Getting Started

About Battery

First, read the document **About_Battery.pdf** in the unzipped folder.
If you did not download the zip file, please download it and unzip it via link below.
https://github.com/Freenove/Freenove_Micro_Rover/archive/master.zip

Get Support and Offer Input

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

- Product quality issues
- Product use and build issues
- Questions regarding the technology employed in our products for learning and education
- Your input and opinions are always welcome
- We also encourage your ideas and suggestions for new products and product improvements

For any of the above, you may send us an email to:

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:

- Keep this product out of the reach of children under 6 years old.
- This product should be used only when there is adult supervision present as young children lack necessary judgment regarding safety and the consequences of product misuse.
- This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.
- When the product is turned ON, activated or tested, some parts will move or rotate. To avoid injuries to hands and fingers, keep them away from any moving parts!
- It is possible that an improperly connected or shorted circuit may cause overheating. Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it cools down! When everything is safe and cool, review the product tutorial to identify the cause.
- Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
- Store the product in a cool dry place and avoid exposing the product to direct sunlight.
- After use, always turn the power OFF and remove or unplug the batteries before storing.

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

http://www.freenove.com

Copyright

FREENOVE ®

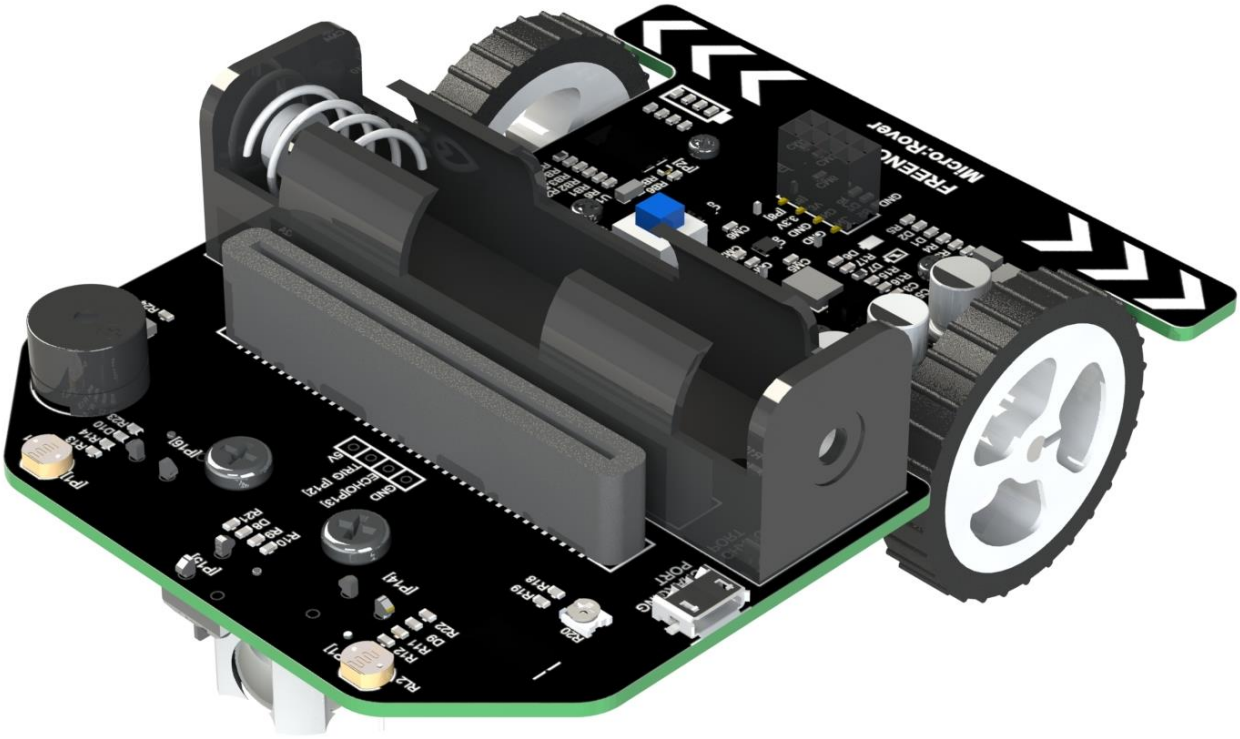
FREE YOUR INNOVATION

micro:bit® is a trademark of Micro:bit Educational Foundation (https://www.microbit.org/).

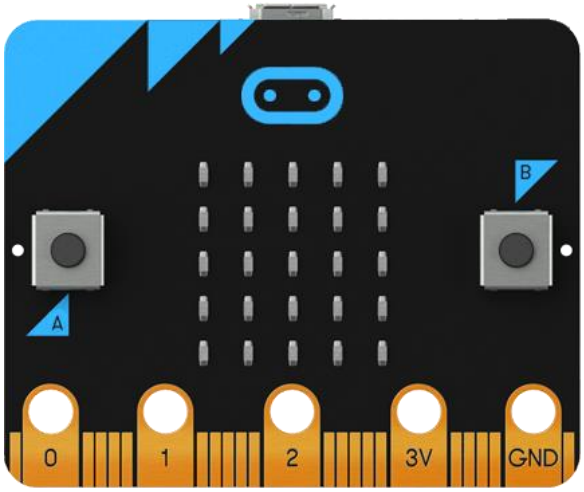
Contents

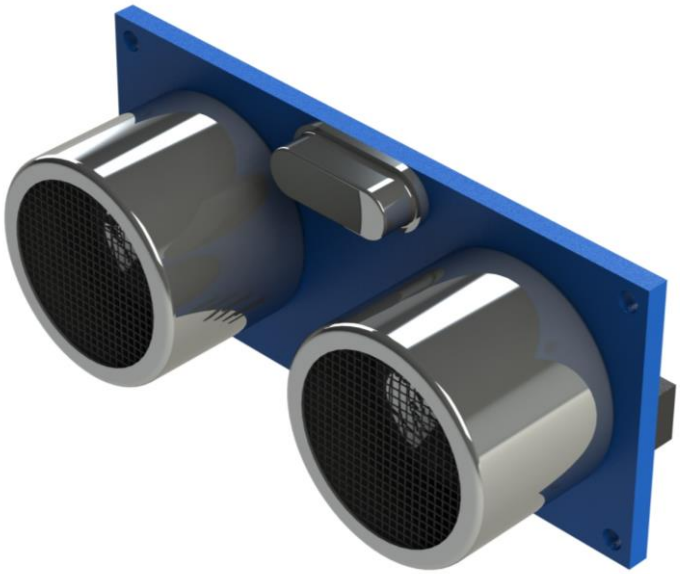
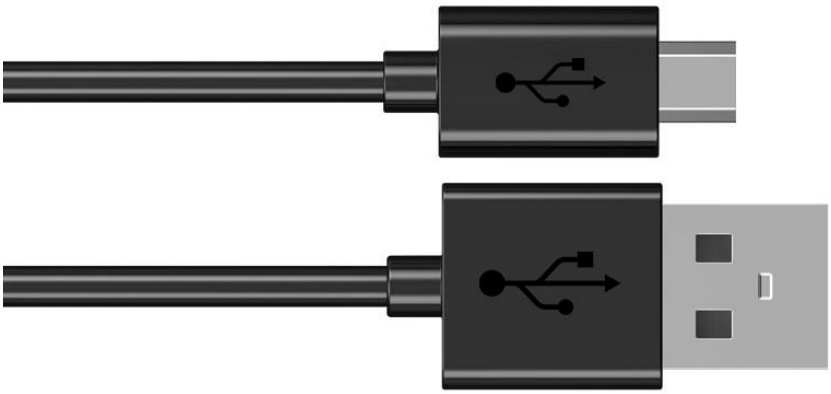
List

Micro:Rover

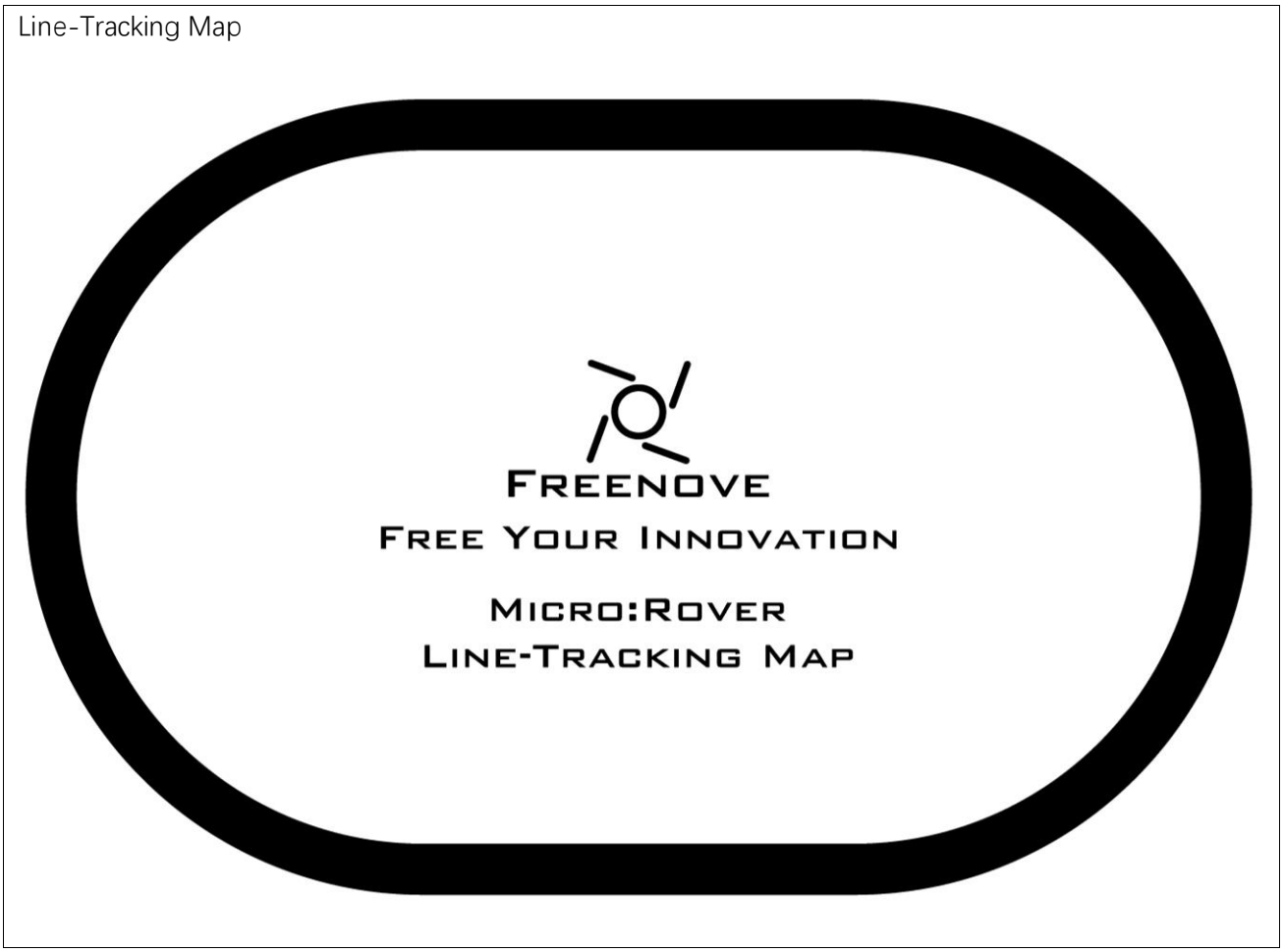


BBC micro:bit



Ultrasonic ranging module 
Micro USB cable 

Line-Tracking Map



Preface

Do you want to learn programming or get a robot programmed by yourself?
Currently, Program is developed into the younger age group, and everyone programming is a trend. From Arduino and Raspberry Pi to micro:bit, simple graphical programming makes programming for kids possible. It doesn't matter even though you haven't heard of them. 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, its simple graphical programming interface allows you to master it easily. It doesn't require any professional programming software; just simply a browser is enough. So, no matter your computer system is Windows, Linux or Mac, they all work. And you can also program it with Python.

It attracts a lot of fans in the world who are keen to exploration, innovation and DIY and have contributed a great number of high-quality open source code, circuit and rich knowledge base, thus helping us realize our own creativity more efficiently by using these free resource. Of course, you can also make contributions to the resource.

And that's why Freenove Micro:Rover emerged. Rover is a programmable car developed by Freenove based on BBC micro:bit. Adopting integrated design, it does not require additional wiring, which makes it easy to use. There are ultrasonic, infrared and photosensitive sensors on it as well as buzzer, RGB LED and other peripherals. Rich hardware resources will help you master more knowledge and skills. You can use your imagination to create more ways to play the robot.

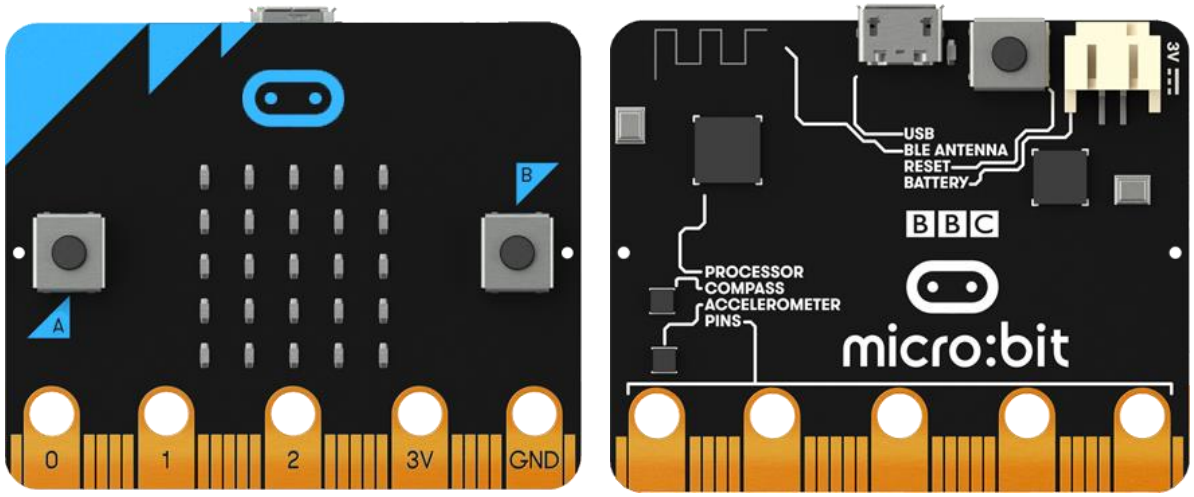
In each learning chapter of this tutorial, we provide program source code with detailed program explanations and burnable binaries, contributed to your understanding of the meaning of each section of program.

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

Welcome to the world of electronic and programming.
Our journey to build and explore Micro:bit and Micro:Rover electronic projects will start with this chapter.

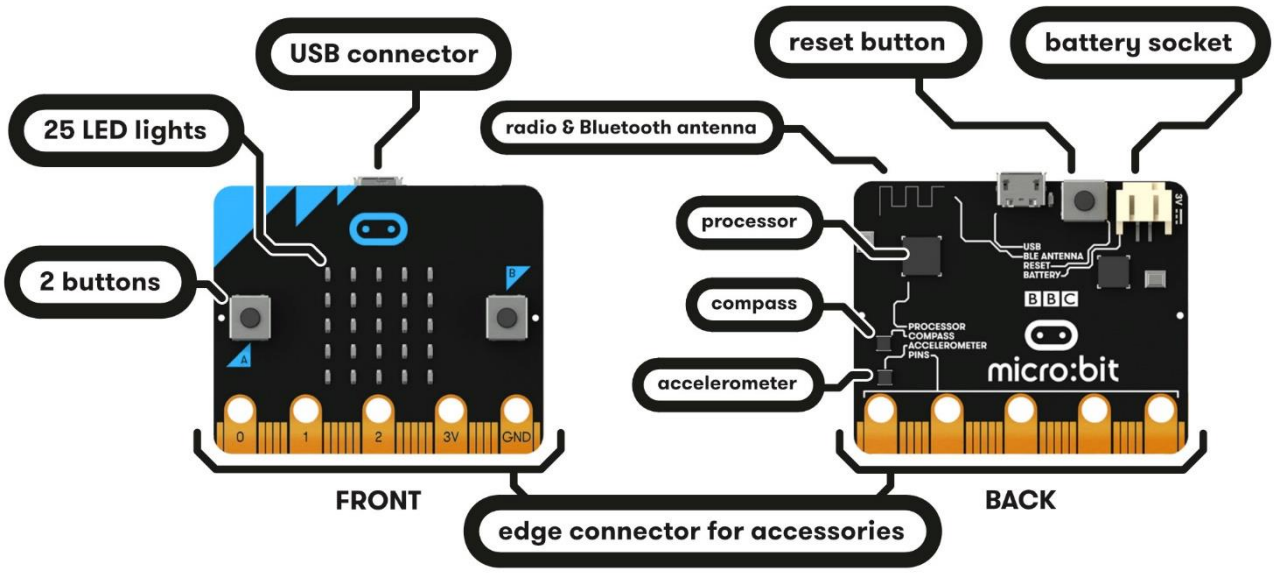
Meet micro:bit



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

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/

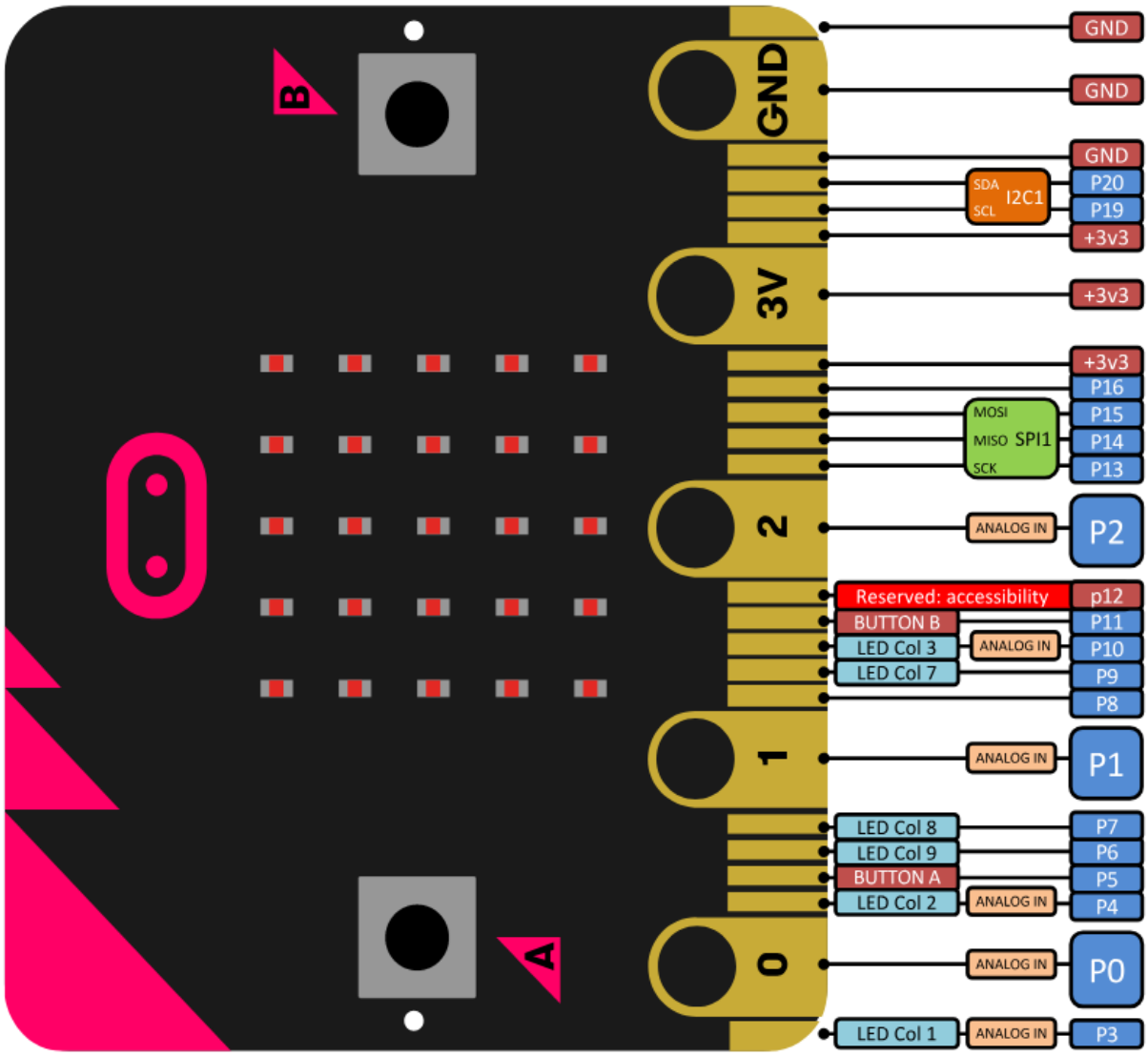
Hardware

It is not required that beginners master this section, but a brief understanding is necessary. However, 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 building Rover, we need a brief understanding of GPIO.

GPIO

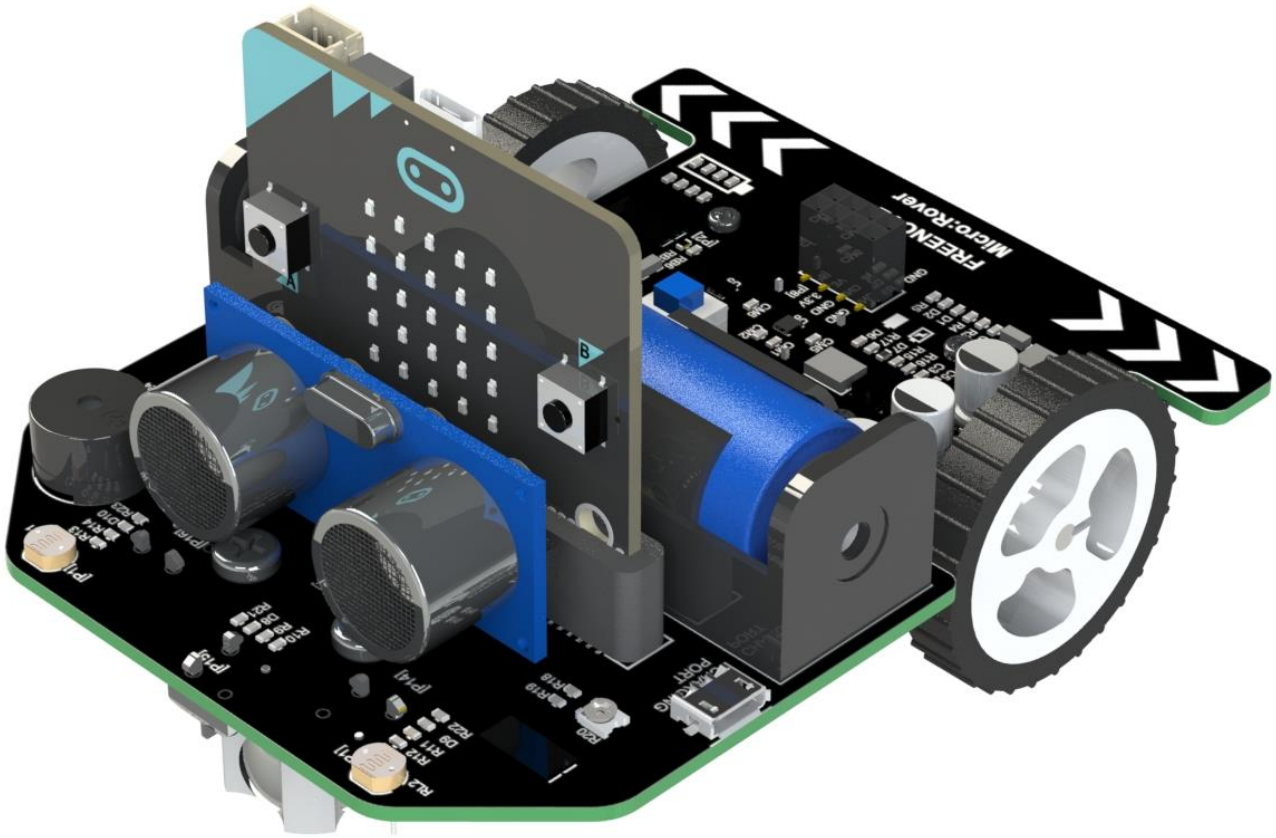
GPIO, namely General Purpose Input/output Pins, is an important part of micro:bit for connecting external devices. All sensors and devices on Rover communicate with each other through micro:bit GPIO. The following is the GPIO serial number and function diagram of micro:bit:



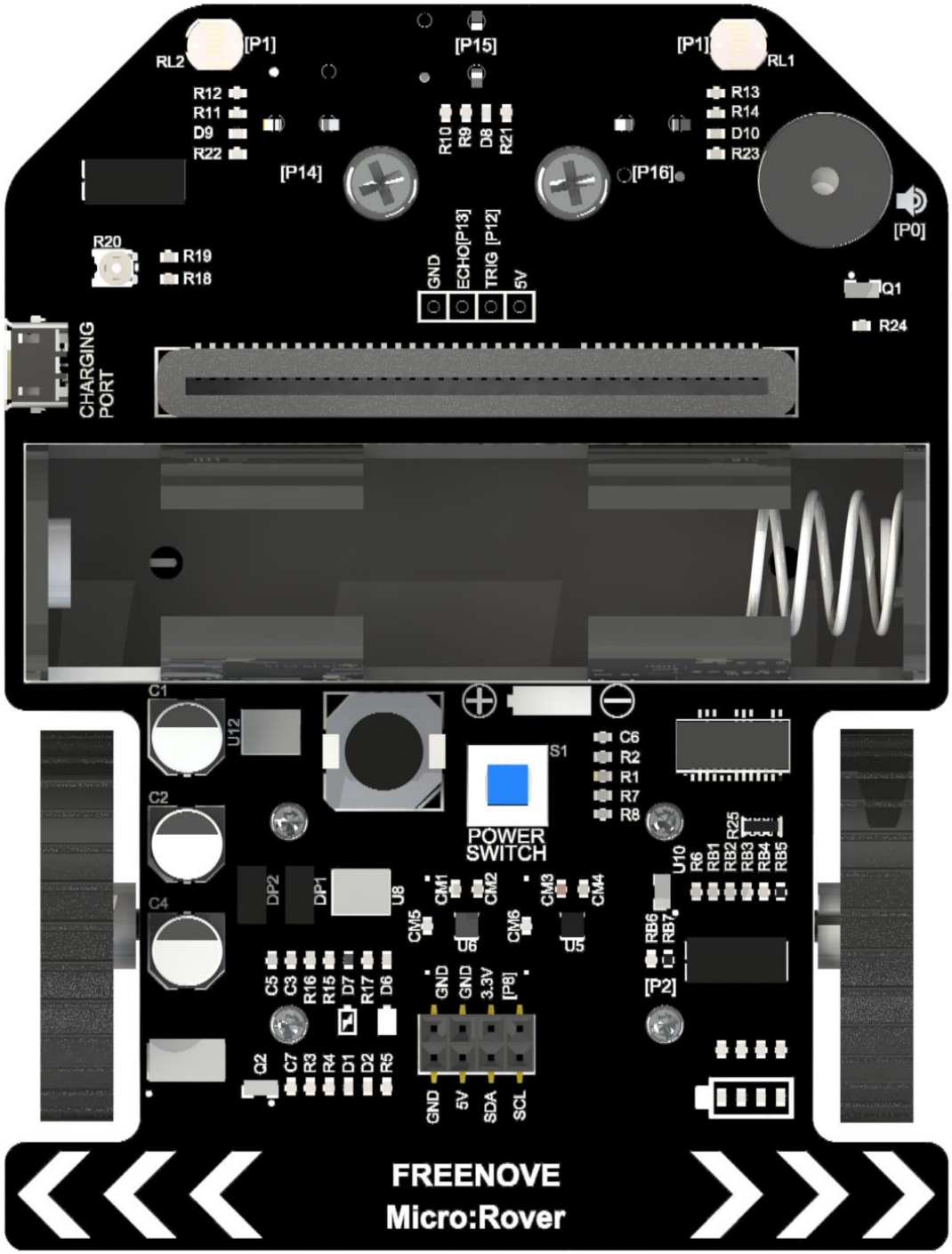
Micro:Rover

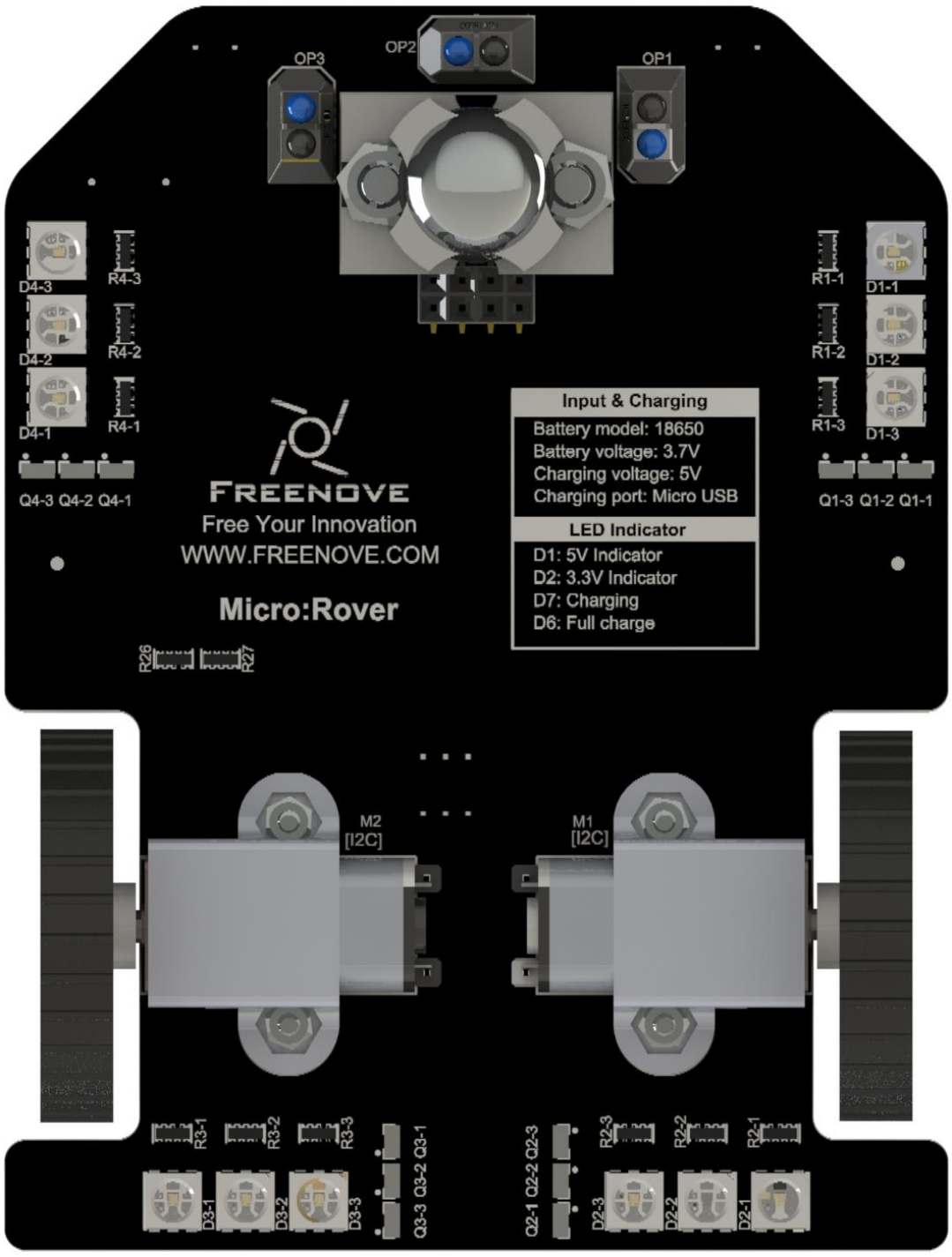
In this chapter, we will introduce the functions and features of Micro:Rover.

Freenove Micro:Rover is a programmable car based on BBC micro:bit. With integrated design, it does not require additional wiring, which makes it easy to use. There are ultrasonic, infrared and photosensitive sensors on it as well as buzzer, RGB LED and other peripherals. Rich hardware resources will help you master more knowledge and skills. You can use your imagination to create more ways to play the robot.



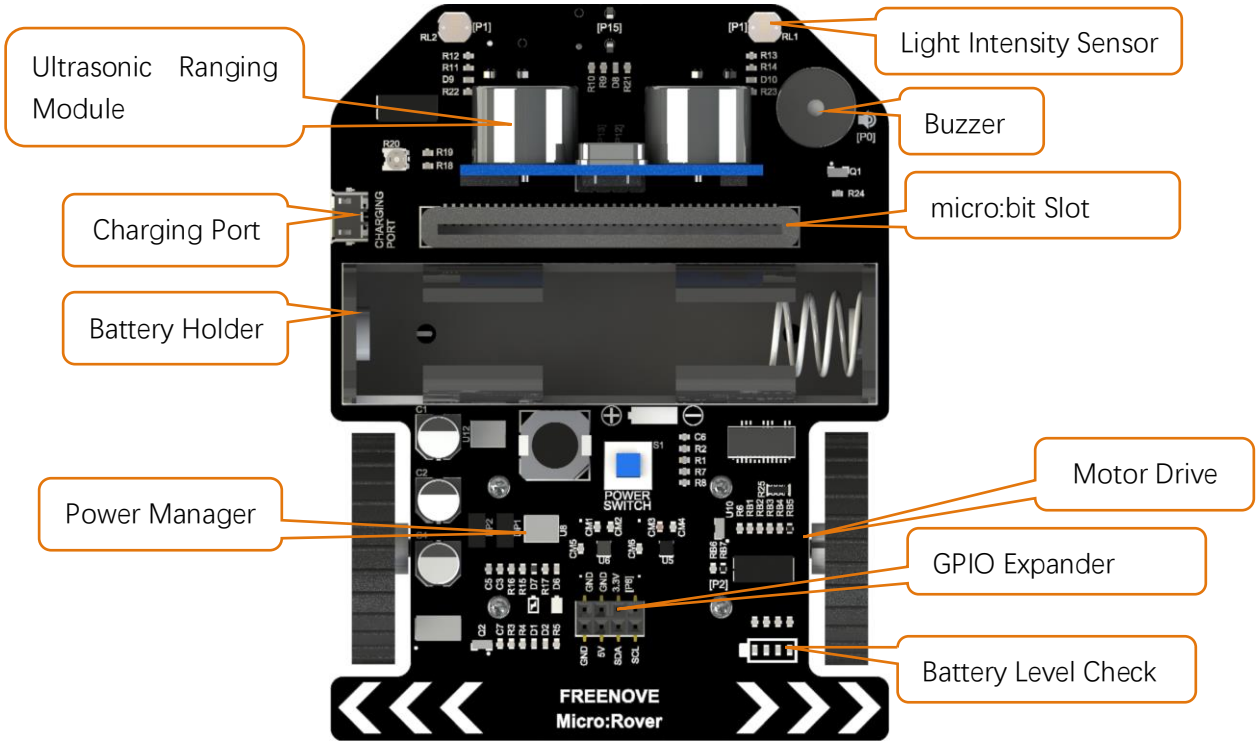
Meet micro:rover

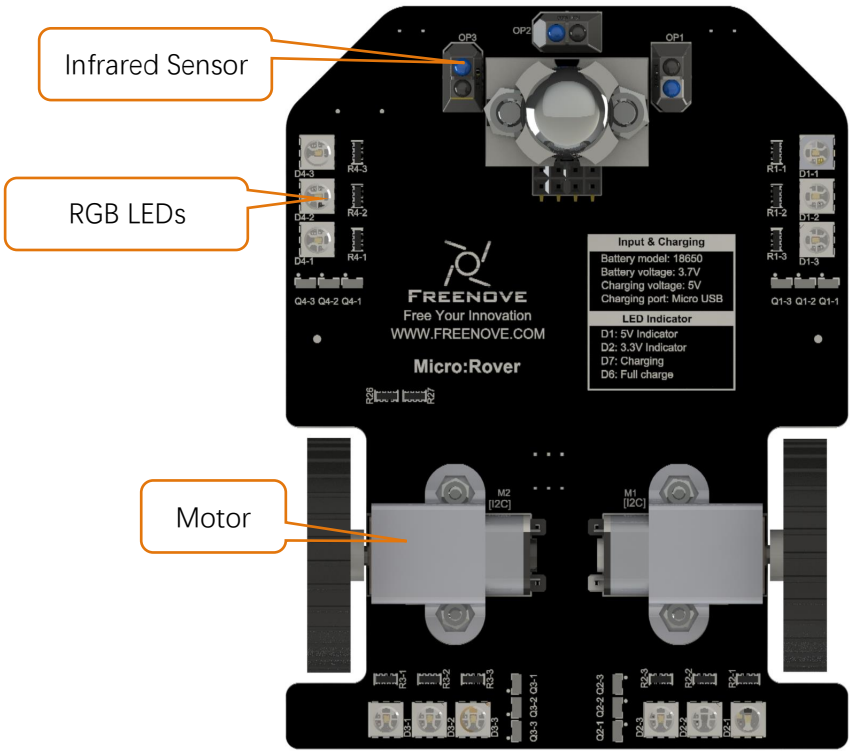




The Freenove Micro:Rover is a multi-functional car based on BBC micro:bit. It has rich sensors and peripherals to help you further learn how to use micro:bit and acquire knowledge about electronics.

Features


Ultrasonic Ranging Module
Light Intensity Sensor
Buzzer
Charging Port
micro:bit Slot
Battery Holder
Motor Drive
Power Manager
GPIO Expander
Battery Level Check
FREENOVE
Micro:Rover


Infrared Sensor
RGB LEDs
Motor
FREENOVE
Free Your Innovation
WWW.FREENOVE.COM
Micro:Rover
Input & Charging
Battery model: 18650
Battery voltage: 3.7V
Charging voltage: 5V
Charging port: Micro USB
LED Indicator
D1: 5V Indicator
D2: 3.3V Indicator
D7: Charging
D6: Full charge

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, no matter it is pointed/flat/with or without protective plate. 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.

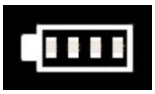
Indicator

Some LEDs are integrated on Rover, different states of each have various meanings. . You can learn the working state of Rover better 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		

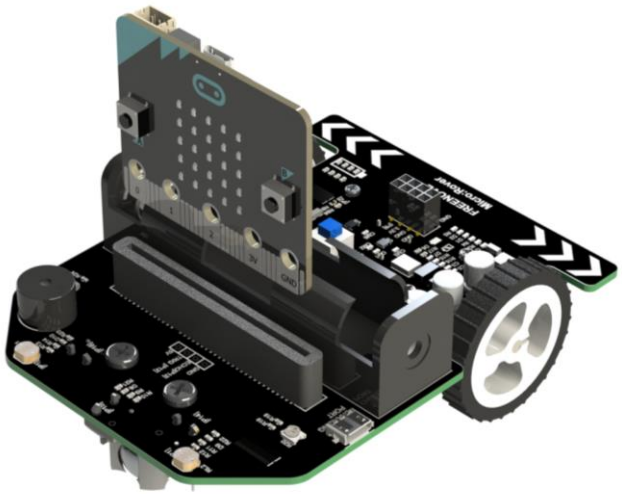
D7 State	D6 State	Indicating meaning
On	Off	Charging.
Off	On	Charging is complete and the battery is full.
Blink	On	The charger did not detect the battery.
Off	Off	The boost circuit is working.

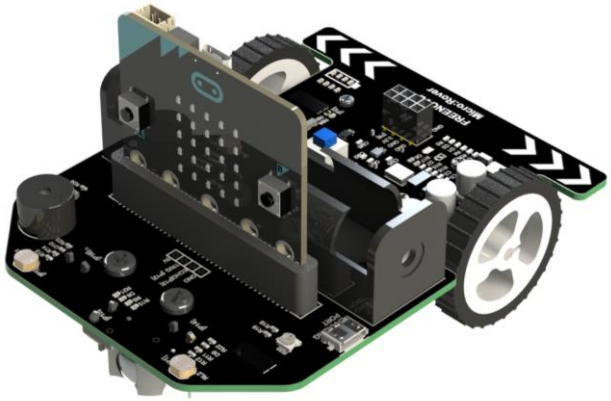
On bottom right corner of Rover, four LEDs are used to indicate battery level, as shown blow.



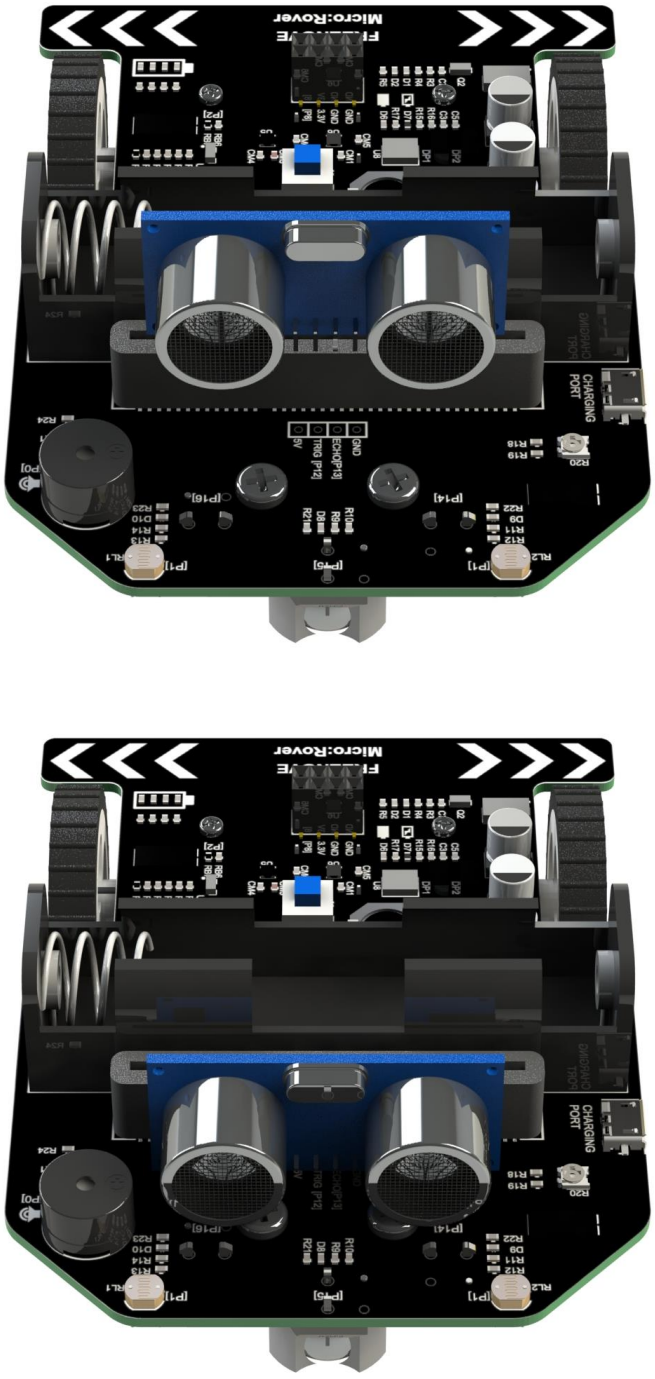
Assemble

Install micro:bit

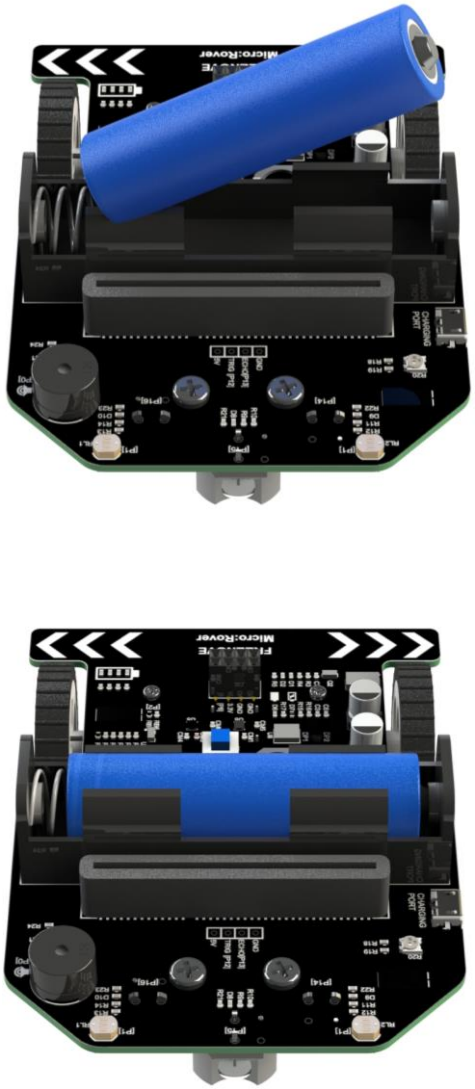




Install ultrasonic ranging module



Install battery



Code & Programming

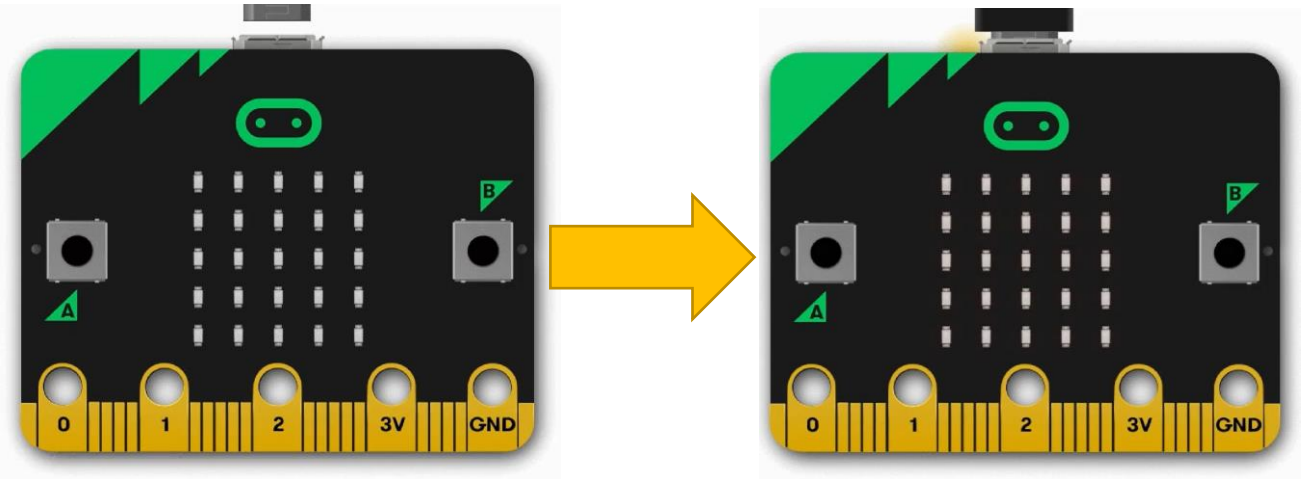
If you have any concerns, please feel free to contact us at **support@freenove.com**

Quick Start

This section describes how to write programs 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: Connecting Micro:bit

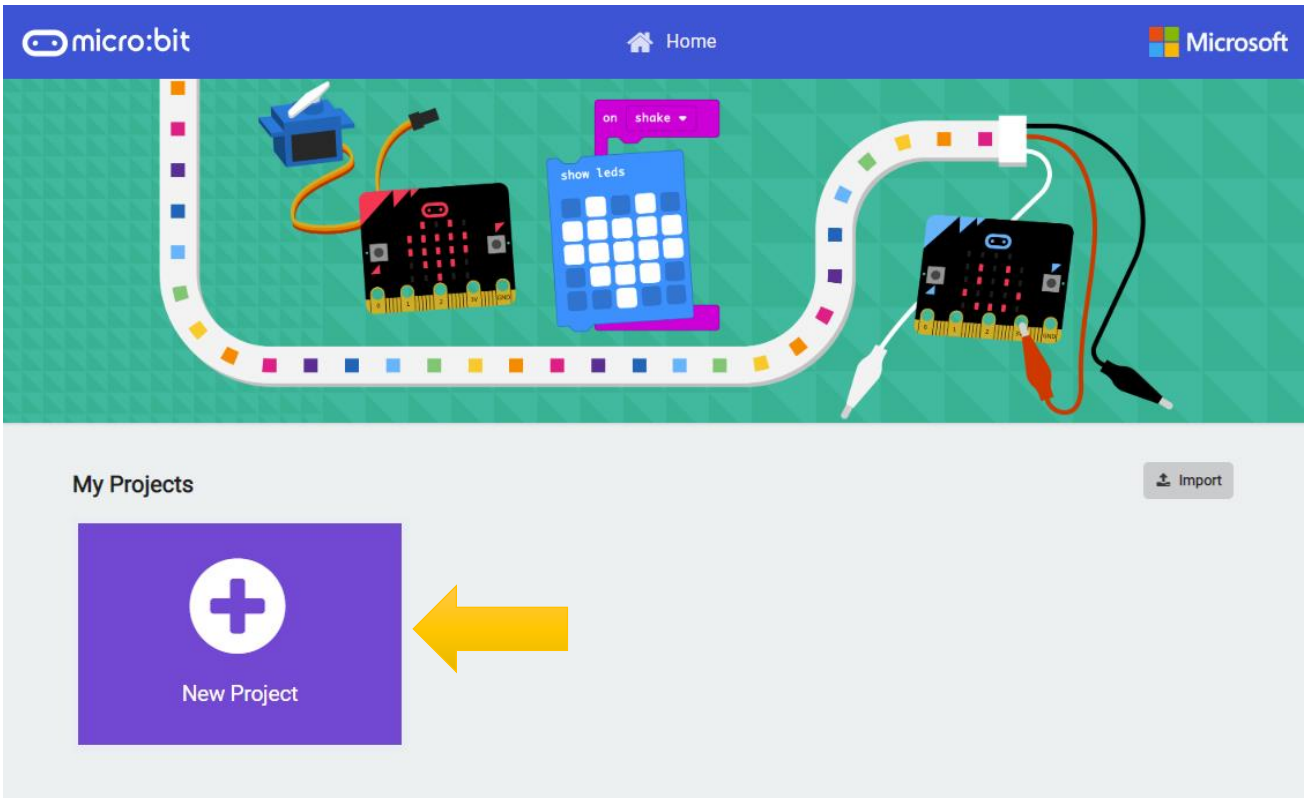
Connect the micro:bit to your computer via a micro USB cable. Macs, PCs, Chromebooks and Linux systems (including Raspberry Pi) are all supported.
It comes with an interesting application, give it a try!



Step 2: Create your Program

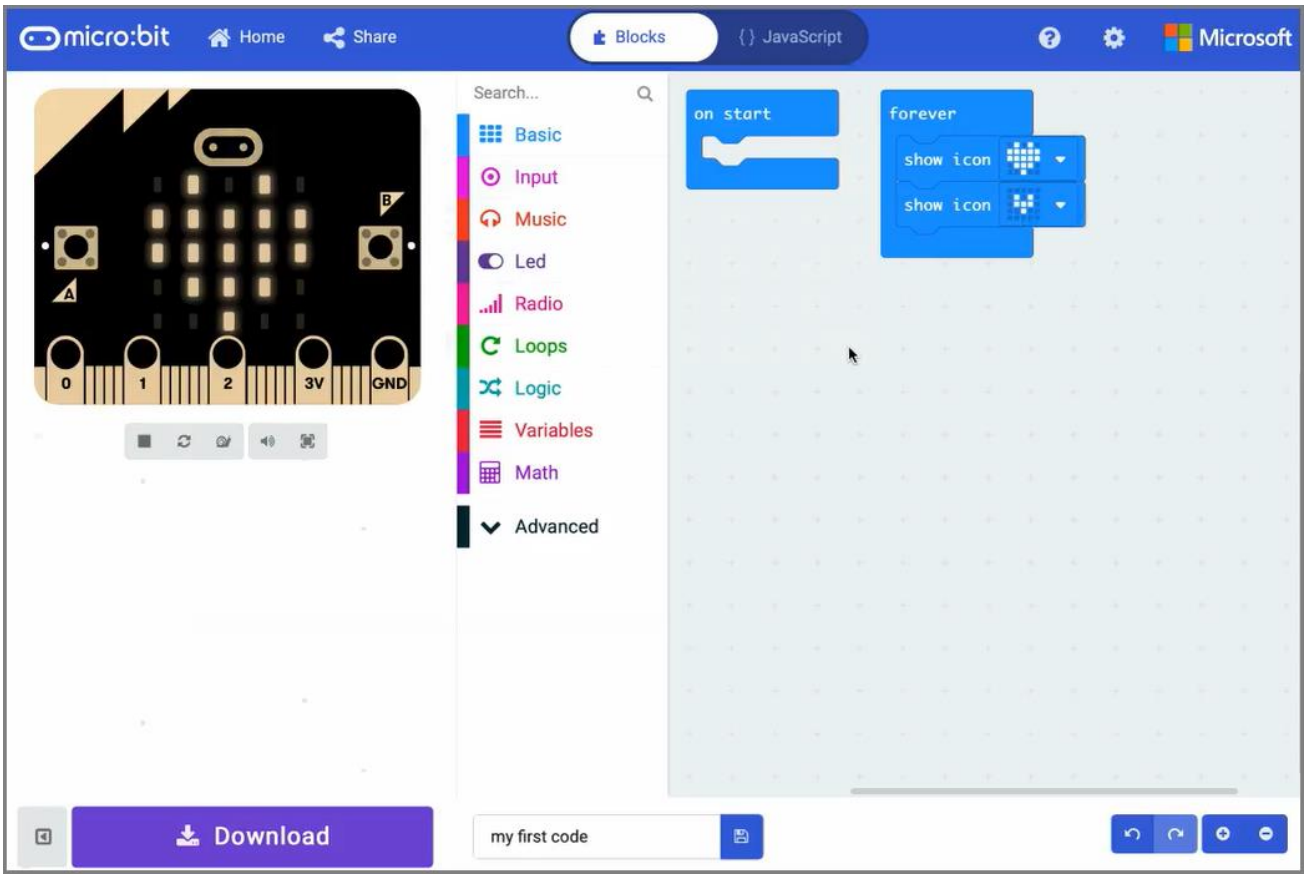
Visit https://makecode.microbit.org/. Then click "New Project" and start programming.

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, as in the image below that shows how to program a Flashing Heart.

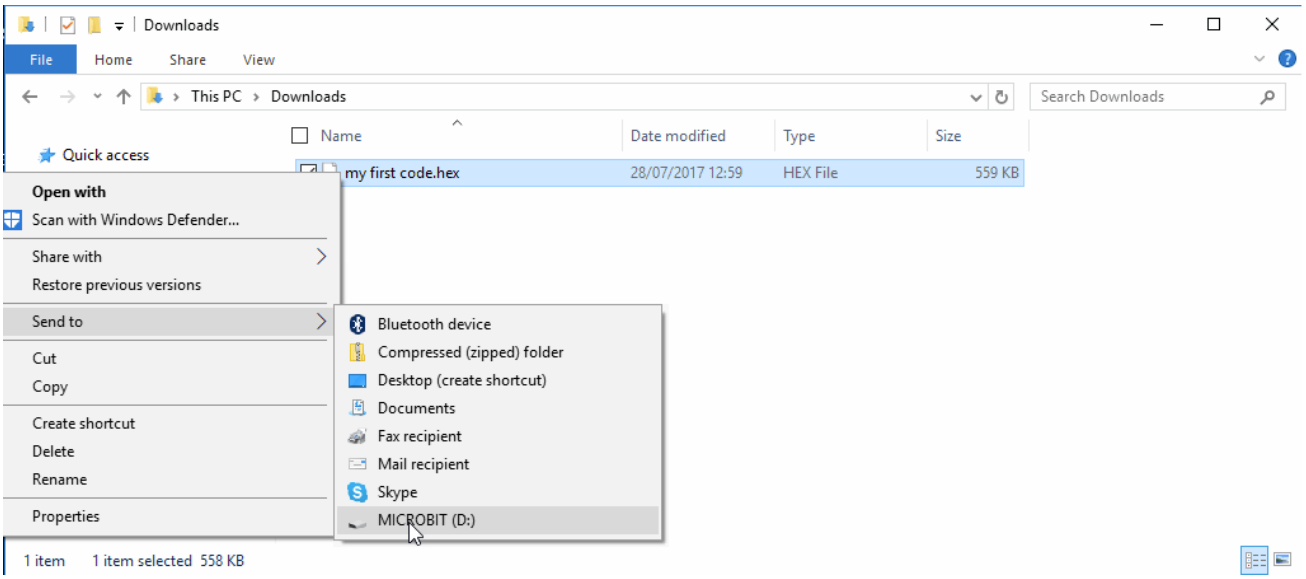
Here is a demo video: https://microbit.org/images/quickstart/makecode-heart.mp4
MakeCode will be further introduced in next section.



Step 3: Flashing program to your Micro:bit

The process of transferring the .HEX file to the BBC micro:bit is called flashing.
If you write program 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 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 been 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: Displaying your Program

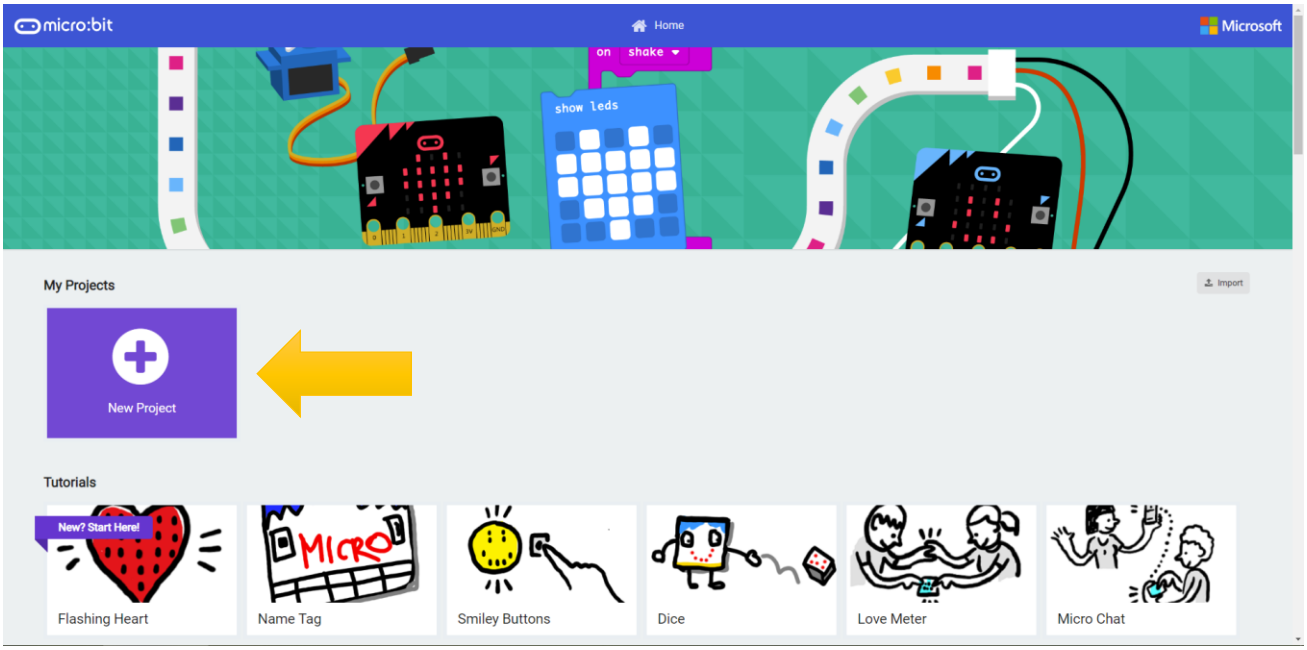
The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is programmed. Once that's finished the code will run automatically!
Note: The micro:bit can only run one program at a time - every time you drag-and-drop a hex file onto the device over USB it will erase the current program and replace it with the new one.

Warning

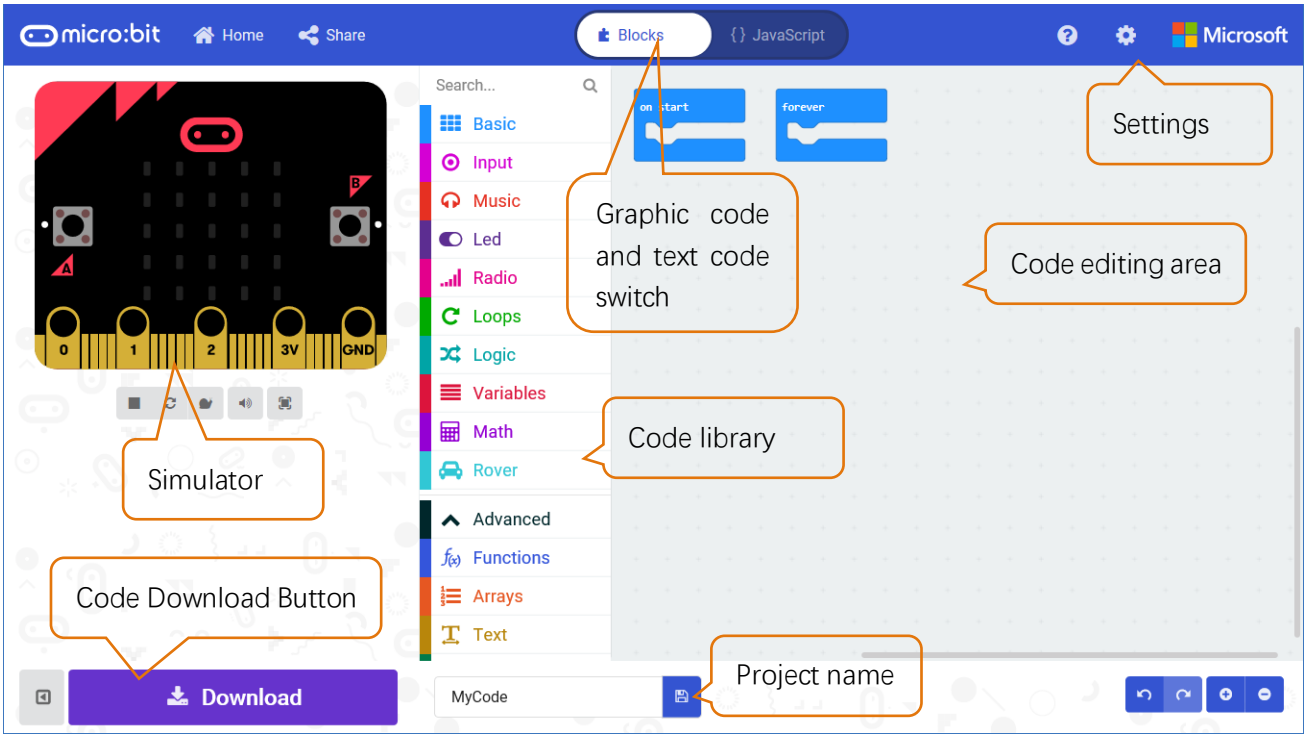
The MICROBIT drive will automatically eject and reconnect each time you program it, but your hex file will be gone. The micro:bit can only receive hex files and won't store anything else!

MakeCode

Open web version of makecode or windows 10 app version.
https://makecode.microbit.org/



Click "New Project", the interface of 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, while the code in "forever" block will be executed circularly.

Quick download

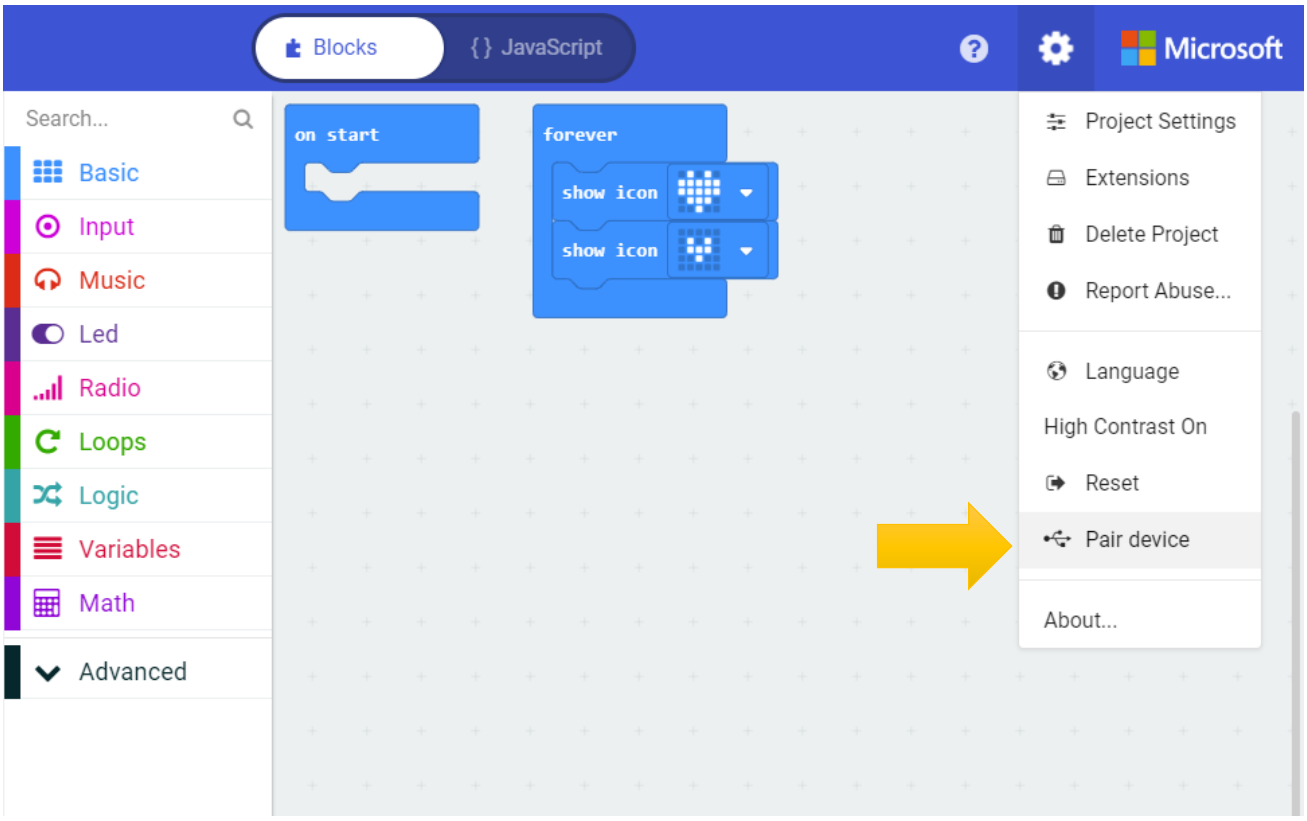
As mentioned earlier, if you use Windows 10 App of MakeCode, you can quickly download the code to micro:bit by clicking the download button.

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 download file quickly.

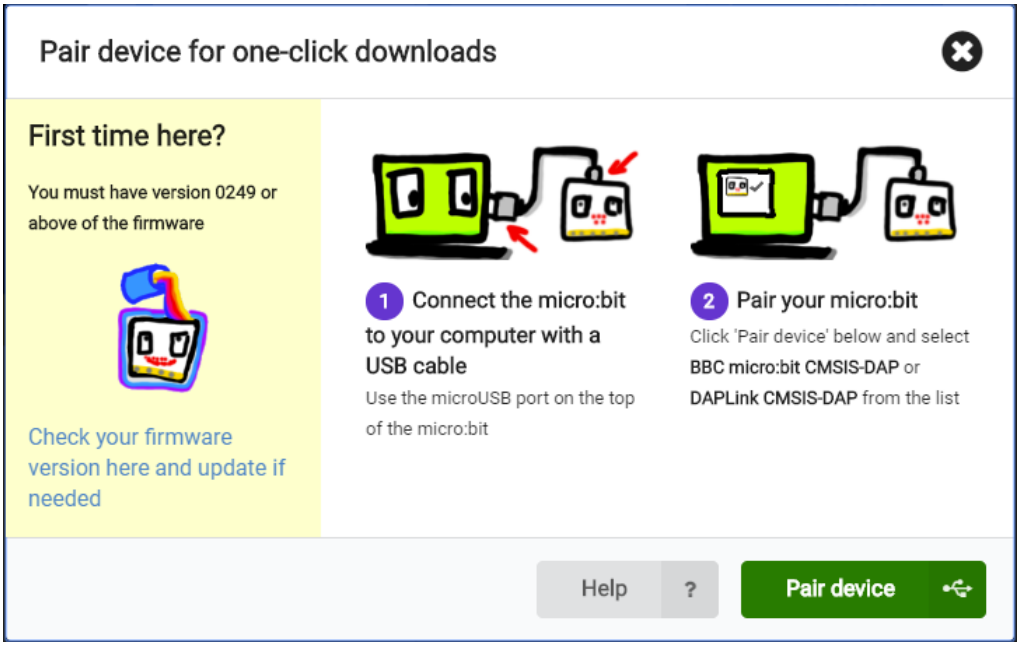
Here we use webUSB feature of Chrome, which allows web pages to access your USB hardware devices. We will complete the connection and pairing of the micro:bit device with the webpage in following steps.

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".



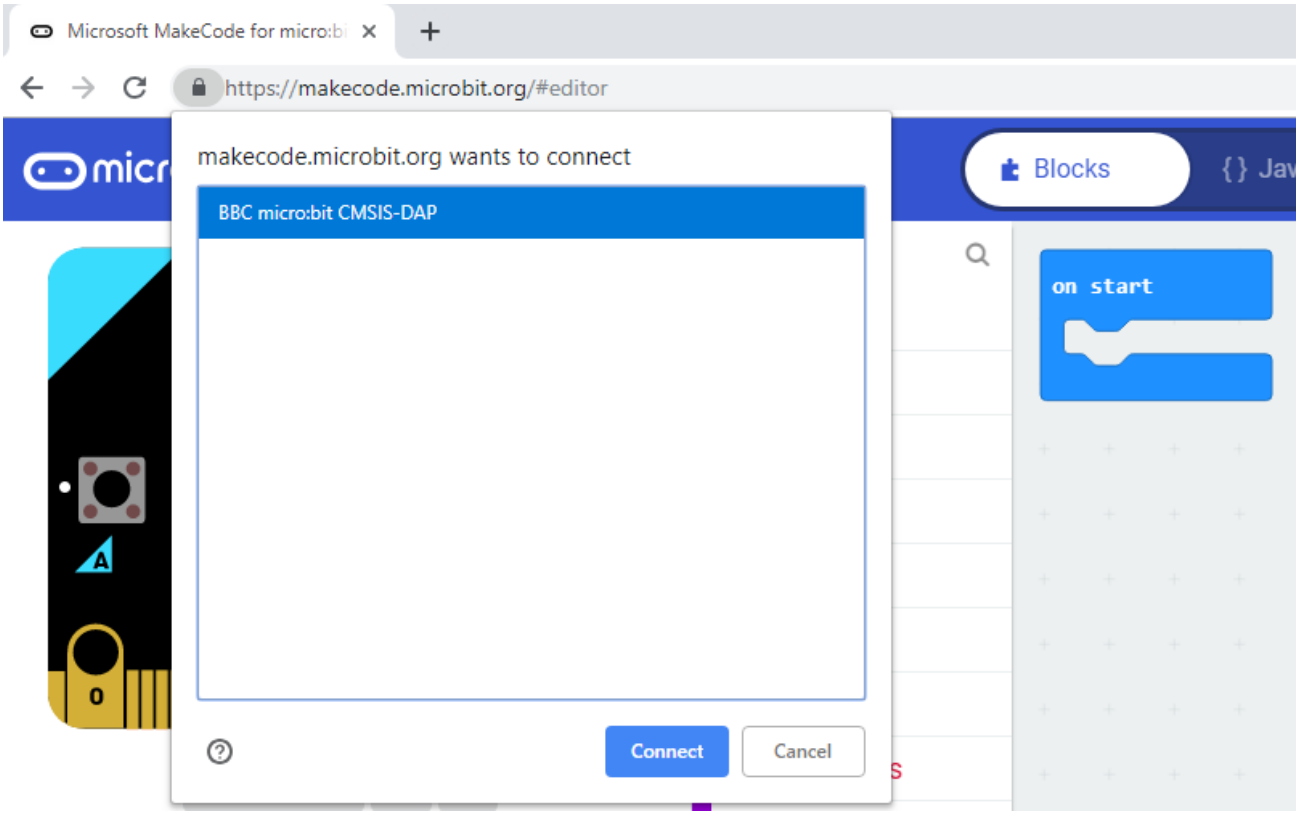
Then continue to click “Pair device” button.



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

We have saved 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 comes from: https://microbit.org/guide/firmware/



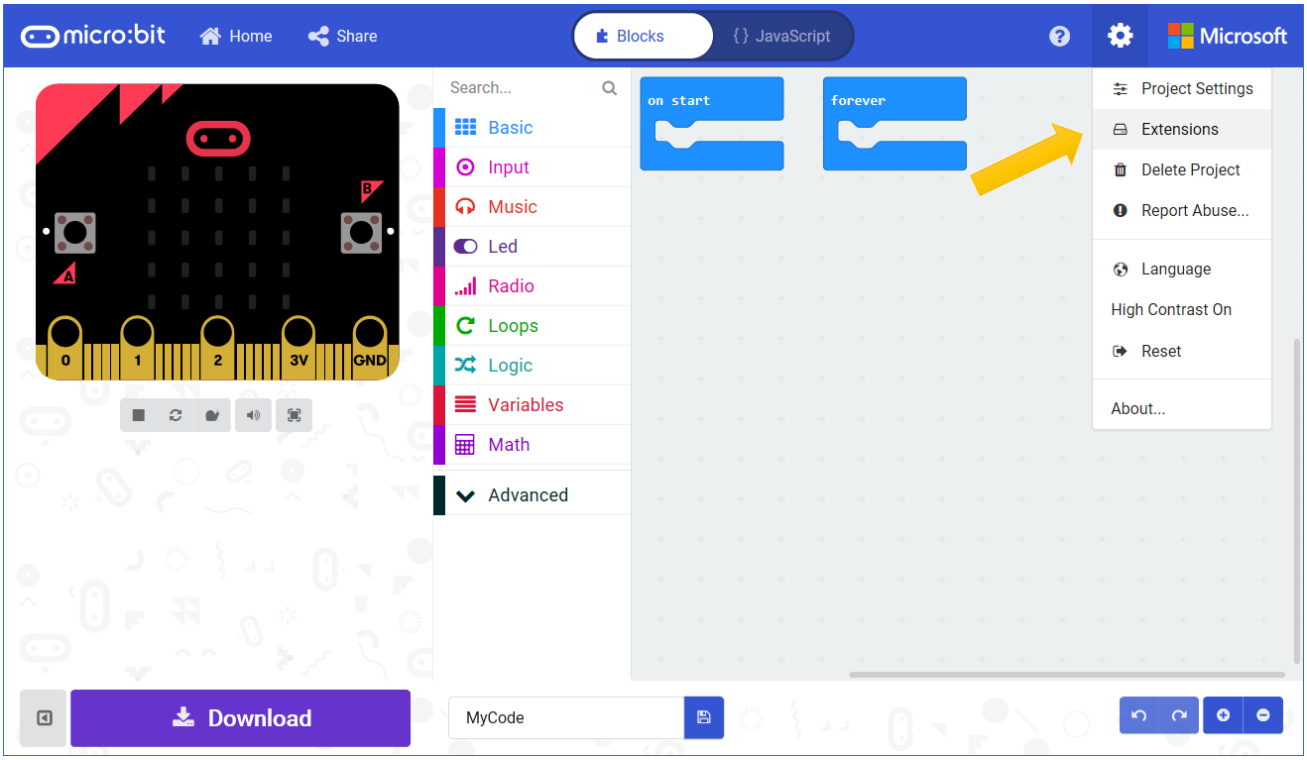
After the connection succeeds, click the Download button and the program will be downloaded directly to Micro: bit.

Extension for MakeCode

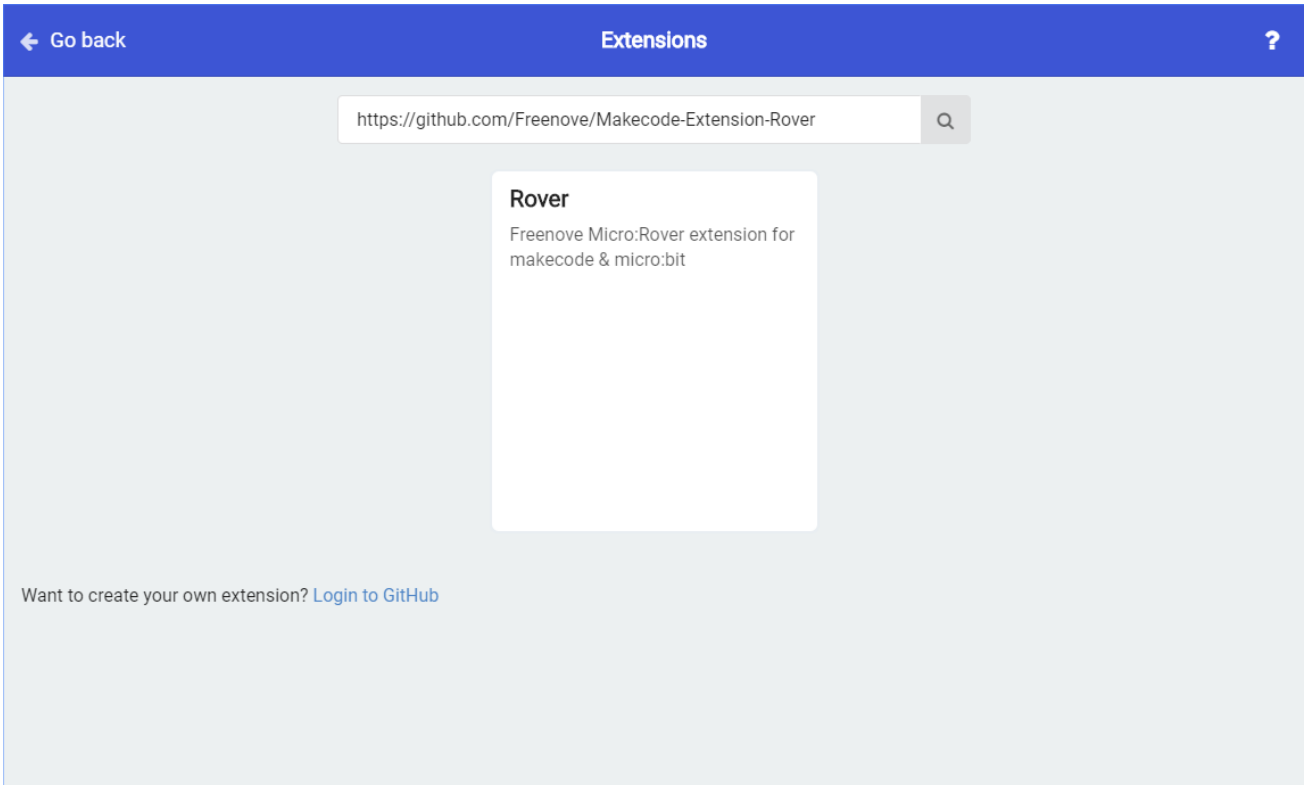
In order to use Rover more easily, we make a MakeCode Extension for Rover.

Add Rover Extension

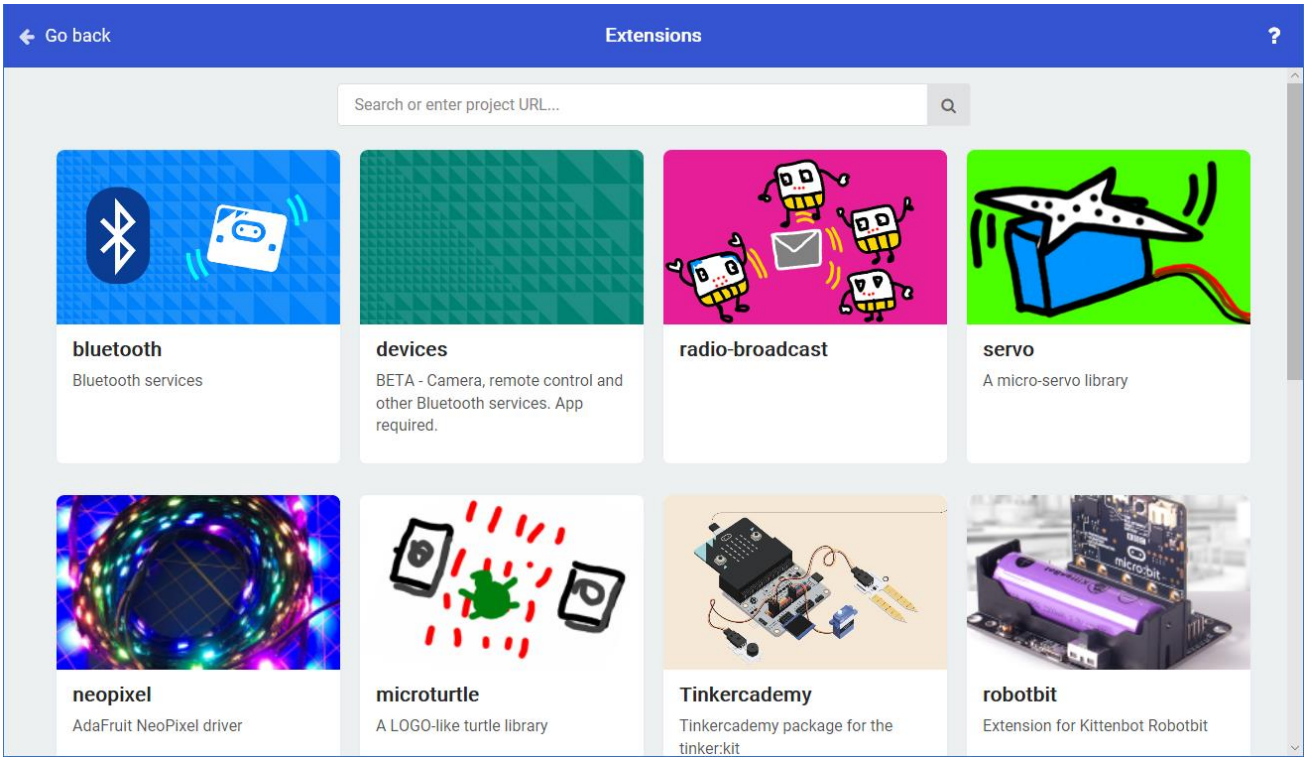
You can add Rover Extension with the following method.
Open MakeCode, click gear icon(settings) →Extensions.



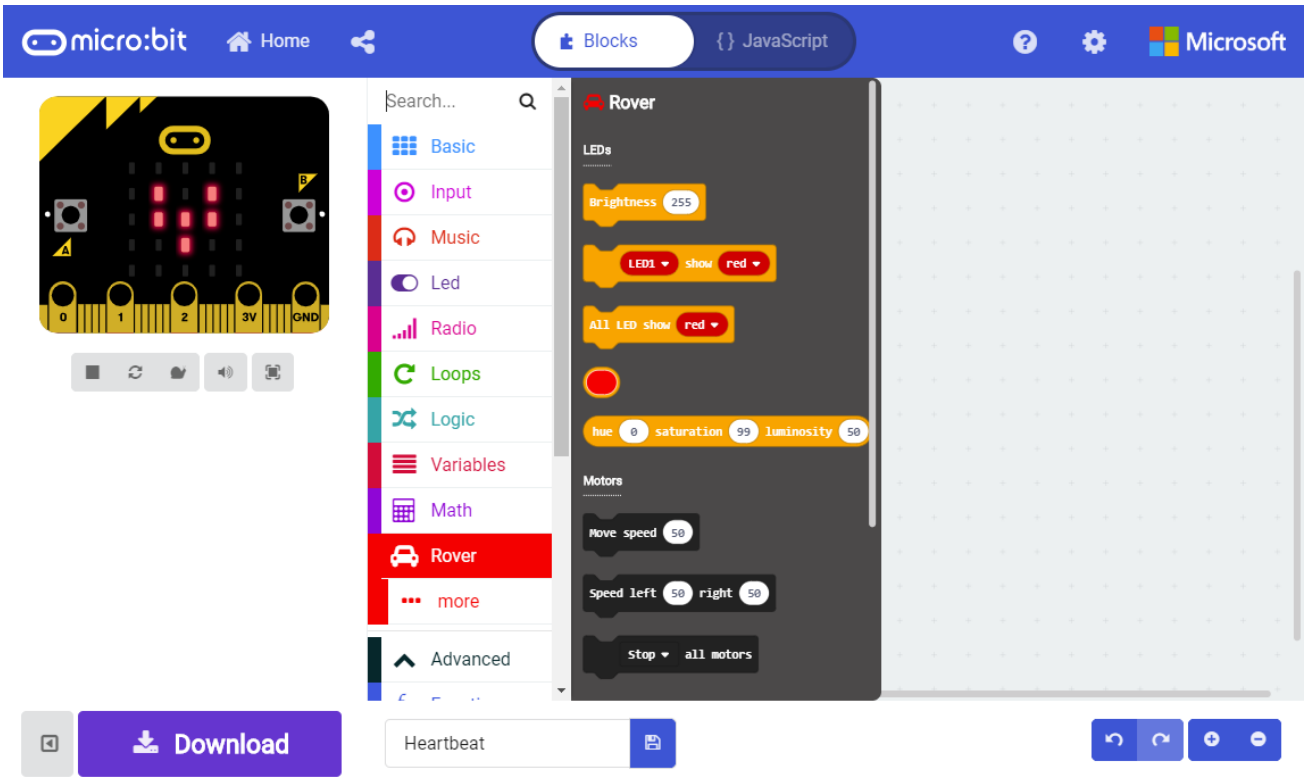
Enter this link in the search box:
https://github.com/Freenove/Makecode-Extension-Rover
Then click search.
Click the searching result Rover to download and install it. The process may take a few seconds.



Before clicking search, you can see some extended libraries on the page, including Bluetooth, Servo, neopixel, etc., which can be added by simply clicking on them.



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



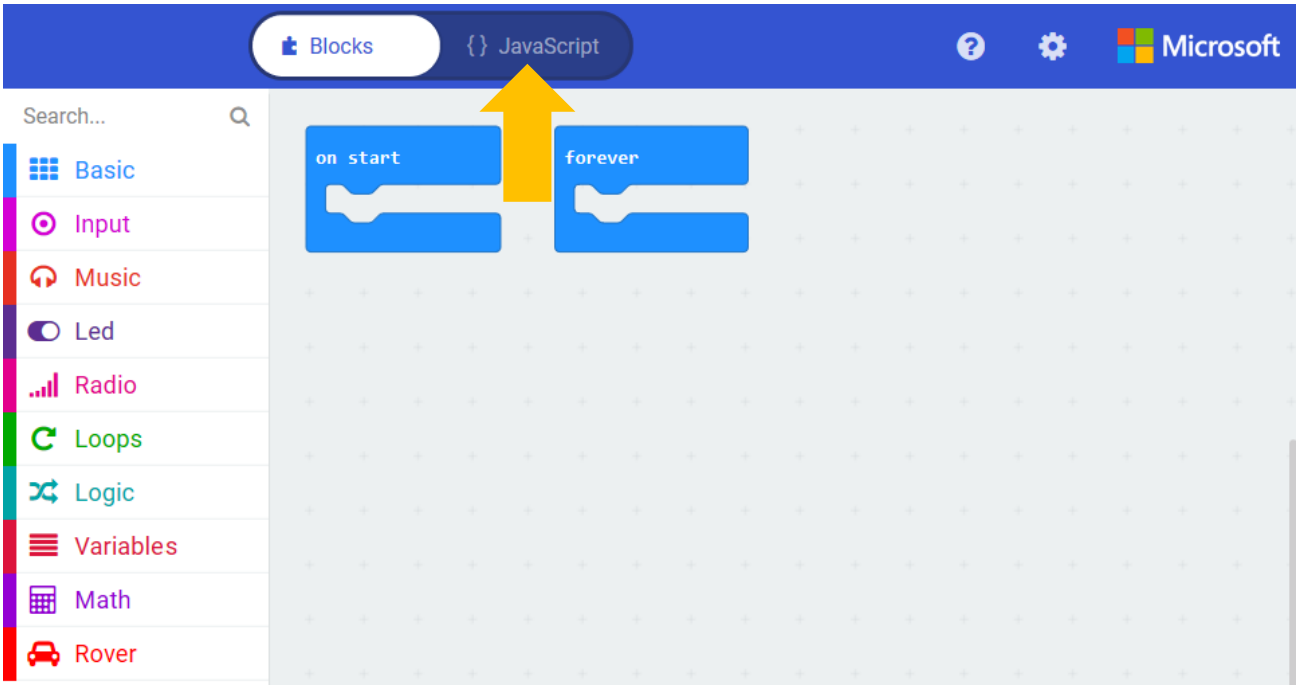
Note:

Extension library added to a project is only available for the particular project, and will not appear in other projects. That is to say, if you want to use this extension library in a new project, you need to add it again

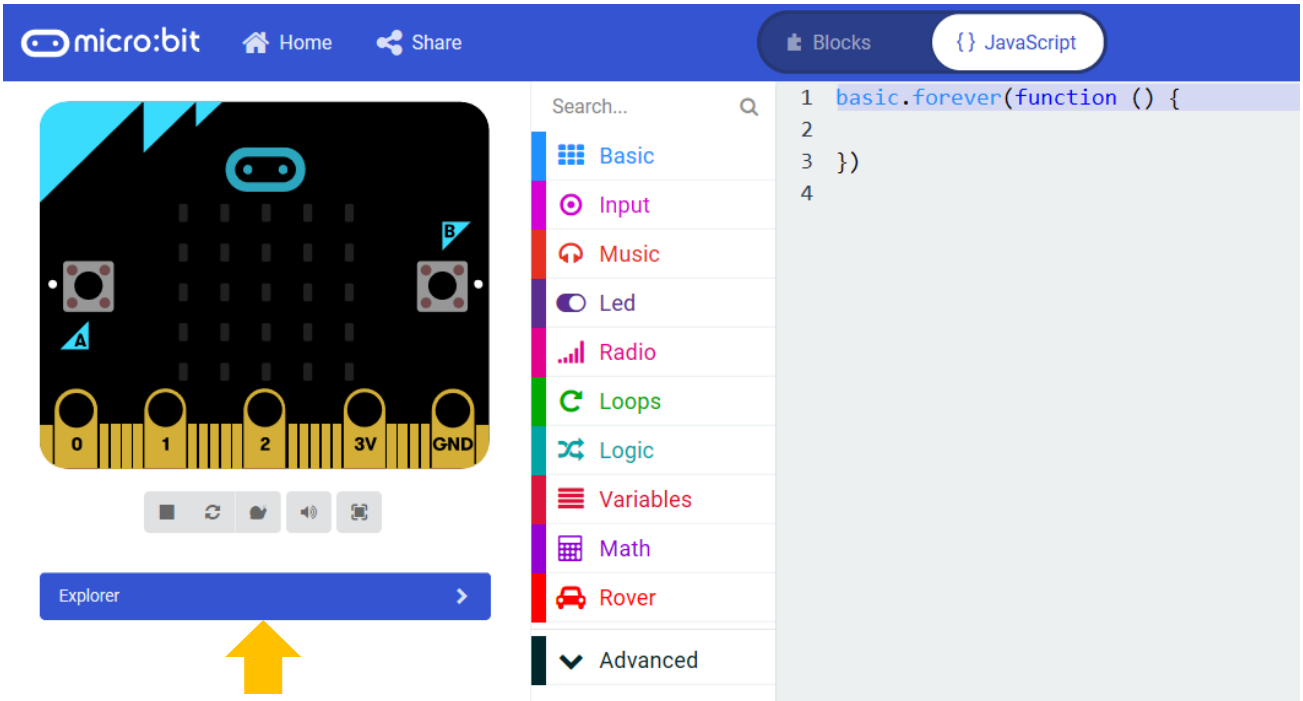
Update or Delete Rover Extension

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

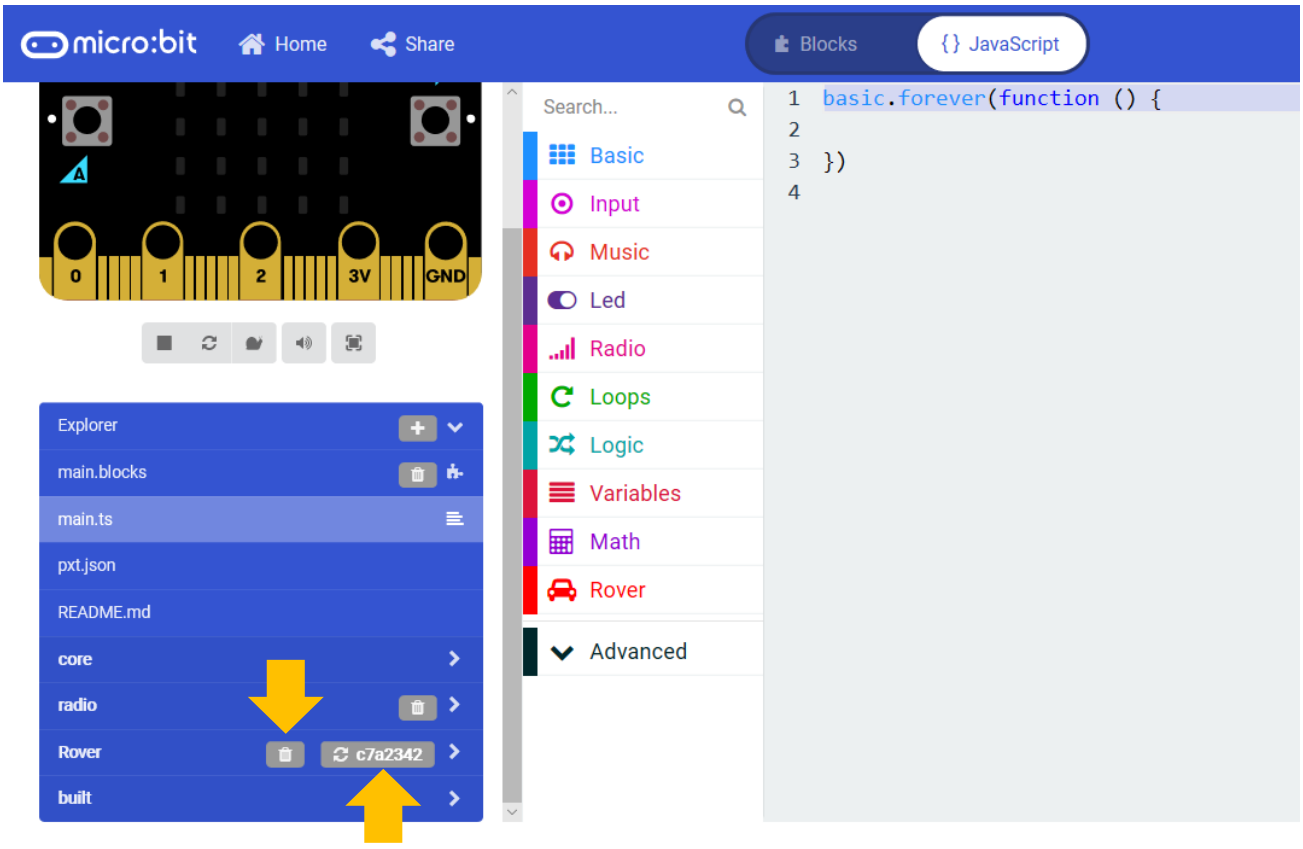
Click the "**JavaScript**" button to switch to text code.



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



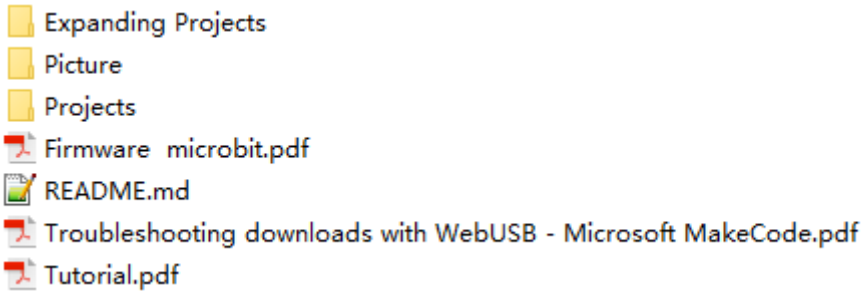
Find Rover in the extended list.
Click on the **trash can icon** to delete Rover extension.
Click on the **refresh icon** to update Rover extension.



Resource and code

Resource and code of this kit can be viewed and downloaded via the links 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 future 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

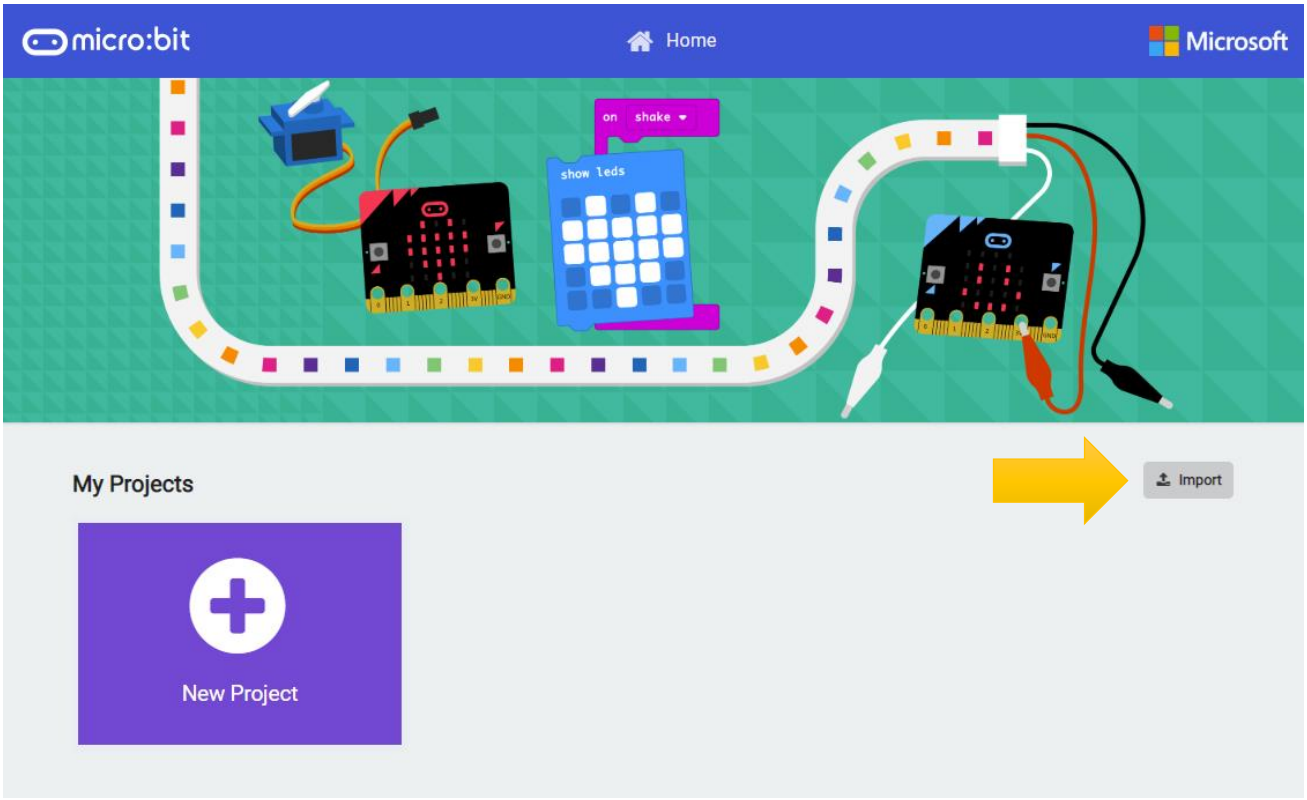
Import Code

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?)**

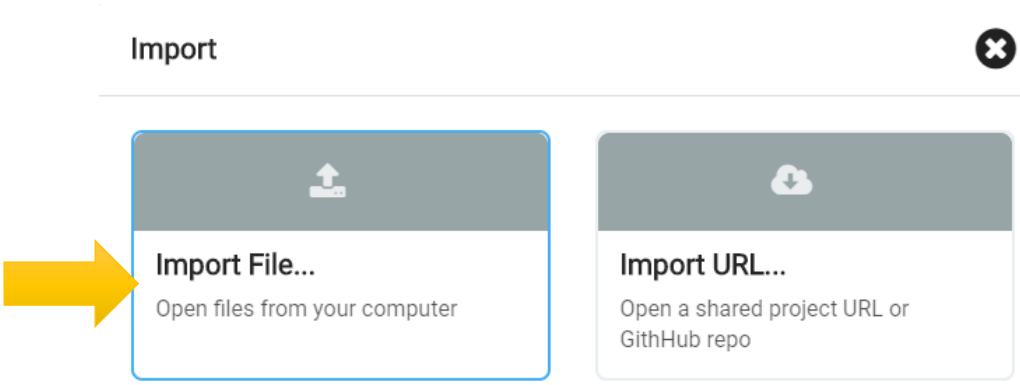
As for simple projects, it is recommended to complete the project by dragging code block.
As for complicated projects, it is recommended to complete the project by importing Hex code file.

Next, we will take "Heartbeat" project as an example to introduce how to load code.

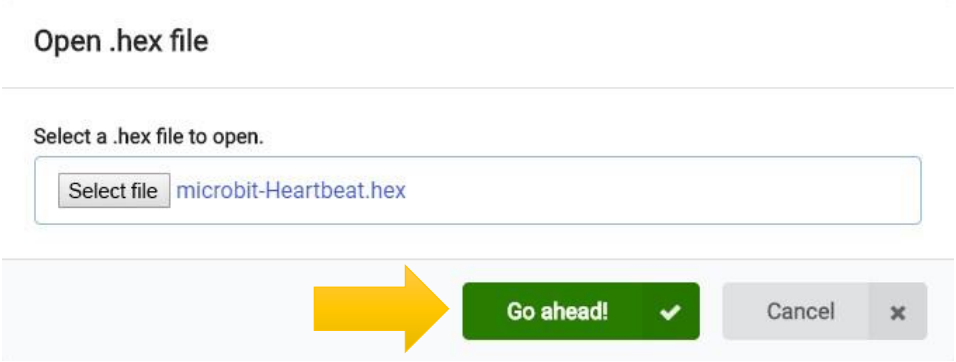
Open web version of makecode or windows 10 app version.
Click "Import" button on the right side of HOME page.



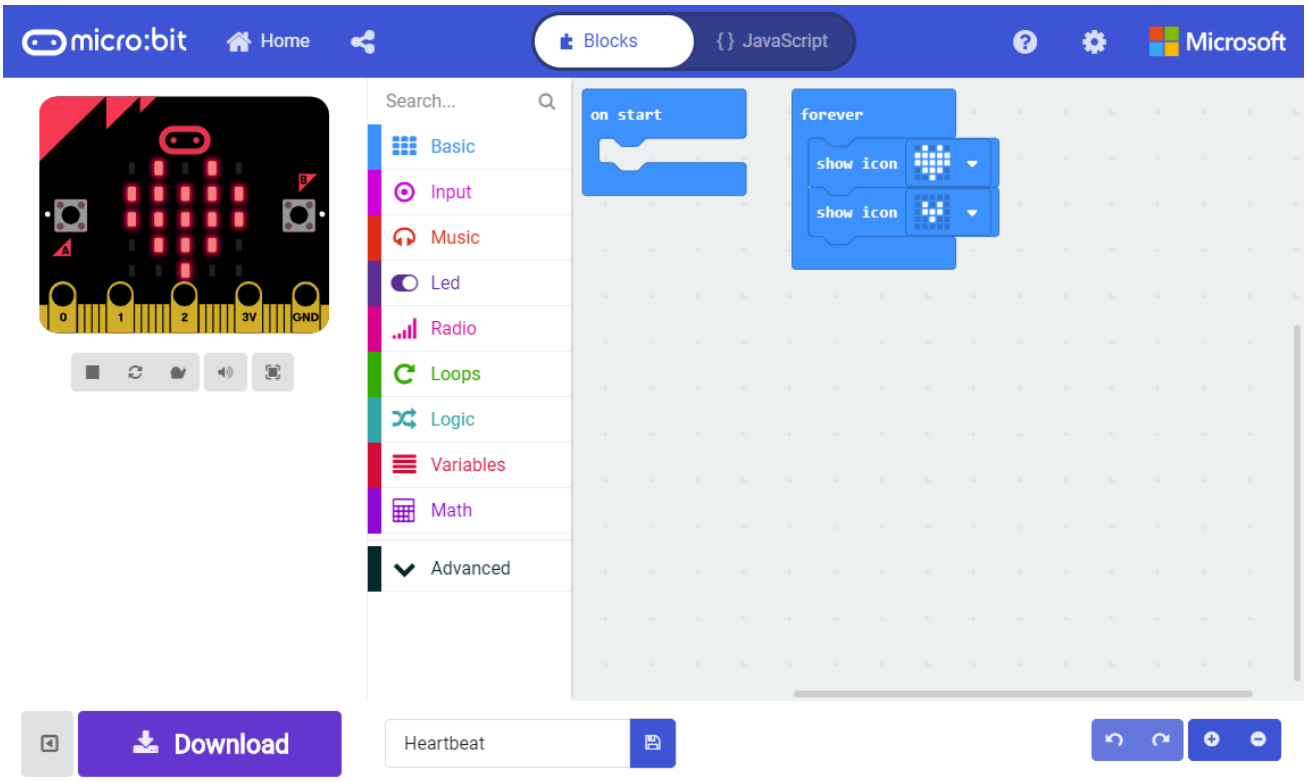
In the pop-up dialog box, click "Import File".



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



A few seconds later, the project is loaded successfully.



App

If you have any concerns, please feel free to contact us at **support@freenove.com**

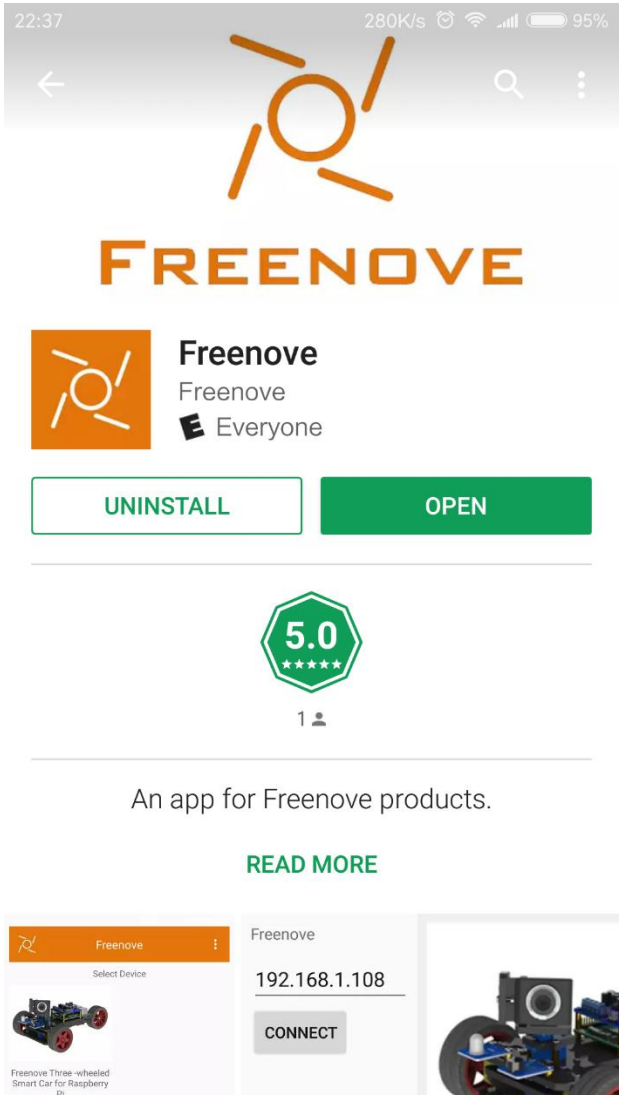
We design an Android app for Rover, which you will use in future projects. Please install it as follows.

Install Freenove Android app

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

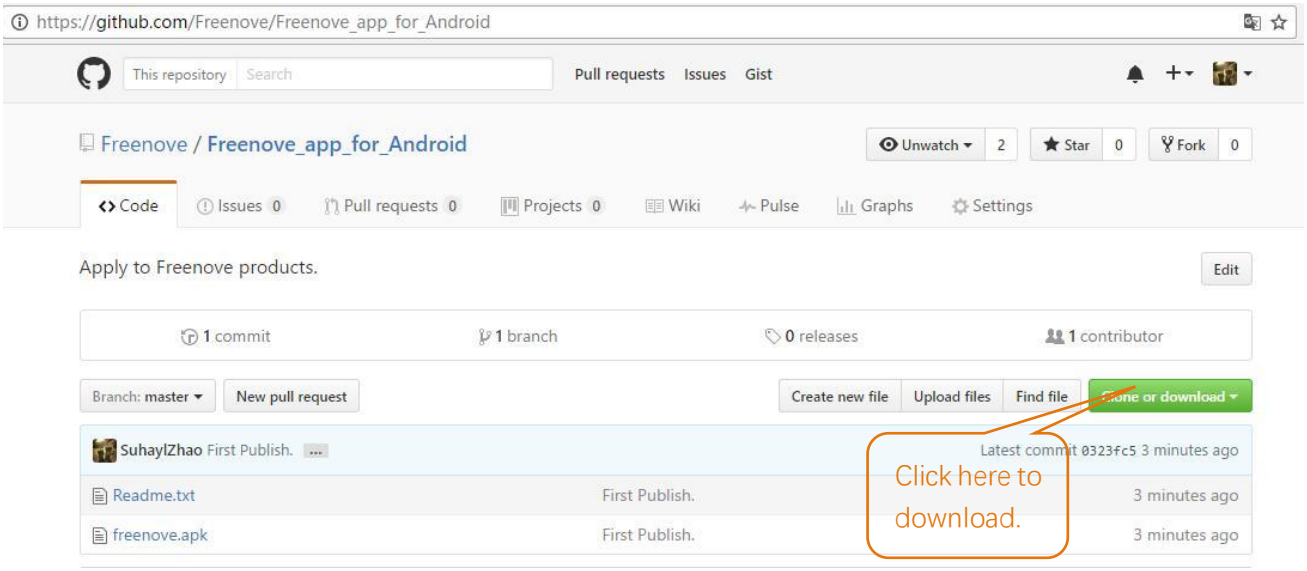
Method 1

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



Method 2

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



Install iPhone iOS app

Search **freenove** in App Store and install it.

If you want to control the car with app directly, please skip to chapter 7.

If you want to learn to control the car step by step, just follow chapter 1 and the following chapters.

Chapter 1 Music

The buzzer integrated on Rover can be used to play music. You can create and play music on Rover with blocks of Music library in MakeCode.
If you have any concerns, please feel free to contact us at **support@freenove.com**

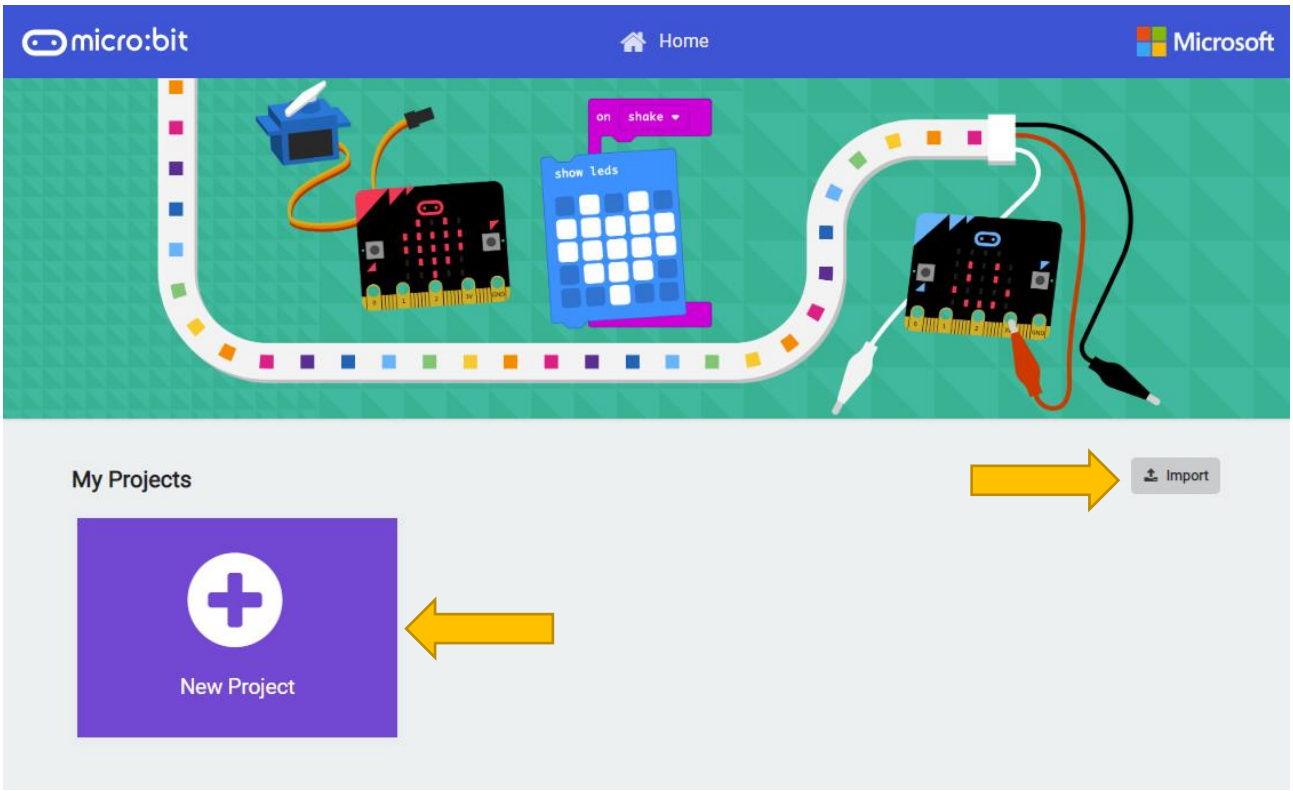
Preparation

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 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”.

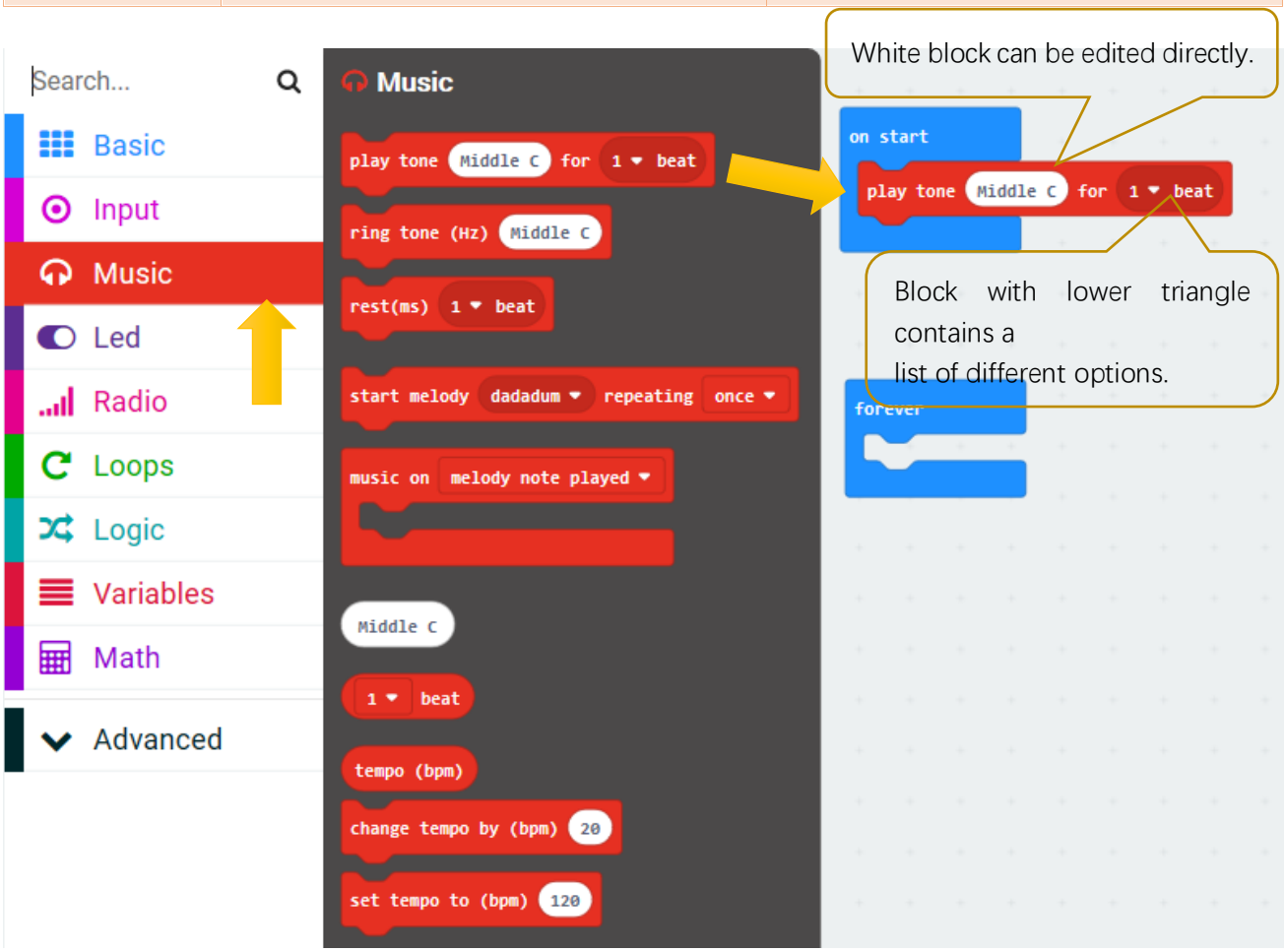


Play a note

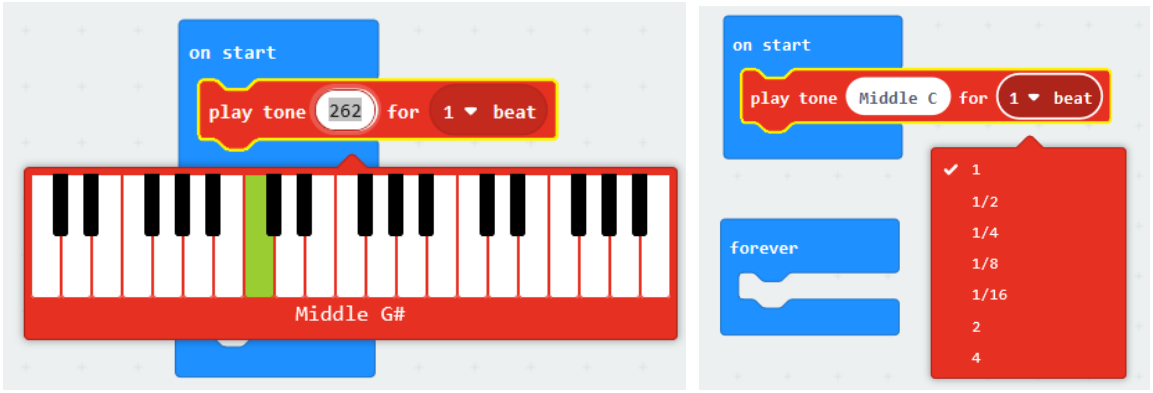
First, we play a note on the Rover.

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

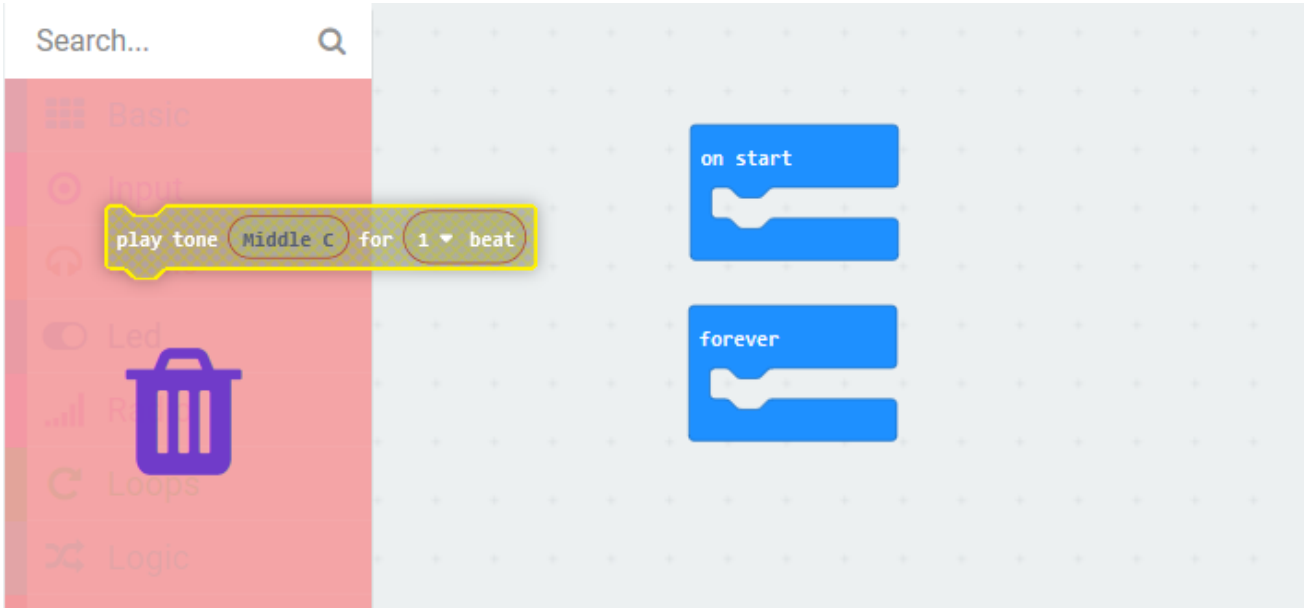
File type	Path	File name
Hex file	../Projects/01.1_Play-a-note	microbit-Play-a-note.hex



The meaning 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.

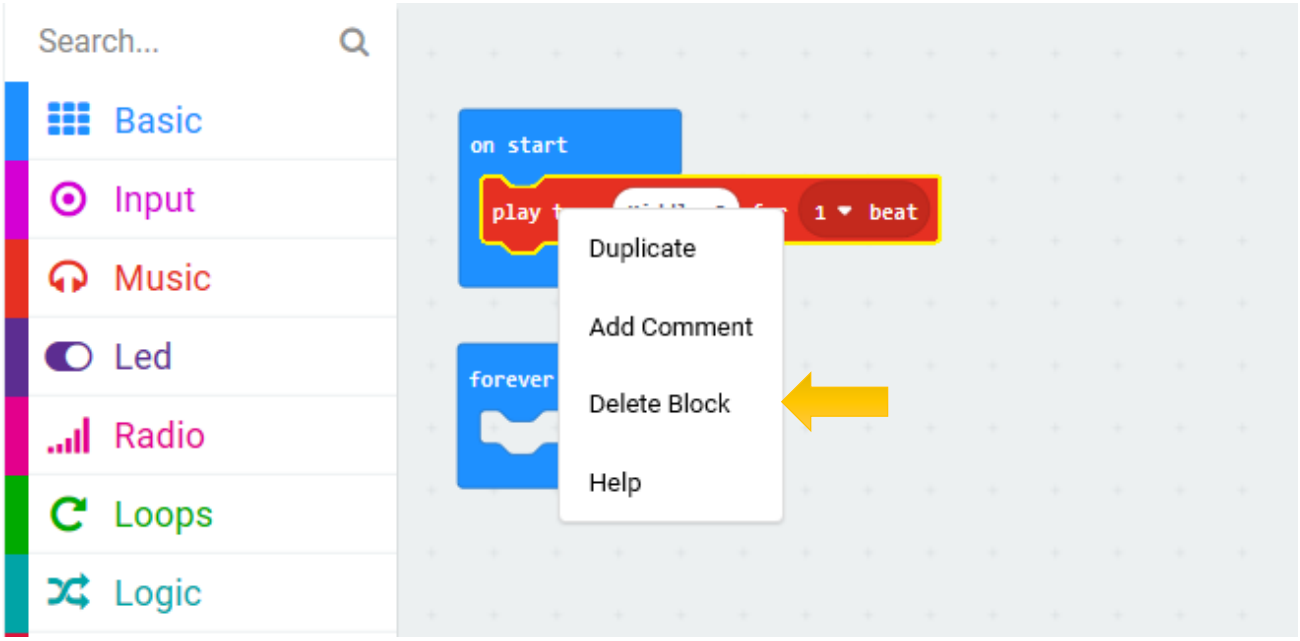


If you want to delete a code block, you can drag the block to the left extension area.



Or select the block in the code area and press "**Delete**" key.

Or right click on the block and select "Delete Block".



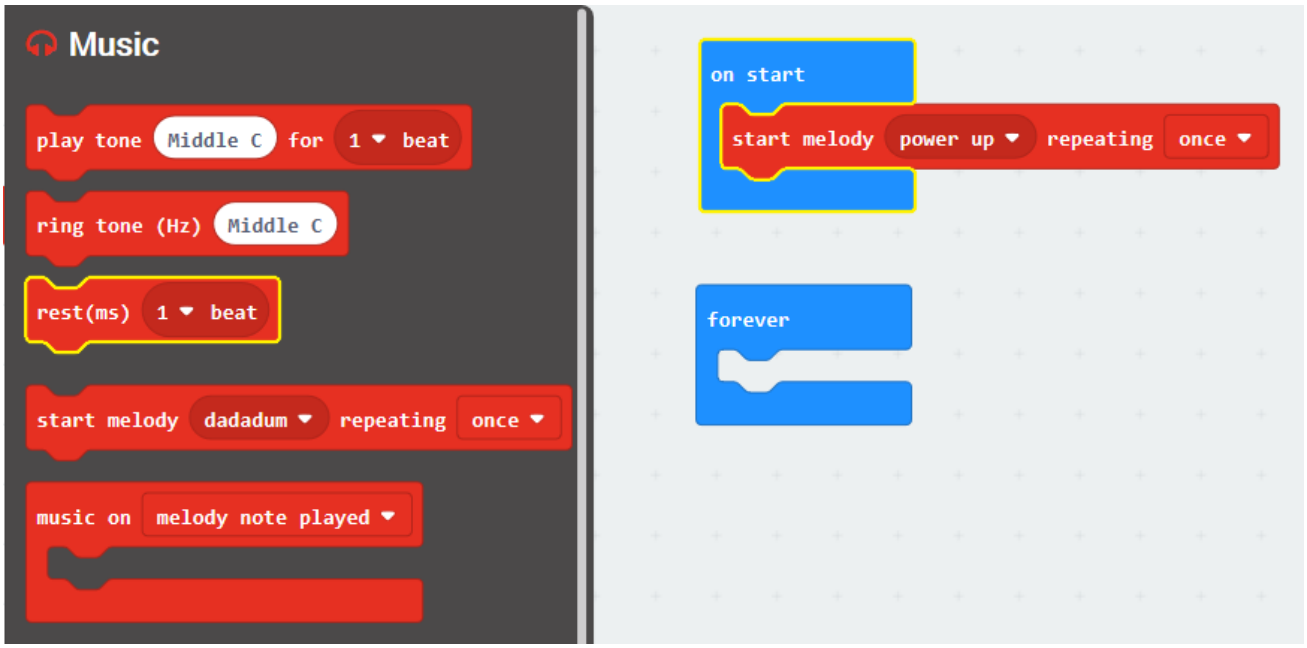
Download the code to micro:bitwith the method mentioned earlier, and Rover will play the notes and rhythms. (How to download? How to quick download?)

If there is a downloading 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 shown in the picture below:

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

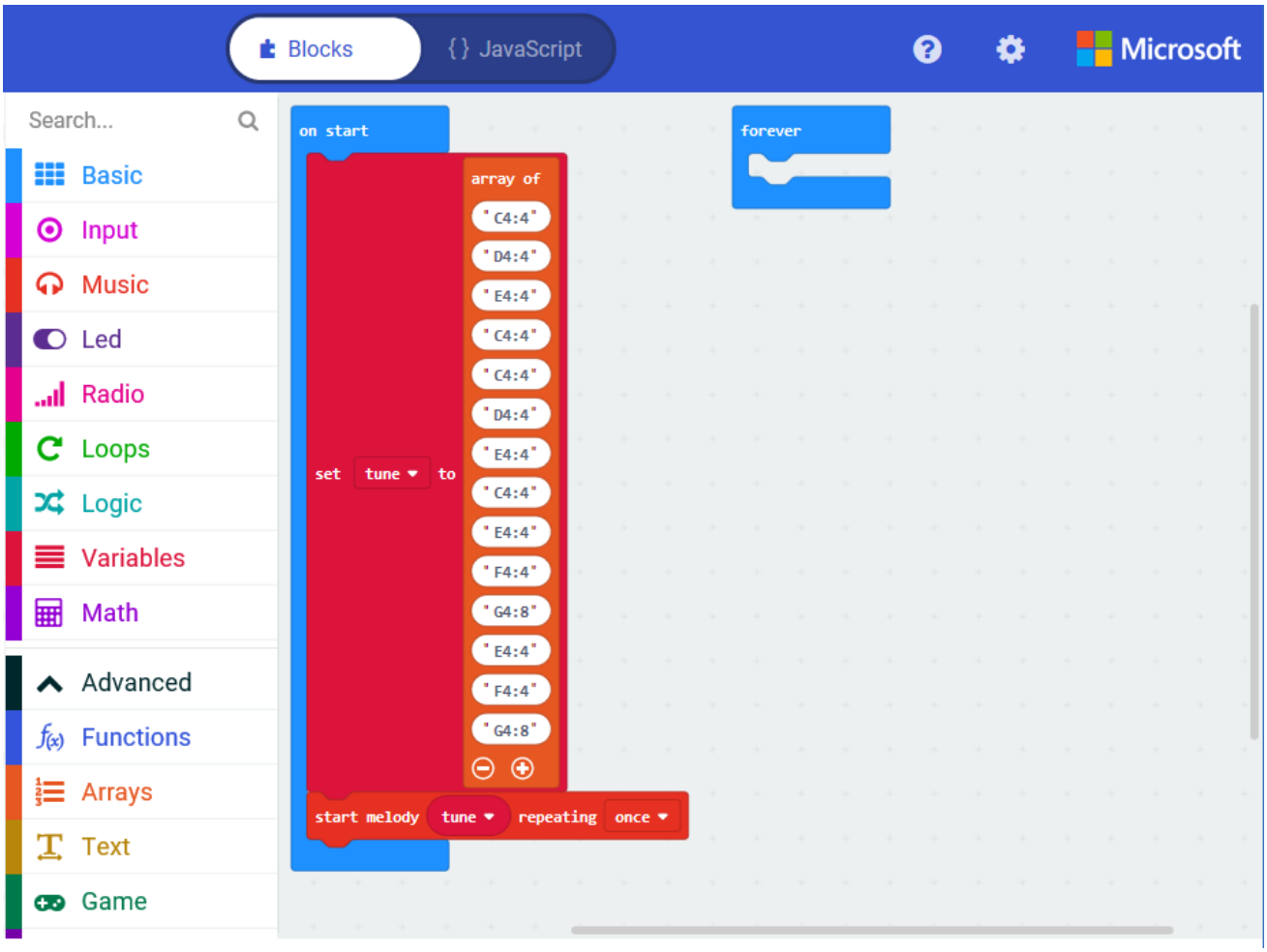


In this drop-down list of this code block, there are dozens of melodies to choose. You can choose any one. Then download the code to micro:bit, and Rover will play melody of the code.

Play custom melody

Load code according to the table below or drag the code block as shown in the 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, and Rover will play melody of the code.

Chapter 2 RGB LED

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

If you have any concerns, please feel free to contact us at **support@freenove.com**

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 a piece of white paper, which will make the color displayed more clearly.

Open web version of MakeCode or windows 10 app version.

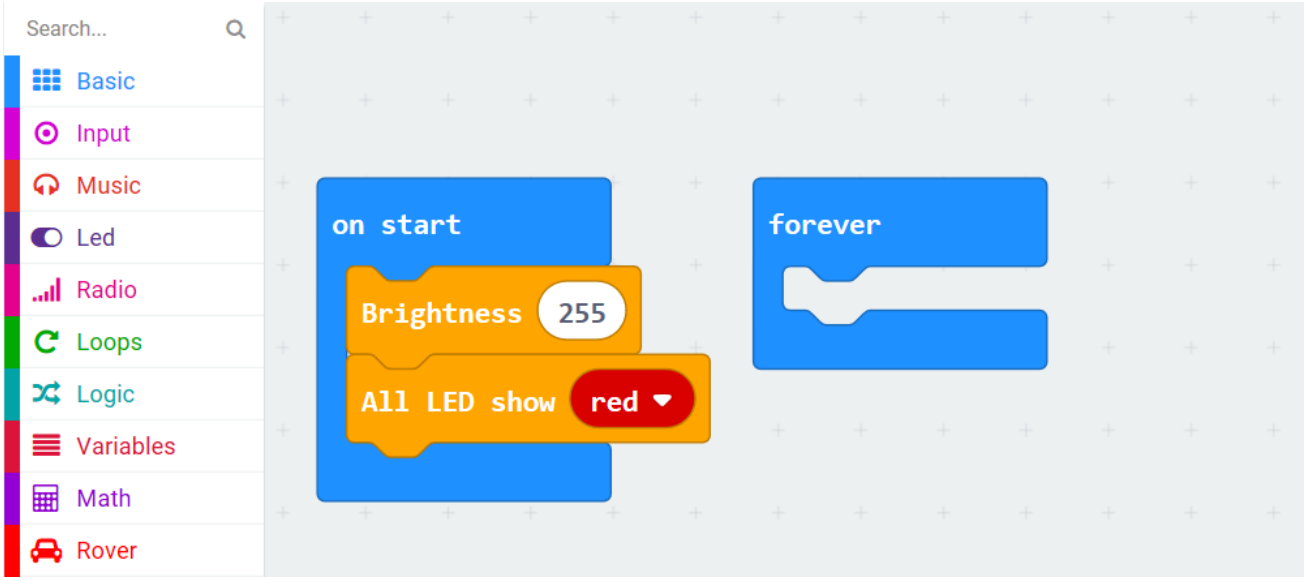
If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.
(How to import?)

If you choose to drag code manually, you first need to add Rover extensions.
(How to add Rover extension?)

Emitting one color of light

First let all RGB LEDs of Rover emit same color and same brightness.
Load code according to the table below or drag the code block as shown in the picture below:

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



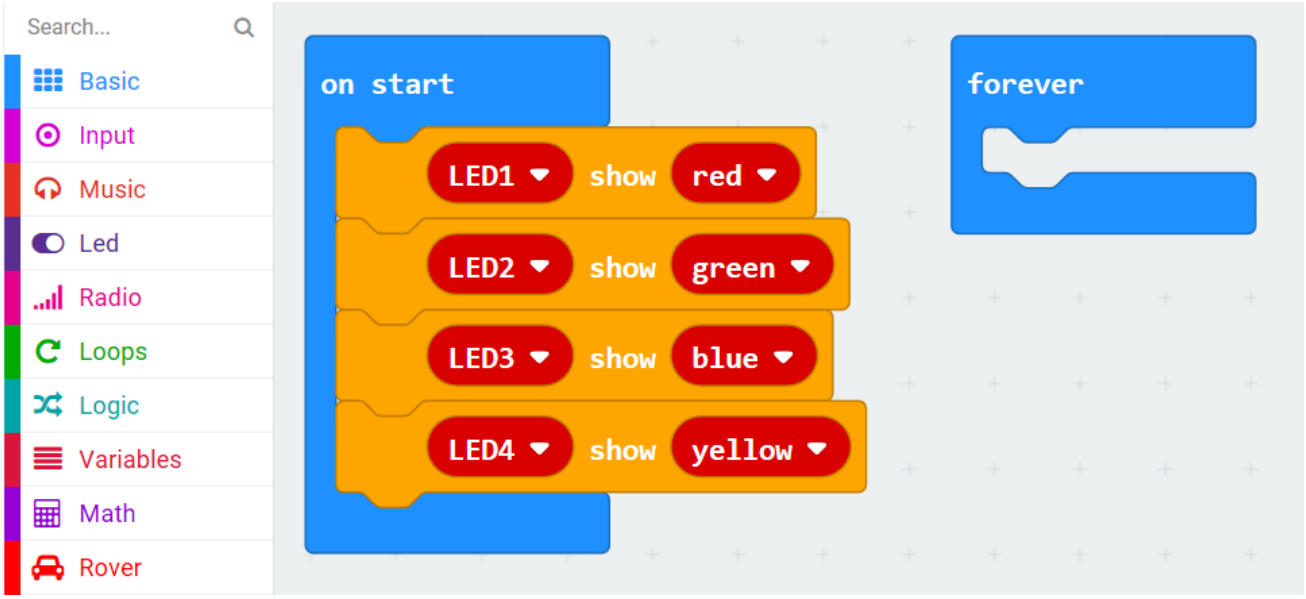
In this code, set brightness of all RGBLED to 255 and set the color for all to red.
You can modify brightness in the range of 0~255 and choose different color.

Download the code to micro:bit. All RGBLEDs on Rover will emit red light.

Emitting different colors of light

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

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

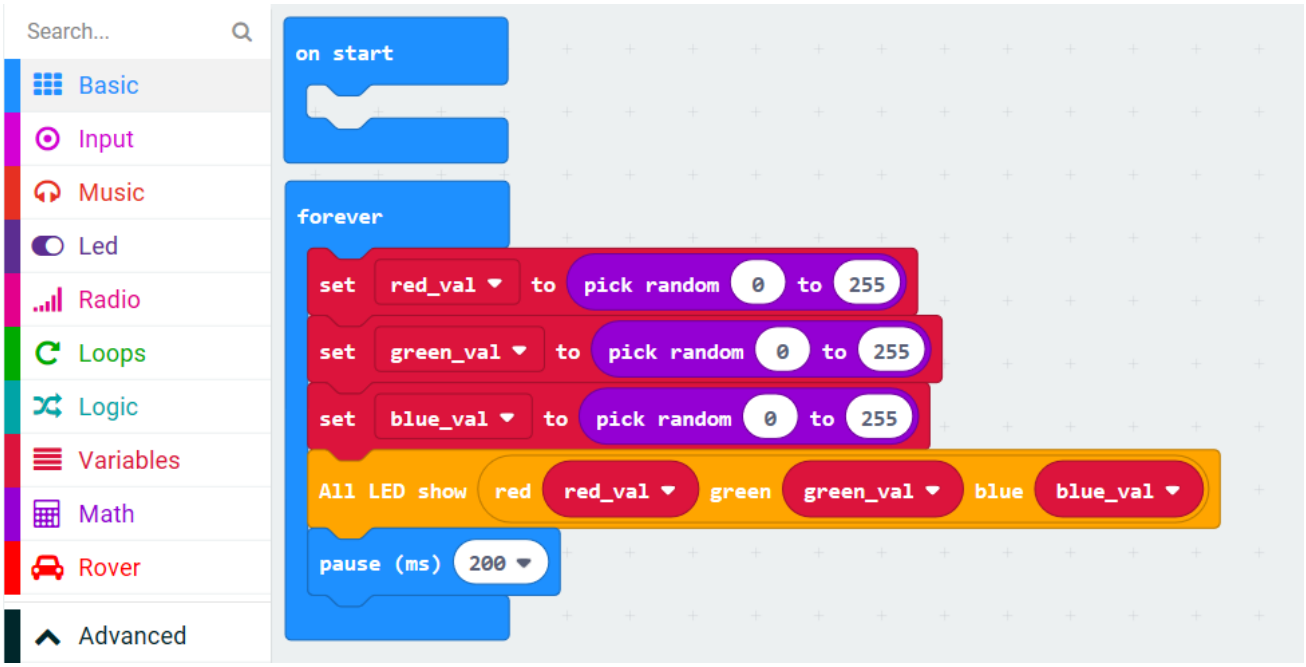


In this code, set color of RGBLED1,2,3,4 to red, green, blue, and yellow respectively.
Download the code to micro:bit. All RGBLEDs on Rover will emit red light.

Emitting random color of light

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

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



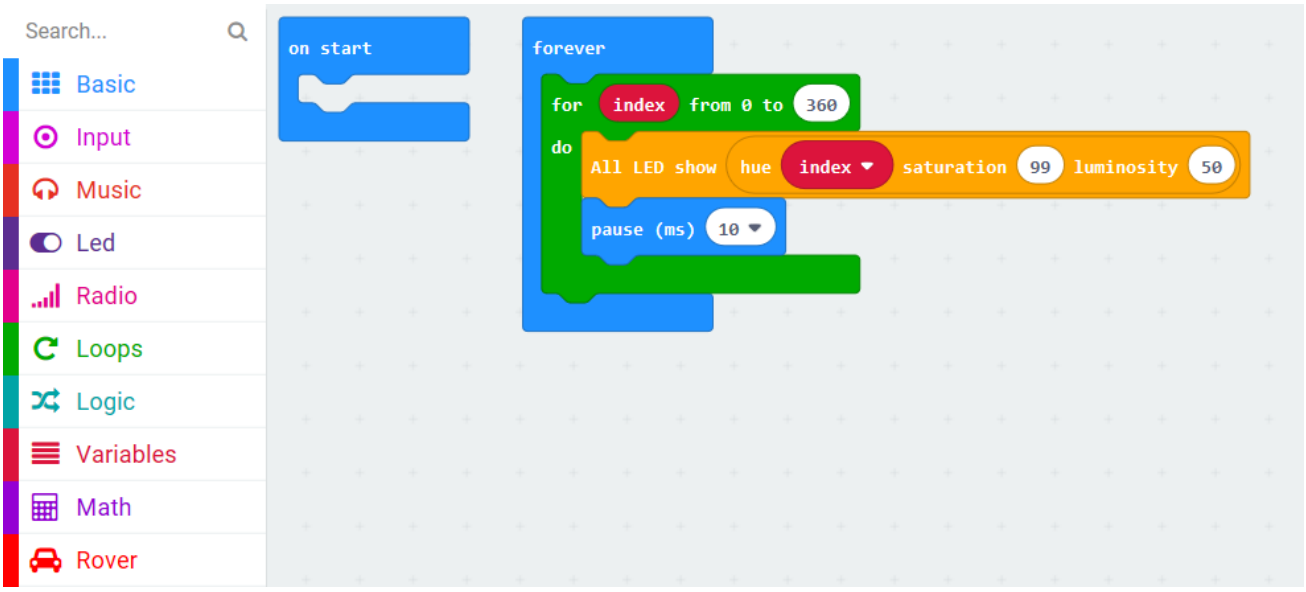
In this code, three variables(red_val, green_val and blue_val) are defined. Three Random values between 0 and 255 are assigned to them respectively. These three values are used as parameters of RGB LED. Then all RGBLEDs will emit random color.

Download the code to micro:bit, and then observe the coloremitted by RGBLED on Rover.

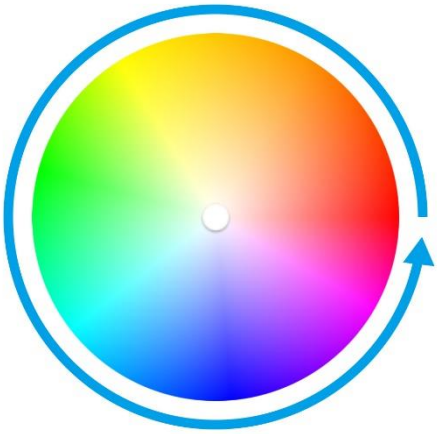
Emitting soft colors

Load code according to the table below or drag the code block as shown in the 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 and then observe the color emmited by RGBLED on Rover.

Chapter 3 Ultrasonic Ranging

The ultrasonic module on Rover can measure the distance between Rover and the obstacle in front of it. Therefore, we can take advantage of this characteristic to make Rover avoid obstacle. When Rover is close to the obstacle in the front, let it turn to avoid the obstacle.

If you have any concerns, please feel free to contact us at **support@freenove.com**

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.

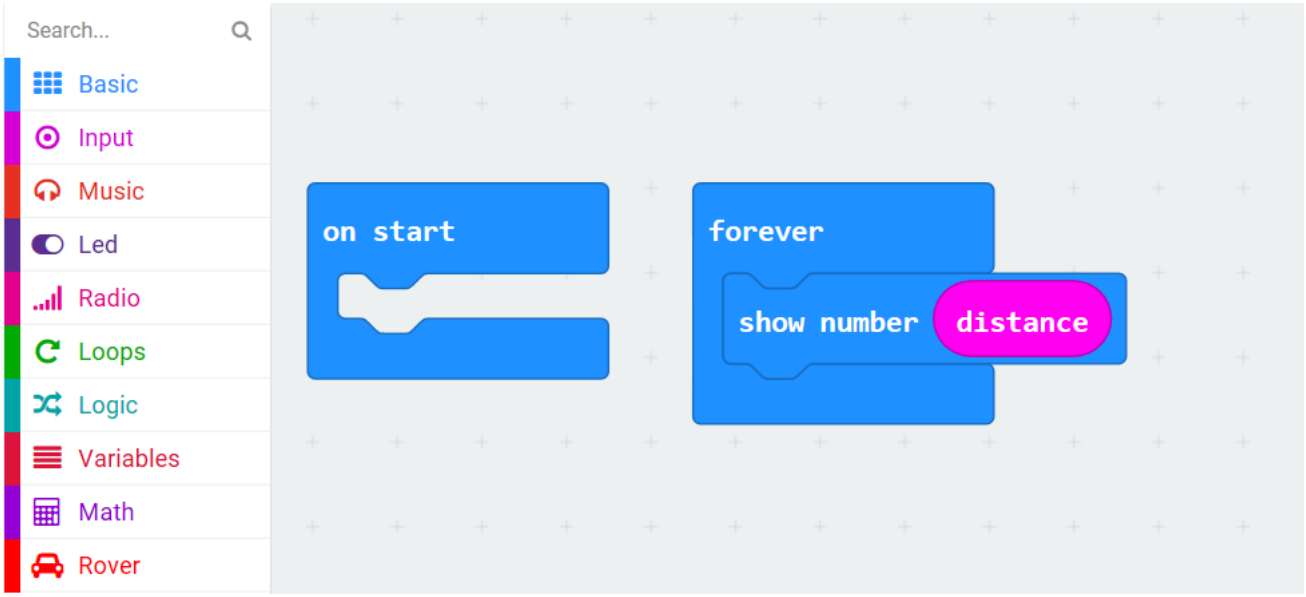
If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.
(How to import?)

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 as shown in the 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 distance value detected by ultrasonic ranging.

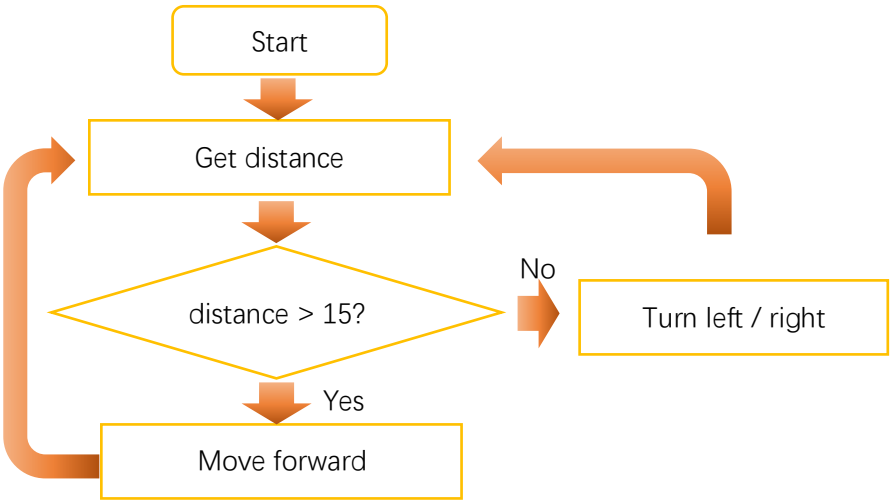
Place an obstacle in front of ultrasonic ranging module. Download the code to micro:bit, and then change the distance between the two objects, and observe the value of showed on LED matrix of micro:bit.

Rover-Obstacle avoidance mode -1

In this project, we will realize the obstacle avoidance mode of Rover.

Flow chart

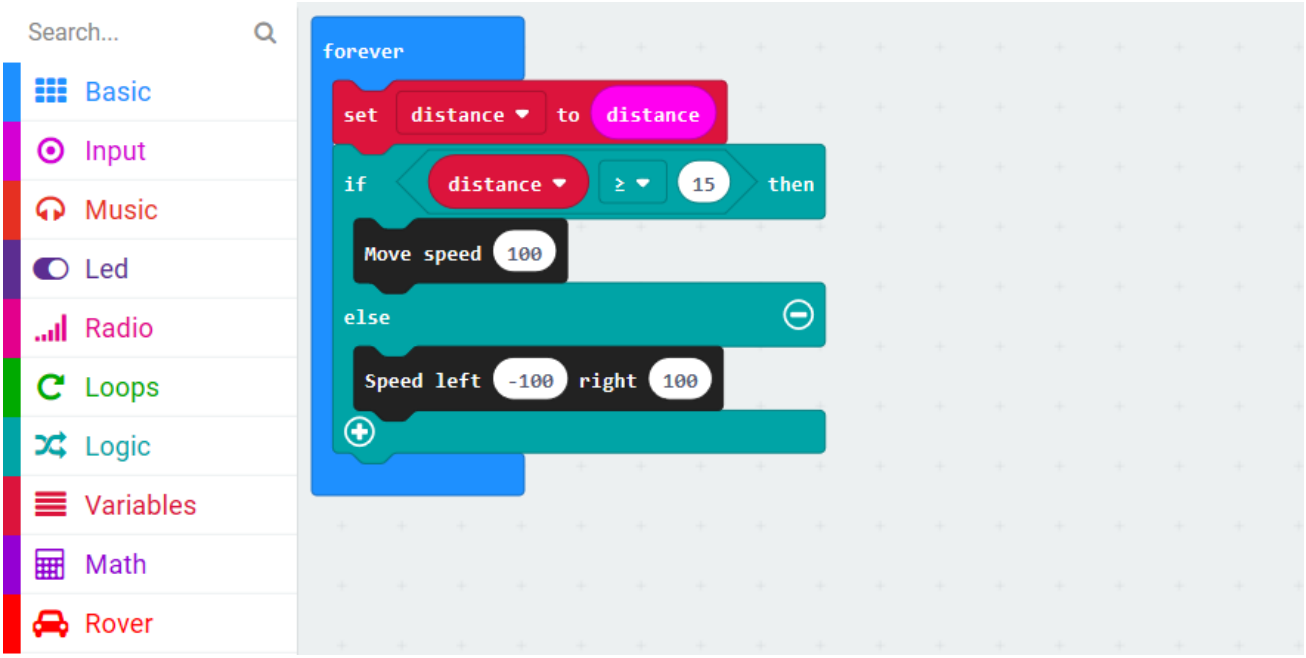
The program code is written according to flow chart, as shown below.

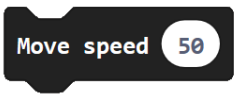


Code

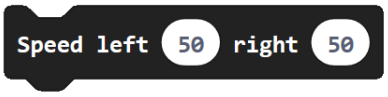
Load code according to the table below or drag the code block as shown in the picture below:

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 moving forward, and negative value means moving 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, and then observe the motion of Rover.

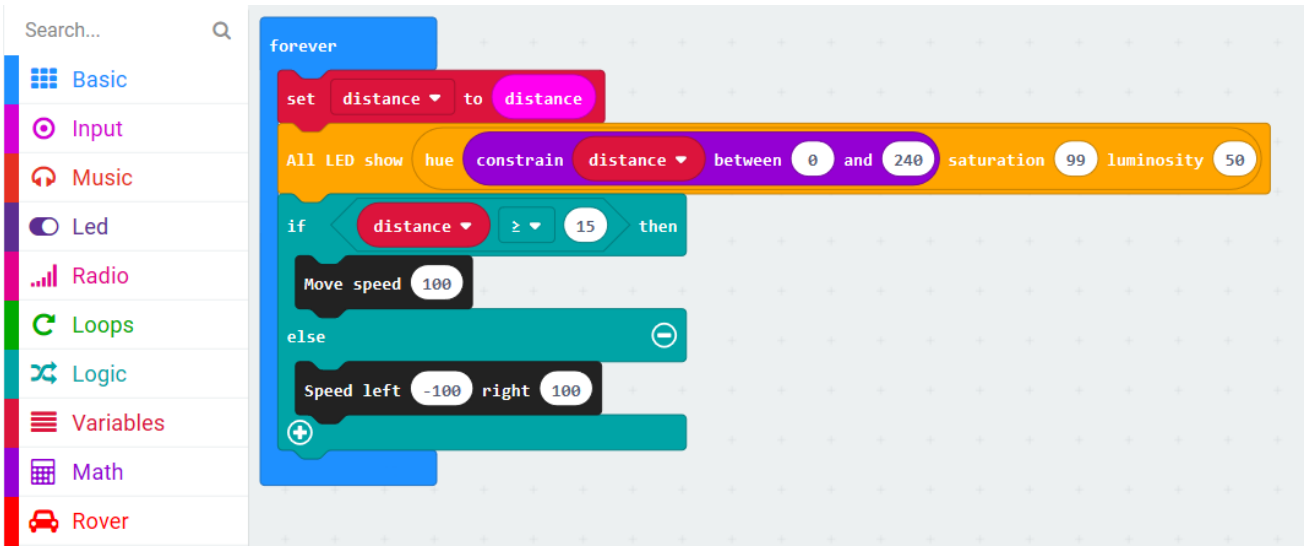
Rover-Obstacle avoidance mode -2

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 shown in the picture below:

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



In this code, the logic 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, and then observe motion of Rover and the display of RGBLED and LED matrix.

Chapter 4 Light tracing

There are two light intensity sensors on Rover, so we can learn which sensor receives more intensive light based on the difference between the values of the two sensors. In this way, Rover can achieve the function of tracking or avoiding light.
If you have any concerns, please feel free to contact us at **support@freenove.com**

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.

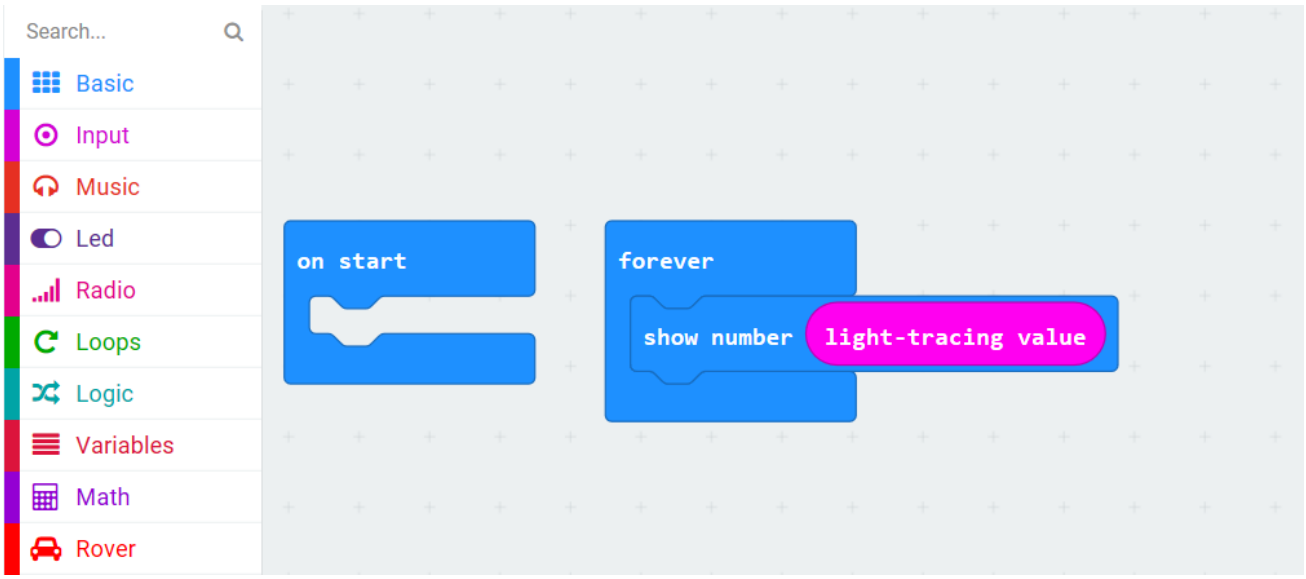
If you choose to load the project by importing Hex file, there is no need to add the Rover extension manually.
(How to import?)

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 as shown in the 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) of the light intensity sensor, which is jointly generated by the two sensors. Its range is 0~1023.

Theoretically,
If β= 512, it indicates that light intensities sensed by two sensors are equal.

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

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

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

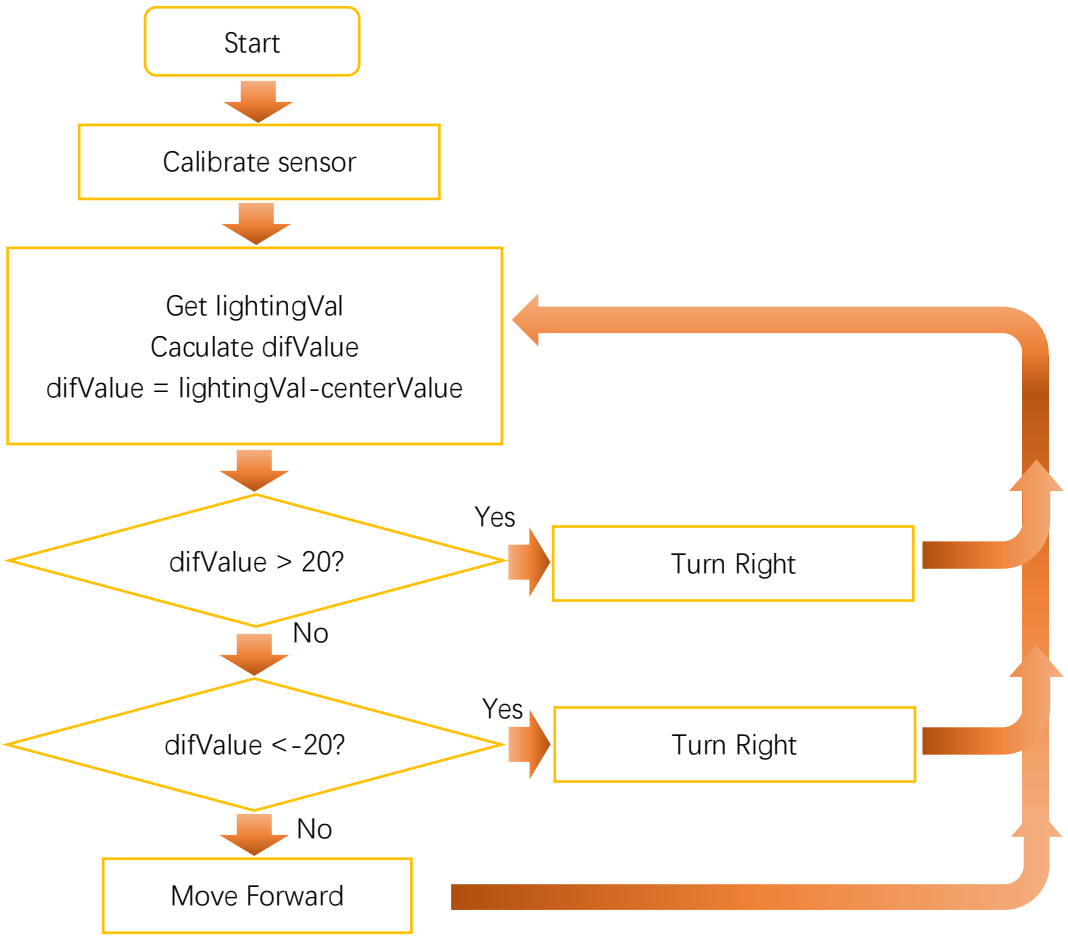
Download the code to Micro:bit. And use a light source to illuminate one light intensity sensor, or use your finger to cover it, and then observe the values showed on micro:bit LED matrix.

Rover-light tracing mode

In this project, we will realize the light tracing mode of Rover.

Flow chart

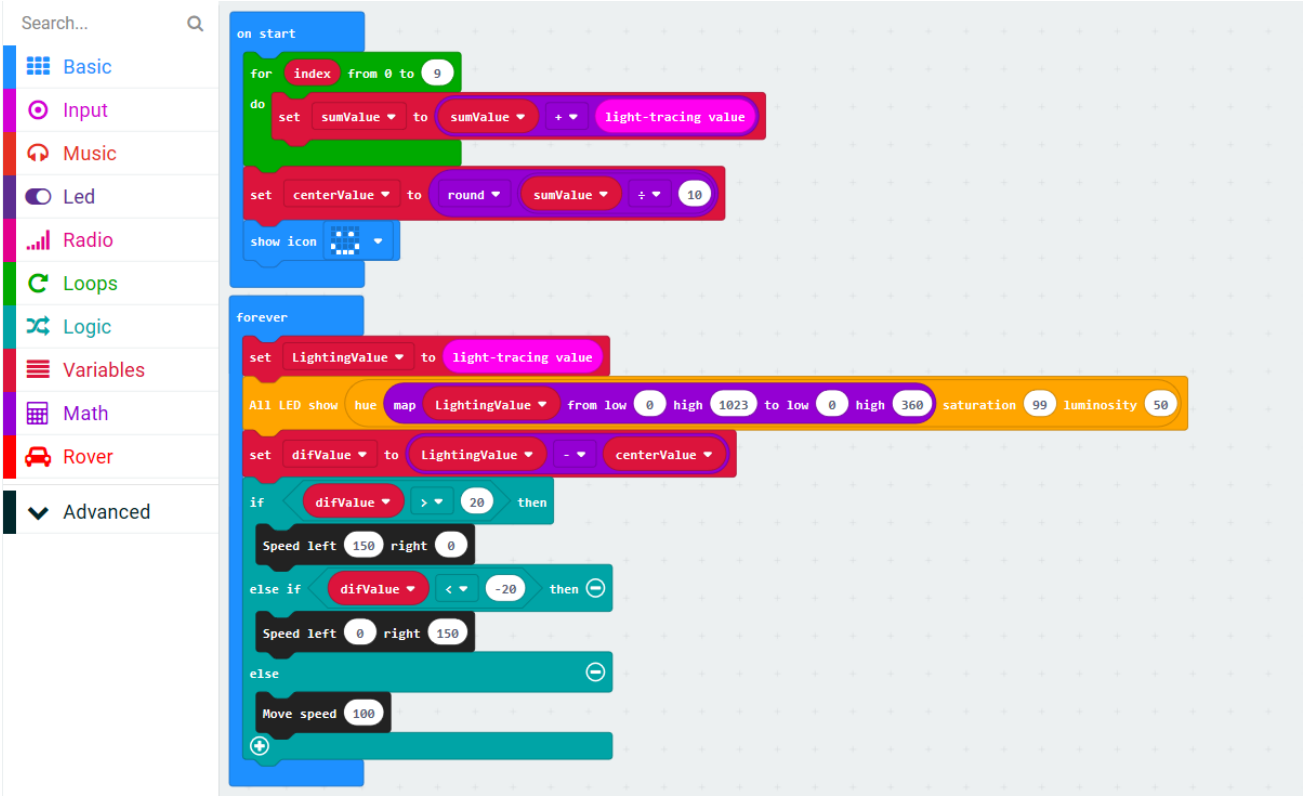
The program code is written according to flow chart, as shown below.



Code

Load code according to the table below:

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



As mentioned before, when using light sensors, we need to calibrate them first. Therefore, in the code, the sensor values are read 10 times after booting up, and their 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 make 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 for 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 the 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 reflected by the objects.. 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.

If you have any concerns, please feel free to contact us at **support@freenove.com**

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 .

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?)

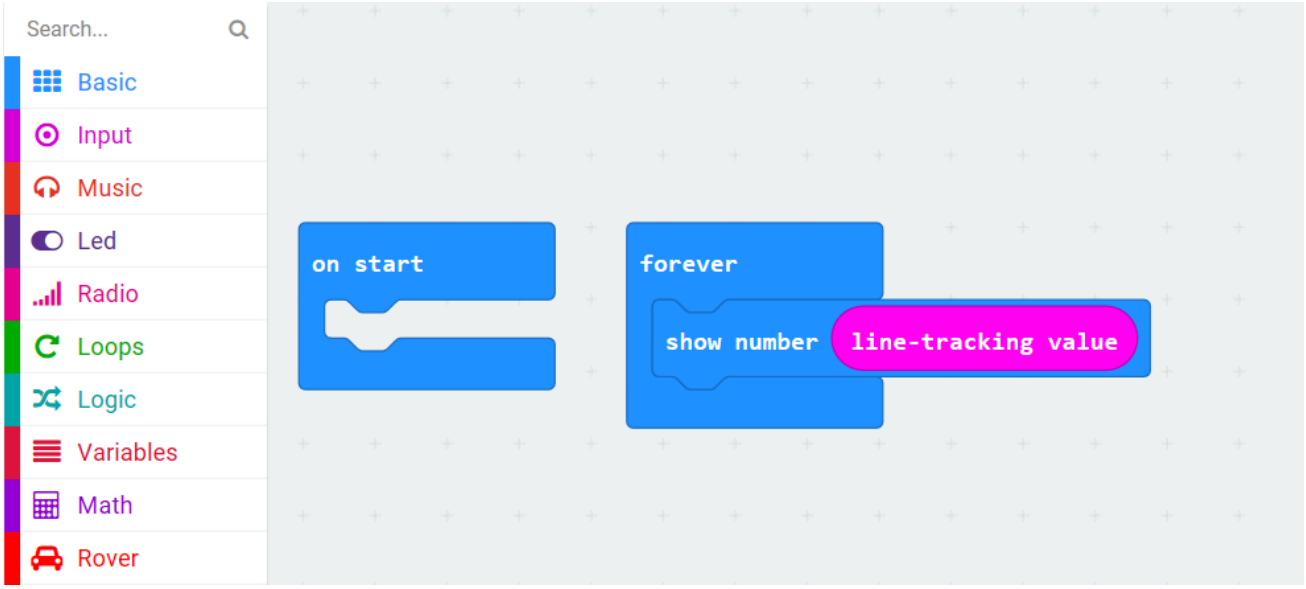
Warning

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 shown in the picture below:

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



Download the code to micro:bit. Approach the Tracking Sensor with black and white objects, respectively, and observe the values showed by the indicator 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, 1 presents black objects or no objects were detected, and 0 indicates white objects were detected.

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

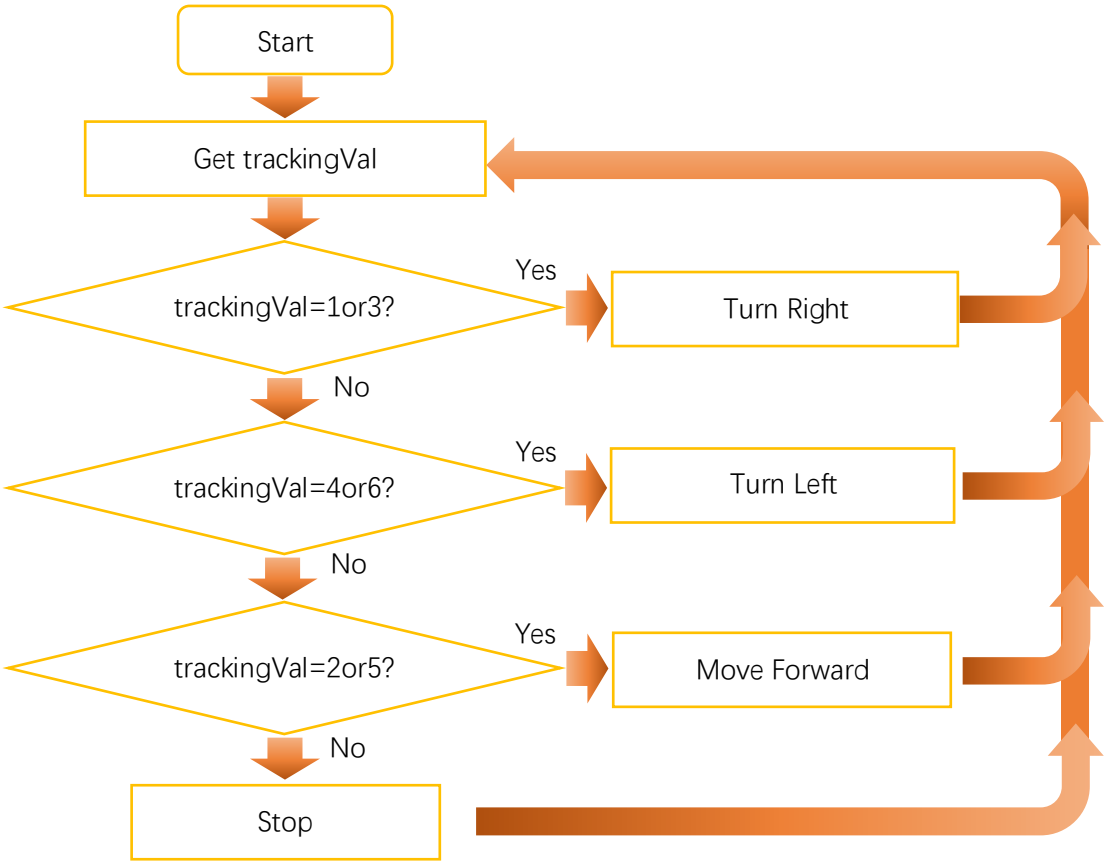
This project will realize line tracking mode of Rover.

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

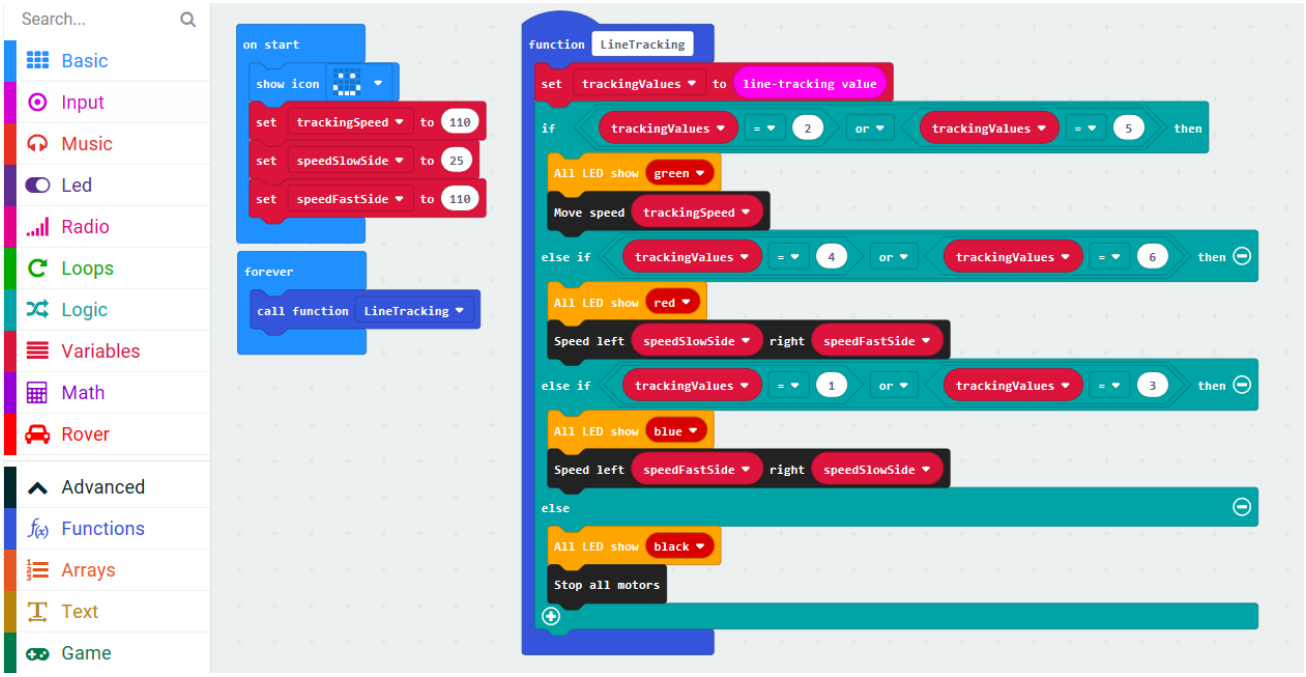
Flow chart is as below:



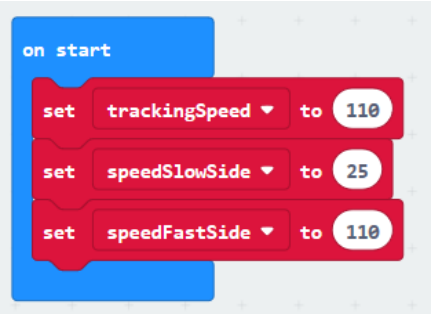
Code

Load code according to the table below:

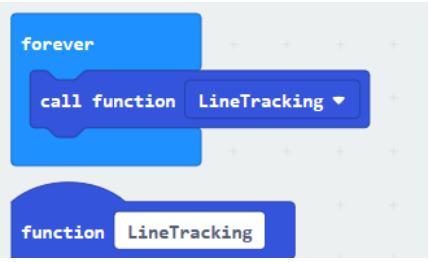
File type	Path	File name
Hex file	../Projects/05.2_LineTracking	microbit-LineTracking.hex



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



Among them, trackingSpeed is used to set the Rover's speed moving forward. It is not recommended to be too large, in order to avoid running out of the runway. Variables speedSlowSide and speedFastSide are used to set the turning speed, and the smaller the difference between them, the smoother the turning, but the harder to pass through sharp turn. The larger the 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 block. And we will see many such usage later.

Download the code to micro:bit, put Rover on Line-Tracking Map comes with in this kit and then observe motion of Rover.

Chapter 6 Bluetooth

The 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 communication between micro:bit and mobile phone, and use app to control Rover's action to realize the function of a wireless remote control.

If you have any concerns, please feel free to contact us at **support@freenove.com**

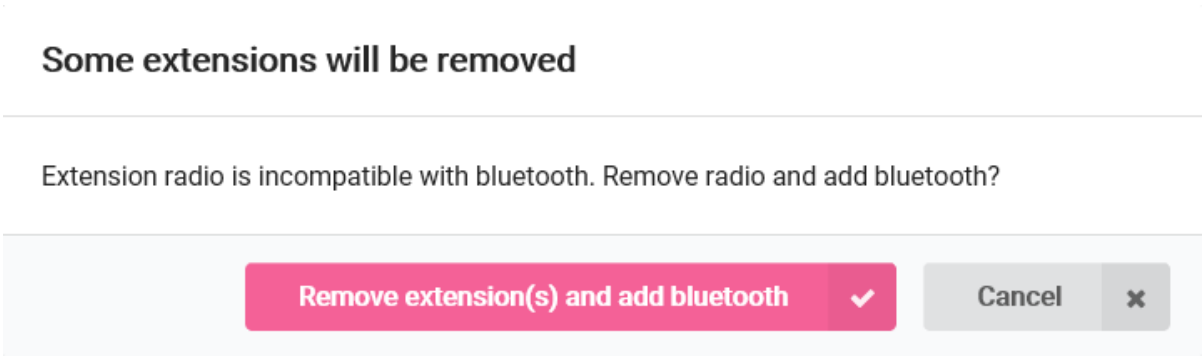
Preparation

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.

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 compatible with each other. When installing Bluetooth extension library, you will be prompted to delete Radio extensions. Just confirm the removing.

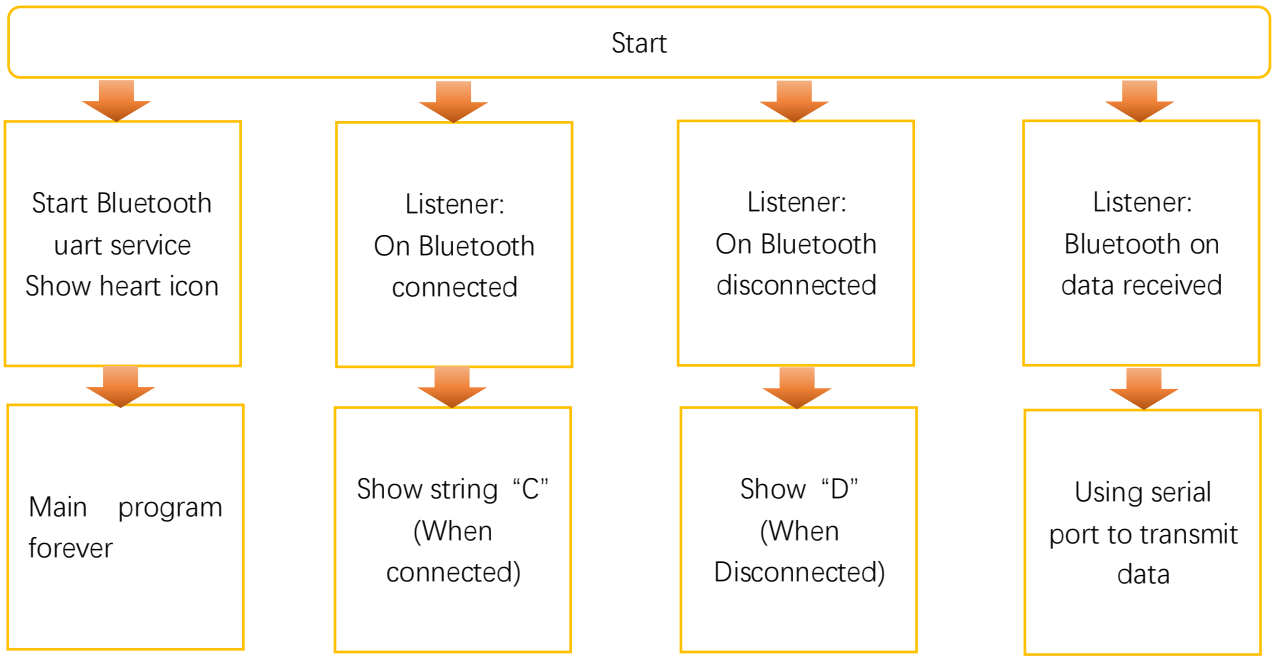


Getting Bluetooth Data

This section is to analyze Bluetooth data, which is theoretical. If you have no interest in it, you can skip to next section. It won't affect the experiencing of future 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 are familiar with uart, you can also use a third-party serial tool.

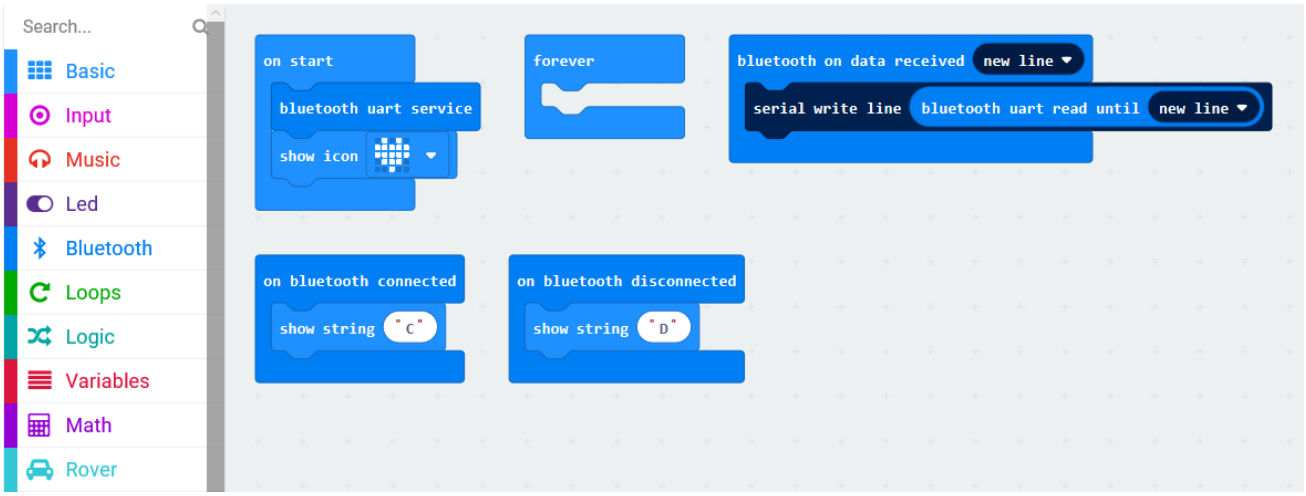
Flow chart



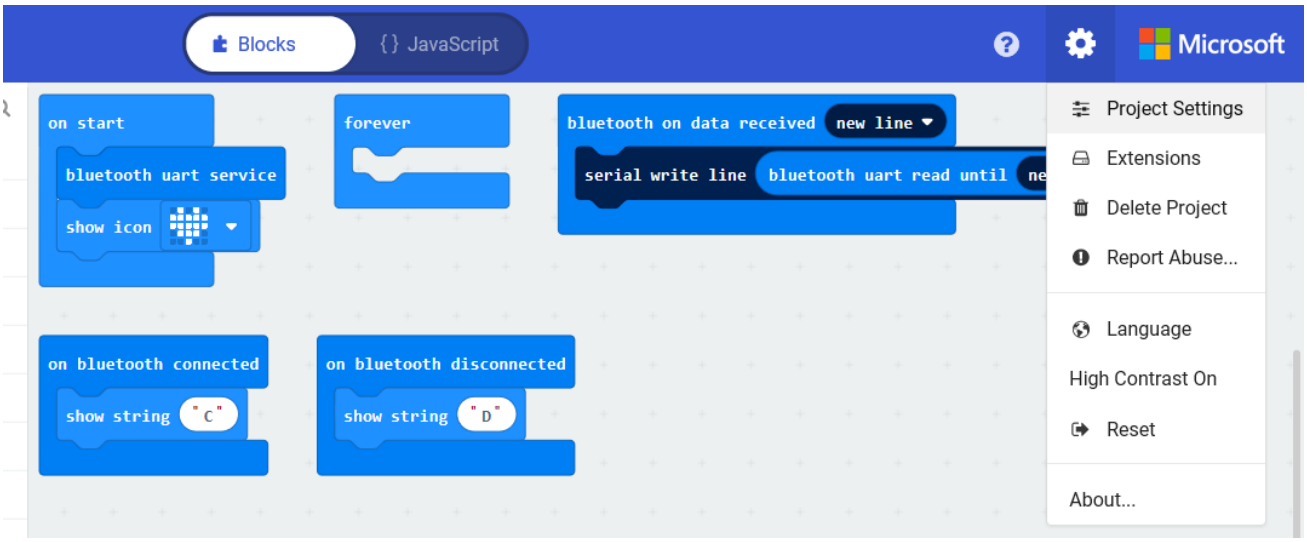
Code

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

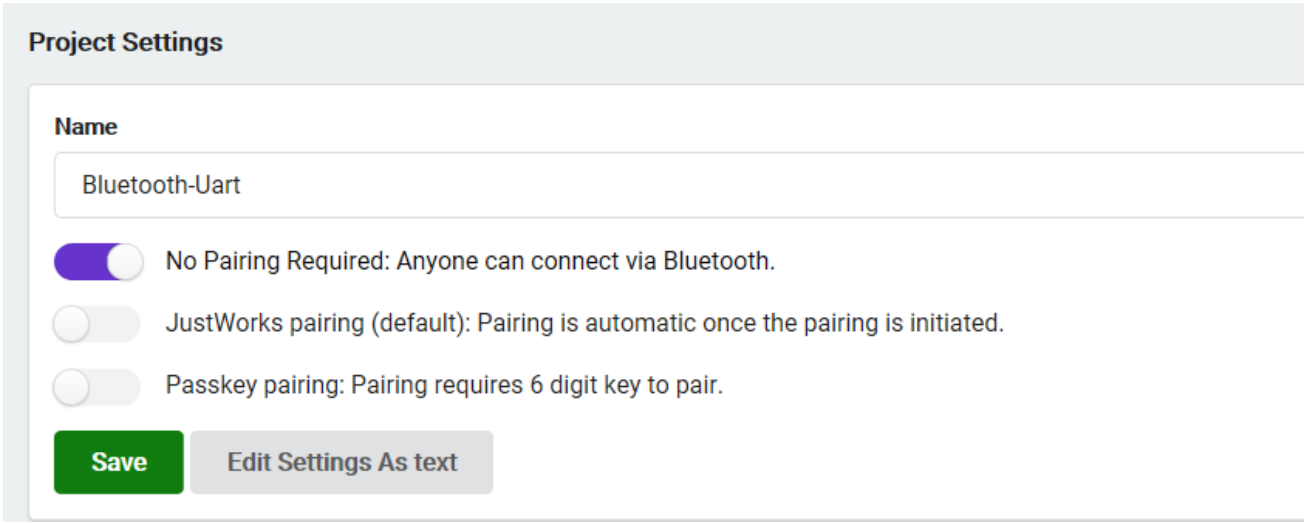
File type	Path	File name
Hex file	../Projects/06.1_Bluetooth-Uart	microbit-Bluetooth-Uart.hex



Then click gear icon(settings)→Project Settings.

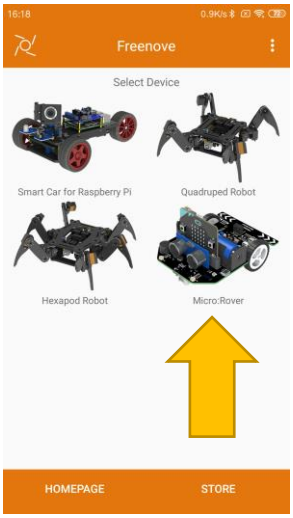


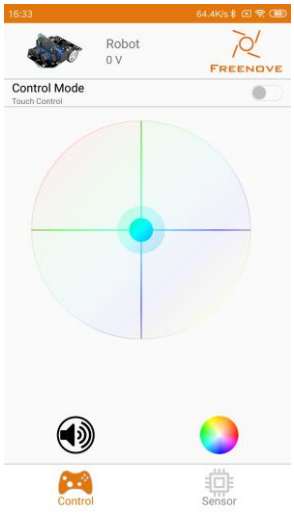
When the following page appears, set "No Pairing Required" to open state as below.

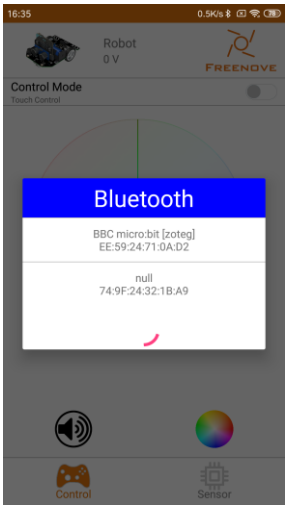


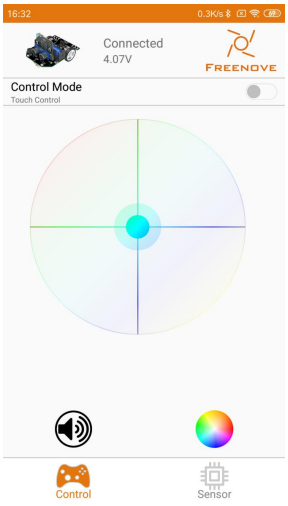
Click Save button.
Download the code to micro:bit. Later, the LED matrix of micro:bit will show heart pattern.

Open Freenove android app. **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. In this process, there may be a pop-up window asking for permission to obtain location. Click OK, otherwise Bluetooth will not be searched.





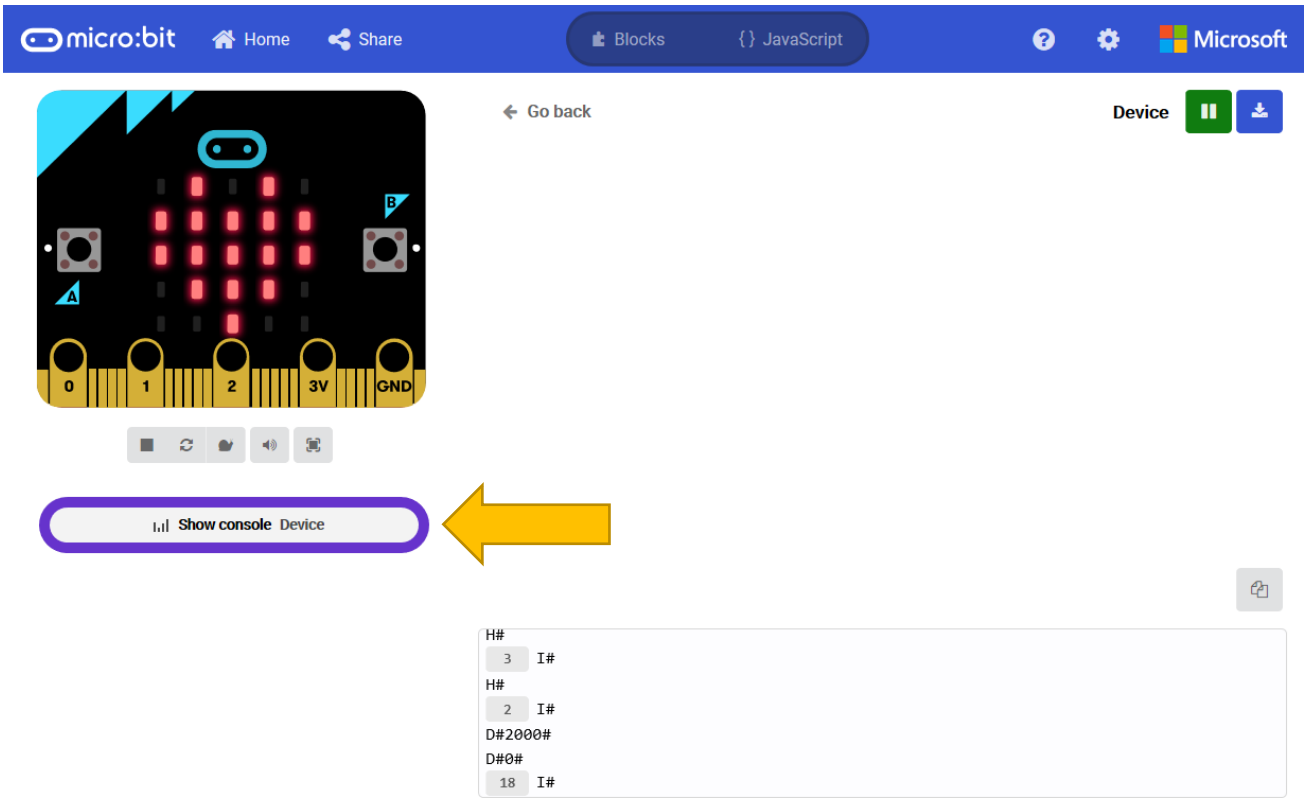




Click or drag the operating disk at the bottom, or click on the speaker, color plate or other controller to view the data received in MakeCode.

In MakeCode, when data is received, there will be a **"ShowConsoleDevice"** button below the simulator. Then click this button to view the received serial data.

If you use web version of MakeCode without webUSB function, you will not see the "ShowConsoleDevice" button.



Data analysis

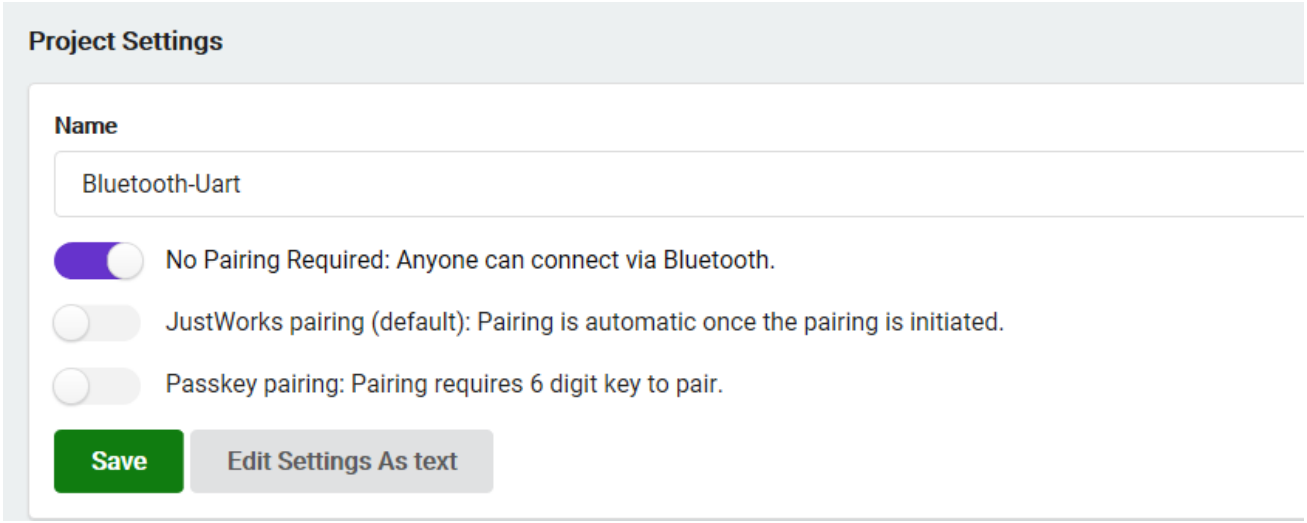
Learning the data commands sent from app will help to understand and analyze later program.
The command format for communication between app and micro:bit is A#xxx#xxx#...xxx#, where # is a separator, the first character A represents the action command, the following xxx represents the parameters of 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 transmission is 20 bytes, and the part exceeding 20 bytes will be lost.

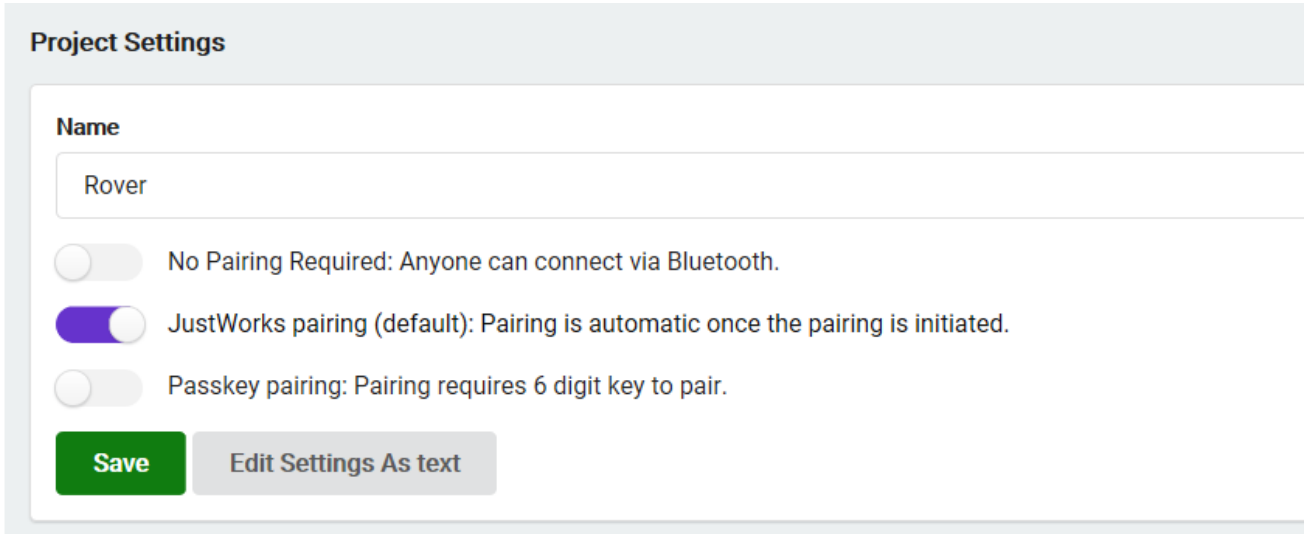
Bluetooth pairing

In previous settings, we switched on the option "no need for pairing", which facilitates us to debug the program.



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 a 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",



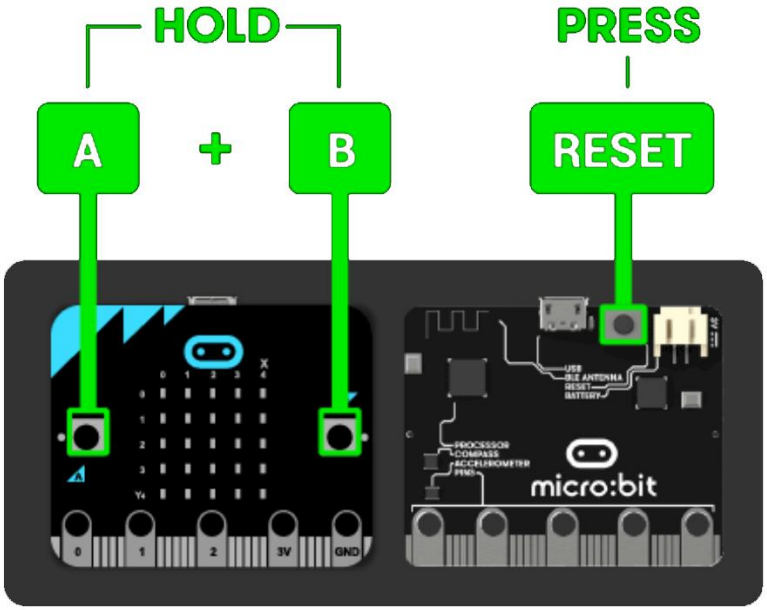
With this option, your micro:bit cannot be found by others. It can only be connected to the Bluetooth device paired before.

Pair

The 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 **short press 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, at this point, if it is then paired with mobile phone B,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.

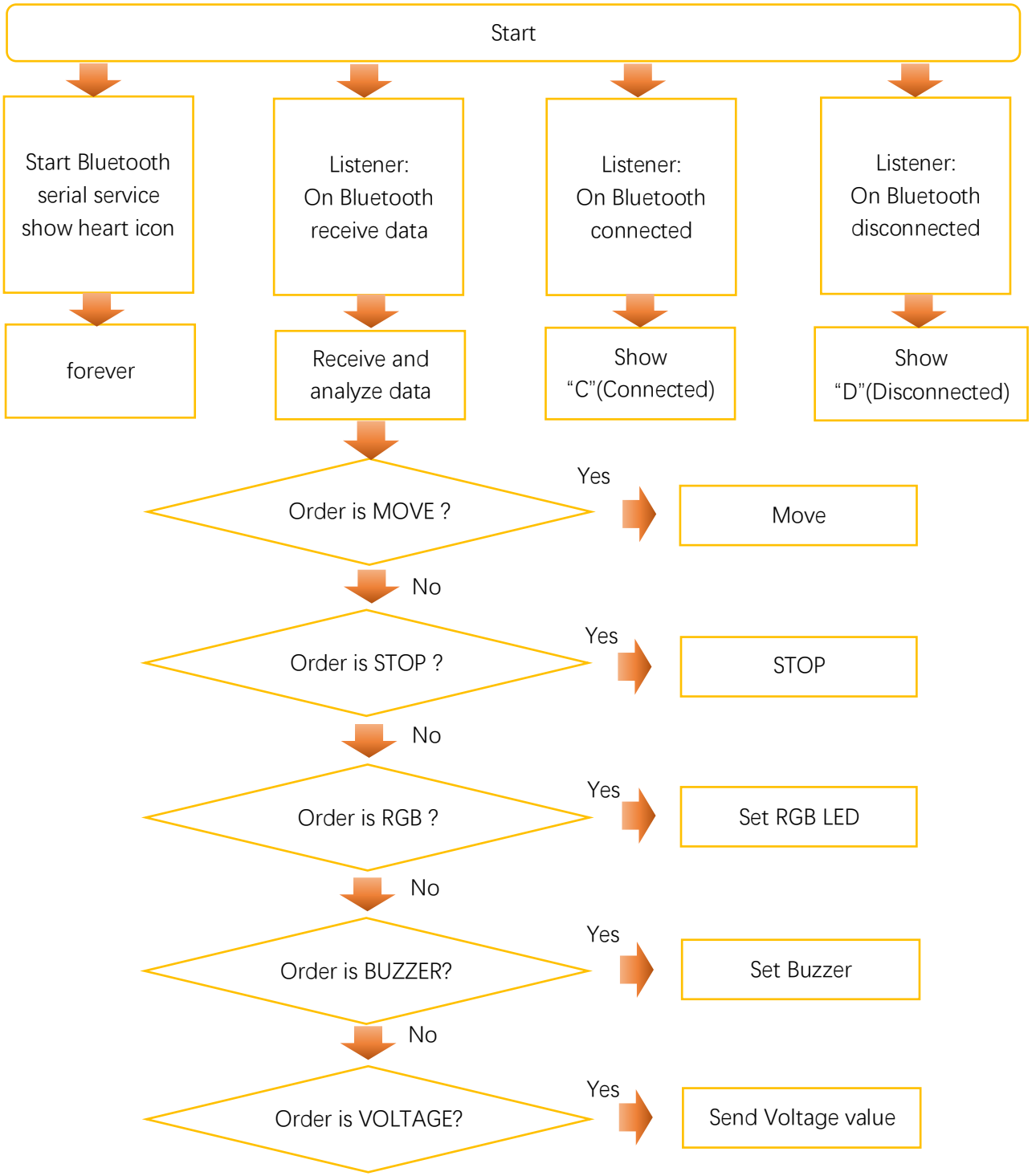
If you want to make multiple phones connected to one micro:bit at any time, you can select "No pairing Required" in the Settings of MakeCode as before.

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

Based on the previous Bluetooth knowledge, let's realize Bluetooth remote control mode of Rover.

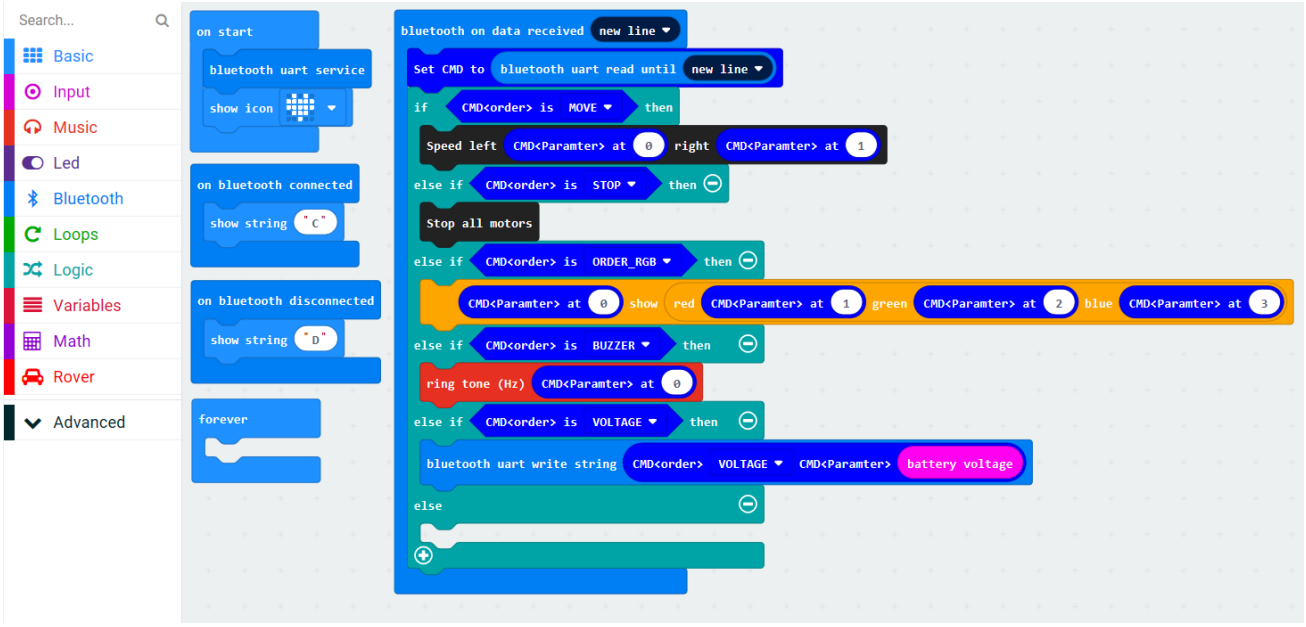
Flow chart



Code

Load code according to the table below:

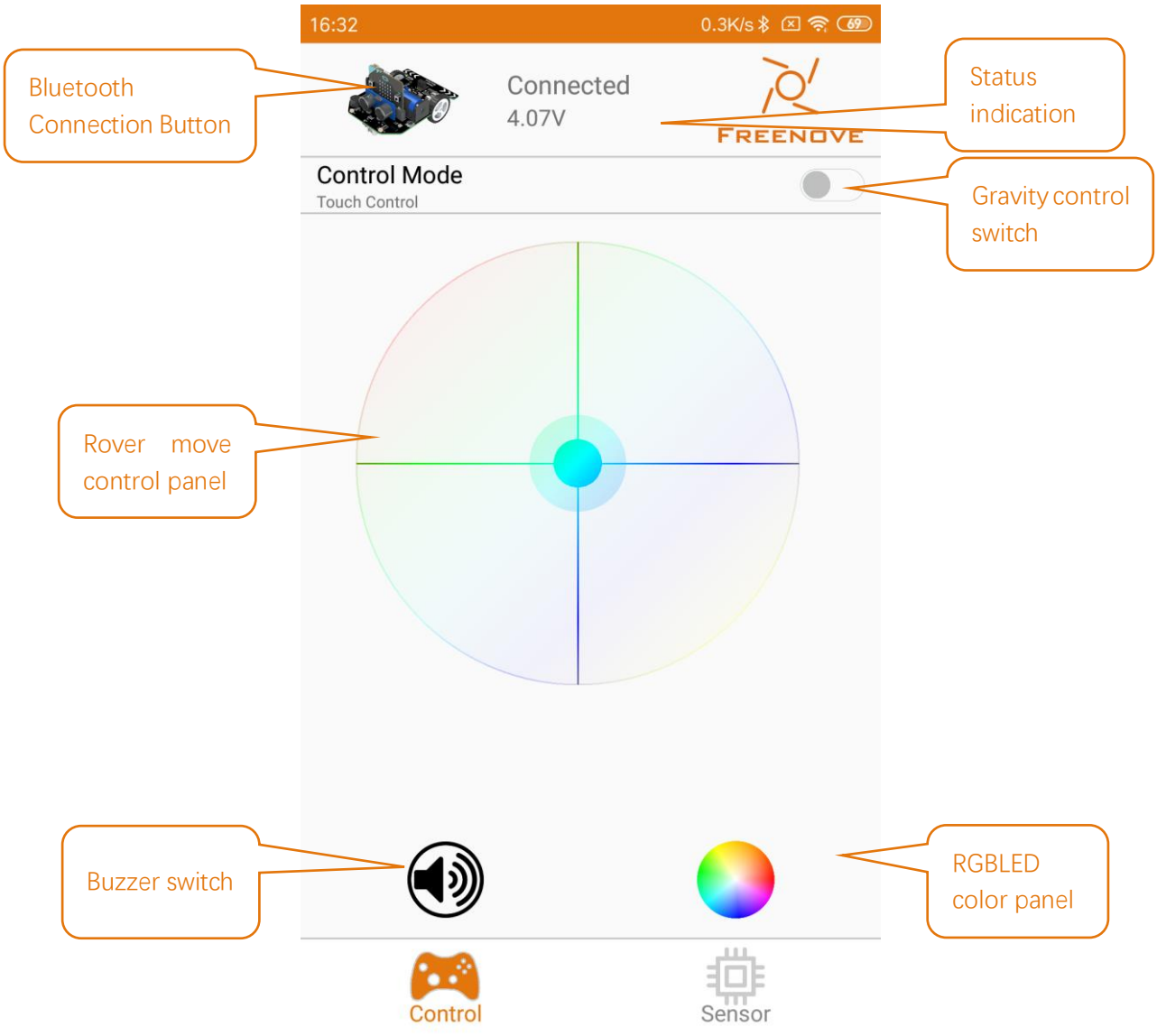
File type	Path	File name
Hex file	../Projects/06.2_Bluetooth-Remote	microbit-Bluetooth-Remote.hex



This code enables micro:bit to 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 psotion 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. We will introduce sensors in next section.



Chapter 7 Android and iOS 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 the app.

If you have any concerns, please feel free to contact us at **support@freenove.com**

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.

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

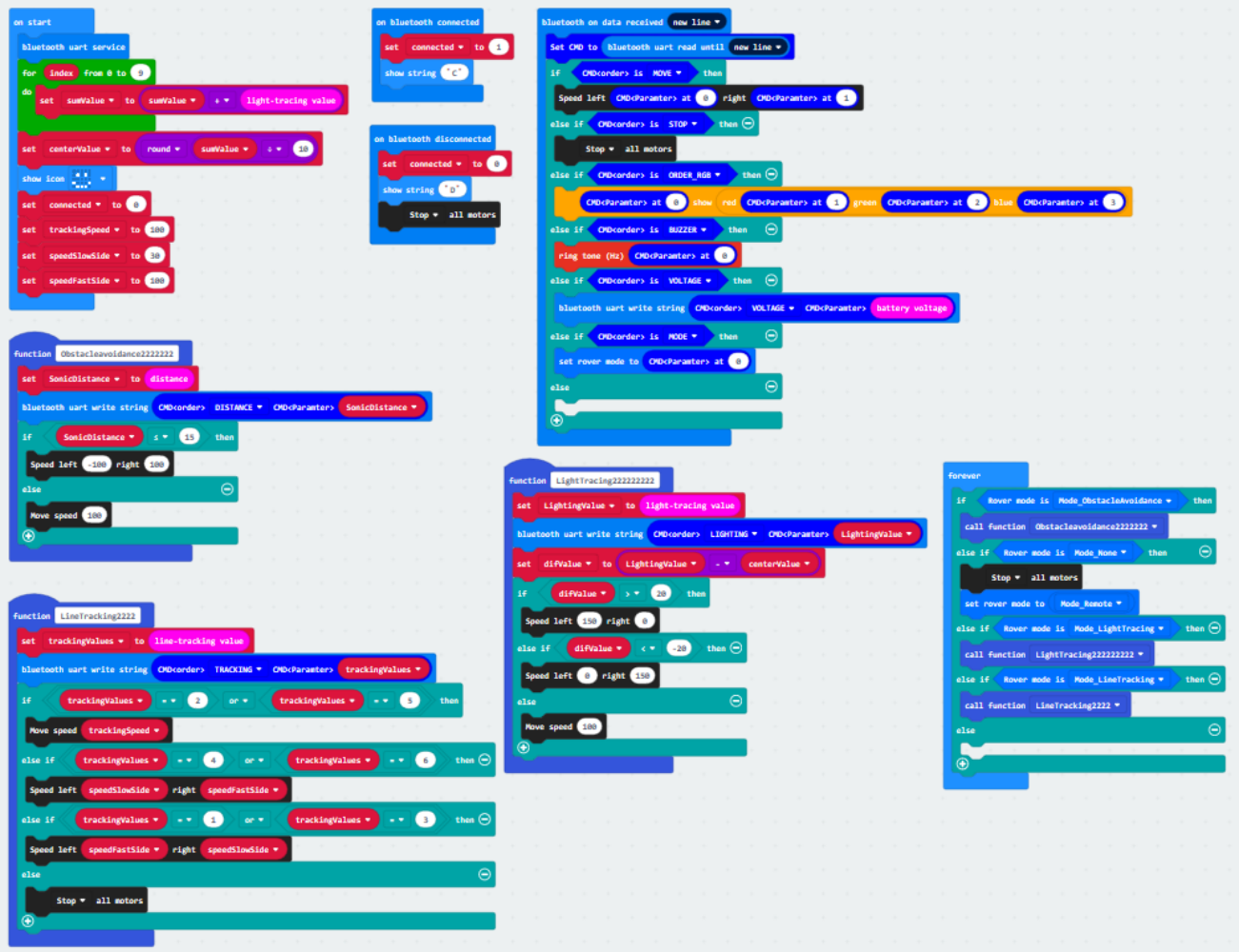
If you choose to drag code manually, you first need to add Rover and Bluetooth extension libraries as before. (How to add Rover extension?)

Rover

Code

Load code according to the table below:

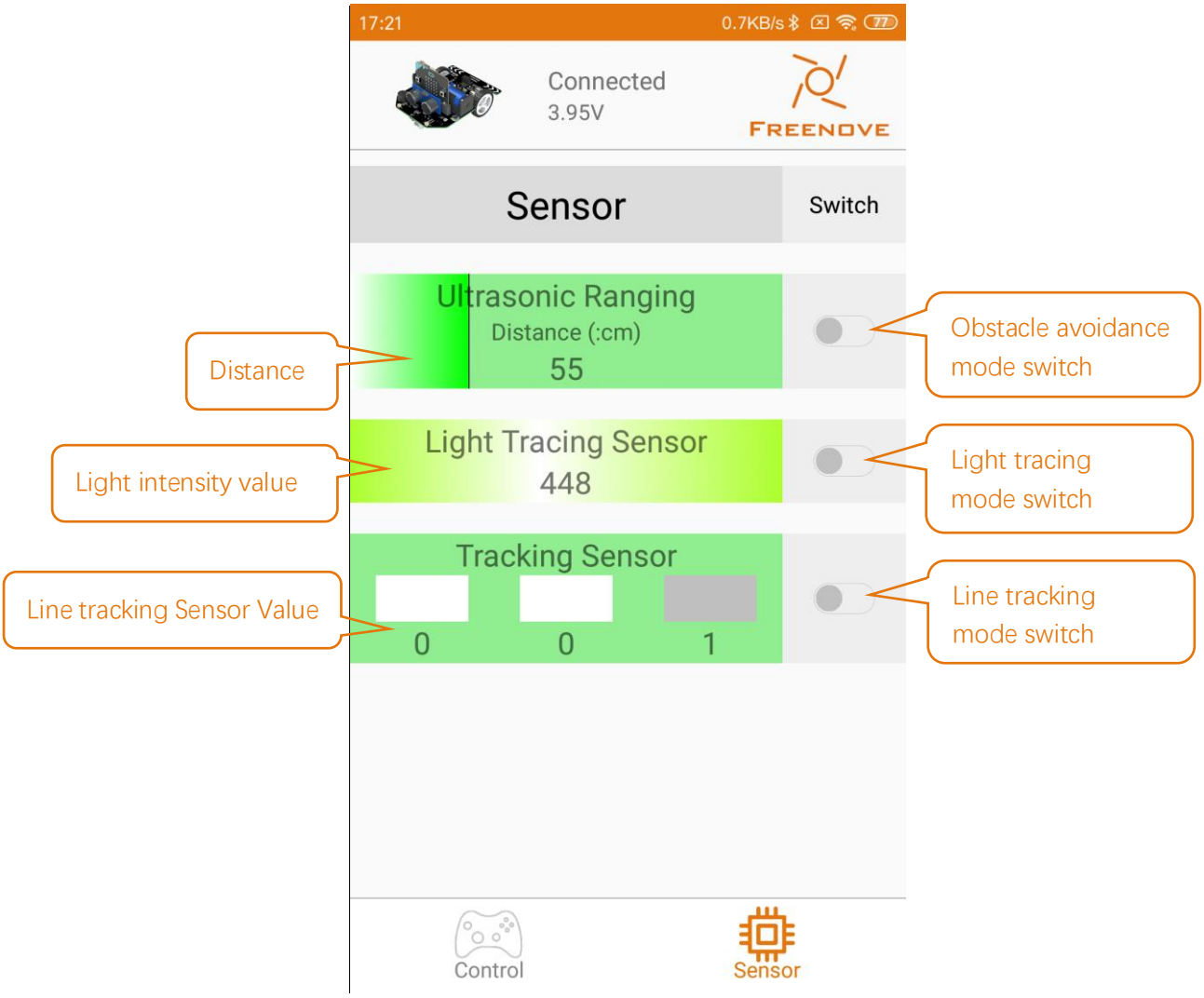
File type	Path	File name
Hex file	../Projects/07.1_Rover	microbit-Rover.hex



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.

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 in it, 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.

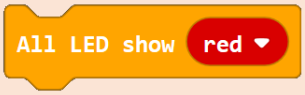
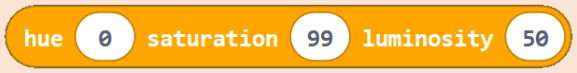
Enjoy your Rover.

Appendix

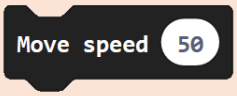
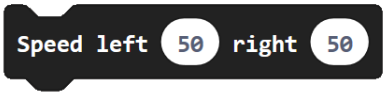
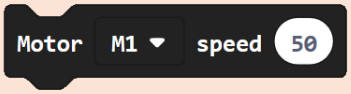
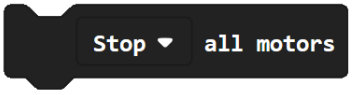
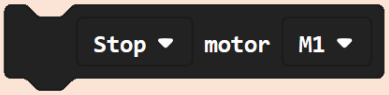
The appendix section introduces the role of each block in Rover Extension.

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.

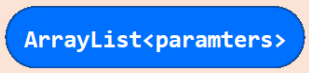
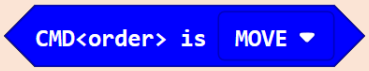
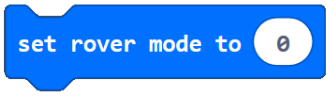
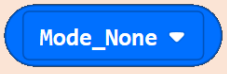
Motors

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<paramters> accordingly.
	Return the parsed command.
	Return an array in which parsed parameters are stored.
	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?

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: support@freenove.com
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.