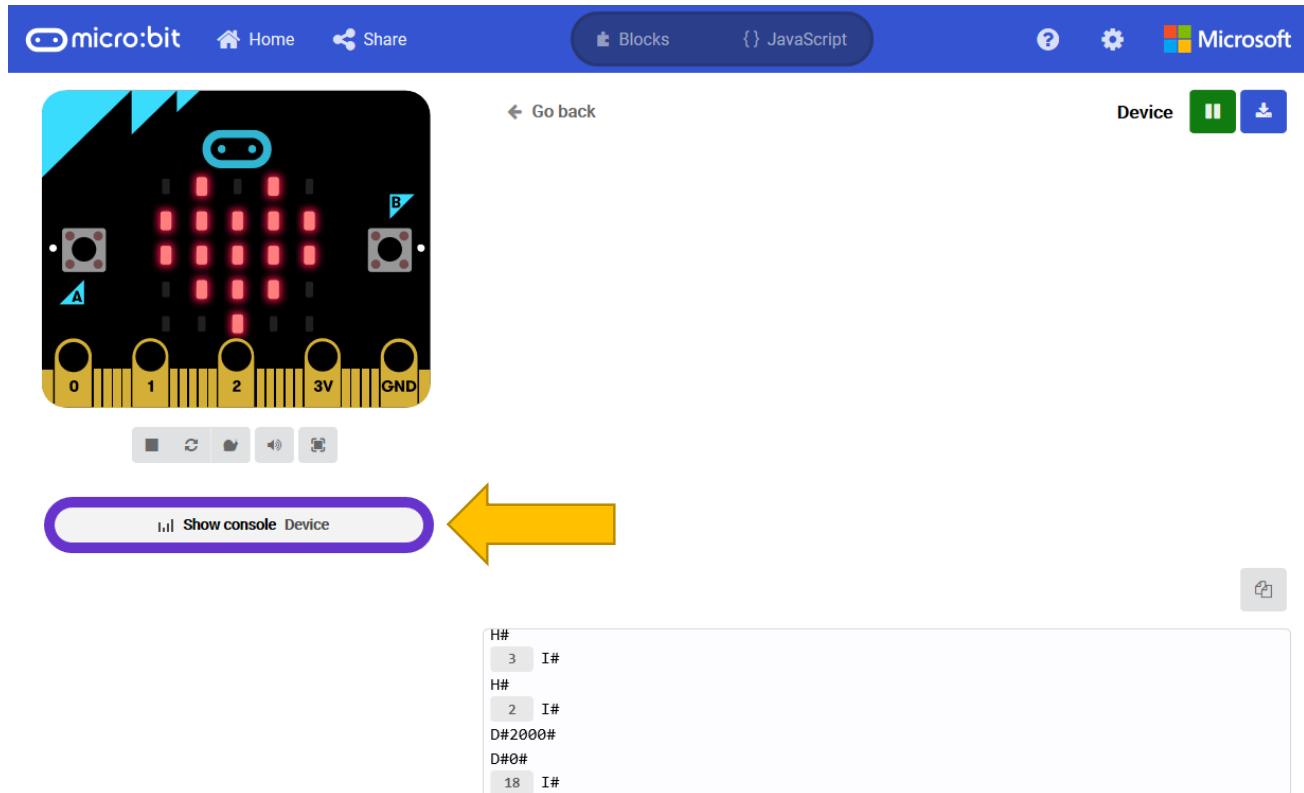


Click or drag the operating disk below, or click on the speaker, color plate or other controller below to view the data received in Makecode.



In Makecode, when data is received, there will be a “**ShowConsoleDevice**” button below the simulator. Then clicks this button to view the received serial data.

If you use web version of makecode, and no webUSB function, you will not see the “ShowConsoleDevice” button.



Data analysis

Learning the data commands sent by app will help to understand and analyze later program.

The command format for communication between app and micro:bit is A#xxx#xxx#...xxx#, where # is a separator, the first character A represents the action command, the following xxx represents the parameters of the action command. And different commands carries different parameters. The command list is as below:

Action command	Description	Command character	Number of parameters (app send/receive)	Format example (app send/receive)
MOVE	Move, parameters are speeds of the two motors	A	2	A#100#100#
STOP	Stop moving	B	0	B#
ORDER_RGB	Control RGBLED, parameters respectively are serial number of LED, red value, green value, blue value.	C	3	C#15#100#150#200#
BUZZER	Control buzzer, and parameter is the frequency.	D	1	D#2000#
DISTANCE	Get the distance of ultrasonic measurement	E	0/1	E# /E#50#
LIGHTING	Get the value of the light sensor	F	0/1	F# /F#512#
TRACKING	Get the value of the tracking sensor	G	0/1	G# /G#2#
MODE	Set motion mode of Rover	H	1	H#1#
VOLTAGE	Get the battery voltage of Rover, parameter is the voltage value, unit mV	I	0/1	I# /I#4100#
ECHO_OK	Answer	J	0	J#
NONE	None	K	0	K#

Warning: the maximum data length supported by micro:bit Bluetooth in one transmit is 20 bytes, and the part which exceed 20 bytes will be lost.



Bluetooth pairing

In previous settings, we switched on the option “no need for pairing”, which is convenient for us to debug the program.

The screenshot shows the 'Project Settings' interface for a project named 'Bluetooth-Uart'. Under the 'Name' section, the name is already filled in. Below it, there are three options for Bluetooth pairing:

- No Pairing Required: Anyone can connect via Bluetooth.
- JustWorks pairing (default): Pairing is automatic once the pairing is initiated.
- Passkey pairing: Pairing requires 6 digit key to pair.

At the bottom, there are two buttons: a green 'Save' button and a grey 'Edit Settings As text' button.

However, it also means that any compatible Bluetooth can be connected to your micro:bit, just like a WIFI without a password. Obviously this is not safe, because micro:bit can only be connected to one Bluetooth device at the same time. So once someone connects to your micro:bit, you won't be able to connect to your micro:bit.

The default option is “JustWorks pairing”,

The screenshot shows the 'Project Settings' interface for a project named 'Rover'. Under the 'Name' section, the name is already filled in. Below it, there are three options for Bluetooth pairing:

- No Pairing Required: Anyone can connect via Bluetooth.
- JustWorks pairing (default): Pairing is automatic once the pairing is initiated.
- Passkey pairing: Pairing requires 6 digit key to pair.

At the bottom, there are two buttons: a green 'Save' button and a grey 'Edit Settings As text' button.

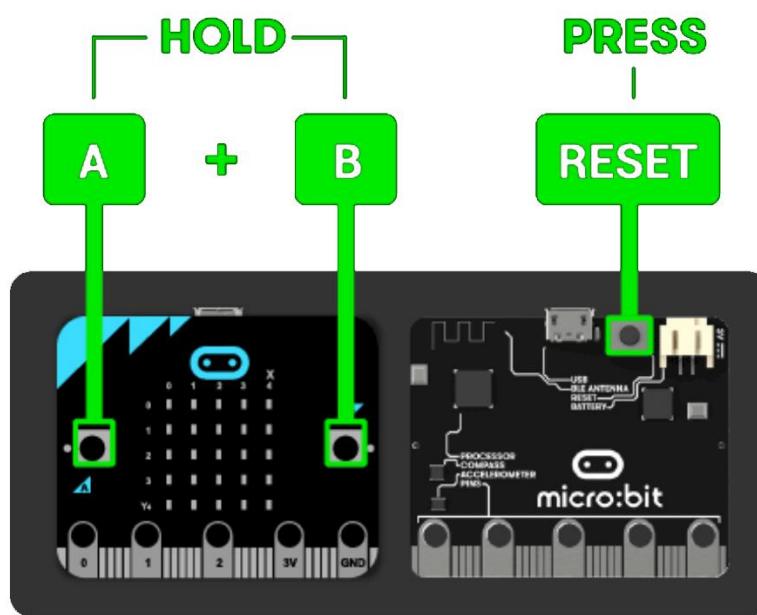
With this option, your micro:bit cannot be found by others. It can only be connected to the paired Bluetooth device before.

Pair

Following is the method to pair micro:bit:

Power the micro:bit. If you are not using USB, press and hold **A+B+Reset** buttons at the same time. About 3 seconds later, the LED matrix on micro:bit starts to light up one by one. **After all LEDs light up, release all buttons.** The LED matrix will show a Bluetooth icon, and then show a pattern which is the pairing code. Different micro:bit shows different pattern.

If you are using USB power supply, press and hold **A+B** buttons at the same time and then **click Reset** button. Then continue to keep A+B pressed. Then the LED matrix of micro:bit starts to light up one by one. **After all LEDs light up, release all buttons.** The LED matrix will show a Bluetooth icon, and then show a pattern which is the pairing code. Different micro:bit shows different pattern..



At this point, the micro:bit Bluetooth can be searched. Open the phone settings→Bluetooth, search Bluetooth, and pair with micro:bit.

During the pairing process, you may need to manually enter the connection key. Pay attention to the change of the LED matrix on the micro:bit. When an arrow-left pointing to the Button A is showed, press the button A, the micro:bit will show the connection key in turn. Record and Input the key in the phone. After the connection is successful, micro:bit will show a check mark, which means the pairing is successful.



Note that the micro:bit can only be paired with one Bluetooth at a time. When micro:bit is paired with other Bluetooth devices, you must re-pair to use. In other words, the same mobile phone can be paired with multiple micro:bits, but the same micro:bit can only be paired with one mobile phone.

If micro:bit(1) is successfully paired with mobile phone A, then it is paired with mobile phone B. Then micro:bit(1) will lose the pairing information with the mobile phone A. And mobile phone A must clear its pairing information and re-pair it with micro:bit(1) to use.

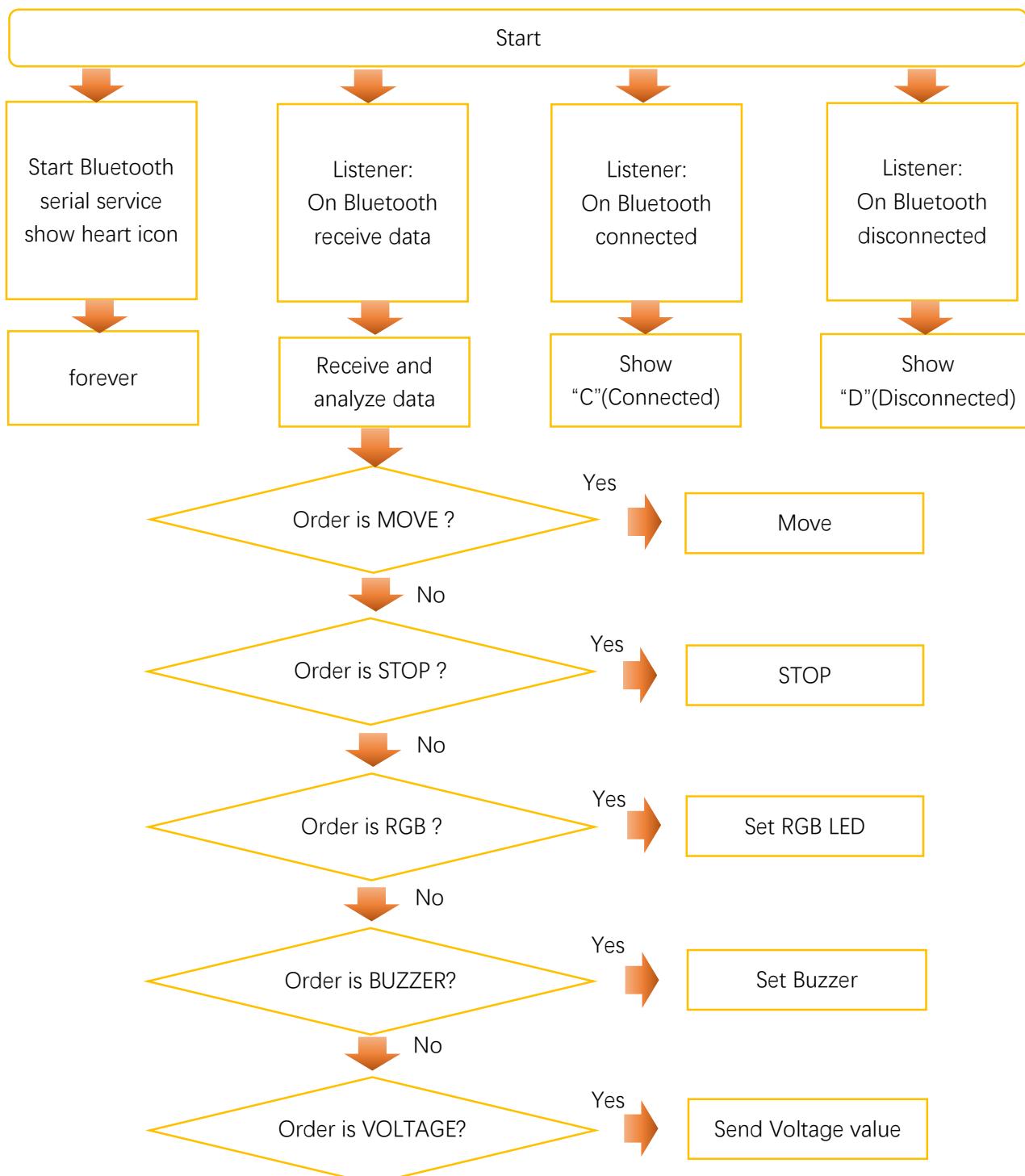
If you want to make multiple phones connected to one micro:bit at any time, you can select "No pairing Required" in the Settings of makecode as before.

You can decide whether you want to choose the pairing requirement. In the following projects, the default is no pairing requirement.

Rover-Remote control mode

According to the previous Bluetooth knowledge, let's realize Bluetooth remote control mode of Rover.

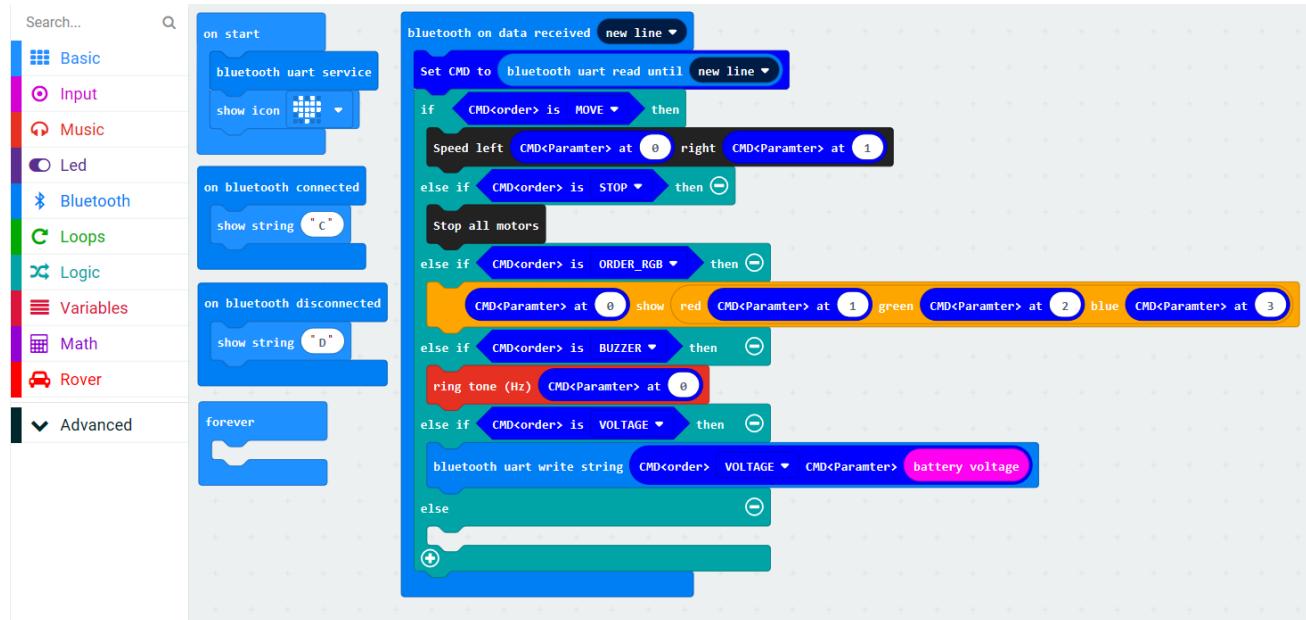
Flow chart



Code

Load code according to the table below:

File type	Path	File name
Hex file	./Projects/06.2_Bluetooth-Remote	microbit-Bluetooth-Remote.hex



This code makes micro:bit receive the command of APP through Bluetooth, and execute corresponding actions according to different commands.

In “Project Settings”, Switch “No Pairing Required” to open state.

Download the code to micro:bit. Later, LED matrix of micro:bit will show heart pattern.

Open Android app Freenove and connect to Rover via Bluetooth according to previous method. When the connection is successful, the LED matrix on micro:bit will show letter C. Click one position on control panel or drag the dot on control panel, then Rover will move accordingly. Click on the icon of the speaker, then Rover's buzzer will sound. Click on the color panel to control the color of RGB LED on Rover. About sensor, we will introduce them in next section.



Chapter 7 Android App

You may have noticed that on Android app. There is a tag "Sensor" that is not introduced. This chapter will combine all previous contents and write all the functions in one program to realize the remaining functions of app.

Preparation

1. Insert micro:bit into Rover correctly.
2. Install ultrasonic ranging module on Rover
3. Install battery into Rover.
4. Turn on Rover power.
5. Connect micro:bit and computer through USB cable.
6. Android mobile phone
7. Android app Freenove. ([How to get android app?](#))

Open web version of makecode or windows 10 app version of makecode.

If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.

[\(How to import?\)](#)

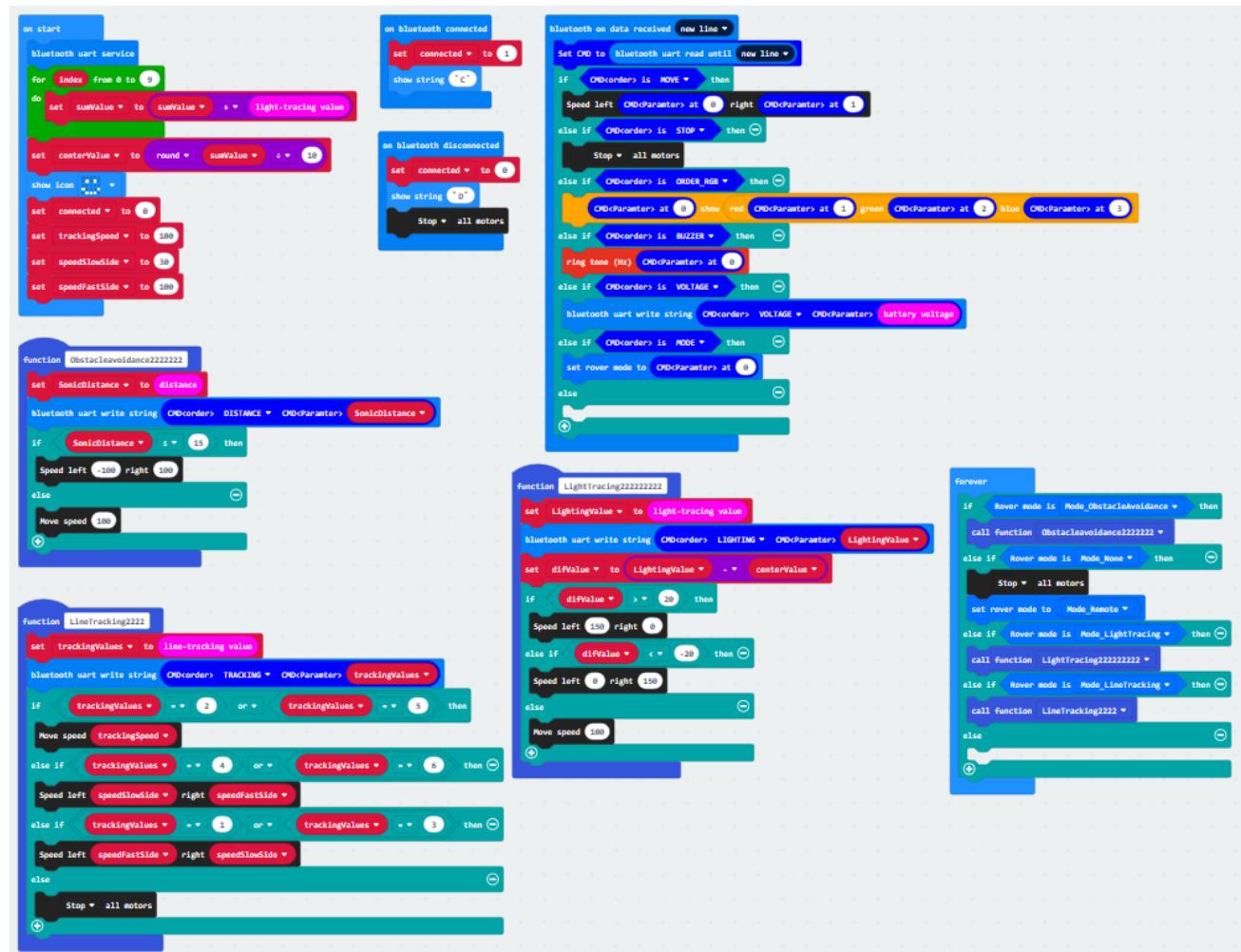
If you choose to drag code manually, you first need to add **Rover and Bluetooth extension** libraries as before. ([How to add Rover extension?](#))

Rover

Code

Load code according to the table below:

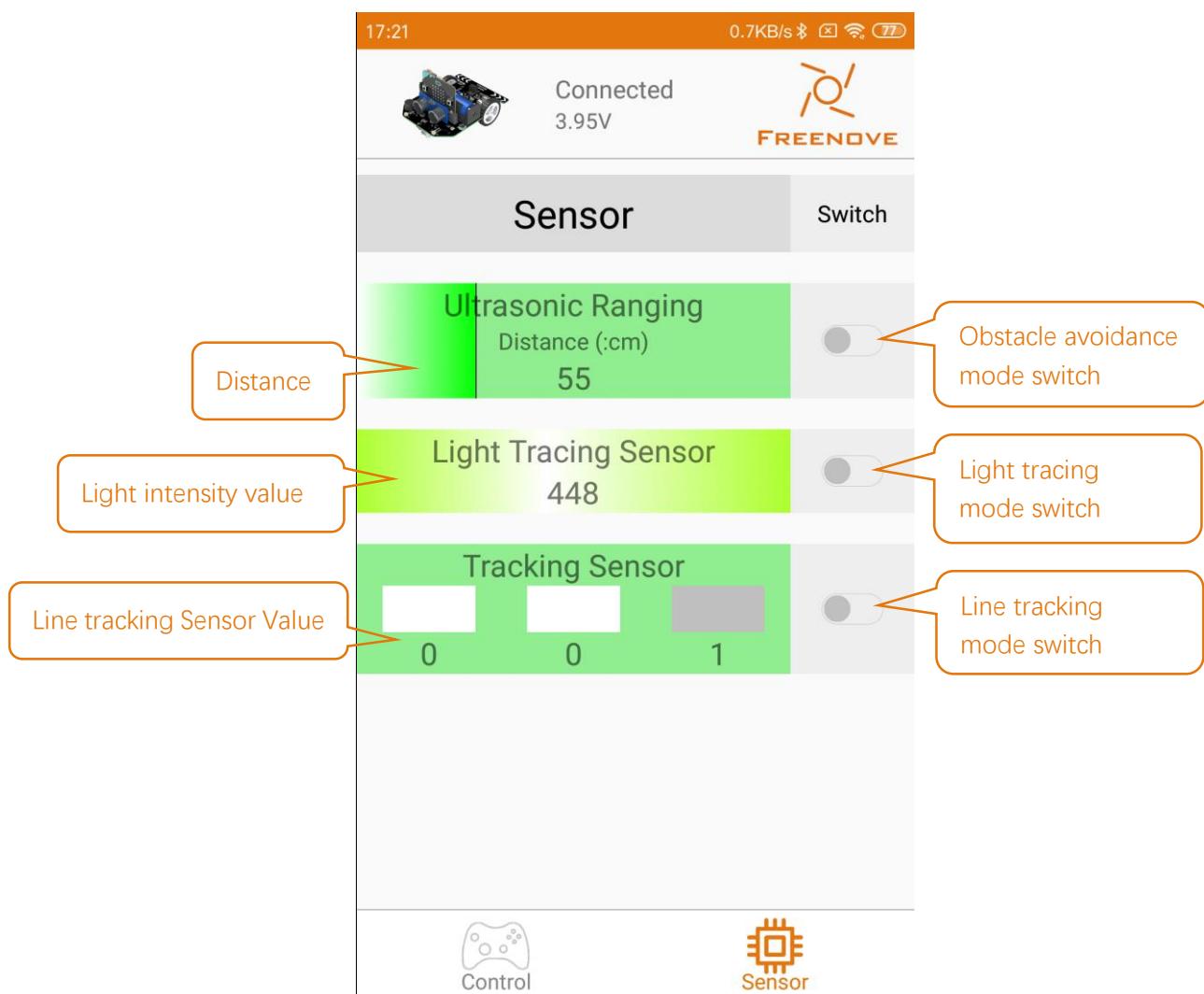
File type	Path	File name
Hex file	./Projects/07.1_Rover	microbit-Rover.hex



This code combines contents of all the previous sections, and its logical structure is still to make Rover execute corresponding actions according to the commands received by Bluetooth.

In “Project Settings”, set “No Pairing Required” to open state.

Download the code to micro:bit. Put the two light sensors in the same light intensity environment when booting. Wait the micro:bit LED Matrix to show a smile, which indicates that the calibration is completed. Open Android app Freenove, then complete Bluetooth connection according to previous method. Then you can use controls on Control and Senor pages to control Rover.



Rover can only work in one mode at a time. It cannot work in many modes at the same time.

This project no demo video.

Next

Here, all projects that do not require additional components have been completed.

We have also prepared an additional free tutorial. It contains several projects. But they need additional components or modules that are not included in the Rover kit. If you are interested, you can find it here:

View: https://github.com/Freenove/Freenove_Micro_Rover_Extended_Projects

Download: https://github.com/Freenove/Freenove_Micro_Rover_Extended_Projects/archive/master.zip

If you do not have these components, you will not be able to complete the next projects.

If you're not interested in extra projects. Never mind. Just leave them alone.

Enjoy your Rover.



Appendix

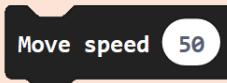
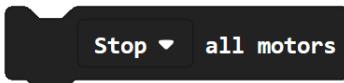
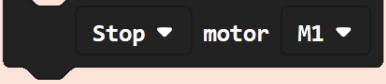
The appendix section introduces the role of each block in Rover Extension.

Rover Extension divides all blocks into four groups according to their functions: [LED, Motors, Sensors, Commands].

LEDs

Block	Function
	Set the brightness of all RGB LEDs on the Rover.
	Set specific RGB LED to a specific color.
	Set all RGB LED to a specific color.
	Color palette provides 16 kinds of colors.
	The HSL color picking model returns the RGB color value.
	RGB LED combination selection list.
	Common color list provides 10 kinds of colors
	The RGB color picking model returns the RGB color value.

Motors

Block	Function
	Set the two motors at the same speed to make the Rover move forward or backward. The positive value is for forward and the negative value is for backward.
	Set the speed of the left and right motors to make the Rover move or turn.
	Set the speed of only one motor (M1 or M2).
	Make two motors stop or brake at the same time.
	Make only one motor (M1 or M2) stop or brake.

Sensors

Block	Function
	Start the ultrasonic ranging module and return the measured distance. This block is a time-consuming block. If you use this block multiple times in a short time, you need use a variable to save the returned distance value.
	Returns the value of the line-tracking Sensor.
	Returns the value of the light-tracing Sensor.
	Returns the battery voltage value.

Commands

Block	Function
	Set CMD as a string and parse it into CMD<order> and CMD<parameters>. And store them in CMD<order> and ArrayList<parameters> accordingly.
CMD<order>	Return the parsed command.
ArrayList<parameters>	Return an array in which parsed parameters are stored.
count of paramters	Returns the number of parsed parameters.
	Determine whether CMD < order > is the currently specified command.
	Gets the specified parameter in the parameter array.
	Combine a specified command with a specified parameter. It is used when sending a command.
	Set a variable to represent the mode of Rover.
	Judge if the Rover mode variable is the specified mode.
	Command/action list.
	Rover mode list.

What's next?

Thanks for your reading.

This tutorial is all over here. If you find any mistakes, missions or you have other ideas and questions about contents of this tutorial or the kit and etc, please feel free to contact us, and we will check and correct it as soon as possible.

If you want to learn more about micro:bit, Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

Thank you again for choosing Freenove products.