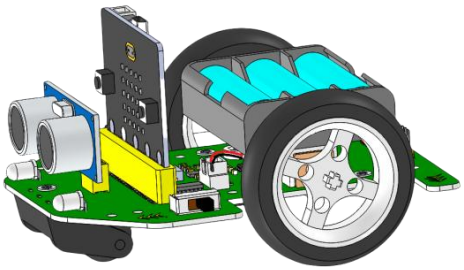
SIYEENOVE

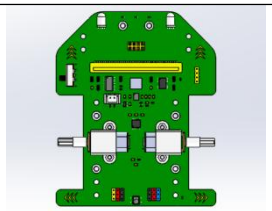
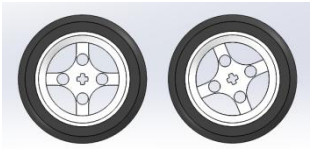
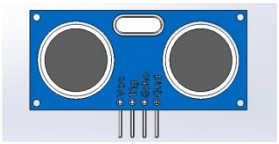
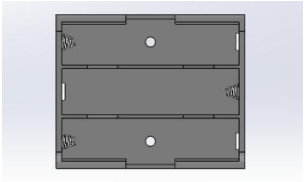
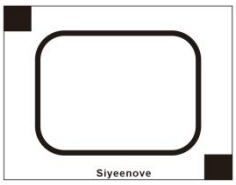
Table of Contents

1. Introduction to mCar

SIYEENOVE mCar robot is a micro:bit car designed around maker education and programming learning. It is very easy to install and multifunctional. Suitable for children over 6 years old, students and electronics enthusiasts.

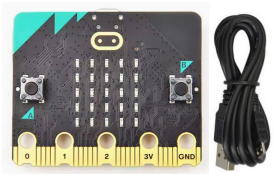


1.1 Product List

PCB base	Wheel	Ultrasonic module
		
AA Battery Case	Infrared remote control	Map
		
3M adhesive sticker		
		

Note: You need to purchase three AA batteries and a Micro:bit v2 board with

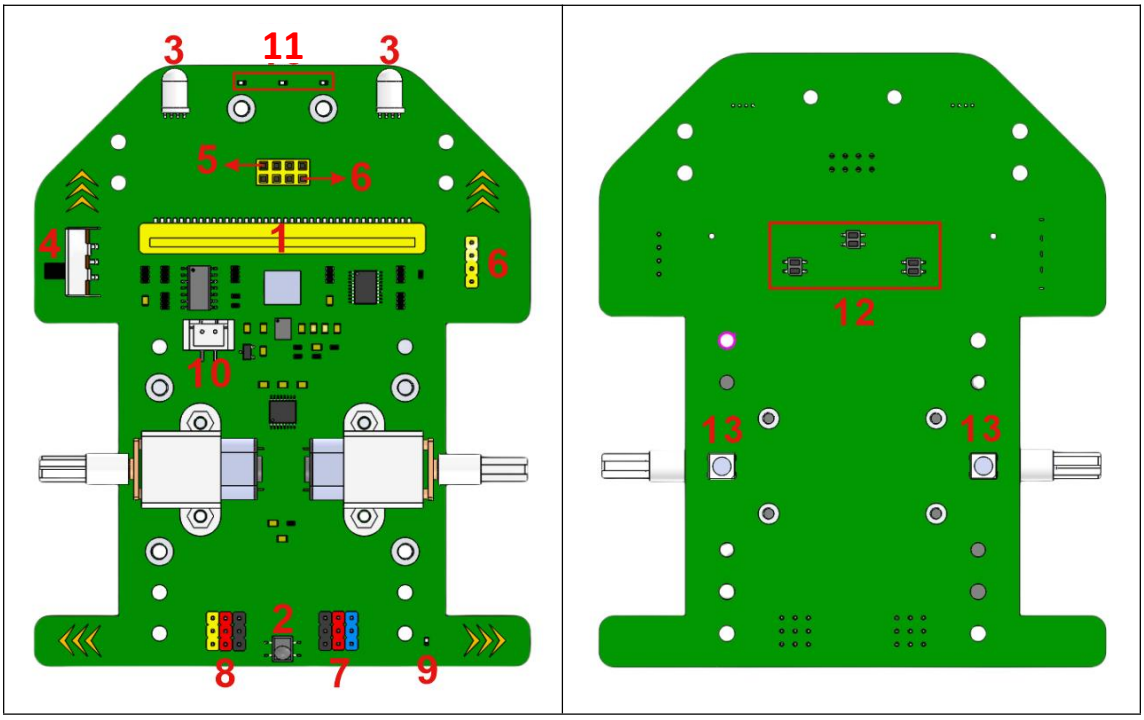
USB cable by yourself!

AA Battery	Micro:bit-v2 with USB cable
	

1.2 Specifications

Control board:	Micro:bit - v2 and v1 (not included)
Programming language:	MakeCode
Battery interface type:	XH2.54 2P, input voltage 3--5.5V, 3 AA batteries are required
Infrared remote control:	NEC, remote control distance ≧5 meters
Motor:	GA12-N20 micro DC gear reduction motor (145 rpm)
Ultrasonic sensor:	HC-SR04 (distance measurement: 2cm-400cm, accuracy: 3mm)
Extension interface type:	2.54mm pitch pin header
Wheel diameter:	53mm
Length and width:	117.5*97*70mm

1.3 PCB base introduction



1: Micro:bit slot

Can be plugged into a Micro:bit v2 and v1 board.

2: Infrared receiving sensor

Using NEC infrared communication protocol, its signal pin is connected to the P9 pin of Micro:bit.

3: Head light

Equipped with two 5mm RGB LED lights on the left and right, respectively controlled by STC8H1K08, and the I2C port of Micro:bit sends relevant commands to STC8H1K08 to control it.

4: Power switch

Controls the power on and off of the battery box.

5: Ultrasonic module interface

V=5V, G=GND, used to connect the HC-SR04 sonar module, the signal pins are connected to the P12 (Echo) and P13 (Trig) pins of Micro:bit respectively.

6: I2C port

It is the I2C communication interface of Micro:bit, SDA=P20, SCL=P19, which can communicate with other I2C devices.

7: Servo interface

Can be connected to 3 servos, which can be used to drive 90, 180, 270 and 360 degree servos.

8: IO expansion port

Can be connected to other sensor modules.

9: Power indicator light

After setting the battery model, this light will be on when the robot is working, and it will flash when the voltage is less than or equal to 20% of the voltage. (This only works if the battery type is set in the program)

10: Battery interface

XH2.54 2P, input voltage 3--5.5V.

11: Infrared line tracking indicator

The light turns off when the sensor detects a black line, and turns on when it detects a white area.

12: Infrared line tracking sensor

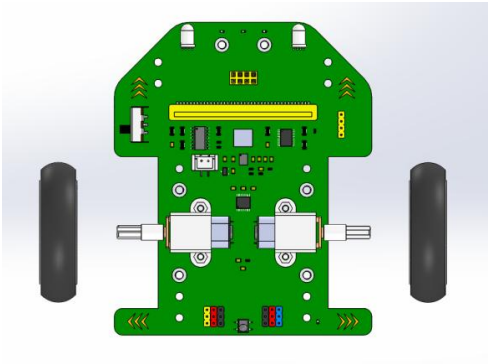
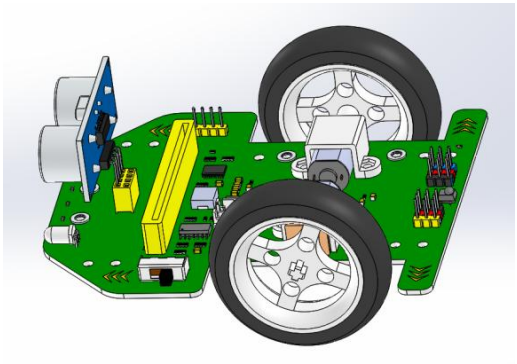
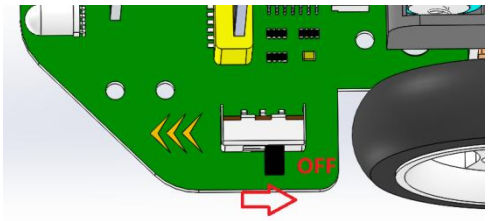
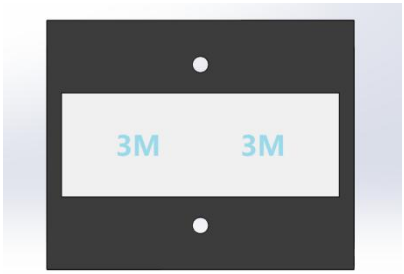
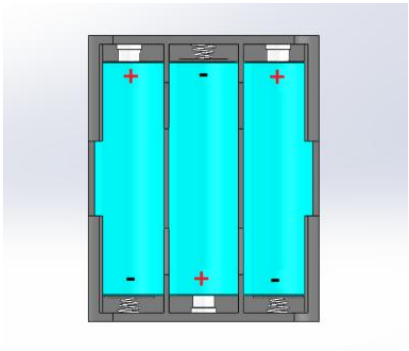
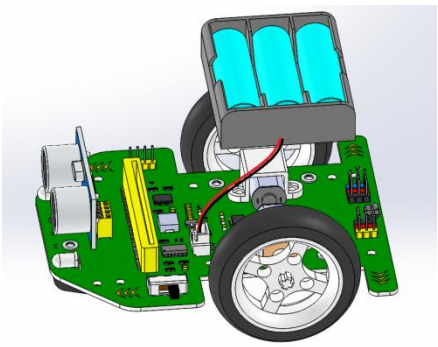
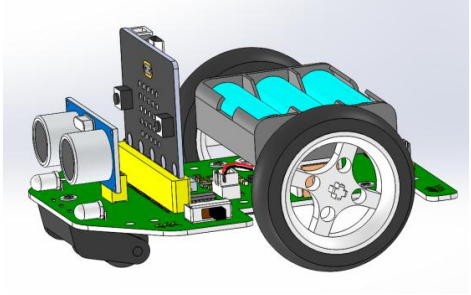
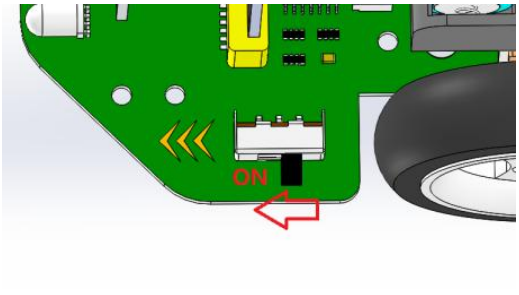
It is an infrared sensor that senses black and white areas through the principle of infrared reflection. There are three infrared line sensors on the left, middle and right bottom of the robot. The signal pins are connected to the P14, P15 and P16 pins of Micro:bit respectively.

13: RGB lights

The two WS2812 RGB lights at the bottom of the robot can emit controllable red, green and blue primary colors. By controlling the intensity of the three primary colors separately, various colors of light can be formed. The signal pin is connected to the P8 pin of Micro:bit.

7

2. Assemble the mCar

Step 1: Install the wheels	Step 2: Install the Ultrasonic Module
	
Step 3: Turn off the power(Note: Turn off the power when plugging or unplugging the Micro:bit V2 to avoid damaging it!)	Step 4: Stick the 3M double-sided tape on the center of the bottom of the battery case
	
Step 5: Install 3 AA Batteries	Step 6: Glue the Battery Box to the Robot
	
Step 7: Insert the Micro:bit V2 board (note that the dot matrix faces forward)	Step 8: Turn on the power switch
	

3. Getting Started with MakeCode

Microsoft's MakeCode editor is the perfect way to start coding with the BBC micro:bit. MakeCode is free and works across all platforms and browsers.

We recommend using Chrome or Edge browsers. WebUSB is a recent and developing web feature that allows you to access a micro:bit directly from a web page. It also lets you directly receive data into the MakeCode editor from the micro:bit. It works in Google Chrome and Microsoft Edge browsers.

WebUSB support for your micro:bit

If you're not using a current version of the Chrome or Microsoft Edge browsers, make sure they are this version or newer:

Chrome (version 79 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

Microsoft Edge (version 79 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

Link to download the latest Google Chrome:

https://www.google.com/chrome/

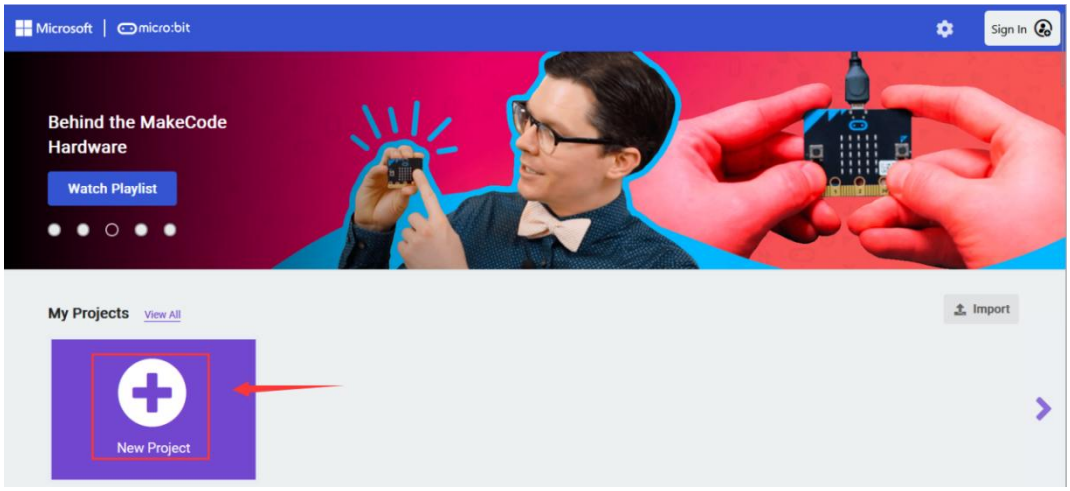
Link to download the latest Microsoft Edge:

https://www.microsoft.com/en-us/edge/download

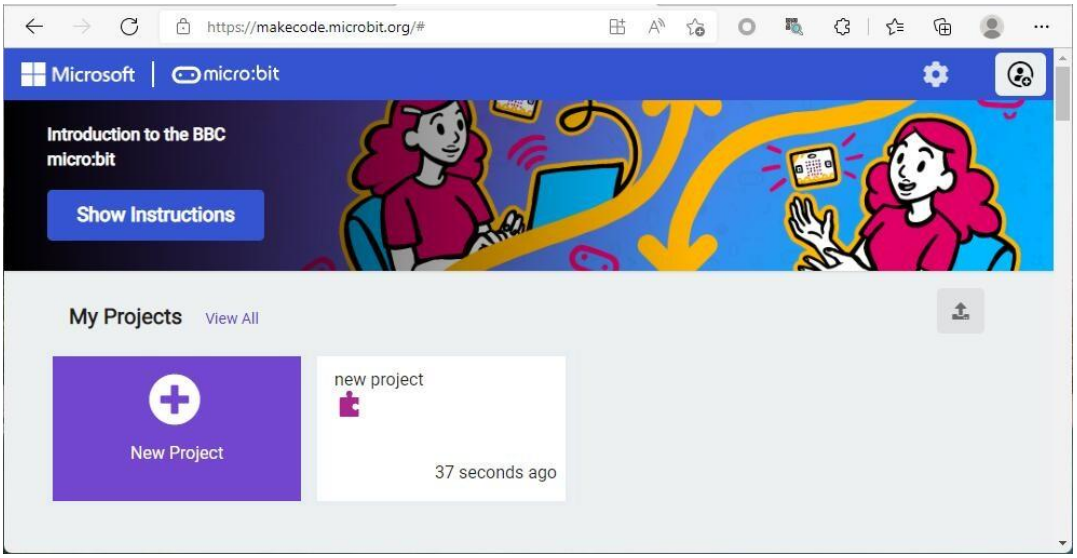
9

3.1 Create a new project

Open the Makecode editor on your browser: https://makecode.microbit.org，click"**New Project**", Then you can **give a name** for your project.

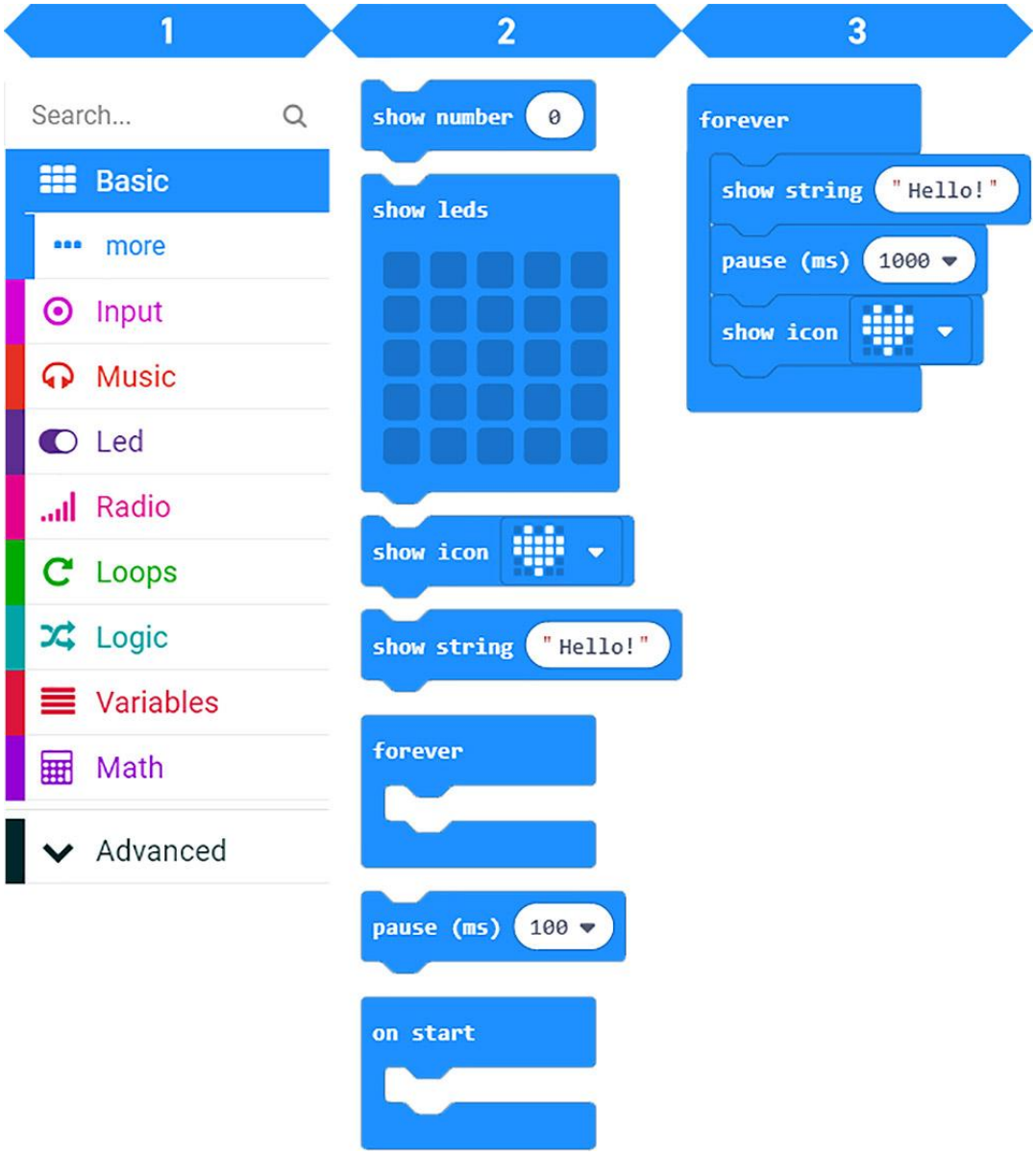


The newly created projects will be saved in the current browser. Just revisit the https://makecode.microbit.org website and find them in the project list.



Selecting Blocks:

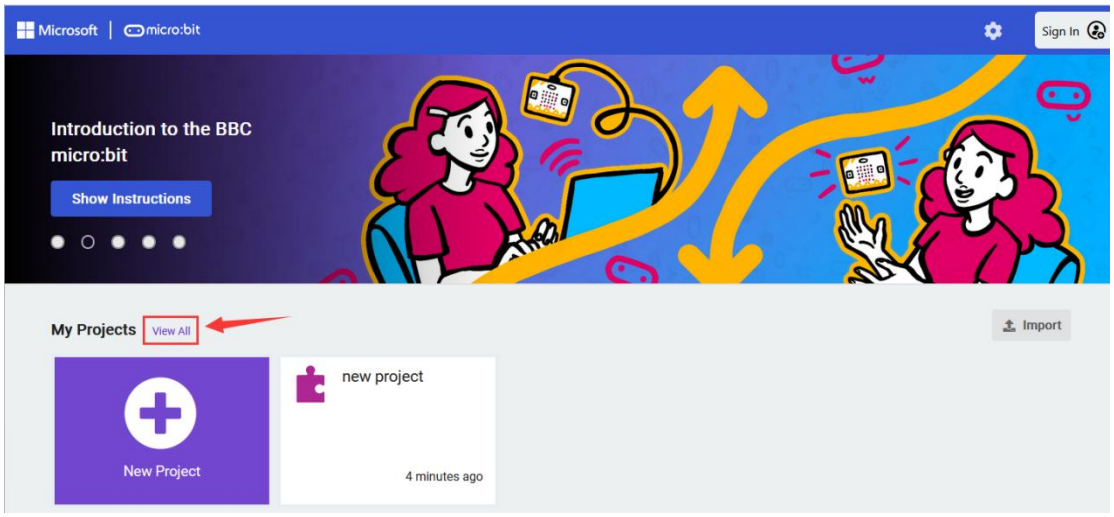
1. Select a block category from the list on the left-hand side of the page.
2. Select a block from the selected category, then drag it to the workspace area on the right.
3. Snap new blocks onto existing blocks in the workspace area. As the new blocks are dragged into the workspace, the editor highlights the connecting parts of each block when they are in a valid position to snap to existing blocks. Also, the shape of the blocks gives you an indication of where they might fit into your code blocks.



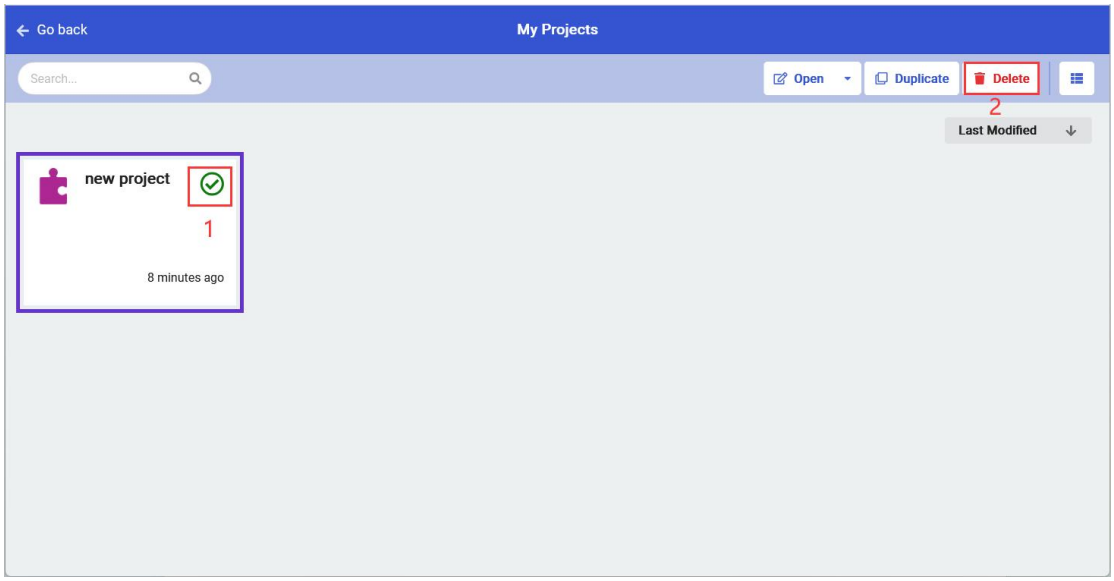
11

3.2 Delete a Project

Click"**View All**":



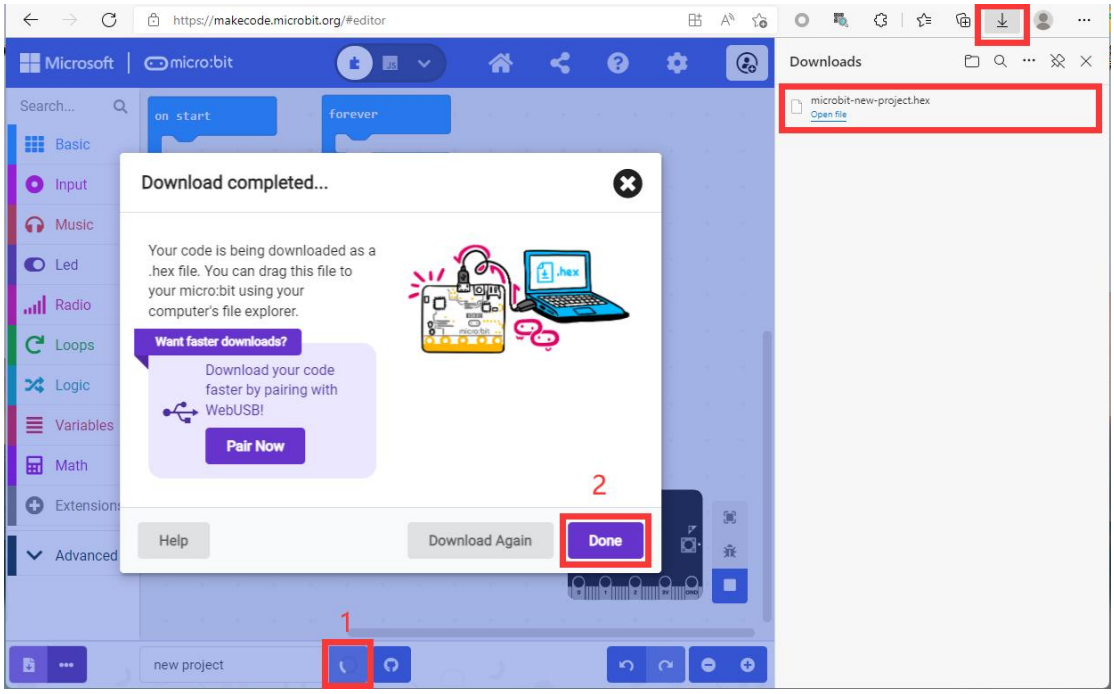
Select the project you want to delete, and click **Delete** button.



12

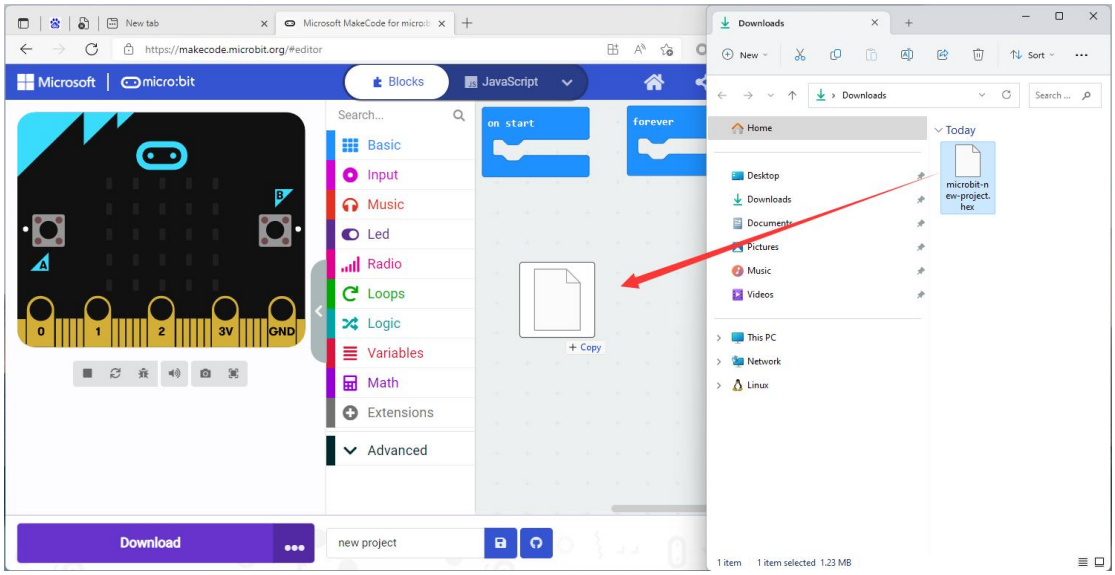
3.3 Save a project

Click the "**Save**" button, and then click the "**Done**" button to save the project to your computer, as shown below:



3.4 Import Files

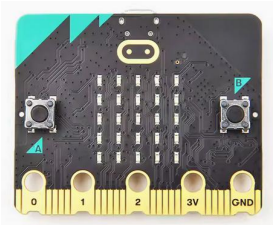
Simply drag the local "HEX" project file to the work area of the MakeCode editor, as shown below:



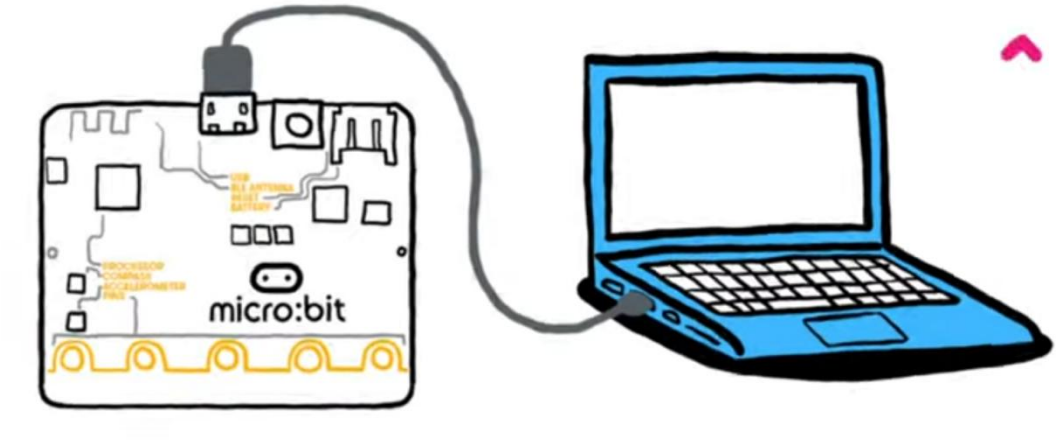
13

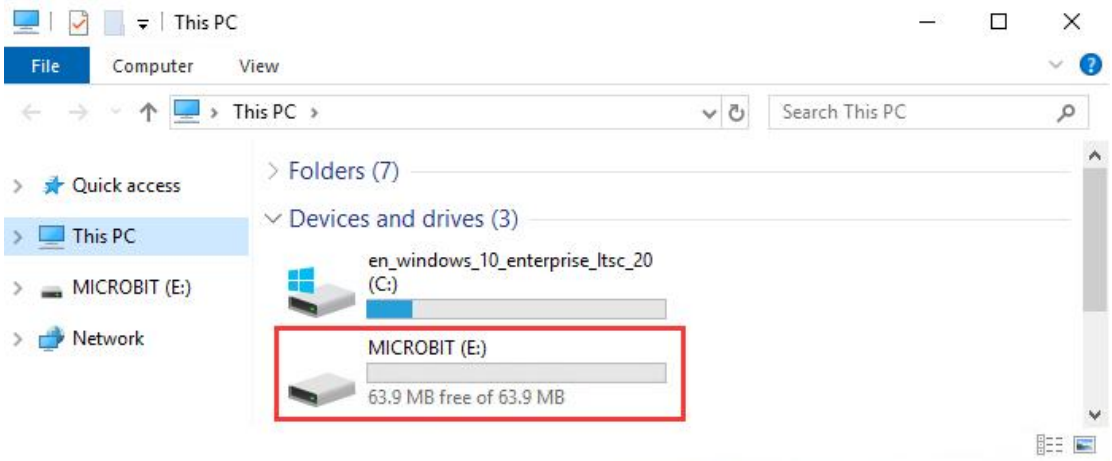
3.5 Upload code

Things you need：

PC	Micro:bit v2.x.x	Micro USB Cable
		

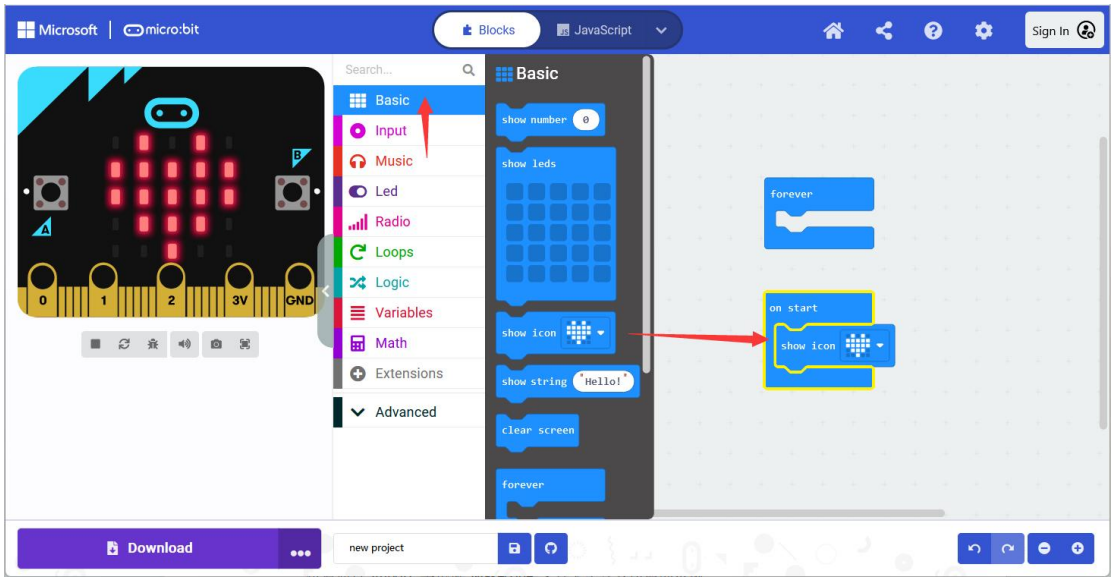
Use a Micro USB cable to connect the micro:bit board to the PC. You will find a new USB disk called MICROBIT on the PC:



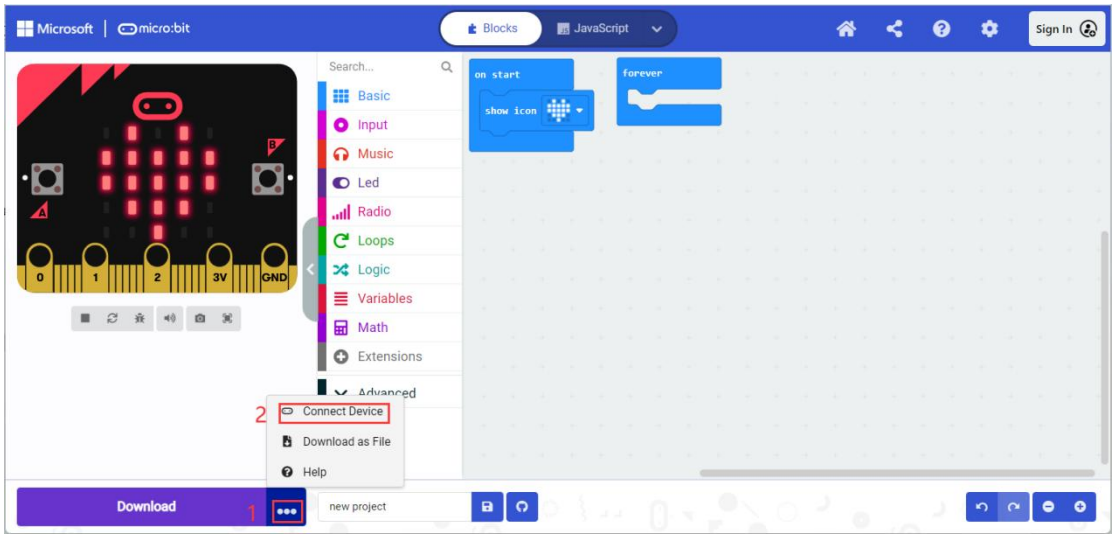


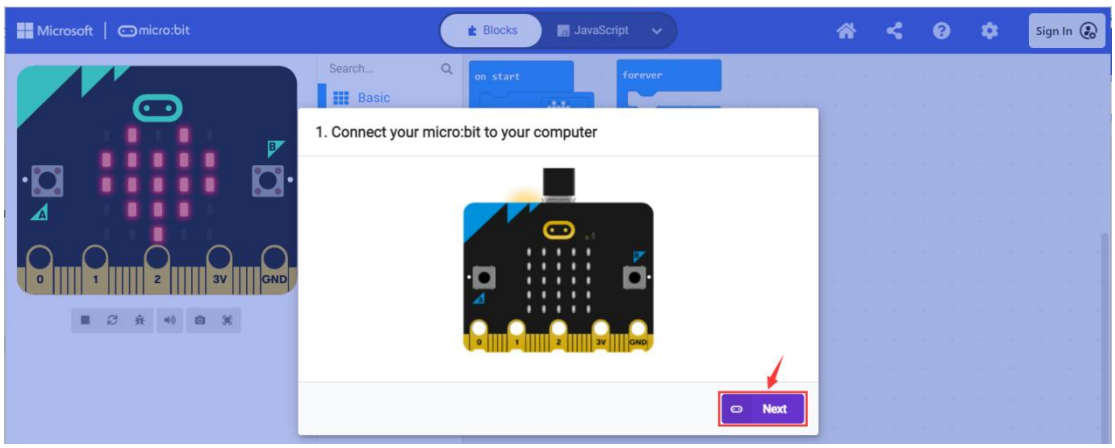
14

Open the MakeCode editor on your browser, hold down the left mouse button, and drag the **show icon** statement on the left to the working area on the right: https://makecode.microbit.org/#editor



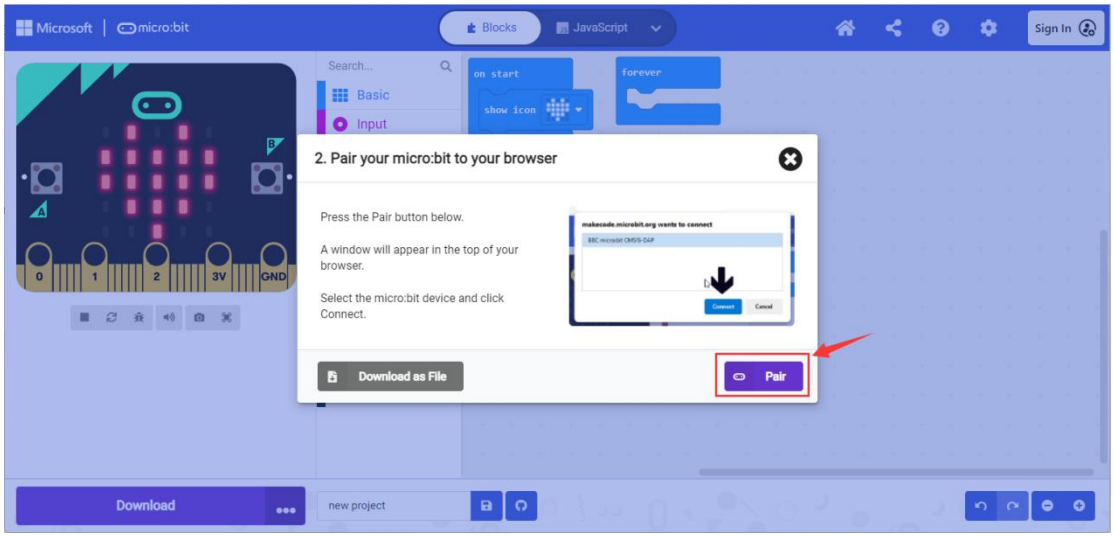
Click on the **three dots** next to the Download button. Then click the "**Connect Device**", and then click the "**Next**" , as shown below:



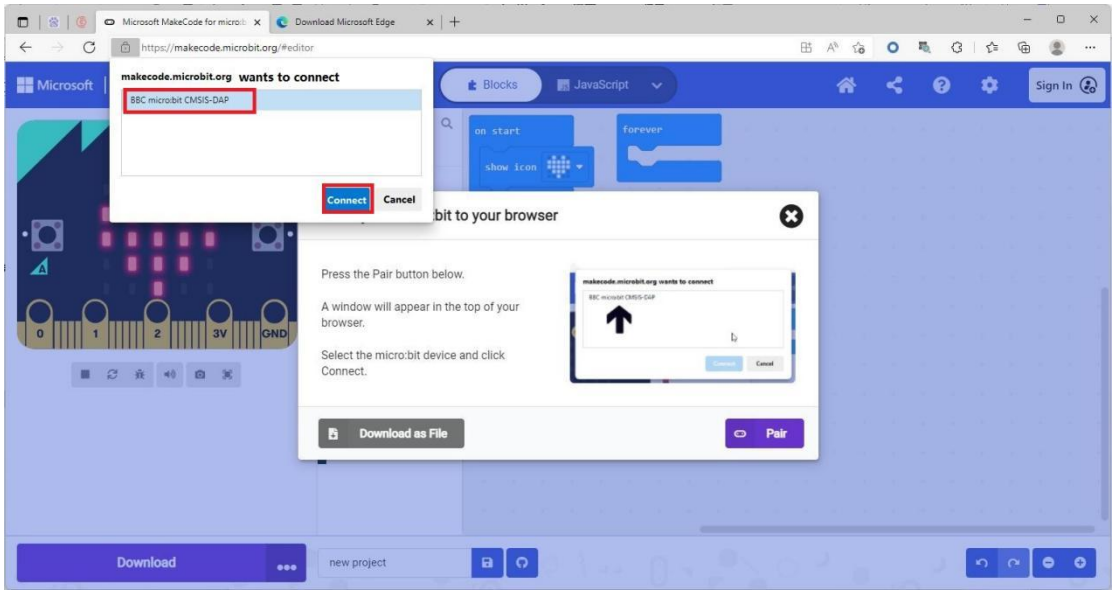


15

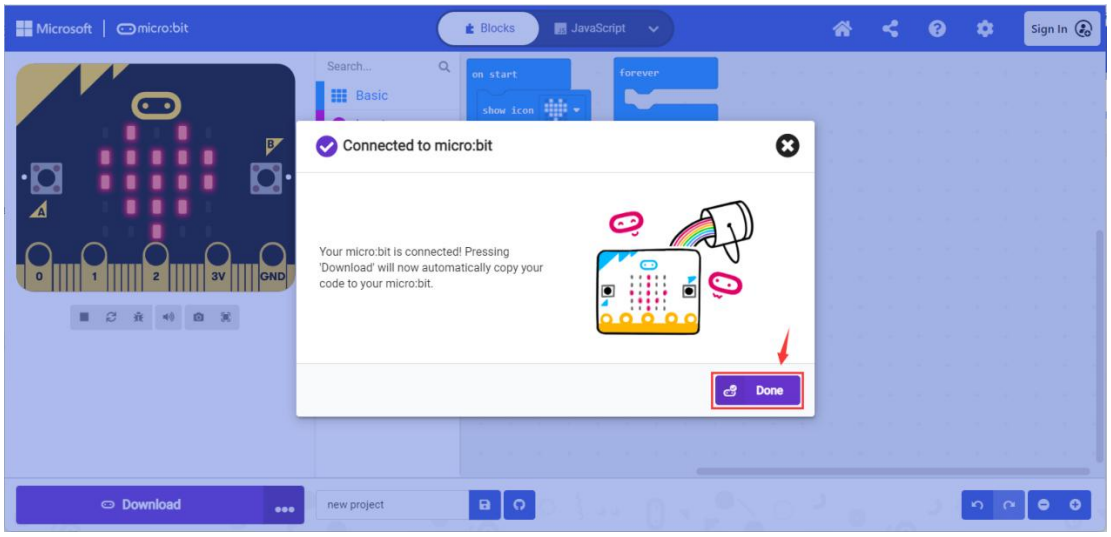
Click the "**Pair**" button:



Select the Micro:bit board you want to connect to, and then click "**Connect**":

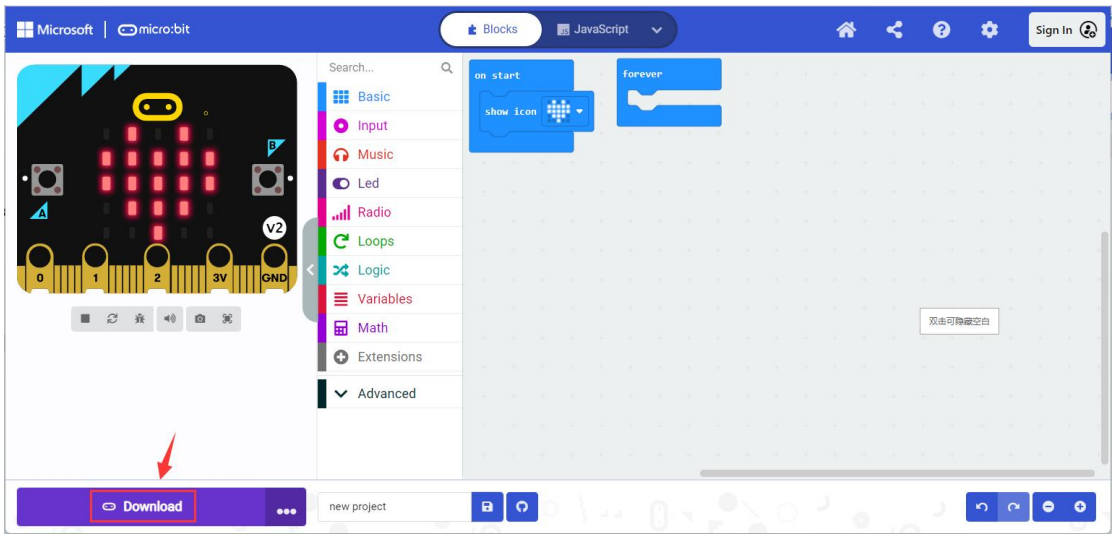


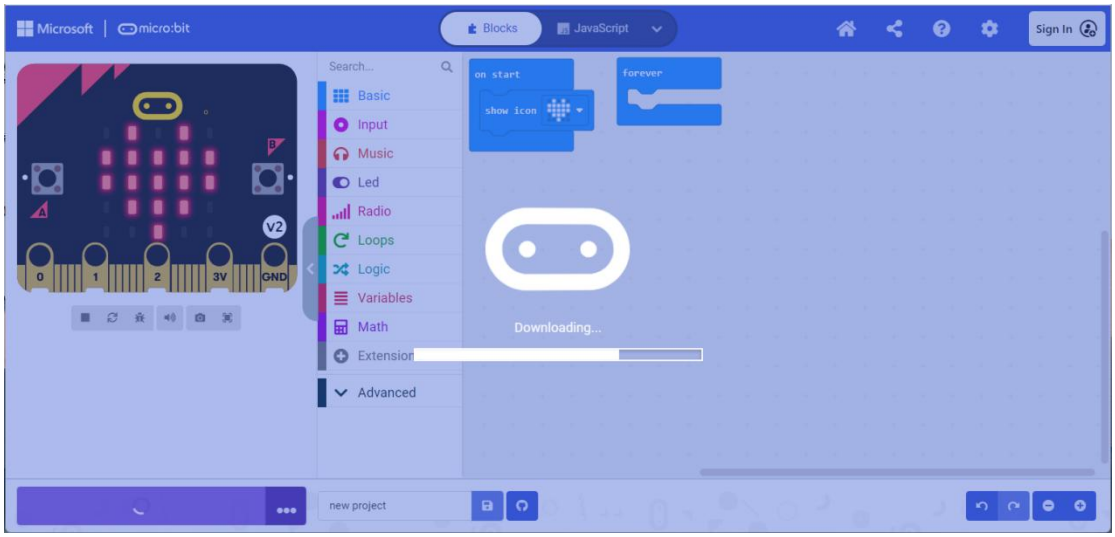
The connection is successful, click"**Done**"：



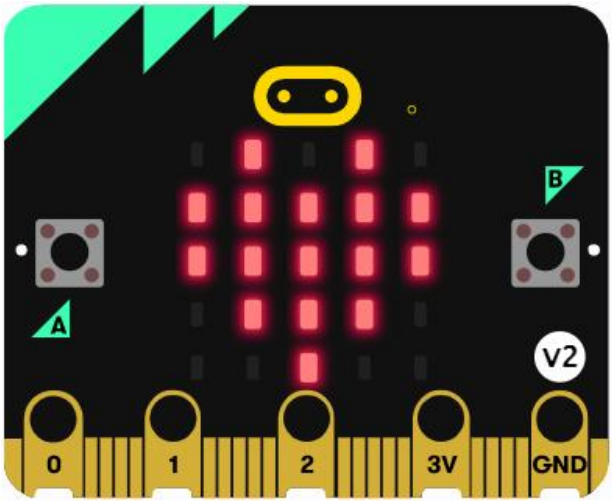
16

Click the "**Download**" button, you can flash the code to the Micro:bit with WebUSB.





After uploading the code, the dot matrix of the Micro:bit board displays a heart shape:

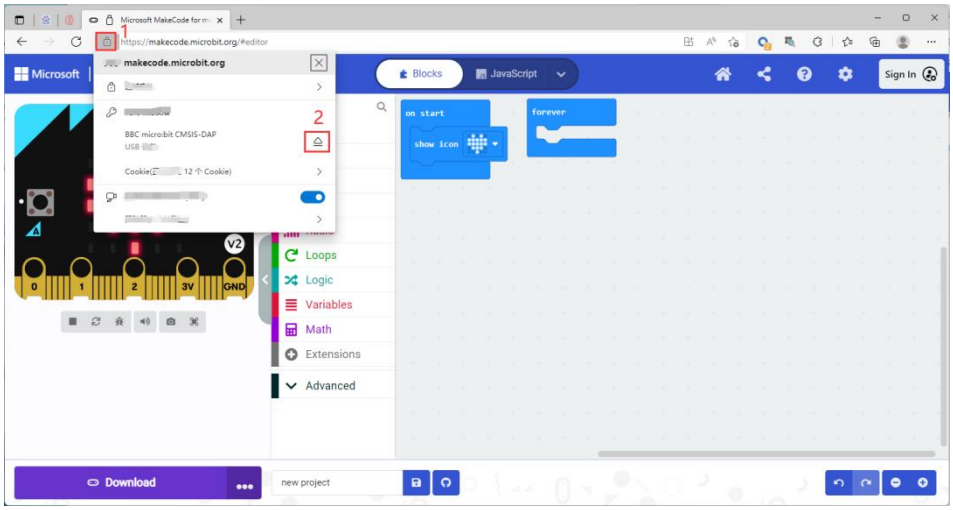


17

3.7 Unpairing Microbit

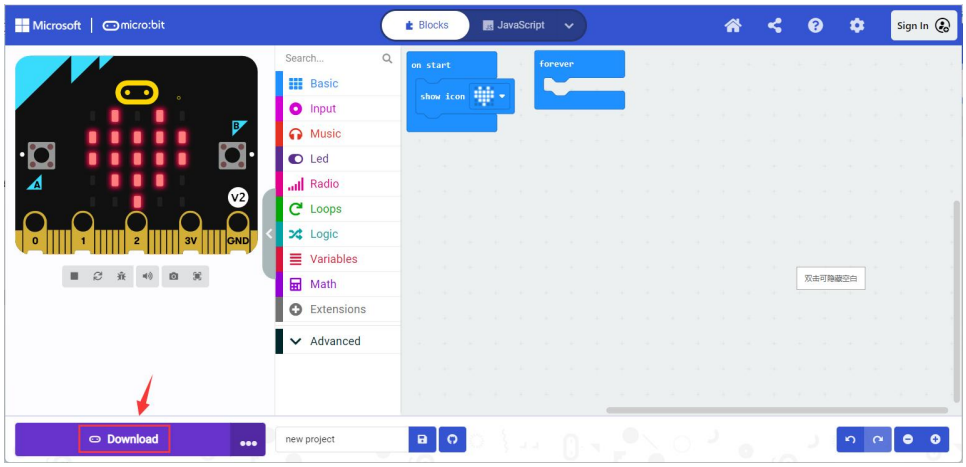
-Click the button to the left of your browser's search box.

-Select the Microbit device you want to disconnect and click the button to the right of it.

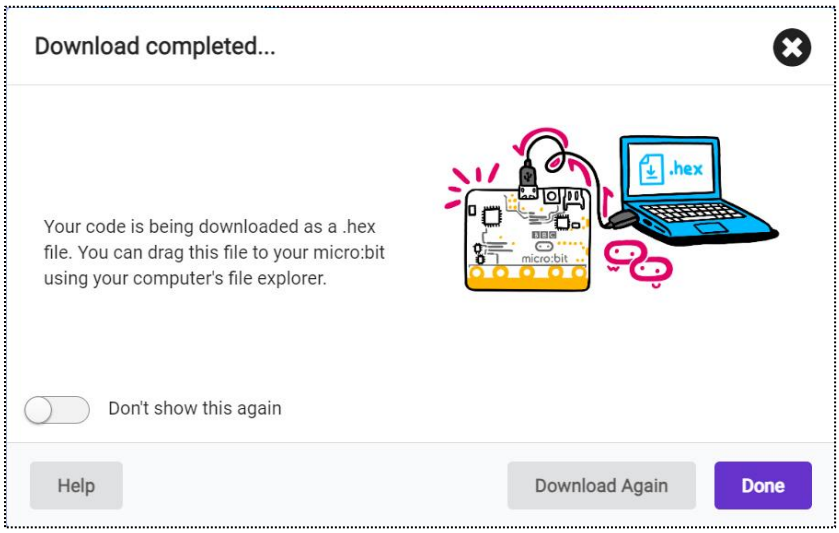


3.8 Upload the HEX file to the Micro:bit

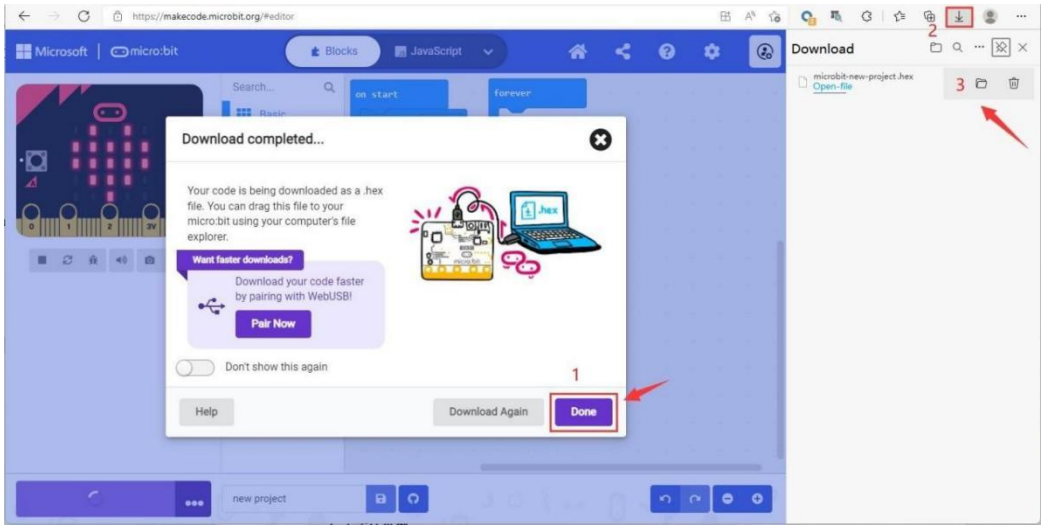
If the Micro:bit is not paired with Microsoft Edge or Google Chrome browser, or if you are using Safari/Firefox/Other browser that may not support WebUSB, directly click the "**Download**" button, the code won't transfer directly to your microbit, it will be downloaded as a .hexfile. Just like click the save icon to save a copy of the hex file to your computer.



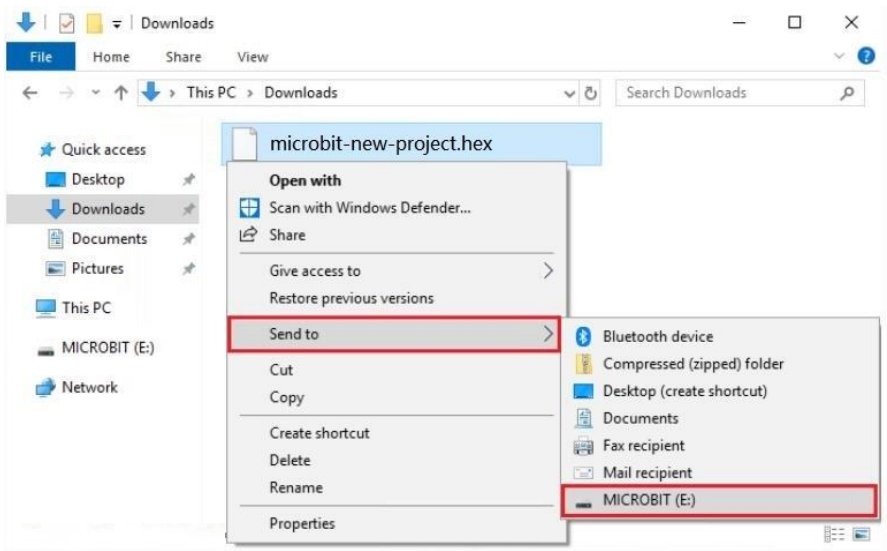
When the following interface appears, click the "⊗" button and click "**Done**":



Find the downloaded hex file in the default save path of browser.

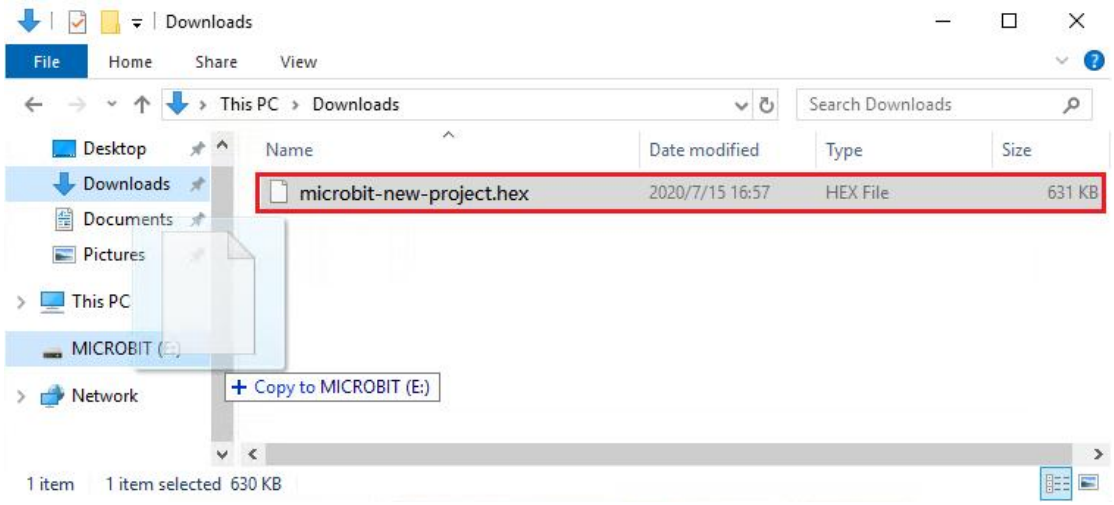


Then select the downloaded hex file，right click the mouse and click "Send to"，then you can send the hex file to your Microbit:

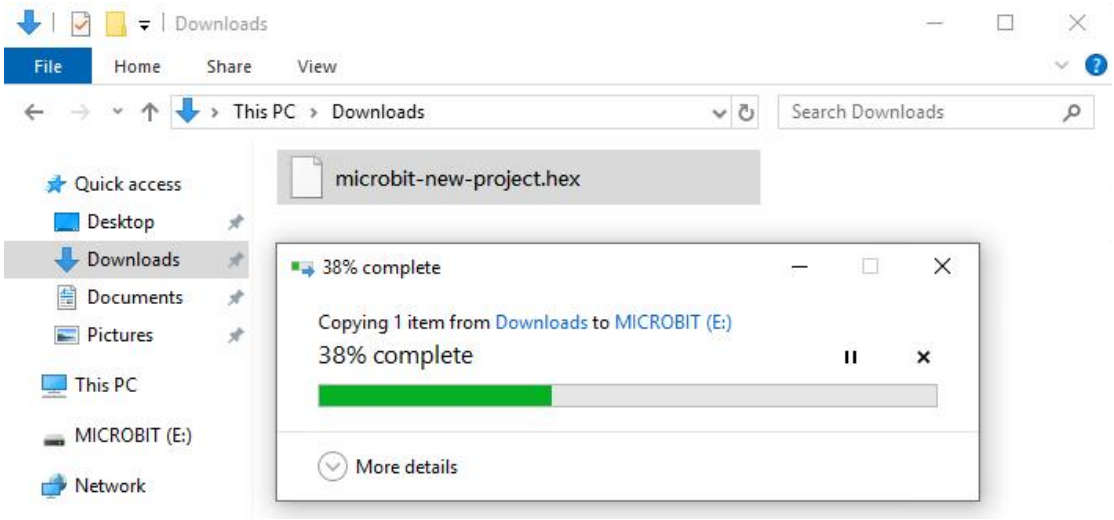


19

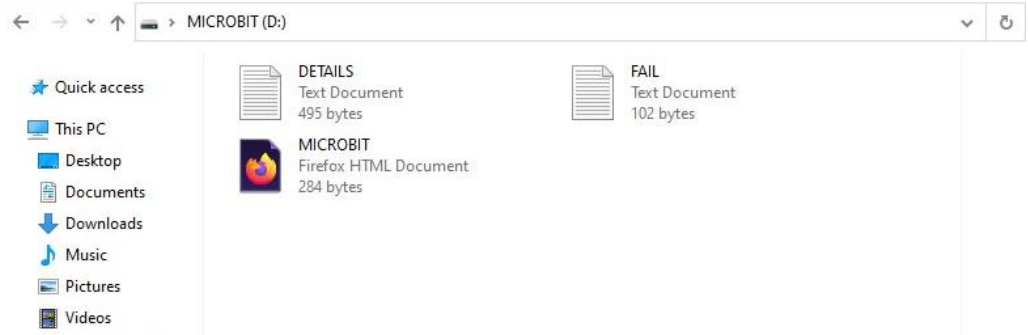
If you do not use the "**Send to**" button, you can directly drag the hex file to the Micro:bit:



The following interface indicates that the code is being uploaded, at the same time, the yellow LED on the back of the Microbit will also flash rapidly until the code upload is complete.



After the code upload is complete, the Micro Bit will disconnect and reconnect. If you look at the contents of the MICROBIT drive, you will not see the .hex file listed, this is normal, but your hex file will start automatically.



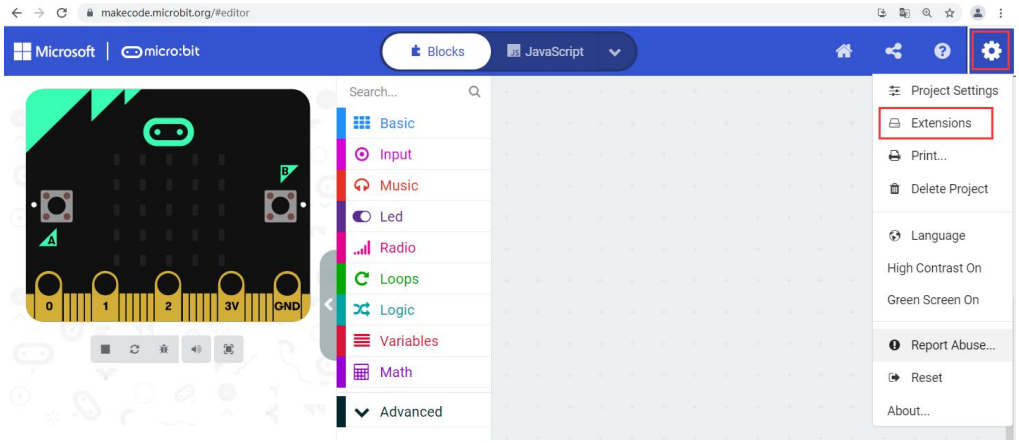
3.9 Learn the basic syntax of Makecode

The Micro:bit platform provides official MakeCode API and device usage documents for your reference.
To use APIs: https://makecode.microbit.org/reference
To use device: https://makecode.microbit.org/device
Logic and data types: https://makecode.microbit.org/blocks

4. Add Extension

4.1 Add mCar extension

We have developed an extension for mCar, which makes it easier for us to use MakeCode to program mCar. The steps to add the extension to the Makecode are as follows:
Click the **Settings** button in the upper right corner of the interface and click "**Extensions**"

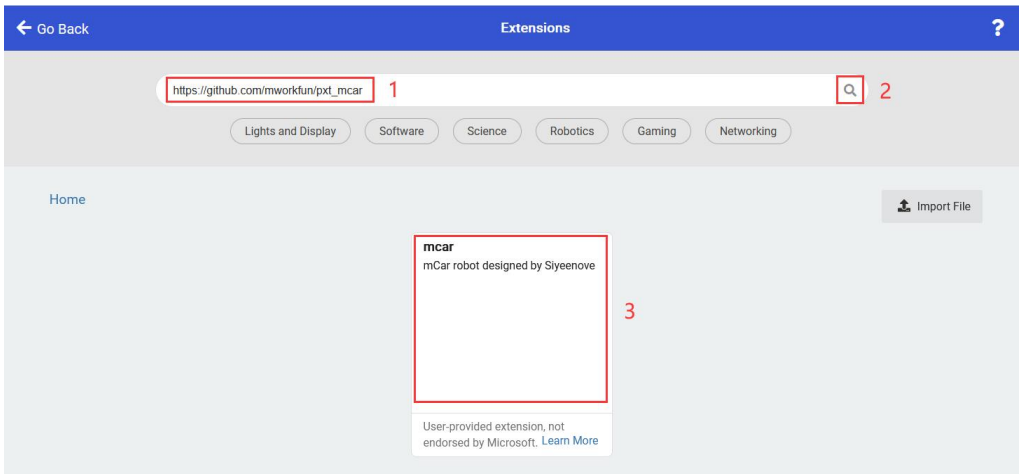


21

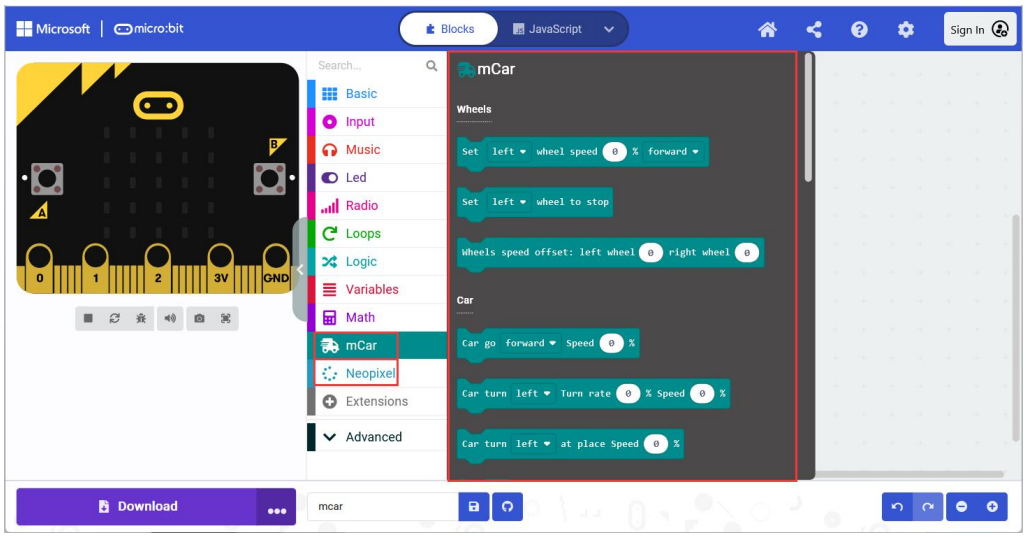
Extension link for mCar：**https://github.com/siyeenove/pxt_mcar**

Copy the above link into the search box on the extension page and click the search button on the right.

Click on the extension named **mcar** in the search results.



After a few seconds the page will jump to the Makecode main interface, and you will see the added **mCar** extension in the toolbox list.

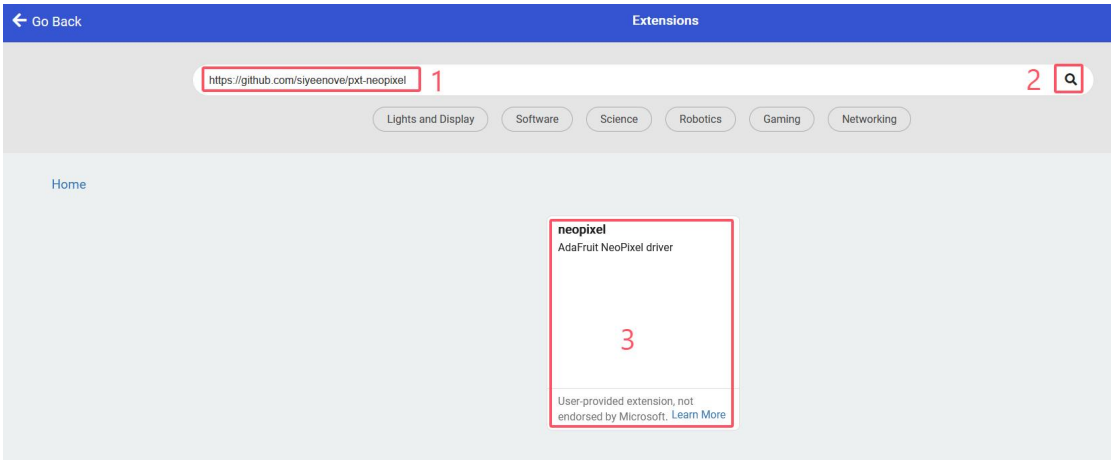


4.2 Add Neopixel extension

Because some other Neopixel expansion is not compatible with bluetooth, so we developed a Neopixel expansion, link:

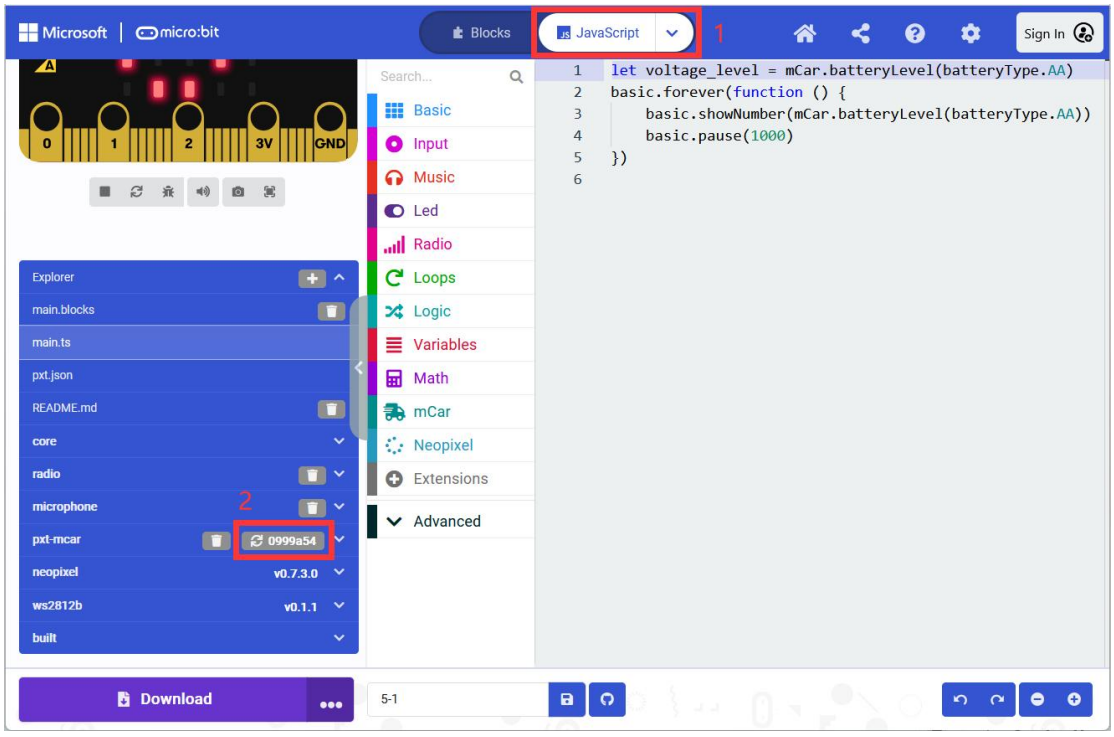
https://github.com/siyeenove/pxt-neopixel

22



4.3 Refresh mCar extension

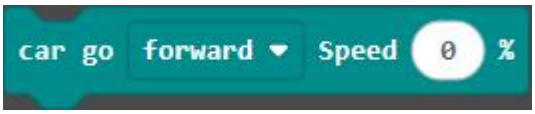
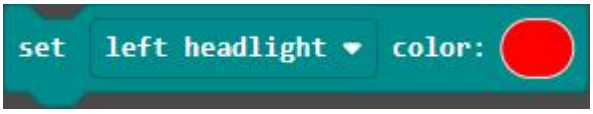
Open the project with the mCar extension added, and switch to the JavaScript programming interface to refresh the extension:



After refreshing, switch back to the "Blocks" interface.

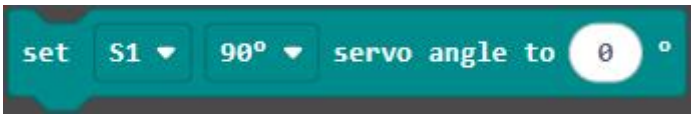
4.3 Parsing of mCar extension statement

All Makecode statements based on mCar are integrated in the mCar extension package. The statement analysis is as follows:

Wheels	 Set the mCar's wheels to move forward or backward, and at what speed.
	 Set the speed and direction of the left and right wheels of the mCar.
	 Set the mCar wheels to stop turning.
	 Adjust the speed of the left and right wheels of mCar and save them permanently inside mCar. This statement can be used to adjust the speed of mCar when it cannot go in a straight line.
Car	 Control the speed at which mCar moves forward or backward.
	 Controls the turning rate and speed of the mCar to turn left or right. The greater the turning rate, the smaller the turning angle.
	 Control the speed at which mCar turns in place.
	 The mCar wheels stopped turning.
RGB LED Headlights	 Colors from the mCar's headlight color palette.

RGB LED headlights	 By changing the three color values, the mCar's headlights can be lit in different colors.
	 Turn off the mCar's headlights.
Tracking sensor	 Read the values detected by the mCar 3 line tracking module and save them in internal variables.
	 Determine whether the value of the 3-way line tracking sensor is equal to the value in the statement tab. Before using this statement, you need to call the "Get the status value of the tracking sensor" statement once.
	 The value of the 3-way line tracking sensor in the variable. Before using this statement, you need to call the "Get the status value of the tracking sensor" statement once.
	 Determine whether the value of one of the three line tracking sensors is equal to the value in the statement tab. Before using this statement, you need to call the "Get the status value of the tracking sensor" statement once.
Sonar sensor	 Read the distance value of the ultrasonic module.
Infrared sensor	 Infrared receiving loop function, this function has been processing the received infrared data.

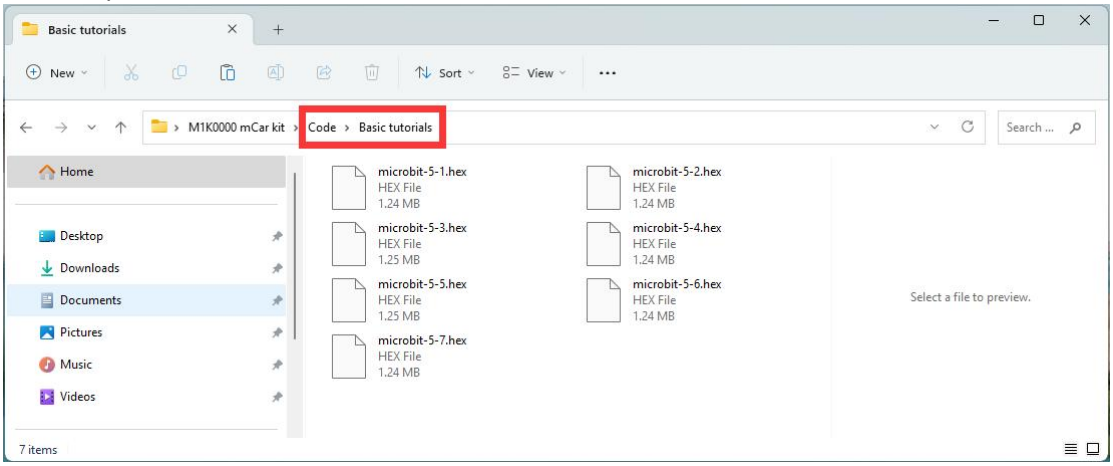
25

Infrared sensor	 Determine whether the data received by the infrared receiver is equal to the key value in the Statement tab.
	 Read the data received by the infrared receiver. Before using this statement, use the "if" statement to determine when the return value is not 0, and then the data will be valid.
Expansion port	 to drive the servo statement, which can drive 90, 180, and 270 degree servos; the interface is the S1, S2, and S3 expansion port on the mCar.
	 The driving statement of 360-degree servo; the interface is S1, S2, S3 expansion port on mCar.
Baterry	 Statement to sets the battery model and reads the battery voltage, returning the battery level between 0 and 100%. **It is recommended to execute this statement once when the machine is turned on.**
APP command	 Send the string to the left or right text display box of SIYEENOVE APP.
	 Determine whether the data received by the Bluetooth of the Micro:bit is the key value of SIYEENOVE APP.
Other	 Read the version number of mCar and return a string value.
Neopixel	 Use the P8 pin of micro:bit to drive the two WS2812 RGB lights at the bottom of mCar.
	 The two WS2812 RGB lights on the bottom of the mCar light up the

Neopixel	colors in the tabs.
	 Set the brightness of the two WS2812 RGB lights at the bottom of the mCar.
	 Turn off all WS2812 RGB lights.
	.....

5. Basic Tutorials

The sample codes for the basic tutorials are all saved in the "Code -> Basic tutorials" file:



Reminder!!!

You don't need to add the mCar extension if you import and use the code we provided. ---> Recommend!

If you create a new project and drag and drop blocks to build code, you need to manually add the mCar extension, please refer to sections 3.1 and 4.1.

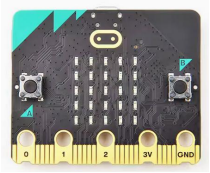
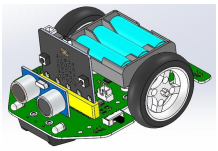
27

5.1 Battery Fuel Gauge

Goal

Set the battery type and read the battery voltage level.

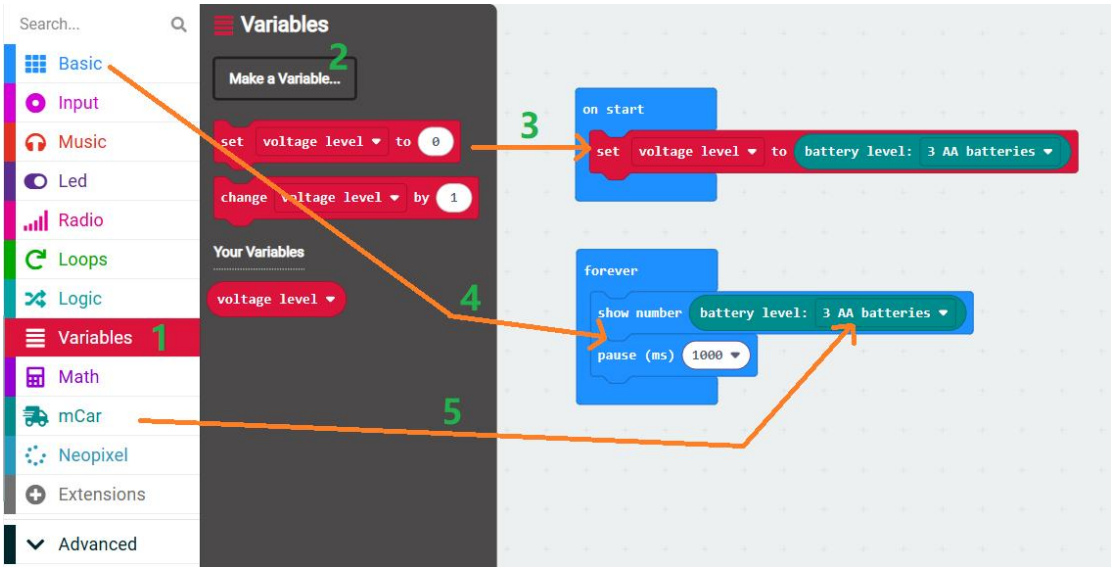
Things you need：

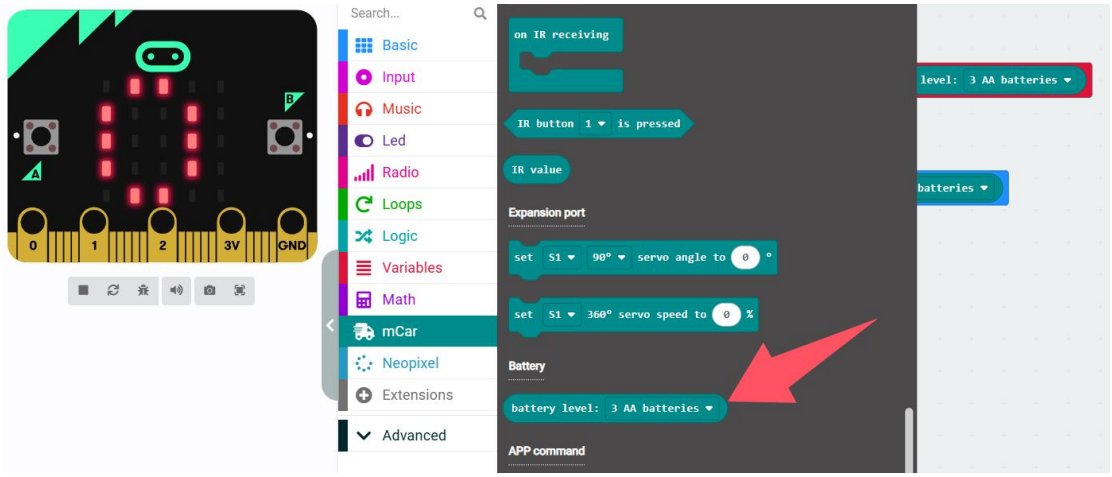
Computer	Micro:bit v2.x.x	Micro USB Cable	mCar
			

Programming

Import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, you need to add the mcar extension.

1. Click "**Variables**">"**Make a Variables**">New variable name: **voltage level.**
2. Drag "**set voltage level to 0"** and stitch it to **"on start".**
3. Click "**mCar**" and drag "**battery level: 3 AA batteries**" behind the voltage level.
4. Click "**Basic**" and drag "**show number**" to stitch it to **"forever".**
5. Click "**mCar**" and drag "**battery level: 3 AA batteries**" behind "**show number**".
6. Click "**Basic**" and drag "**puase(ms)100**" below the "**show number**" , change the number 100 to 1000.





• Create a variable.

• Set the battery type in the "on start" block and read the battery voltage level;

• Display the battery voltage level in the "forever" block.

Consequence

The dot matrix of Micro:bot displays the voltage level, with a value range of 0--100%. When the battery voltage level is less than or equal to 20%, the battery indicator light is always on; otherwise, it flashes.

Note

The statement to read the battery voltage level must be executed once in the program to set the battery type, otherwise the battery voltage indicator will not work properly!

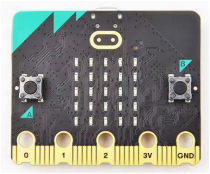
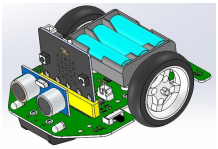
29

5.2 Headlights

Goal

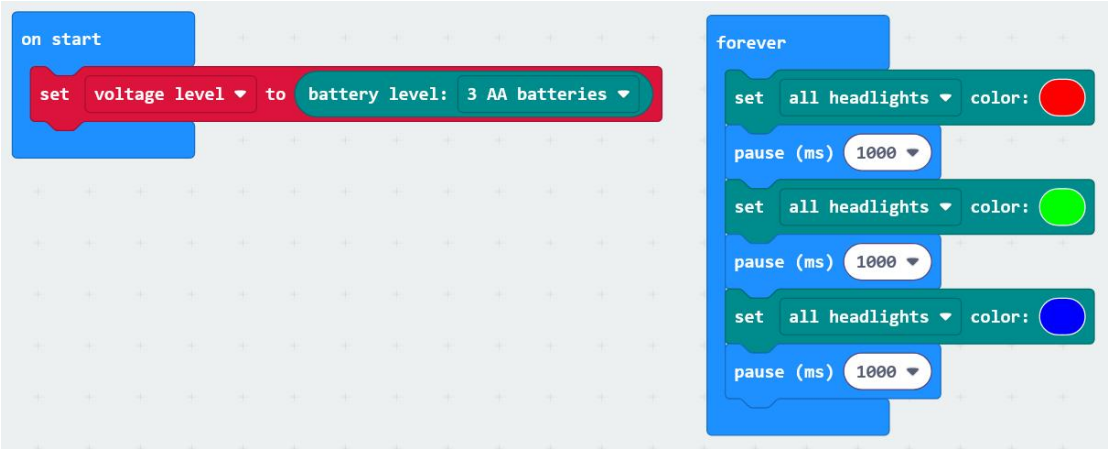
Make the mCar's headlights flash.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



• Set the battery type in the "on start" block.

• Light up both headlights in the "forever" block.

Consequence

The two headlights of the mCar light up in a cycle: red -> delay 1 second -> green -> delay 1 second -> blue -> delay 1 second.

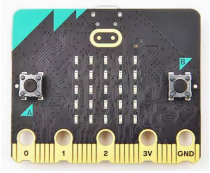
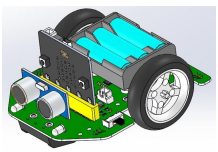
30

5.3 WS2812 RGB Ambient Lights

Goal

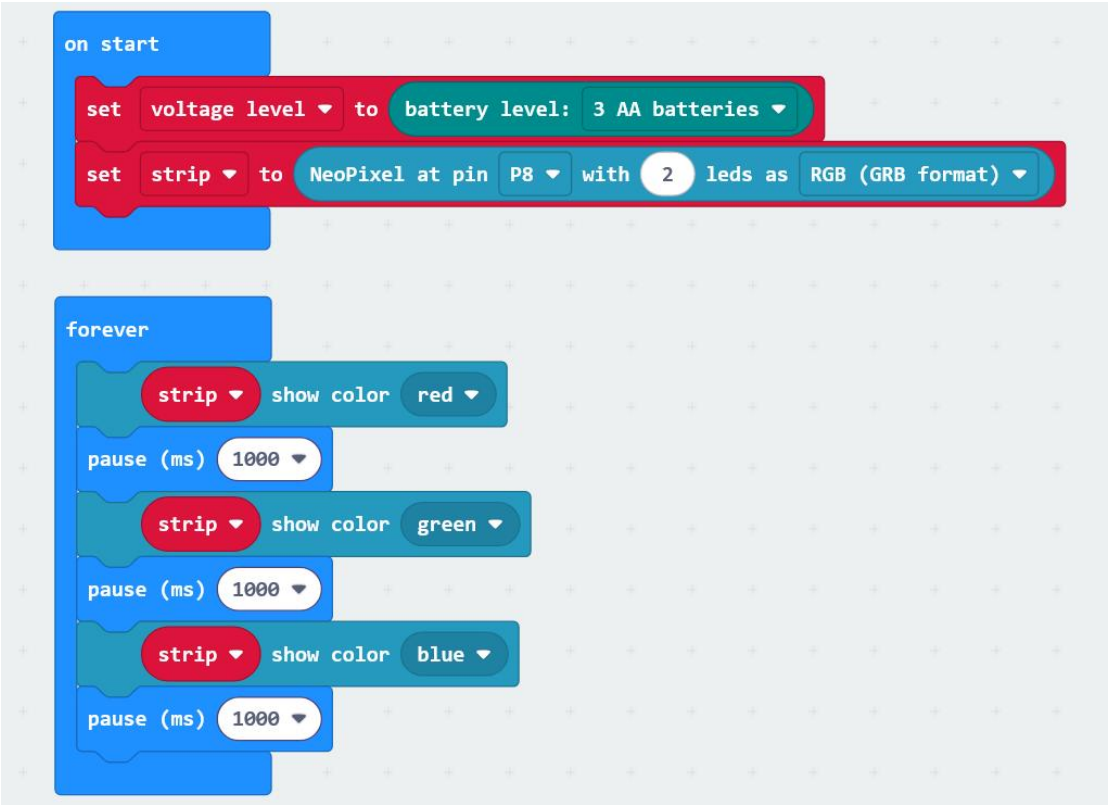
Make the two WS2812 RGB lights on the bottom of the mCar light up.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



• Set the battery type in the "on start" block.

• Light up the two WS2812 RGB ambient lights at the bottom of the mCar on the "forever" block.

31

Consequence

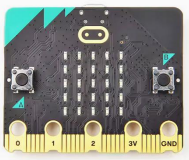
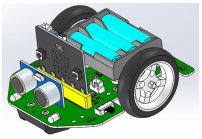
The two WS2812 RGB ambient lights at the bottom of the mCar light up in a cycle: red -> delay 1 second -> green -> delay 1 second -> blue -> delay 1 second.

5.4 Infrared Receiver

Goal

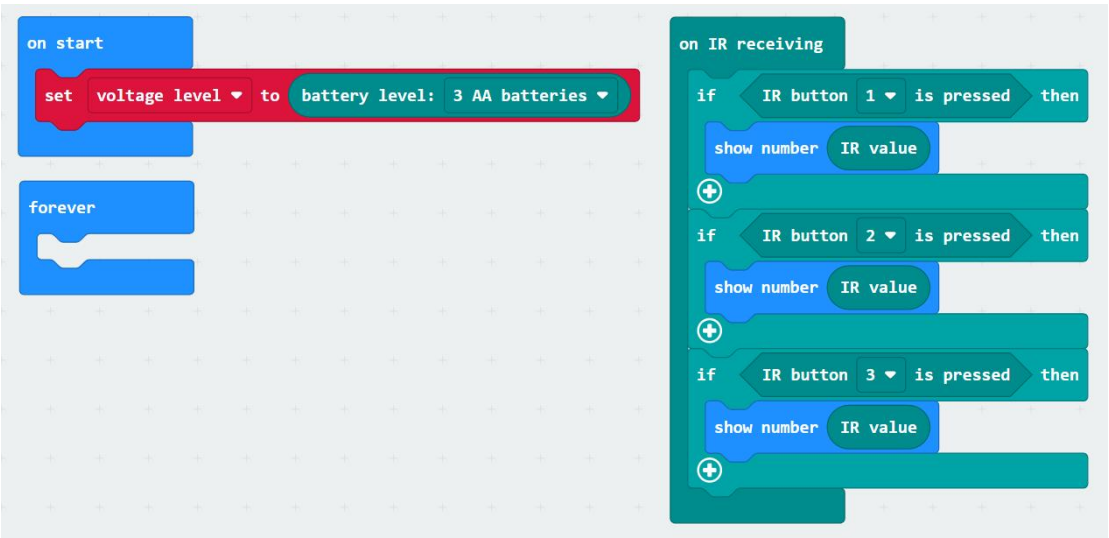
Read the key values of the infrared remote controller through the infrared receiver of mCar.

Things you need:

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	IR Remote
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mCar extension.



32

• Set the battery type in the "on start" block.

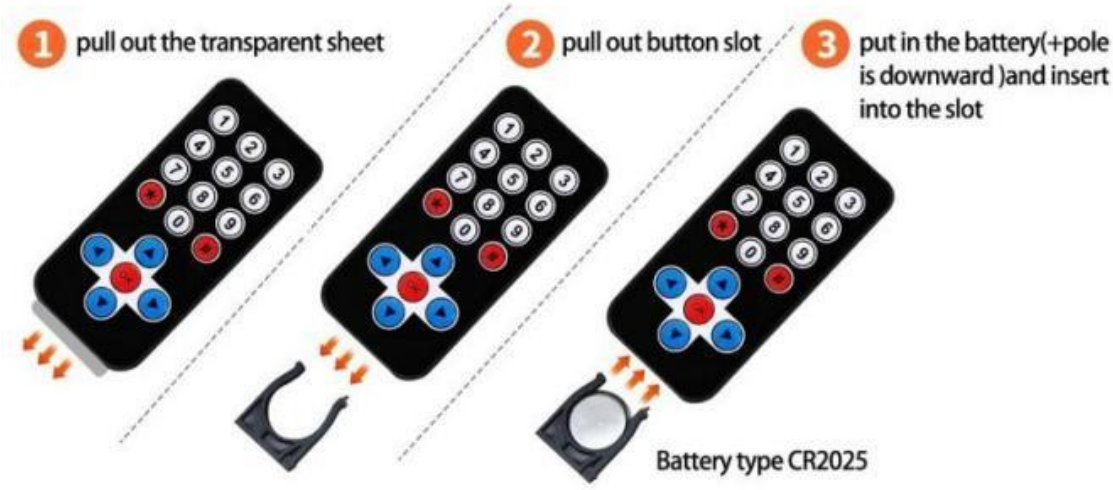
• On the "forever" block, use 3 judgment statements in the infrared data receiving block to check whether the 1, 2 or 3 button is pressed. If a button is pressed, display the button value.

Consequence

Use the infrared remote controller to press the button toward the infrared receiver on the mCar. The dot matrix on the Micro:bit will display the infrared remote controller button value.

Note：

When using the infrared remote control, the plastic battery cover at the bottom must be removed and the batteries must be installed correctly.

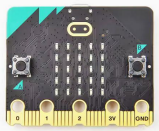
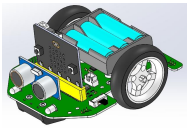
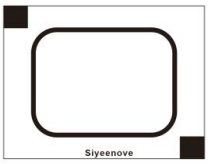


5.5 Infrared tube sensor (line tracking sensor)

Goal

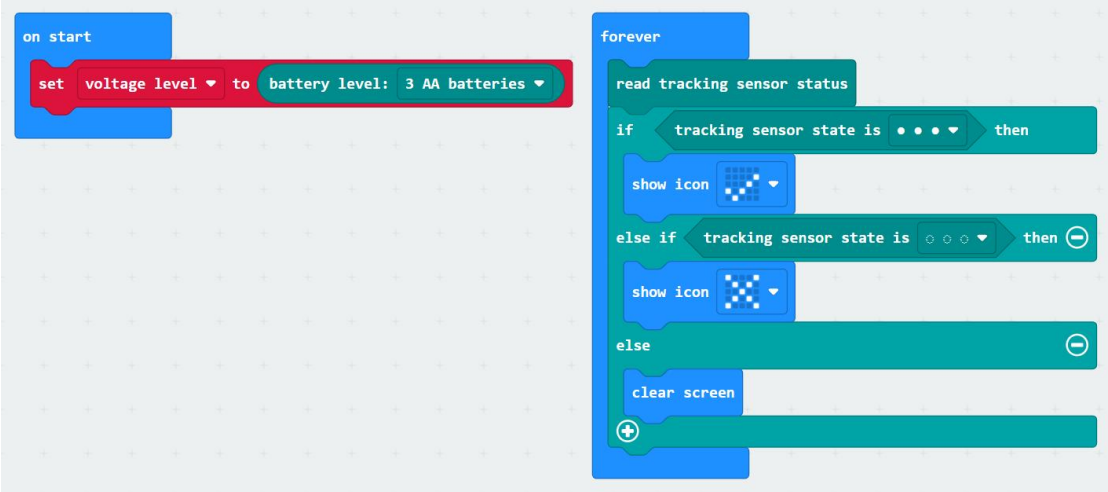
Let mCar's three infrared black and white line sensors identify black and white area.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Map
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mCar extension.



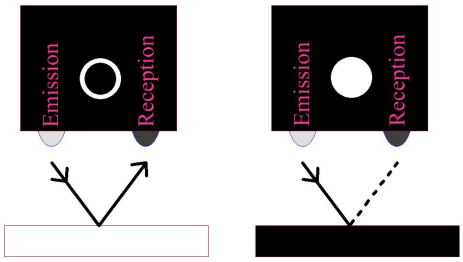
• Set the battery type in the "on start" block.

• Read the values of the three sensors in the "forever" block, and then use three judgment statements to process whether a black line is detected. If all three sensors detect black, "√" is displayed, and if none of them detect black, "×" is displayed.

34

Consequence

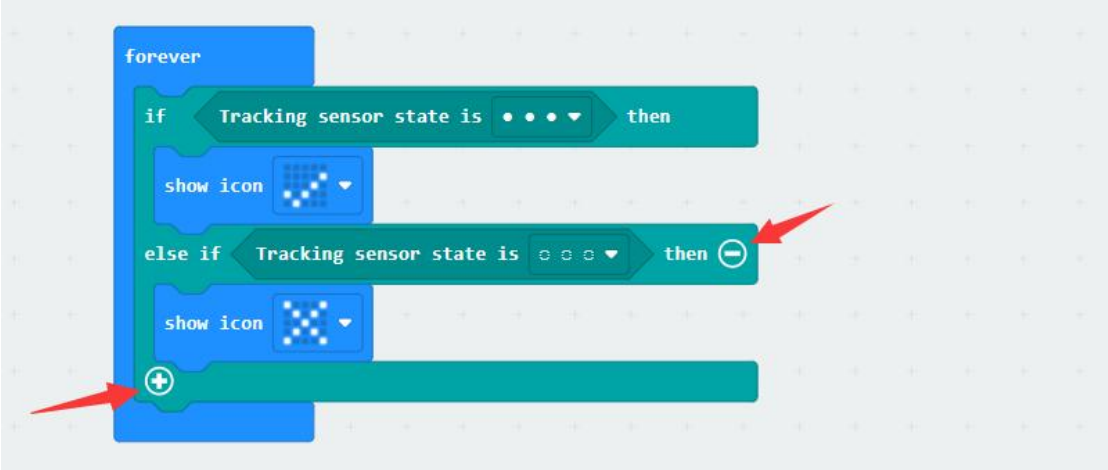
Black objects with rough surfaces have a good absorption effect on infrared rays. When the infrared rays emitted by the sensor hit the black line, the infrared rays are absorbed; when it hits the white object, the infrared signal is reflected back.



After uploading the code, put all three infrared sensors of mCar in the black area of the map, and the Micro:bit dot matrix will display "√"; if all three infrared sensors are placed in the white area of the map, the Micro:bit dot matrix will display "×";

Note

By clicking the "⊕" or "⊖" button, you can switch to an "if...", "if .. else if..." or "if ... else if ... else" statement:



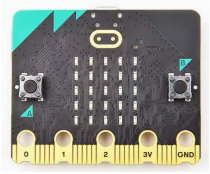
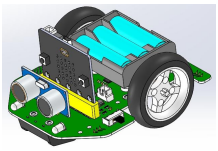
35

5.6 Ultrasonic Sensor

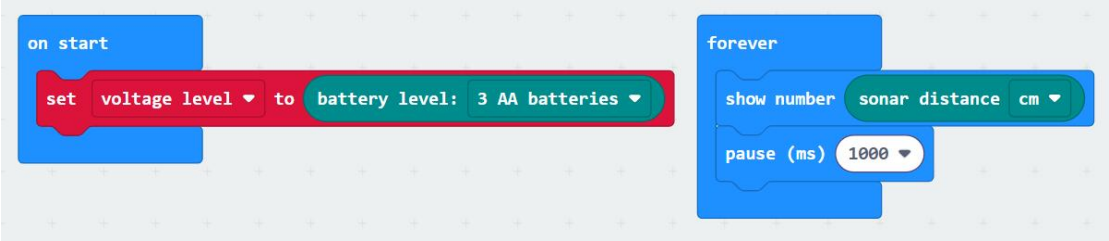
Goal

The ultrasonic module measures the distance to obstacles.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

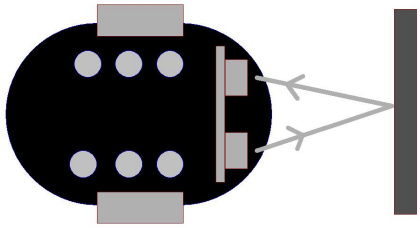
Programming



• Set the battery type in the "on start" block.

• Read the distance measured by the ultrasonic module in the "forever" block and display it in the dot matrix.

Consequence



The dot matrix of Micro:bot displays the measured obstacle distance in centimeters.

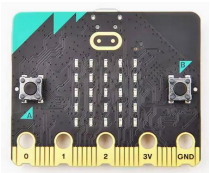
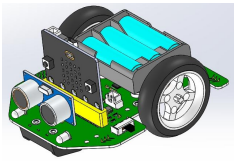
36

5.7 Drive wheels

Goal

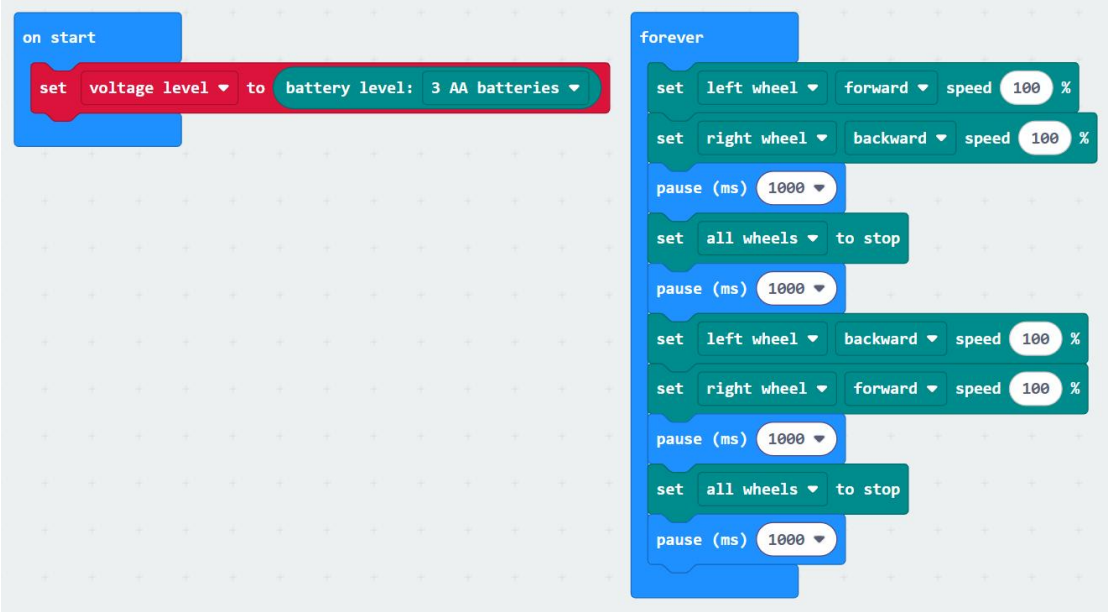
Learn how to drive the mCar's 2 motors

Things you need:

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



• Set the battery type in the "on start" block.

• In the "forever" block, let the two motors rotate in opposite directions for one second and stop for one second.

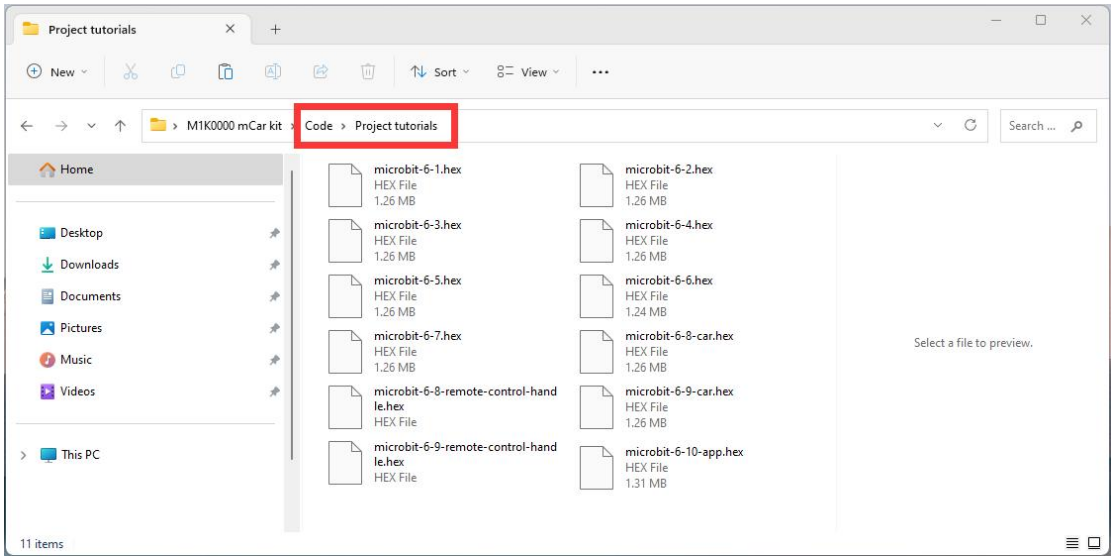
Consequence

mCar rotates left and right on the spot.

37

6. Play with mCar

The sample codes for the project tutorials are all saved in the "Code -> Project tutorials" file:



Reminder!!!

You don't need to add the mCar extension if you import and use the code we provided. ---> Recommend!

If you create a new project to generate code, you need to manually add the mCar extension, please refer to sections 3.1 and 4.1.

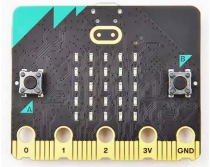
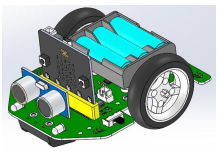
6.1 Evenly accelerate and go straight

Goal

If the motor speed of mCar is very fast, the front universal wheel will leave the ground at the moment of starting.

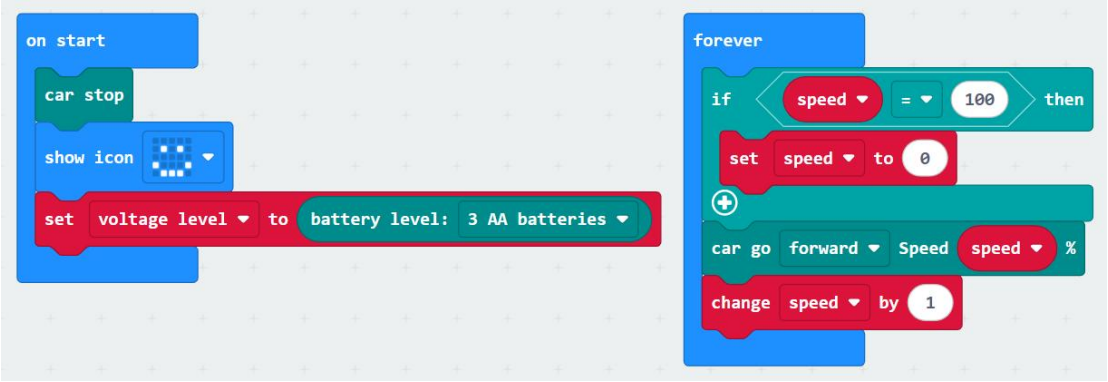
In this case, we let mCar accelerate evenly and complete a smooth start.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



• In the "on start" block, stop the car, add a smiley face icon, and then set the battery type;

• Insert the car forward speed block in the "forever" block, set the speed value to speed, and then increase the speed by 1.

• And if the speed is equal to 100, the speed is already the maximum value, set the speed to 0, and start again.

39

Consequence

The car starts at a constant speed, stops at a certain speed and then starts again at a constant speed, so that the front wheels will not leave the ground due to excessive speed.

Note

If the mCar cannot go straight, you can add a motor speed control statement in the "on start" block. The parameters of this statement will be permanently saved in the mCar.

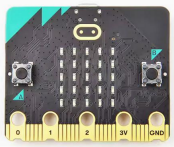
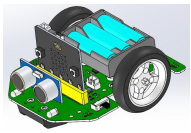
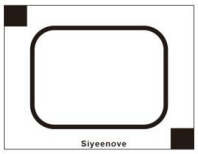
This language block adjust the speed of the left and right wheels of mCar and save them permanently inside mCar. This statement can be used to adjust the speed of mCar when it cannot go in a straight line.

6.2 Follow the black line

Goal

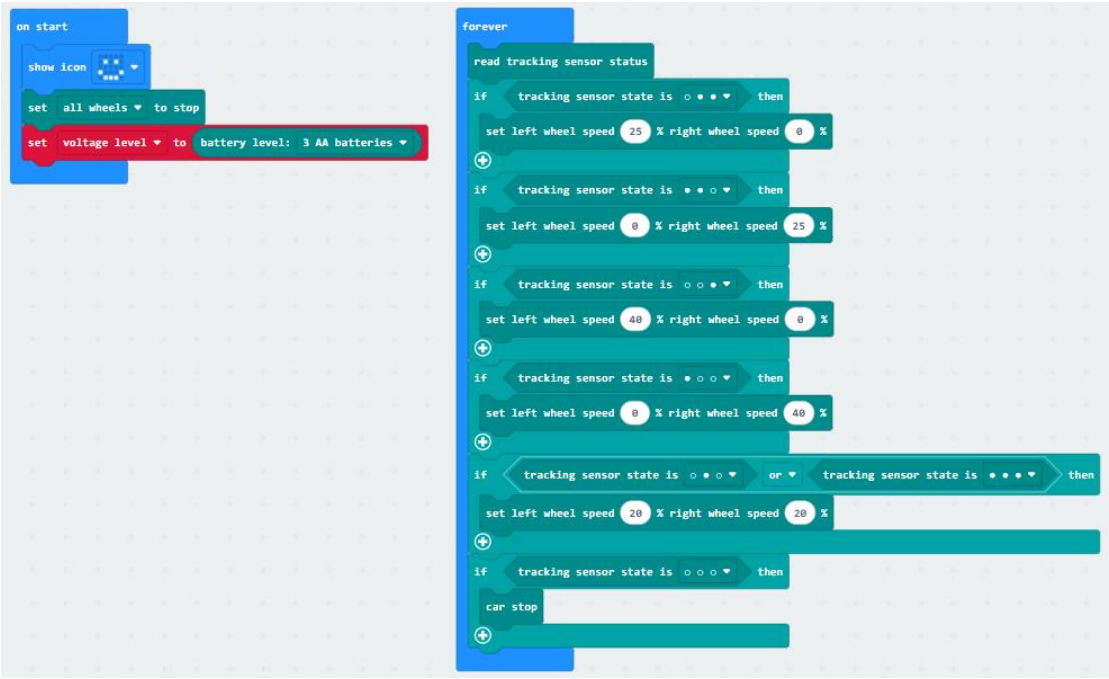
Let mCar run in circles along the black line.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Map
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•Add a smiley face icon to the "on start" block, stop the wheels, and set the battery type.
•In the "forever" block, read the values of the three sensors first, then insert the "if (judgment)" block five times.
•First, determine whether the sensor status is ●○○, that is, the left sensor does not detect the black line, and the other two sensors detect the black line.
•Set the left wheel speed to 25 and the right wheel speed to 0, use the speed difference to

make mcar turn right and then return to the black line.

•Then determine whether the sensor status is ○○● , that is, the right sensor does not detect the black line, and the other two sensors detect the black line.

• Set the right wheel speed to 25 and the left wheel speed to 0, and use the speed difference to make mcar turn left and return to the black line

•First, determine whether the sensor status is ●●○, that is, the right sensor detects the black line, and the other two sensors do not detect the black line.

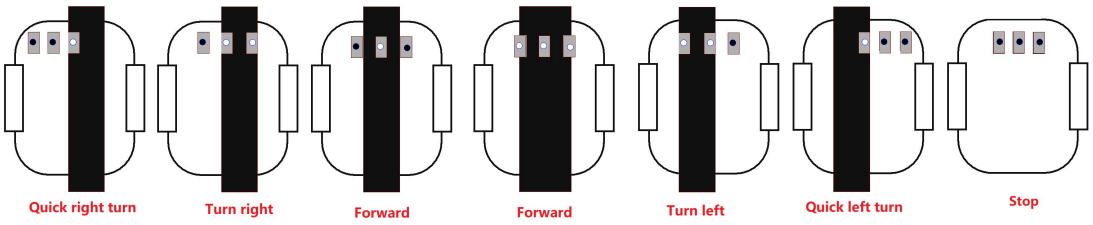
•Set the left wheel speed to 40 and the right wheel speed to 0, and use the speed difference to make mcar turn right and return to the black line.

•Then determine if the sensor state is ○●● , that is, the left sensor detects the black line, and the other two sensors do not detect the black line.

•Set the right wheel speed to 40 and the left wheel speed to 0, and use the speed difference to make mcar turn left and return to the black line.

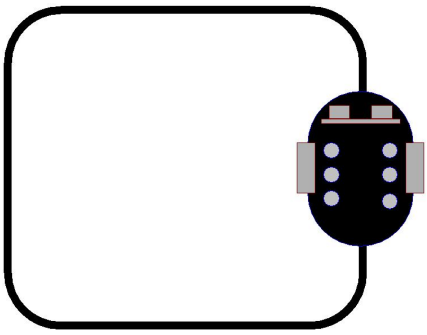
•Then determine if the sensor is in ●○● and ○○○ states, which proves that the car is on the black line, and mcar will go straight at a speed of 20.

•Then determine if the sensor is ●●● , if it is, it proves that the car has moved away from the black line, and the car stops.



●Indicates that no line is detected ○indicates that a black line is detected.

Consequence



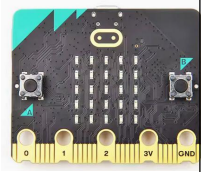
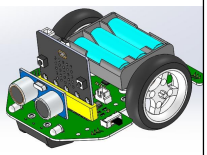
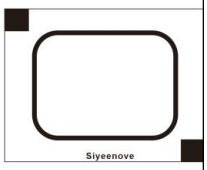
The mCar will move forward at a constant speed following the black line on the map.

6.3 Don't cross the black line

Goal

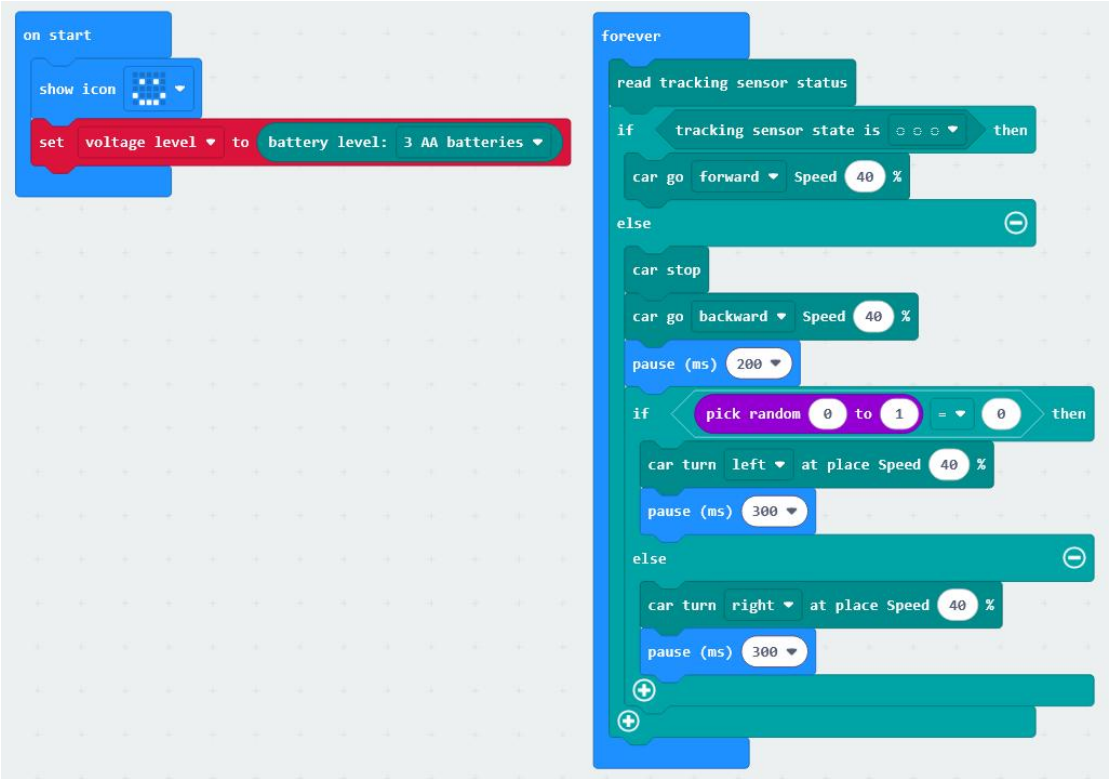
The mCar is restricted to the area within the black line.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Map
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mCar extension.



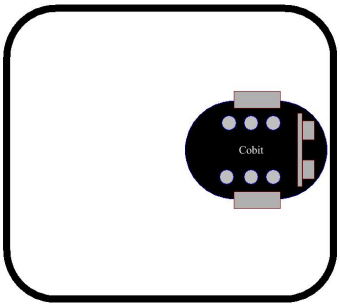
•Add a smiley face icon to the "on start" block and set the battery type.

•First read the values of the three sensors in the "forever" block, and then insert the "if (judgment)" block. If none of the three sensors detects a black line, the mcar moves

43

forward at full speed. Otherwise, the mcar stops immediately, then reverses at a speed of 60, and then randomly turns left or right at a speed of 60.

Consequence



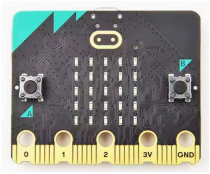
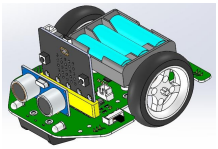
The mCar will always run within the black line and will not cross the black line.

6.4 Follow your hand at a fixed distance

Goal

Let mCar follow your hand at the same distance.

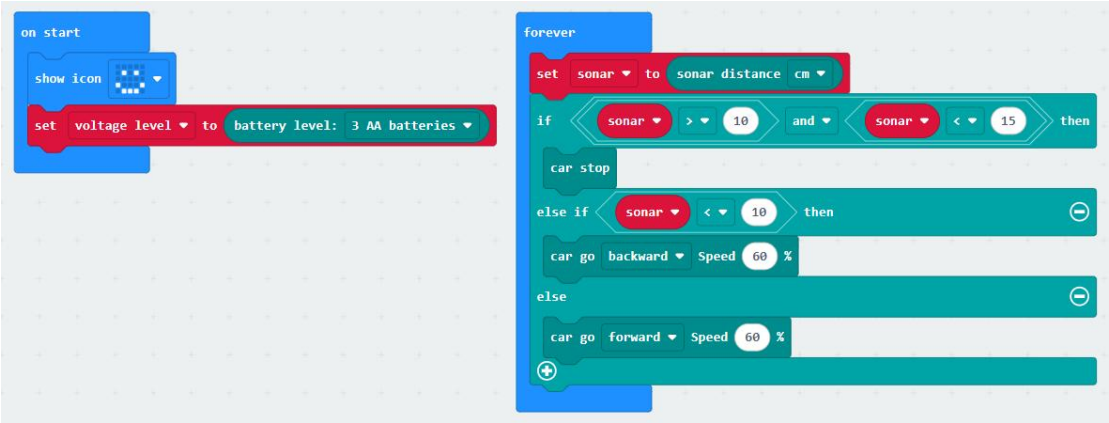
Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

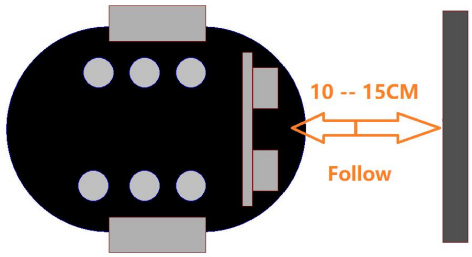
Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.

44



•Add a smiley face icon to the "on start" block and set the battery type.
•Set a sonar variable in the "forever" block to store the CM value returned by the ultrasonic module.
•When the ultrasonic module returns a value greater than 10 and less than 15, the car stops.
•When the return value of the ultrasonic module is less than 10, the robot is too close to the object and will move backwards.
•If neither, mcar is too far from your hand, move forward to catch up with your hand and maintain proper position.

Consequence



When the mcar is too far away from your hand, it will move forward. When it is too close, it will move backward. When the distance is right, it will stop.

Note

Q: When using mCar, if you find that the car is running normally, but cannot run after connecting the ultrasonic sensor.

A: Please check whether the ultrasonic sensor is plugged into the wrong interface. It should be plugged into the sonar interface instead of the I2C interface behind.

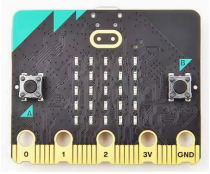
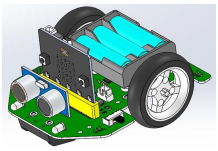
45

6.5 Obstacle Avoidance

Goal

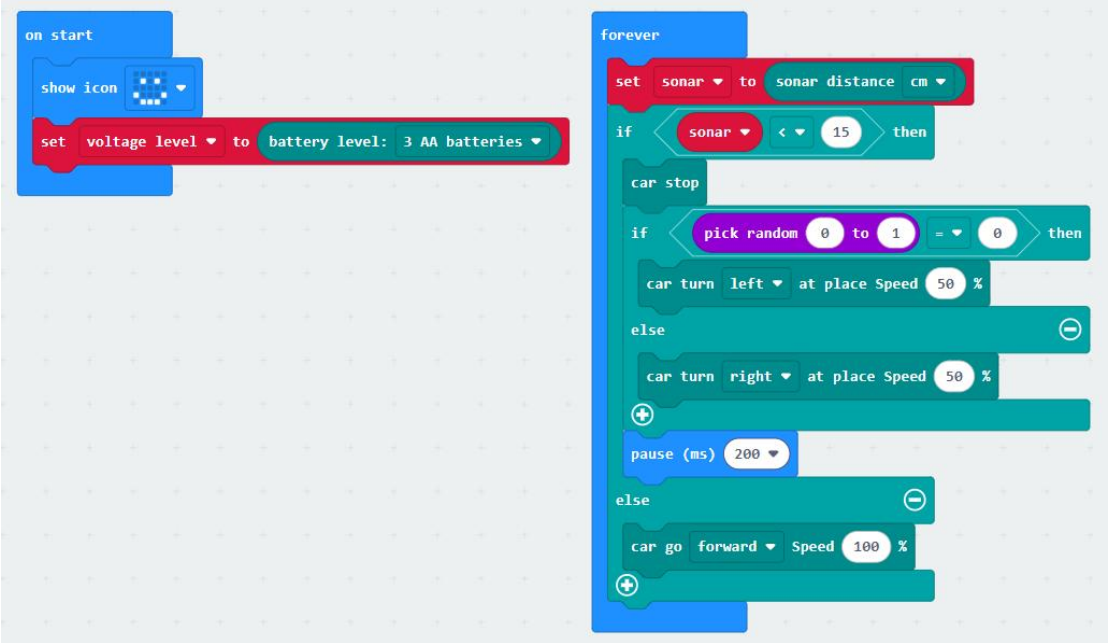
Let mCar achieve autonomous driving and obstacle avoidance.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar
			

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mCar extension.



•Add a smiley face icon to the "on start" block and set the battery type.

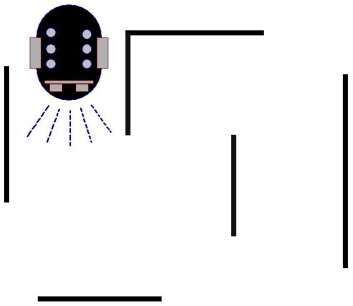
•Set a sonar variable in the "forever" block to save the CM value returned by the ultrasonic module. When the return value of the ultrasonic module is less than 15, it proves that an obstacle has been detected 15CM ahead, and the car

46

randomly turns left or right to avoid the obstacle.

•Else, mcar goes full speed ahead.

Consequence



After the code is uploaded, the mcar car moves forward at full speed. When it detects an obstacle within 15cm, it randomly rotates left or right by an angle and continues to move forward.

Note

Q: When using mCar, if you find that the car is running normally, but cannot run after connecting the ultrasonic sensor.

A: Please check whether the ultrasonic sensor is plugged into the wrong interface. It should be plugged into the sonar interface instead of the I2C interface behind.

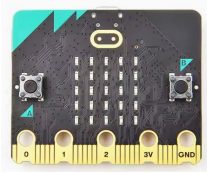
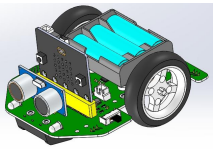
47

6.6 Follow the Light

Goal

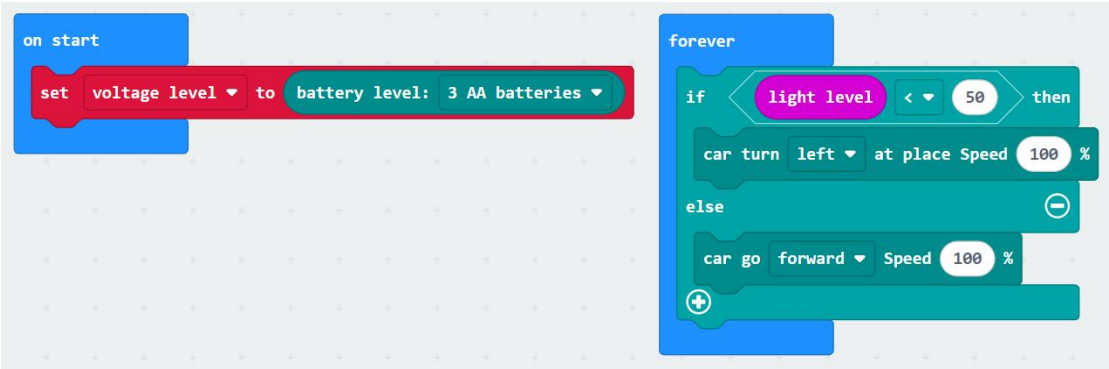
The dot matrix of the Micro:bit can sense the intensity of light. We use this function to allow mCar to automatically find light sources in a dark environment.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	flashlight
				

Programming

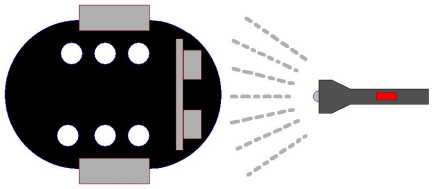
Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mCar extension.



• Set the battery type in the "on start" block.

• In the "forever" block, determine the brightness level. When the brightness level is less than the set threshold, the car turns left at full speed. When the brightness level is greater than the set threshold, the car moves forward at full speed.

Consequence



Shine a flashlight on the dot matrix of the Micro:bit. When the light level is detected to be less than 50, the car rotates in place. When the light level is detected to be greater than 50, the car moves forward at full speed.

Note

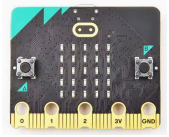
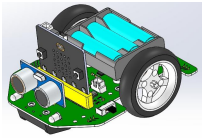
This may not work in brightly lit environments.

6.7 Infrared remote control

Goal

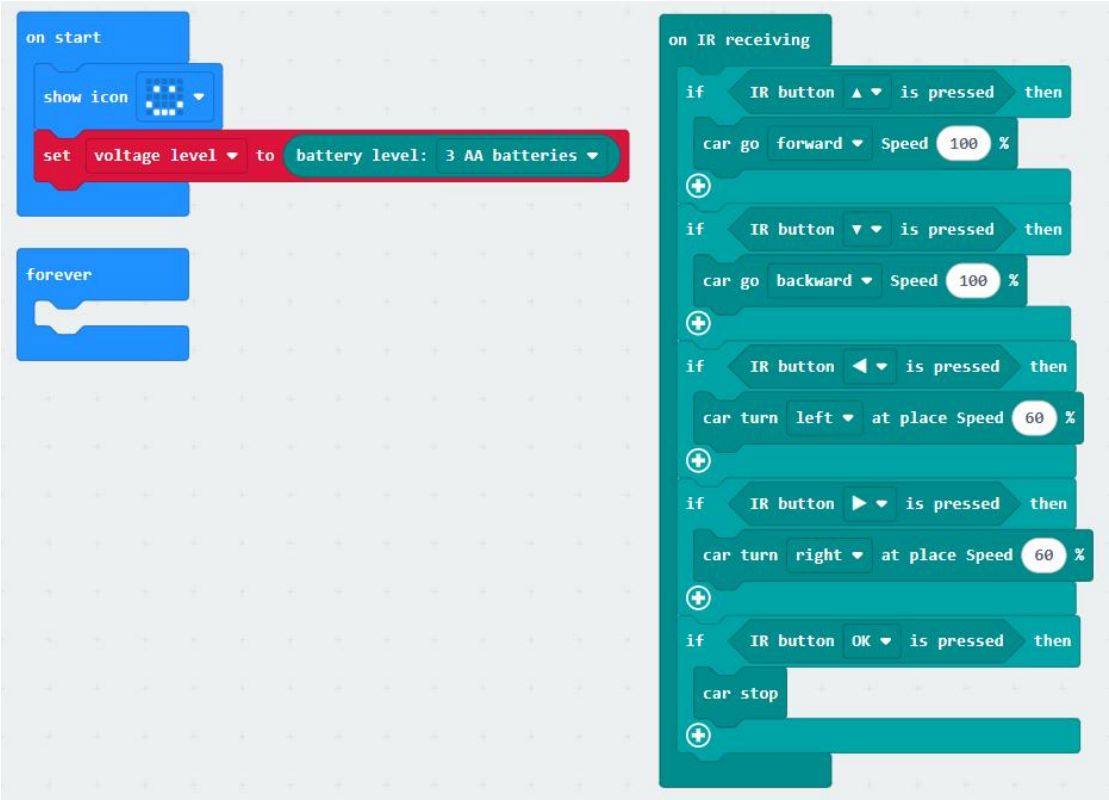
Use the infrared remote control to control the mCar.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Infrared remote control
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•Add a smiley face icon to the "on start" block and set the battery type.

•When mcar receives an infrared signal, it determines which button is pressed; when the ▲ button on the infrared remote control is pressed, the car moves forward at full speed; when the ▼ button on the infrared remote control is pressed, the car reverses at full speed; when the ◀ button on the infrared remote control is pressed, the car turns left at full speed; when the ▶ button on the infrared remote control is pressed, the car turns right at full speed; when the OK button on the infrared remote control is pressed, the car stops immediately.

Consequence

Use the infrared remote control to control the mCar to move forward, backward, turn left, turn right, and stop.

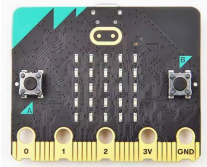
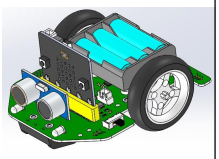
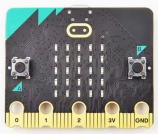
6.8 Control mCar with another Micro:bit

Note: This is an optional project, the kit does not include any Micro:bit, if you have two Micro:bit, you can try it.

Goal

- Use another Micro:bit as a remote controller to wirelessly control mCar.
- Both Micro:bits need to upload programs.

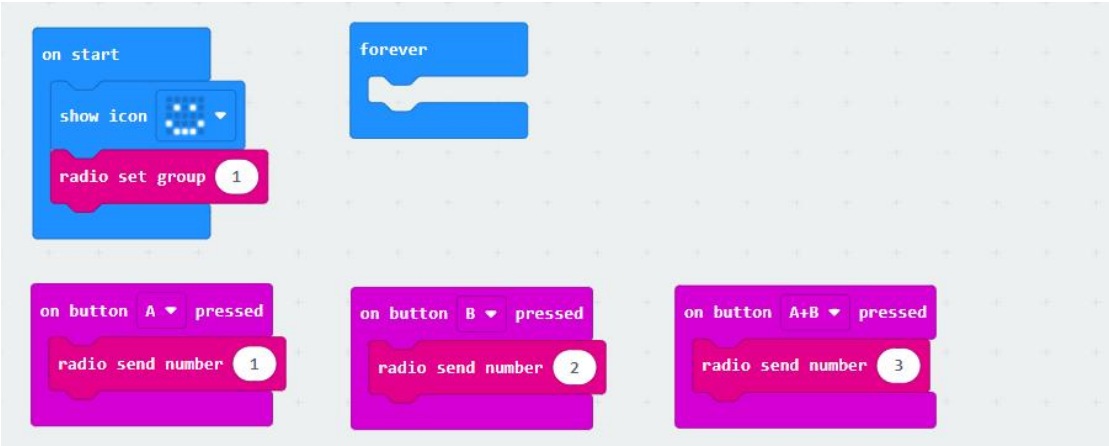
Things you need:

Computer	Micro:bit v	Micro USB cable	mCar	Micro:bit v
				

Programming

-Upload code to the Microbit as a remote controller

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•Add a smiley face icon to the "on start" block and set the radio group to 1;

•When Microbit button A is pressed, the number 1 is wirelessly sent;

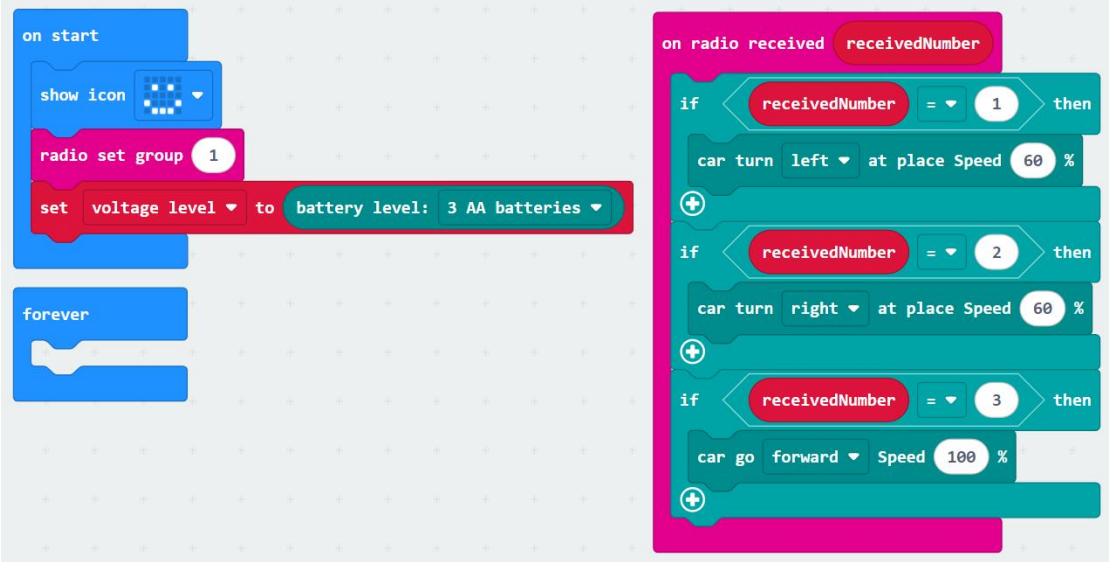
51

•When Microbit button B is pressed, the number 2 is wirelessly sent;

•When Microbit buttons A+B are pressed simultaneously, the number 3 is wirelessly sent;

-Upload code to mCar's Microbit

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•Add a smiley face icon to the "on start" block, set the radio group to 1, and set the battery type. ;

•Insert the display icon block into the "on start" block and set the radio group to 1. Make sure it is set to the same group as the remote control, otherwise they will not match.

•Then insert three judgment statements in the "radio received" block to determine whether the radio reception value is 1, 2 or 3 respectively;

•When the number received by the radio is 1, mcar turns left;

•When the number received by the radio is 2, mcar turns right;

•When the number received by the radio is 3, mcar goes straight.

52

Consequence

- Press the A+B buttons on the remote control Micro:bit, and the car will go straight.
- Press the A button on the remote control Micro:bit, and the car will turn left.
- Press the B button on the remote control Micro:bit, and the car will turn right.

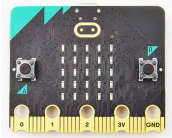
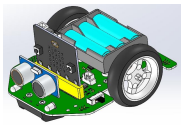
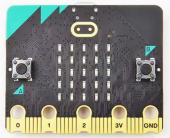
6.9 Accelerometer wireless control

Note: This is an optional project, the kit does not include any Micro:bit, if you have two Micro:bit, you can try it.

Goal

- Use the accelerometer of another Micro:bit to wirelessly control mCar, which can control the 360-degree direction and speed of the car.
- Both Micro:bits need to upload programs.

Things you need：

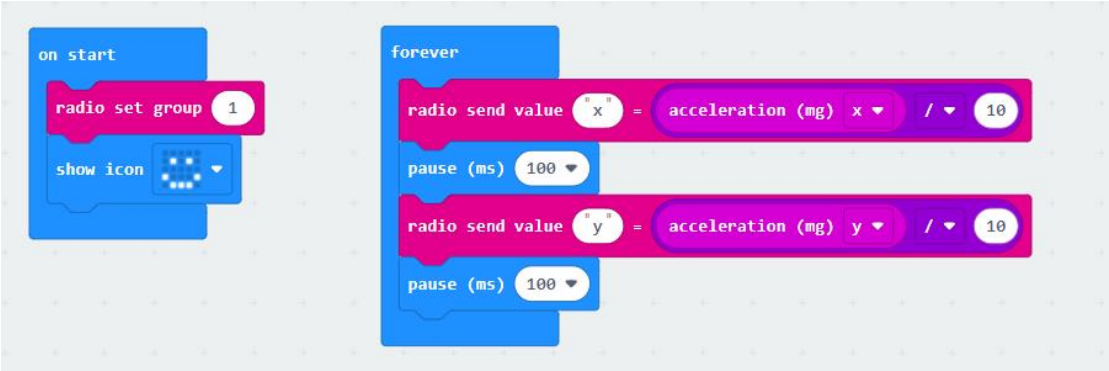
Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Another Micro:bit v2.x.x
				

Programming

-Upload code to the Microbit as a remote controller

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.

53



• Set the radio group to 1 in the "on start" block;

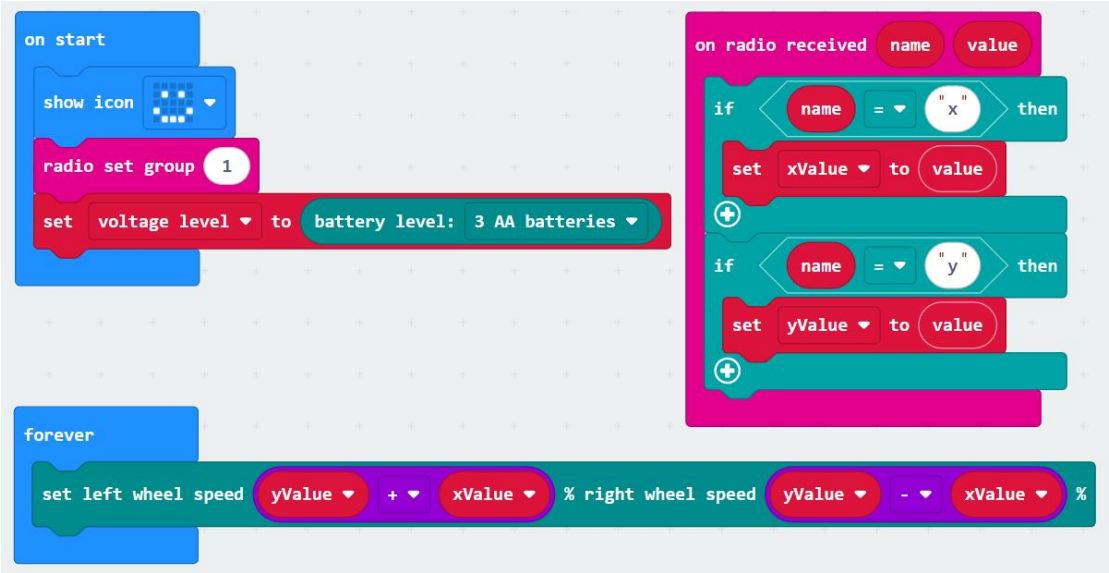
•In the "forever" block, the value of radio send value x is the x-axis acceleration value divided by 10.

•In the "forever" block, the value of radio send value y is the y-axis acceleration value divided by 10.

•Because the acceleration value range is -1024~0~1024, after dividing by 10 it can be approximated as -100~0~100.

-Upload code to mcar's Microbit

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•Add a smiley face to the "on start" block; set the radio group to 1. Make sure it is the same group as the remote control, otherwise they will not match; then set the battery type.

54

•Then insert two judgment statements in the "radio received" block to judge whether the radio received value name is x or y;

•When the name value received by the radio is x, it is the accelerometer X-axis data, and the value is saved to the xValue variable;

•When the name value received by the radio is y, it is the accelerometer Y-axis data, and the value is saved to the yValue variable;

•In the "forever" building block, set the left wheel speed to yValue+xValue and the right wheel speed to yValue-xValue.

Consequence

- Tilt the micro:bit on the control end in different directions to control the mCar to move in different directions.
- Use the tilt angle of the micro:bit on the control end to control the speed of the mCar.

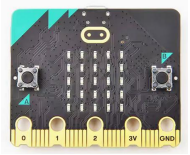
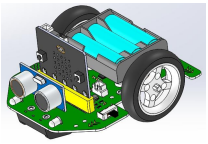
55

6.10 Bluetooth APP control

Goal

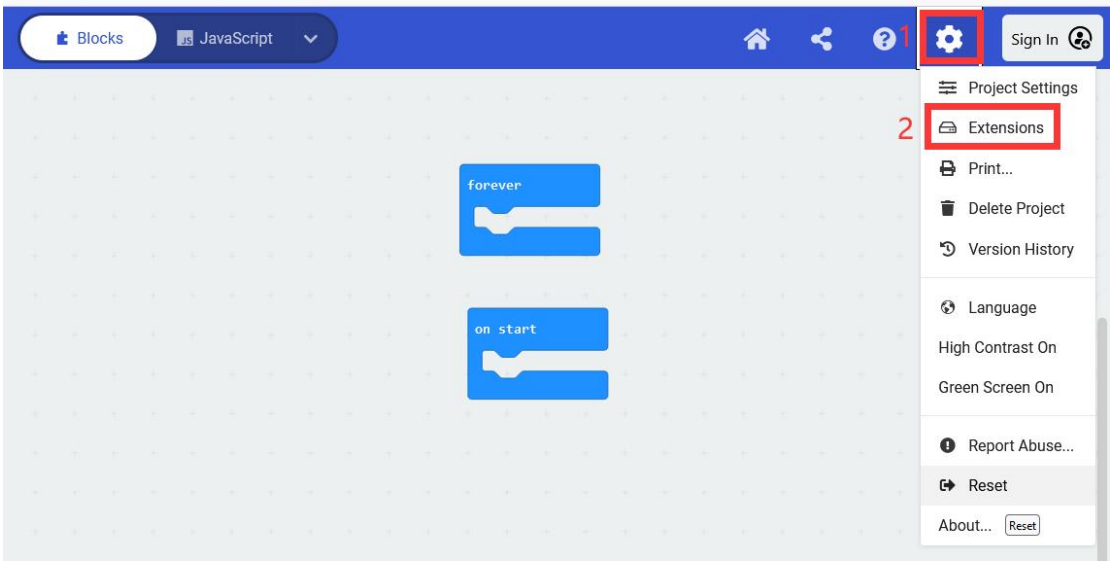
- Use Bluetooth APP of Android phone to control mCar.
- Add Bluetooth extension

Things you need:

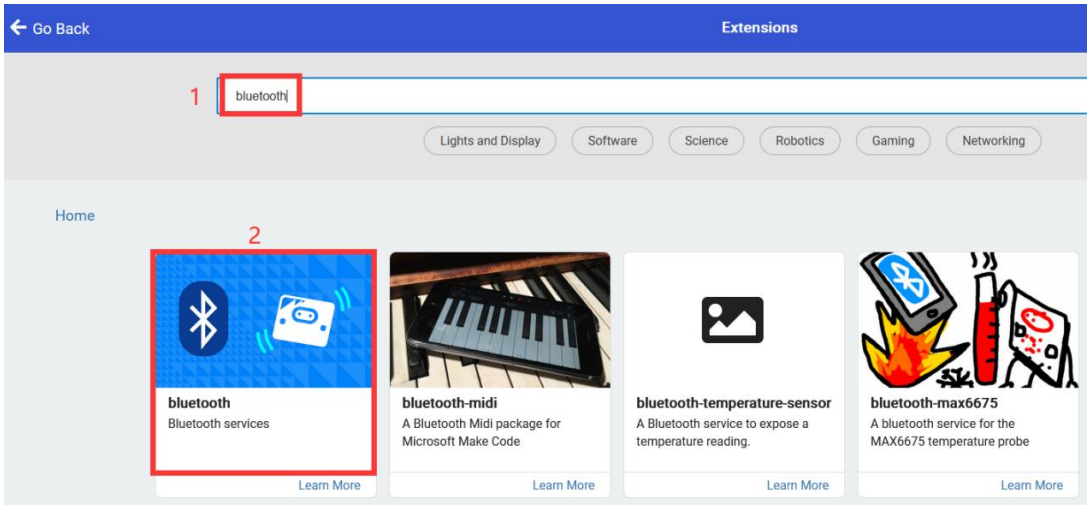
Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Android phone
				

Add Bluetooth expansion in Makecode

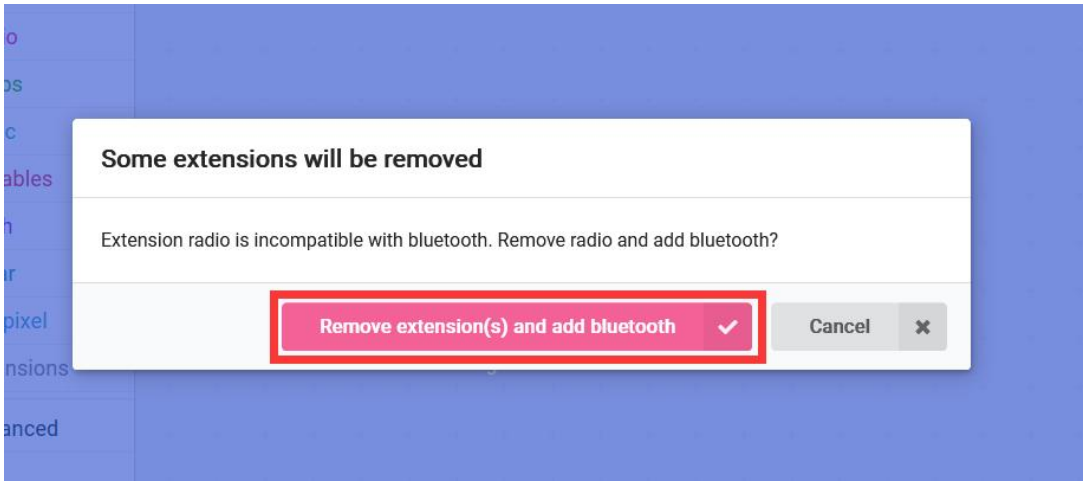
Search and add the Bluetooth extension:



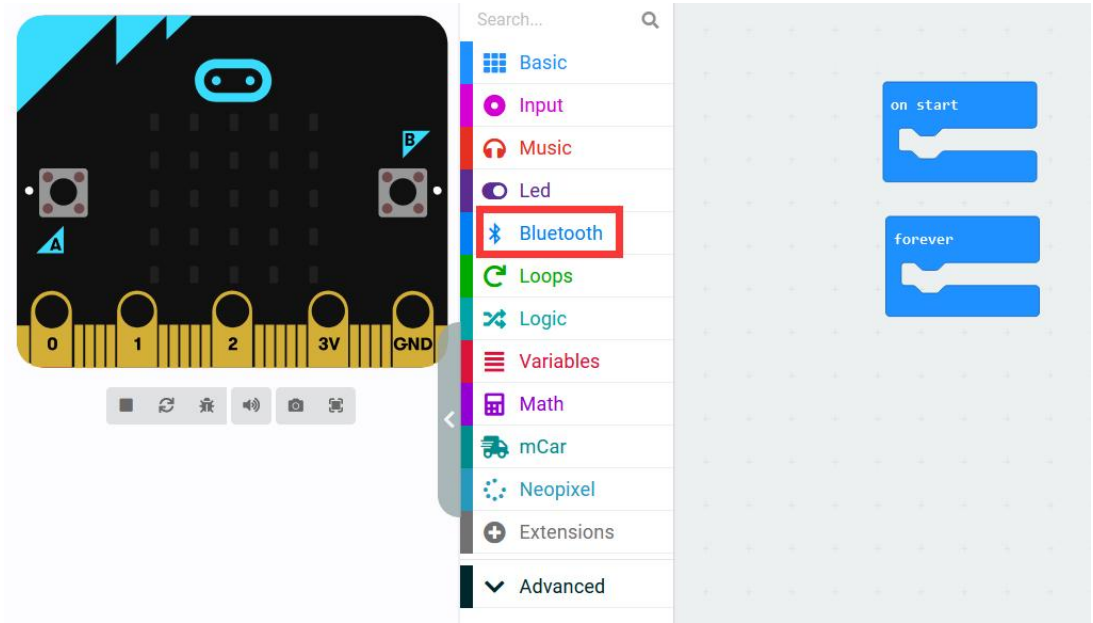
56



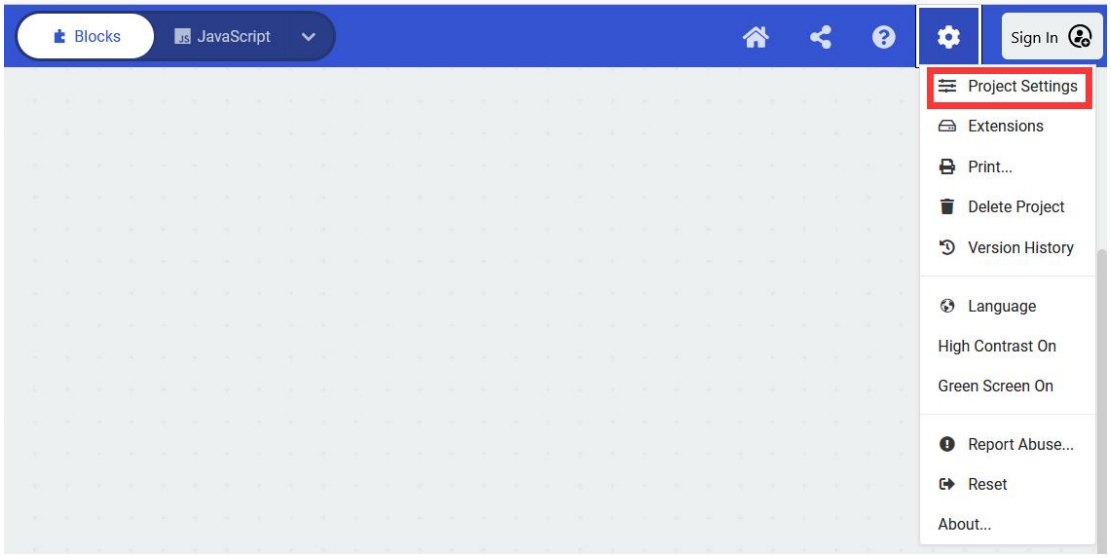
Extension radio is incompatible with bluetooth, Remove radio and add bluetooth.

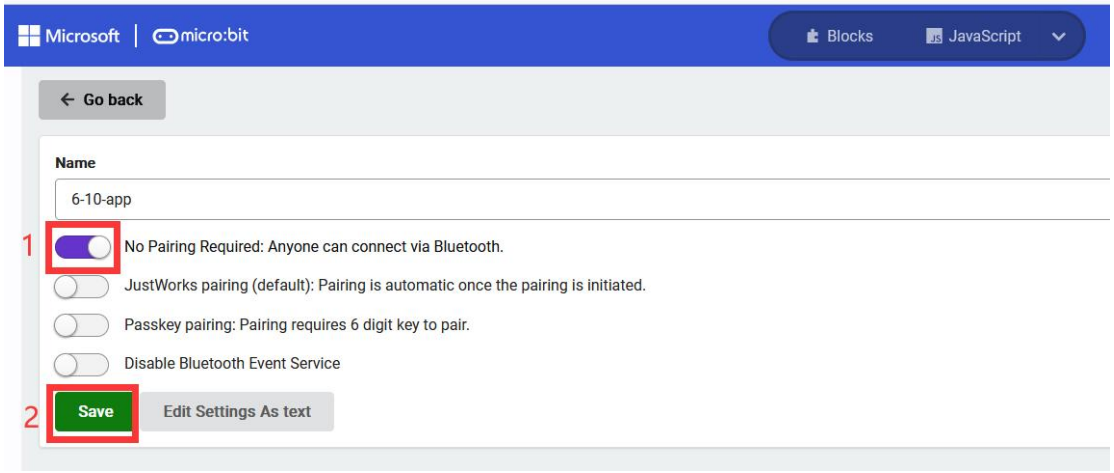


After successfully adding the Bluetooth extension, it will appear in the toolbox list.



Set the micro:bit Bluetooth to a password-free connection:

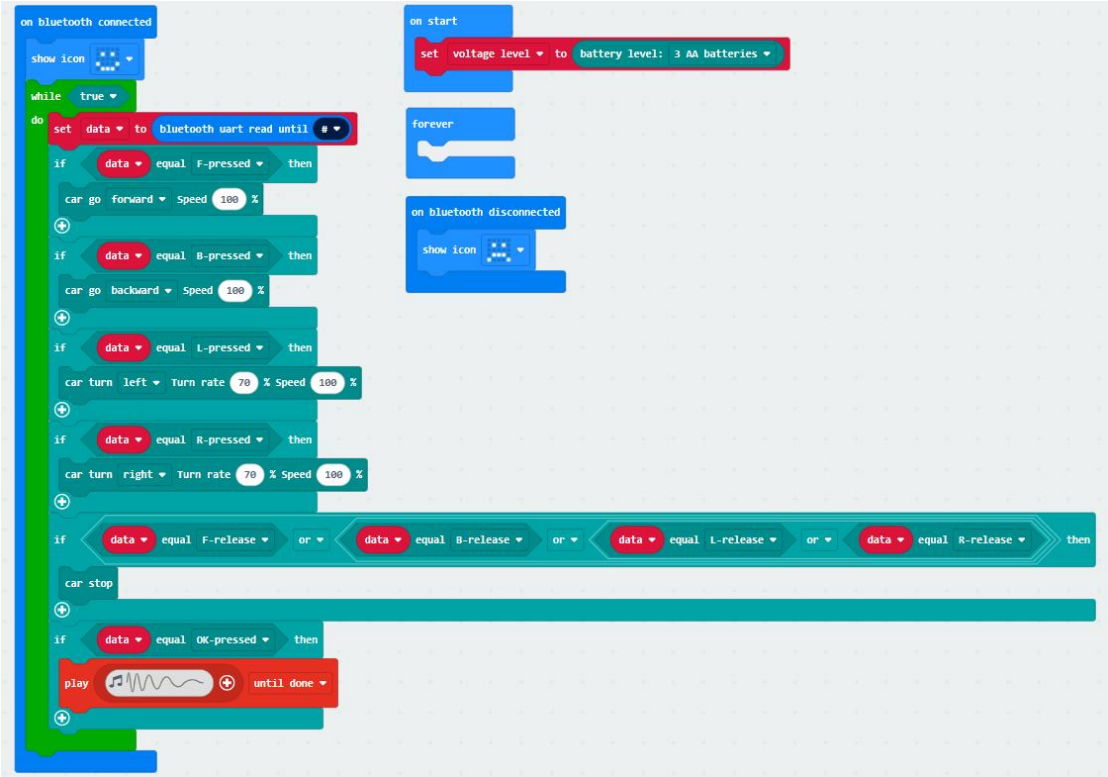




58

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



•When successfully connected to the Bluetooth of Micro:bit, the 5*5 dot matrix displays a smiley face.

•Then enter an infinite loop.

•In the dotted loop, read the data received by the Micro:bit Bluetooth.

•If the "F" key value of the APP is received, the car moves forward.

•If the "B" key value of the APP is received, the car moves backward.

•If the "L" key value of the APP is received, the car turns left.

•If the "R" key value of the APP is received, the car turns right.

•When the pressed button is released, the car stops.

•If the "OK" key value of the APP is received, the buzzer sounds.

•When the Bluetooth of Micro:bit is disconnected, the 5*5 dot matrix displays a sad face.

59

Install SIYEENOVE APP

We provide Android and IOS versions of the APP, please refer to the "**Download and install Bluetooth APP**" in the tutorial folder to install and use.

Download and install Bluetooth APP

Consequence

After the APP and micro:bit Bluetooth are successfully connected, you can use the App to control the mcar, below are the instructions corresponding to the app buttons

Button	F	B	L	R	OK
Function	Press>forward Release>stop	Press>backward Release>stop	Press>turn left Release>stop	Press>turn right Release>stop	Press>buzzer sounds

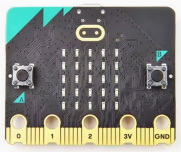
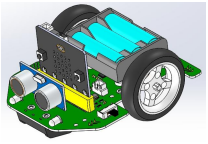
60

6.11 Bluetooth APP Plus control

Goal

- Use Bluetooth APP of Android phone to control mCar.
- Add more app control functions.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	Android phone
				

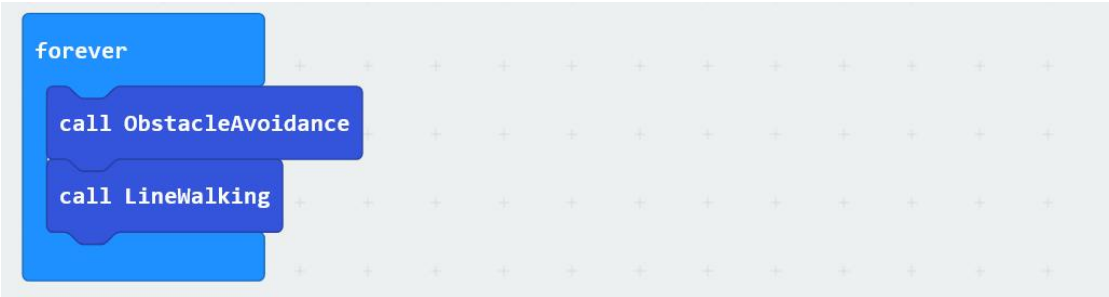
Programming

1. Micro: bit startup code:

•Set mCar to use 3 AA batteries.

•Set up mCar to control 2 RGB ambient lights using the P8 pin of the micro: bit.

2. Micro: bit loop code after startup:

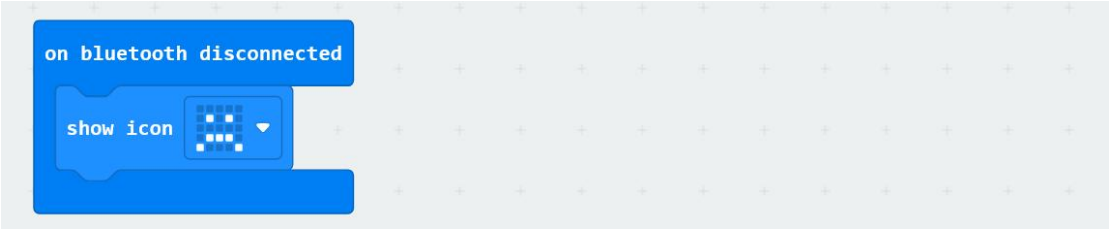


•Execute obstacle avoidance function.

•Execute the detection line function.

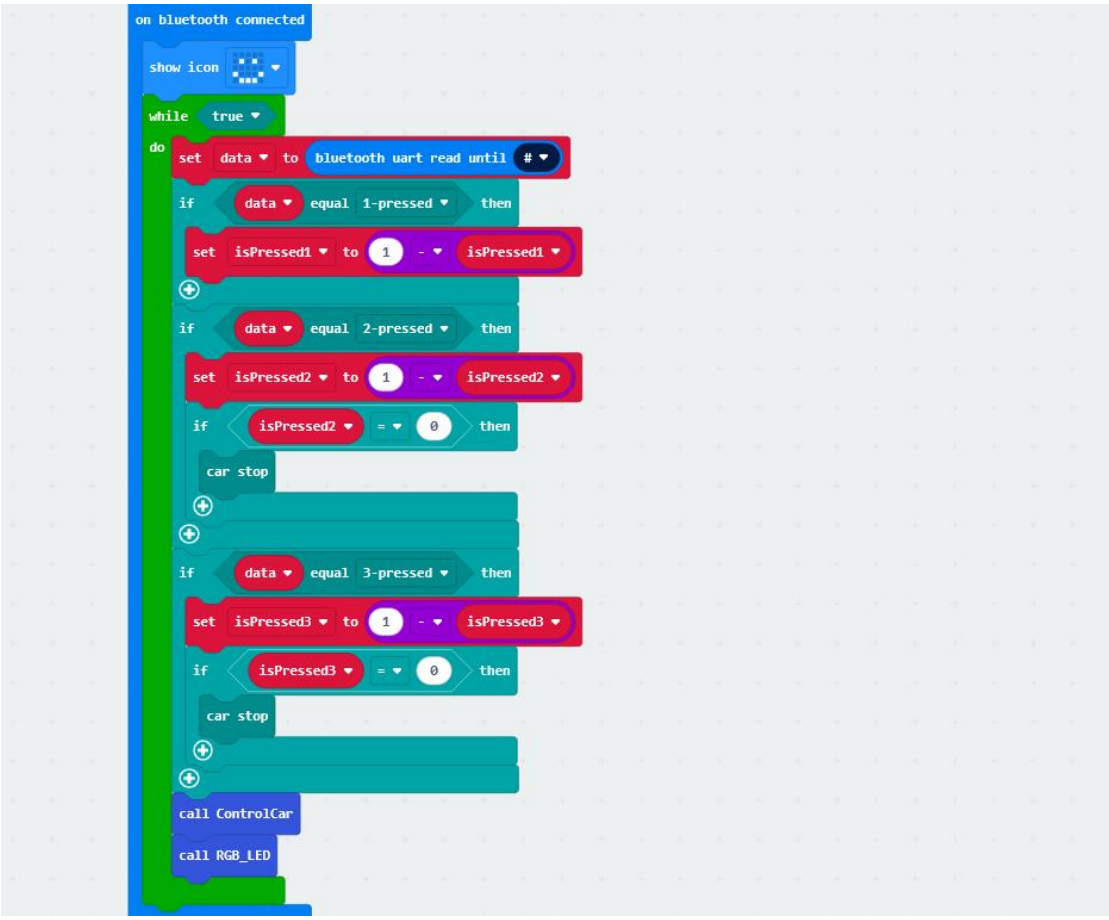
61

3. Bluetooth disconnection code



•When Bluetooth is disconnected, the micro: bit dot matrix displays a sad face.

4. Bluetooth successful connection code



•When Bluetooth is successfully connected, the micro: bit dot matrix displays a smiling face.

•Receive APP instructions in a while loop.

•If button 1 is pressed, change the value of isPressed1 to either 0 or 1.

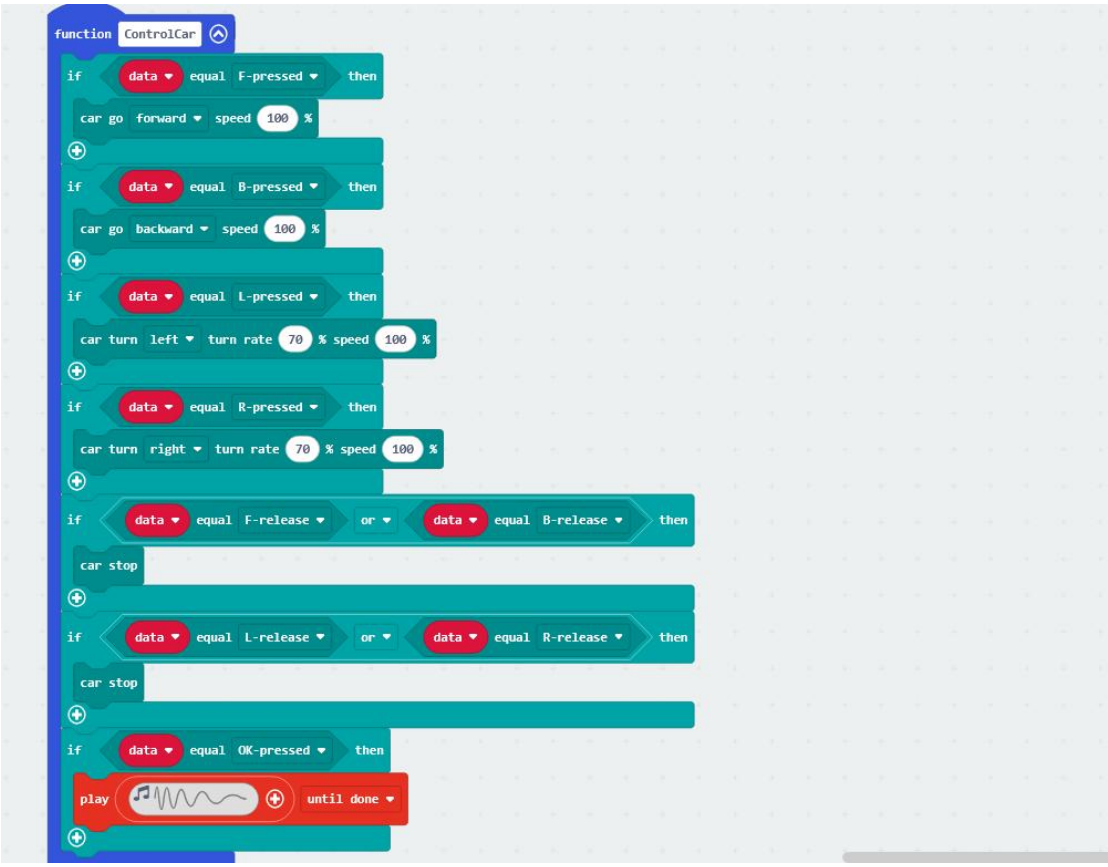
•If button 2 is pressed, change the value of isPressed2 to 0 or 1. If it is equal to 0, mCar stops.

•If the 3 keys are pressed, change the value of isPressed3 to 0 or 1. If it is equal

to 0, mCar stops.

- Execute control car functions
- Execute RGB_LED function

5. Control mCar function



•If the F button is pressed, mCar runs forward at 100% speed.

•If the B button is pressed, mCar runs backwards at 100% speed.

•If the L button is pressed, the mCar turns left at 70% turn rate and 100% speed.

•If the R button is pressed, the mCar turns right at 70% turn rate and 100% speed.

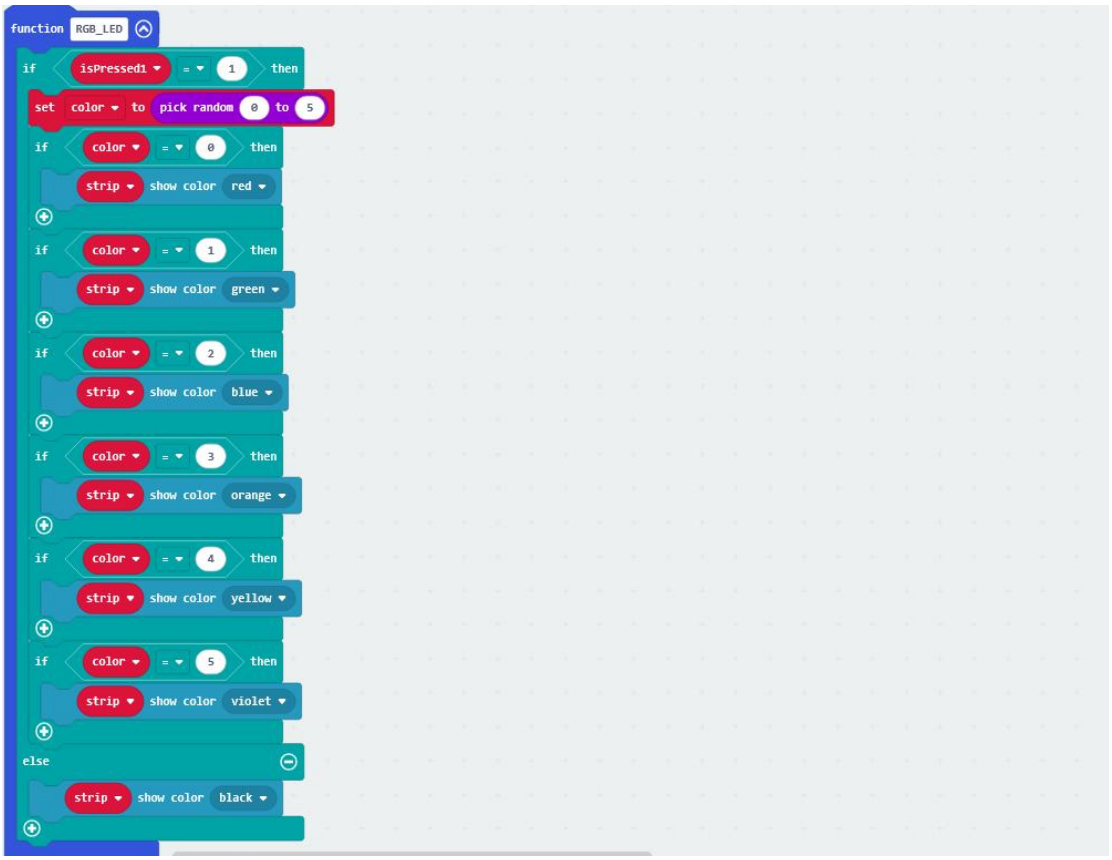
•If the F and B buttons are released, mCar will stop.

•If the L and R buttons are released, mCar will stop.

•If the OK button is pressed, mCar will beep.

63

6. RGB_LED ambient light function



•If the isPressed1 variable is equal to 1, determine which color the light is on.

•Randomly generate any number between 0-5.

•The random number is 0, and the ambient light is red.

•The random number is 1, and the ambient light is green.

•The random number is 2, and the ambient light is blue.

•The random number is 3, and the ambient light is orange.

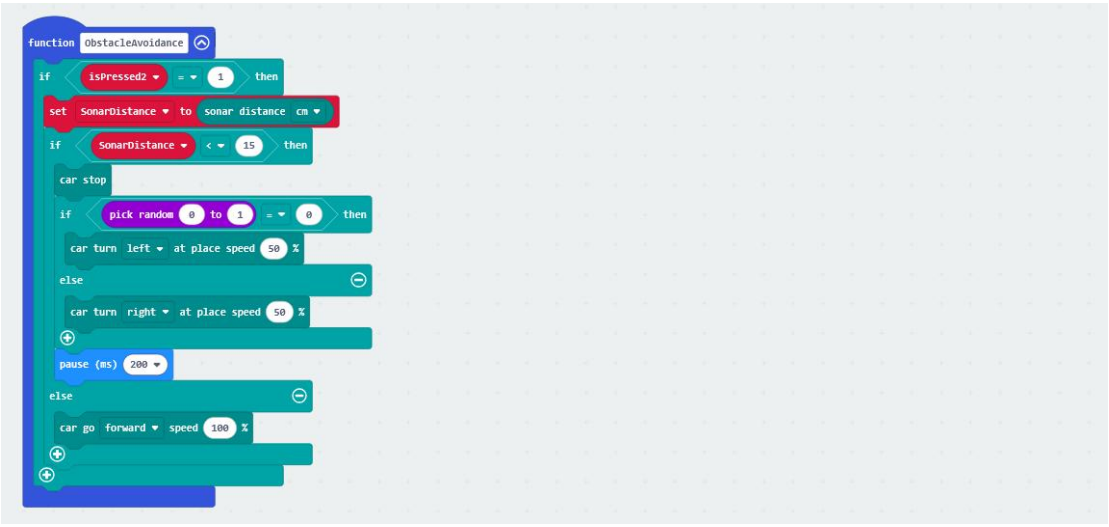
•The random number is 4, and the ambient light is yellow.

•The random number is 5, and the ambient light is violet colored.

•If the isPressed1 variable is not equal to 1, turn off the light.

64

7. Obstacle avoidance function



•If the isPressed2 variable is equal to 1, measure the distance and then avoid obstacles.

Set a SonarDistance variable to store the values returned by ultrasound waves.

•When the ultrasound return value is less than 15CM, it proves that an obstacle has been detected 15CM ahead, and mCar randomly turns left or right.

• Delay of 200ms

•If the ultrasonic return value is greater than or equal to 15CM, mCar will move forward at full speed.

8. Line checking function

If the isPressed3 variable is equal to 1, enter the line checking function, and the code function is the same as in section 6.2.

Install SIYEENOVE APP

We provide Android and IOS versions of the APP, please refer to the "**Download and install Bluetooth APP**" in the tutorial folder to install and use.

Download and install Bluetooth APP

Note: If you have already performed the above steps, you can skip them.

66

Consequence

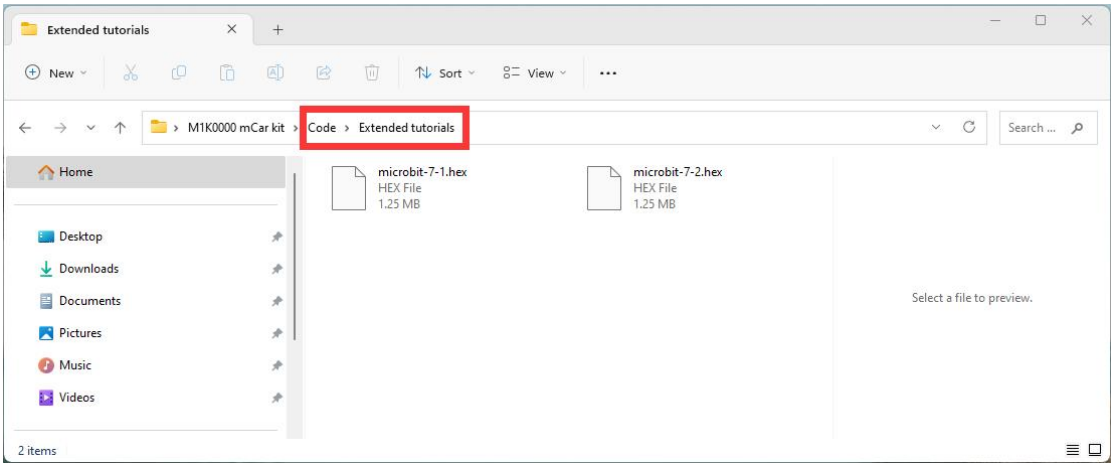
After the APP and micro:bit Bluetooth are successfully connected, you can use the App to control the mcar, below are the instructions corresponding to the app buttons:

Button	F	B	L	R
Function	Press>forward Release>stop	Press>backward Release>stop	Press>turn left Release>stop	Press>turn right Release>stop
Button	**OK**	**1**	**2**	**3**
Function	Press>buzzer sounds	Turn on or off the ambient light with every press.	Turn on or off the obstacle avoidance function with every press.	Turn on or off the line checking function with every press.

67

7. mCar Extended Tutorial

The sample codes for the extended tutorials are all saved in the "Code -> Extended tutorials" file:



Reminder!!!

You don't need to add the mCar extension if you import and use the code we provided. ---> Recommend!
If you create a new project and drag and drop blocks to build code, you need to manually add the mCar extension, please refer to sections 3.1 and 4.1.

We only provide project examples. The Lego accessories used in the following tutorials are not included in the kit. If you want to do this project, you need to purchase it yourself.

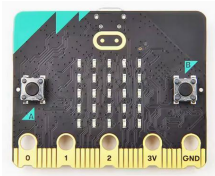
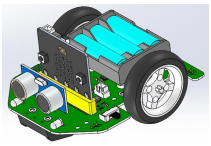
68

7.1 Drive a 270 degree servo

Goal

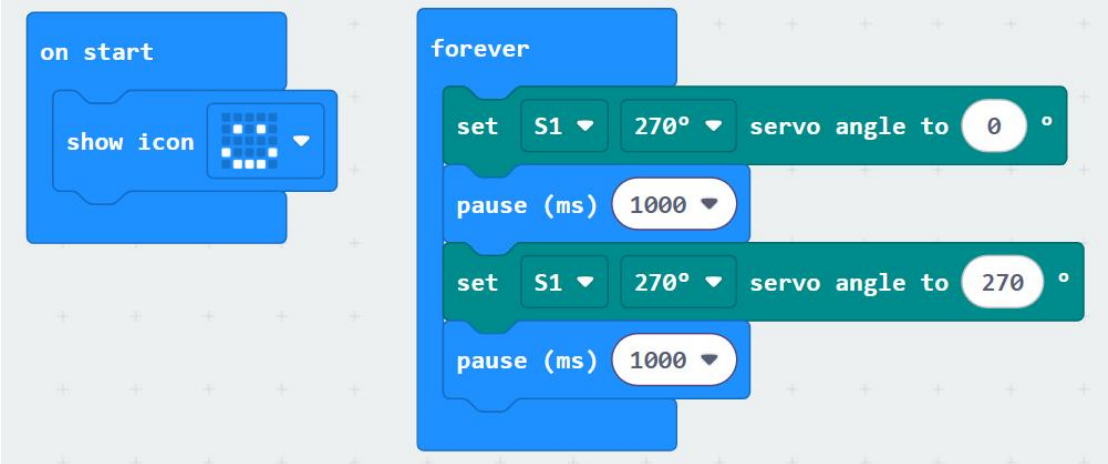
Learn how to drive a 270-degree Lego servo, and get some inspiration for expanding the capabilities of your mcar.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	270 Lego servo
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



Consequence

The 270 degree Lego servo shaft rotates cyclically between 0 and 270 degrees.

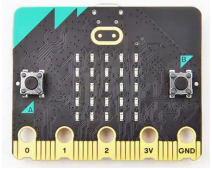
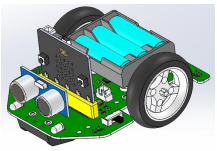
69

7.2 Drive a 360 degree servo

Goal

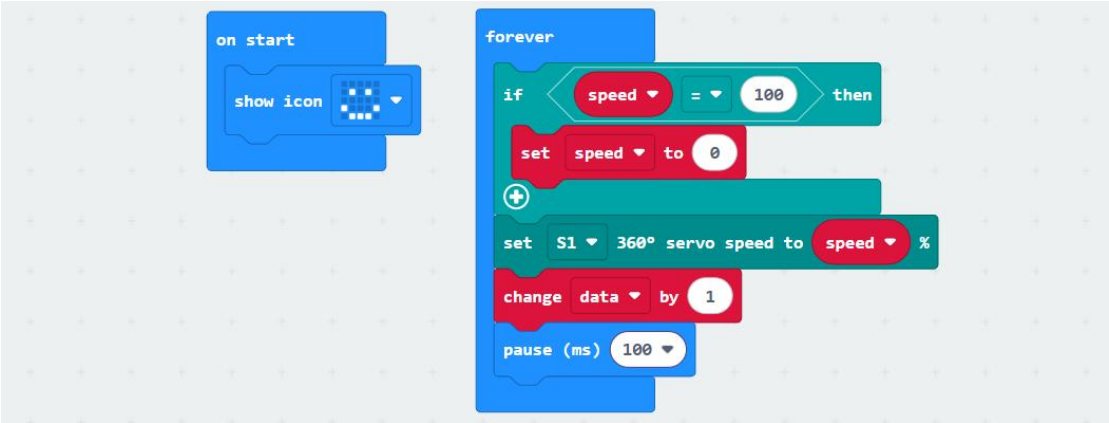
Let the 360-degree Lego servo rotate evenly and rapidly in the positive direction.

Things you need：

Computer	Micro:bit v2.x.x	Micro USB cable	mCar	360 Lego servo
				

Programming

Directly import the hex file code we provided into the Makecode editor and download it to Micro:bit. If drag to build a new code, remember to add the mcar extension.



Consequence

The 360-degree Lego servo rotate evenly and rapidly in the positive direction.

Questions to think about

How to drive 90-degree and 180-degree servos?

How to use Lego block to expand the functions and appearance of mcar?

1. QA

8.1 Unable to upload code to Micro:bit

☛Do you use a USB cable with data communication function?

☛Does the USB cable connect well?

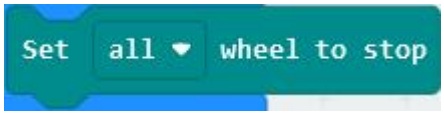
8.2 The Micro:bit disk name is not displayed correctly

☛If the disk name is displayed as: MAINTENANCE (Normal is MICROBIT), it means that it has entered the firmware update mode. If you press and hold the micro:bit reset button and then power on the micro:bit, the micro:bit will enter the firmware update mode. At this time, we can update the firmware and the Microbit disk name will return to normal. You can refer to the following link:

https://microbit.org/get-started/user-guide/firmware/

8.3 After uploading a new code, the motor still rotates

☛ This is because the car stop block is not called again:

8.4 Neither the motor nor the headlights work

☛Make sure the micro:bit is fully plugged into the mCar slot.

☛Make sure the micro:bit Edge Connectors are clean.

☛Make sure the mCar slots are clean.

71

8.5 Other issues

☛Please check if the assembly steps are correct?

☛Please check if the battery power is sufficient?

☛Please check if the battery model used is correct?

9.Contact us

If you encounter technical problems or want to share your ideas and opinions with us, please contact us at any time.

✉ siyeenove@outlook.com

http://siyeenove.com