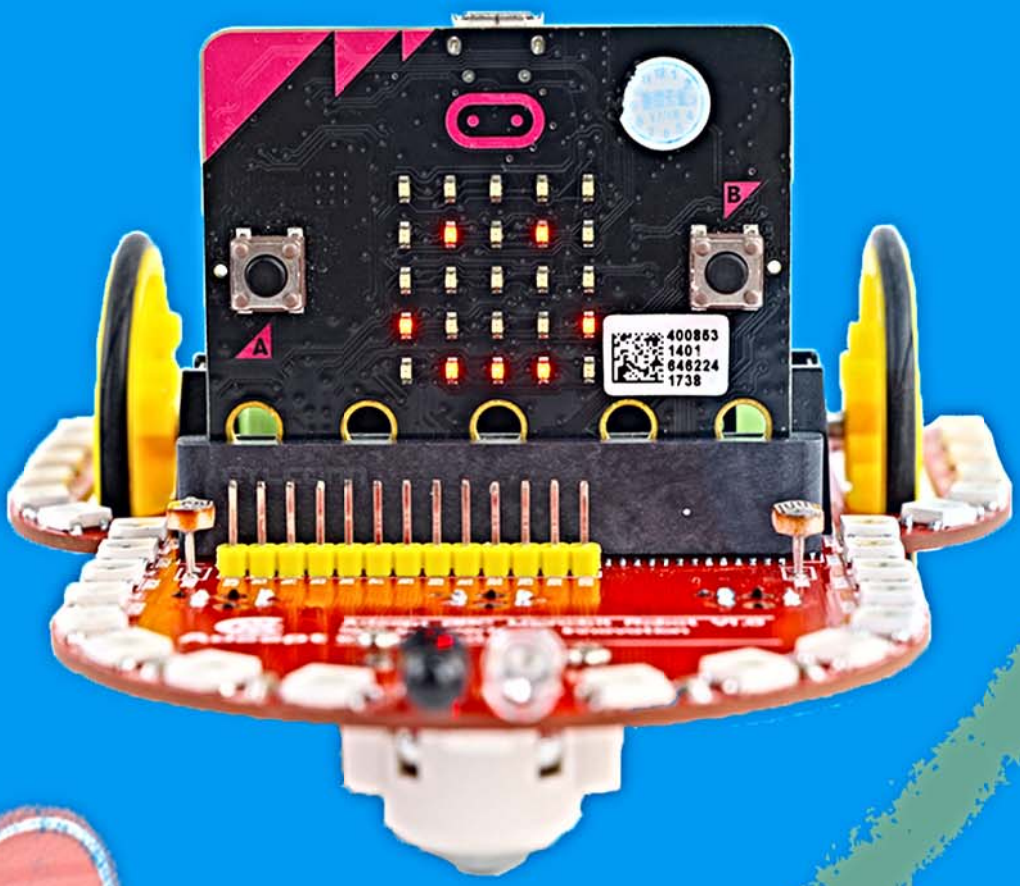
Adeept

Adeept BBC Micro:bit Robot Starry:bit

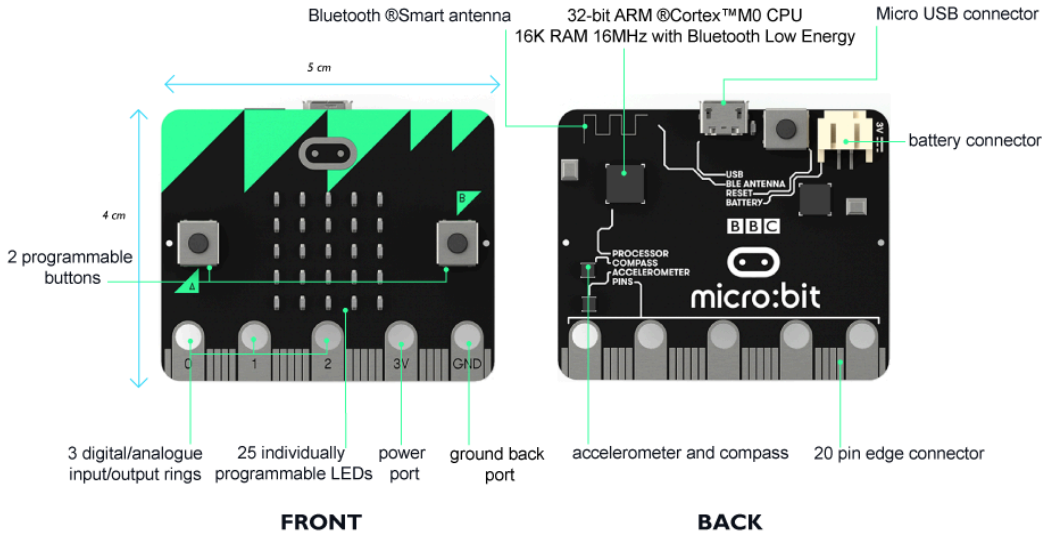


Catalogue

About Micro bit

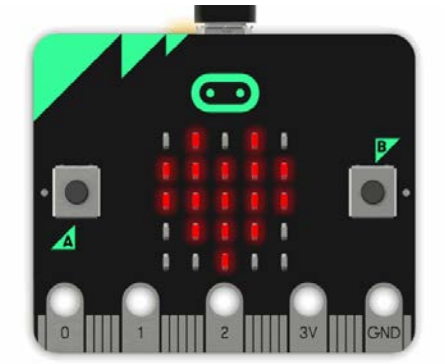
From dancing robots to banana keyboards, your micro: bit has all the features you need to code awesome stuff - the possibilities are endless!

In 2015, BBC launched the second programmable micro computer, and it is a super mini computer which can be put in your pocket, meanwhile, it can help young people learn basic programming knowledge with low cost. In 2016, 100 million micro computers have been provided for middle and primary school students in Britain.



Your micro: bit has the following physical features:

LEDs

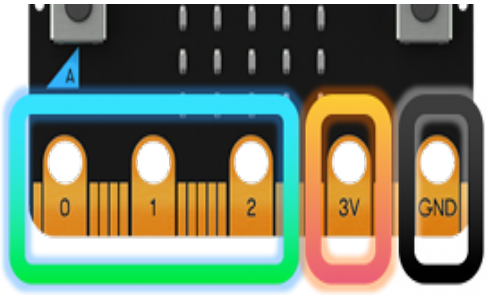


LED stands for Light Emitting Diode. The micro: bit has 25 individually-programmable LEDs, allowing you to display text, numbers, and images.

Buttons

There are two buttons on the front of the micro: bit (labelled A and B). You can detect when these buttons are pressed, allowing you to trigger code on the device.

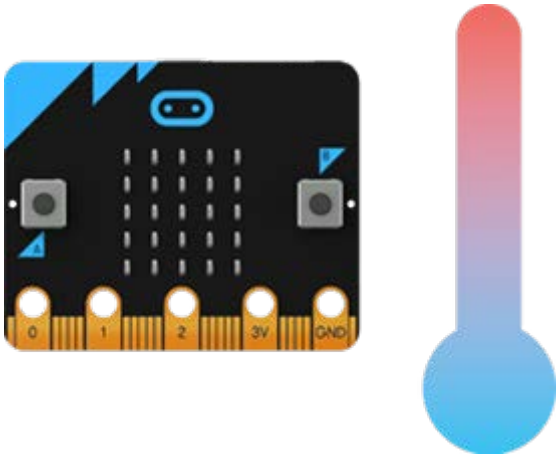
Pins



There are 25 external connectors on the edge connector of the micro: bit, which we refer to as 'pins'. Program motors, LEDs, or other electrical components with the pins, or connect extra sensors to control your code! The five big pins connect with the annular hole, and marked as 0, 1, 2, 3V, GND, representing P0, P1, P2, 3V power output

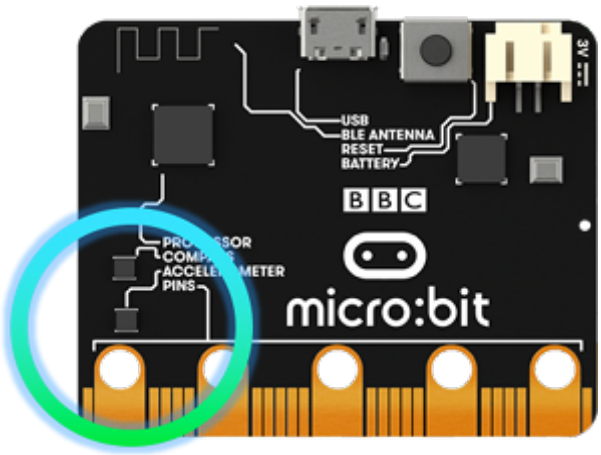
By reversing the LEDs of the screen to become an input, the LED screen works as a basic light sensor, allowing you to detect ambient light.

Temperature Sensor



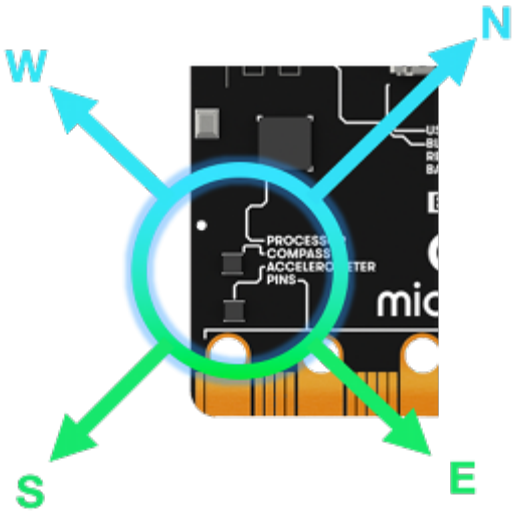
This sensor allows the micro: bit to detect the current ambient temperature, in degrees Celsius.

Accelerometer



An accelerometer measures the acceleration of your micro: bit; this component senses when the micro: bit is moved. It can also detect other actions, e.g. shake, tilt, and free-fall.

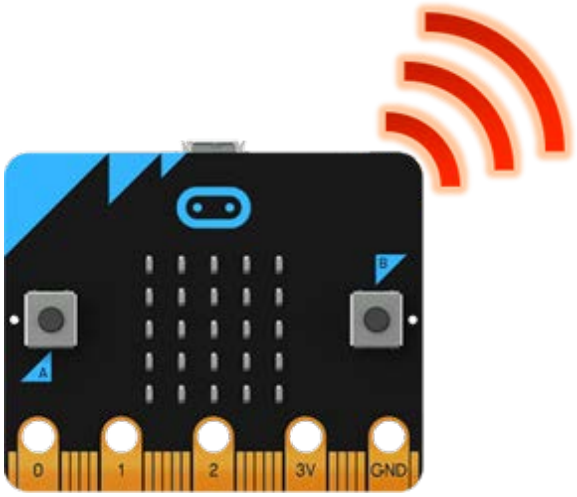
Compass



The compass detects the earth's magnetic field, allowing you to detect which direction the micro: bit is facing. The compass has to be calibrated before it can be used.

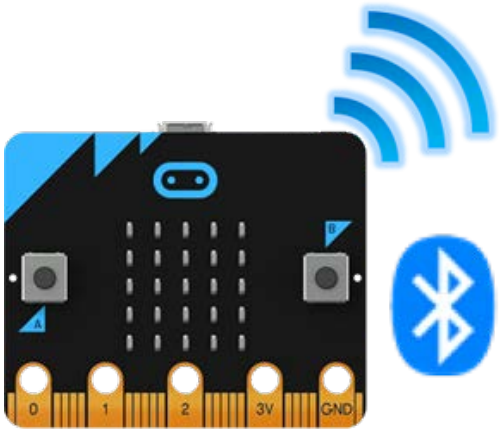
'Calibrating' the compass ensures the compass results are accurate. For the JavaScript Blocks Editor, use the 'calibrate compass' block. To calibrate the compass in Python use compass.calibrate().

Radio



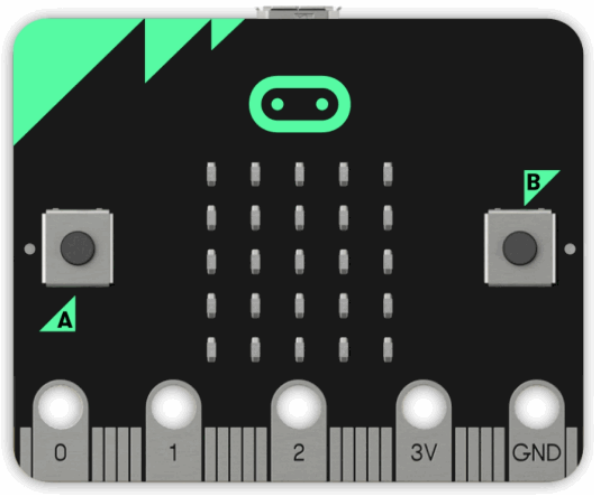
The radio feature allows you to communicate wirelessly between micro: bits. Use the radio to send messages to other micro: bits, build multiplayer games, and much more!

Bluetooth



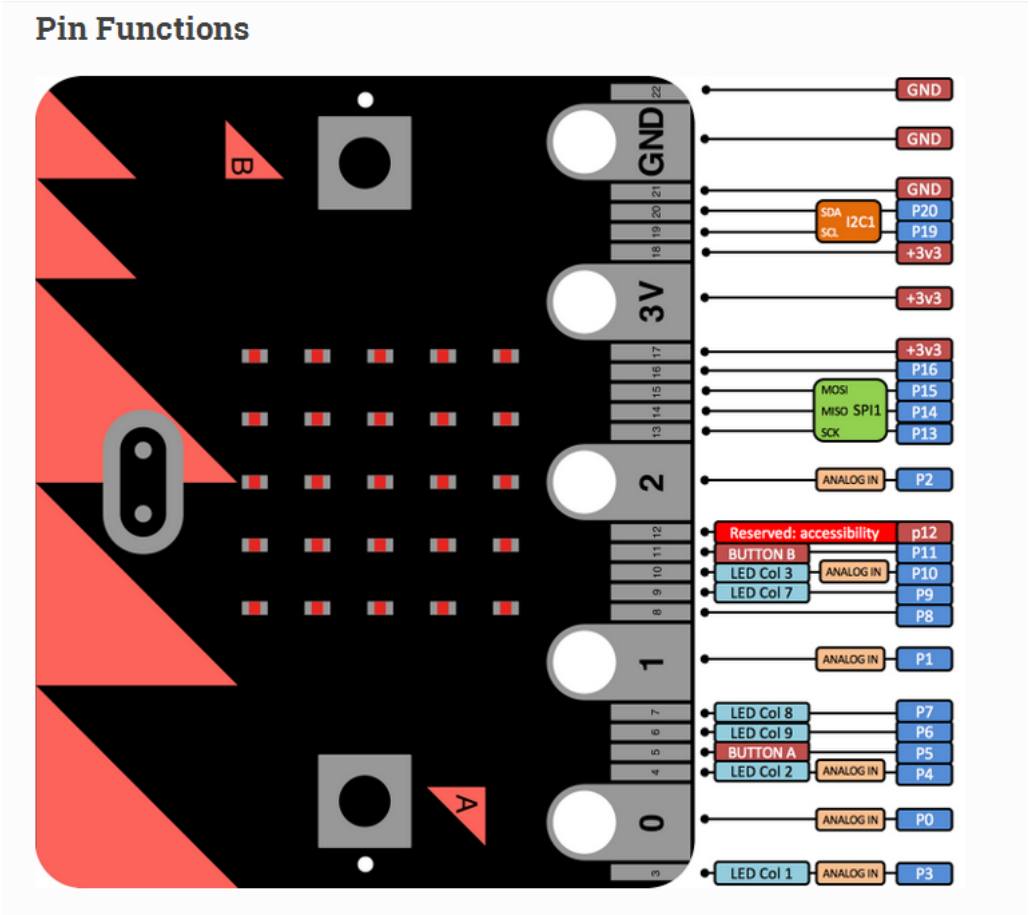
A BLE (Bluetooth Low Energy) antenna allows the micro: bit to send and receive Bluetooth signals. This allows the micro: bit to wirelessly communicate with PCs, Phones, and Tablets, so you can control your phone from your micro: bit and send code wirelessly to your device from your phone!

USB Interface



The USB interface allows you to connect the micro: bit to your computer via a micro-USB cable, which will power the device and allow you to download programs onto the micro: bit.

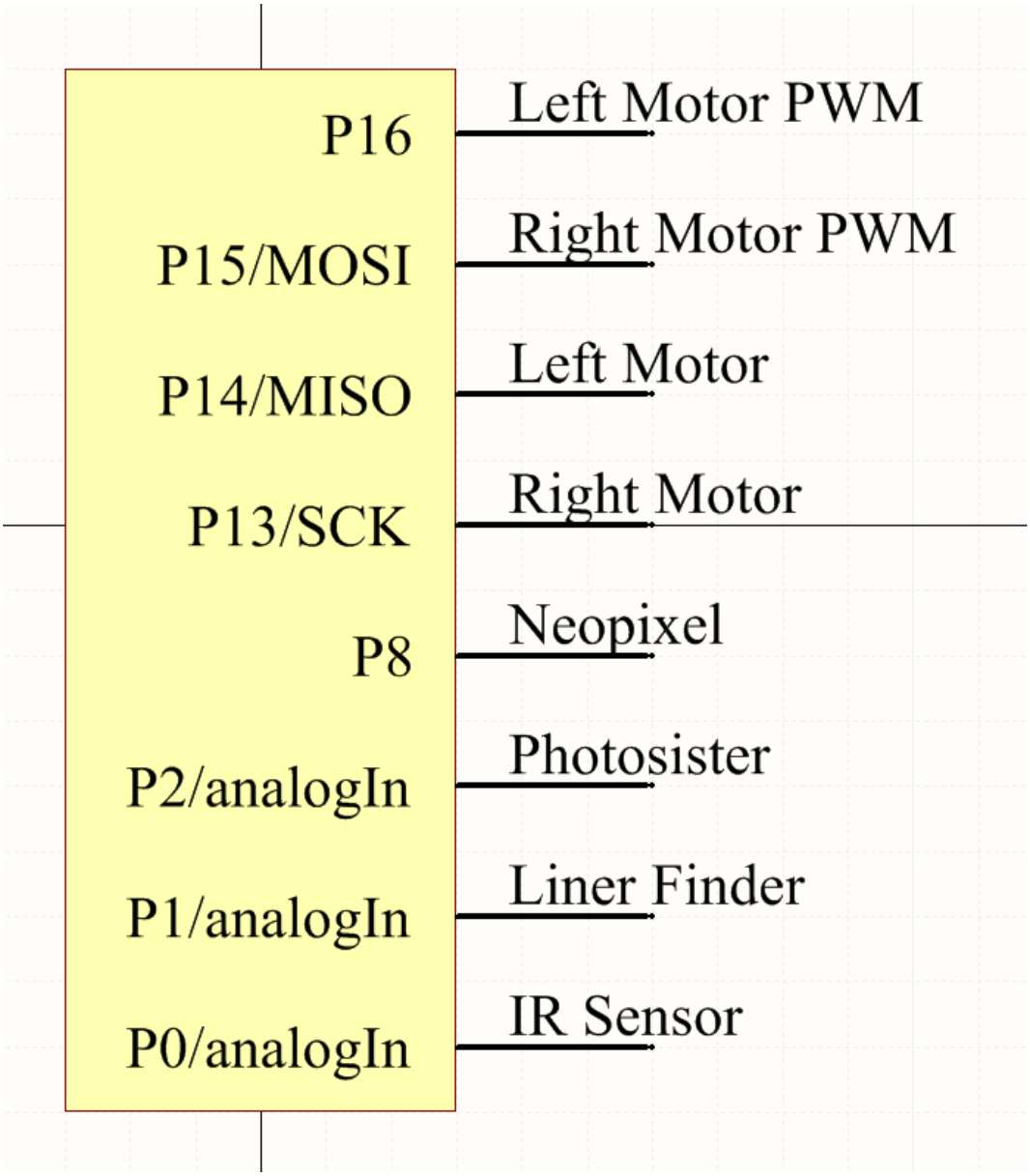
Microbit pins introduction



Product Features:

This is a micro:bit-based programmable robot for which we provide a variety of gameplay and code. The car contains infrared obstacle avoidance module, Neopixel colorful RGB LEDs, DC gear motor, tracking sensor, photoresistor and so on. Also we will expand some IO ports for you to do extended experiments and achieve more gameplay.

Schematic of hardware connection of the car



P0 is connected to the infrared sensor. The obstacle encountered during the driving of the car will be detected and processed by micro: bit.

P1 is connected to the tracking sensor. Any one of the three sensors will give different values as feedback when it touches the black line.

P2 is connected to the photoresistor. The two photoresistors can feed back the detected resistance value to the microbit.

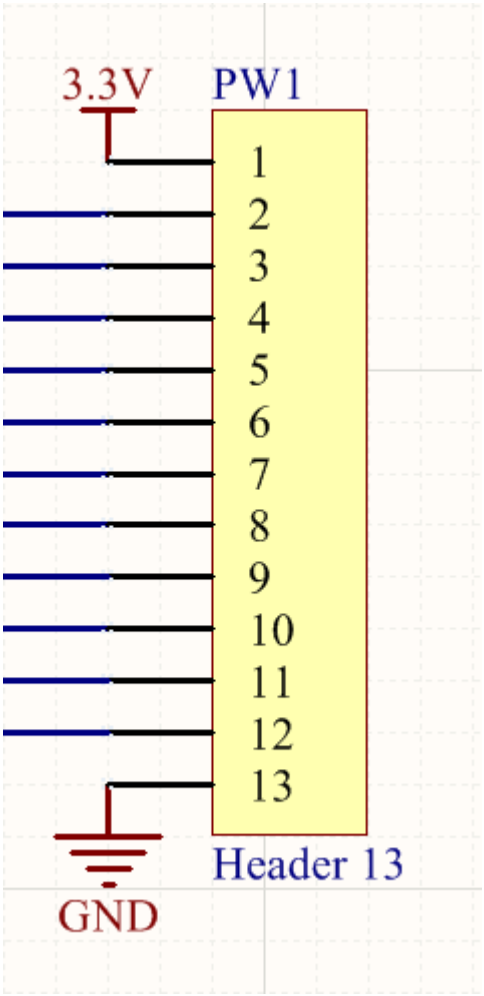
P8 is connected to Neopixel. It can control 40 LEDs of the car.

P13 is connected to the left motor. It can control the left motor to forward and reverse.

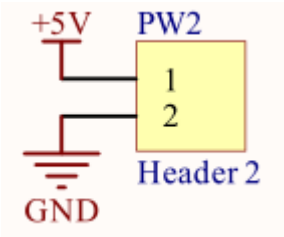
P14 is connected to the right motor. It can control the right motor to forward and reverse.

P15 is connected to the left motor and can control the speed of it.

P16 is connected to the right motor and can control the speed of it.

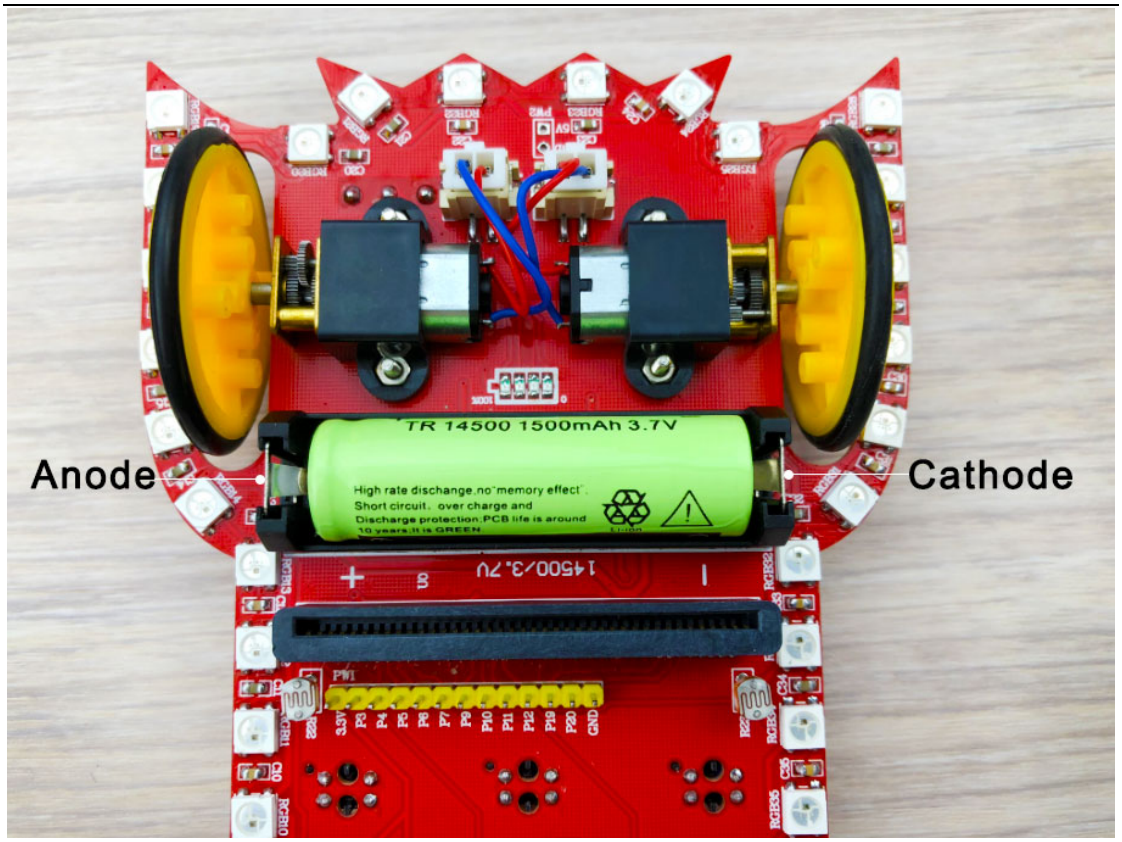


PW1 is an extension interface reserved on the car, which is convenient for you to use for extended experiments.



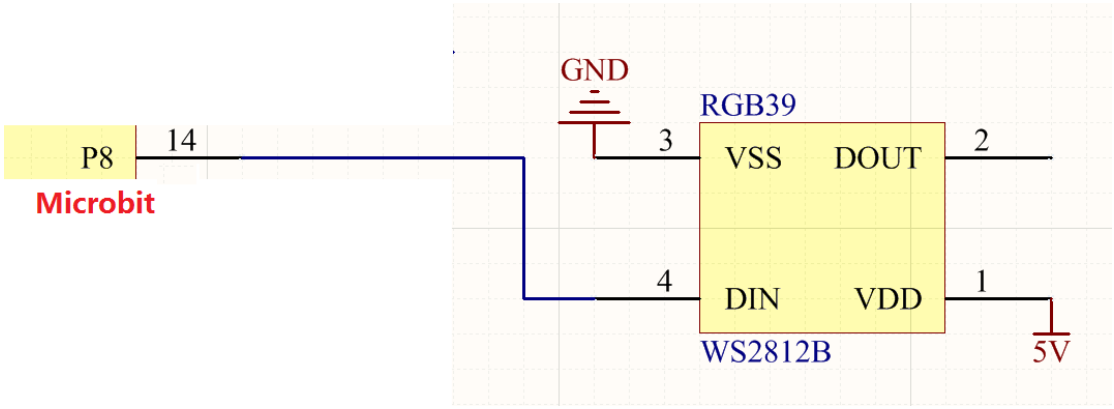
The 5V power supply interface reserved at the rear of the car body.

Note: Though the battery socket has anti-reverse protection, you should not try to reverse the power supply during operation.


Anode
Cathode
TR 14500 1500mAh 3.7V
High rate dischange,no"memory effect".
Short circuit、over charge and
Discharge protection.PCB life is around
10 years it is GREEN.
14500/3.7V

Project 1 Neopixel

Rainbow

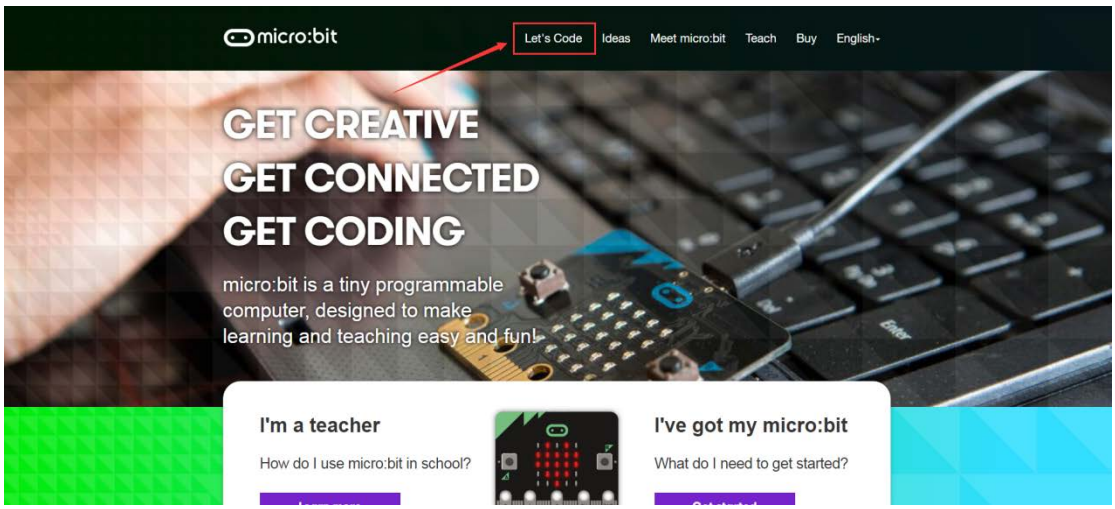


There are 40 LEDs on the starry:bit. Here takes three one as example. They are wired in the same way: DIN connects to the micro:bit P8 port, VDD connects to the anode of power supply, VSS connects to the cathode of power supply, DOUT connects to the DIN of the rear LEDs in series, and so on.

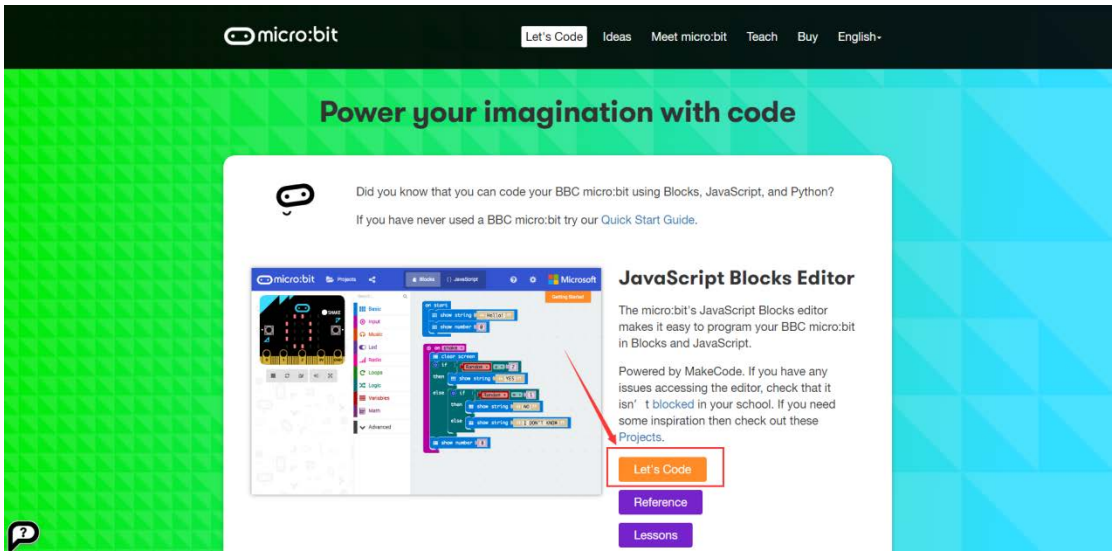
Makecode

First enter the official website：www.microbit.org

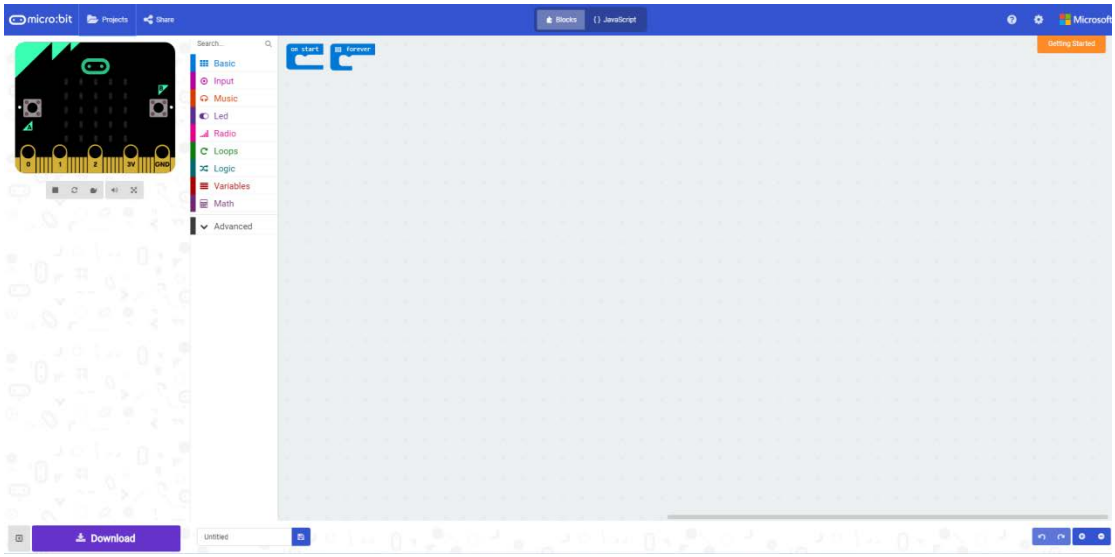
Click let's code



Click lets code



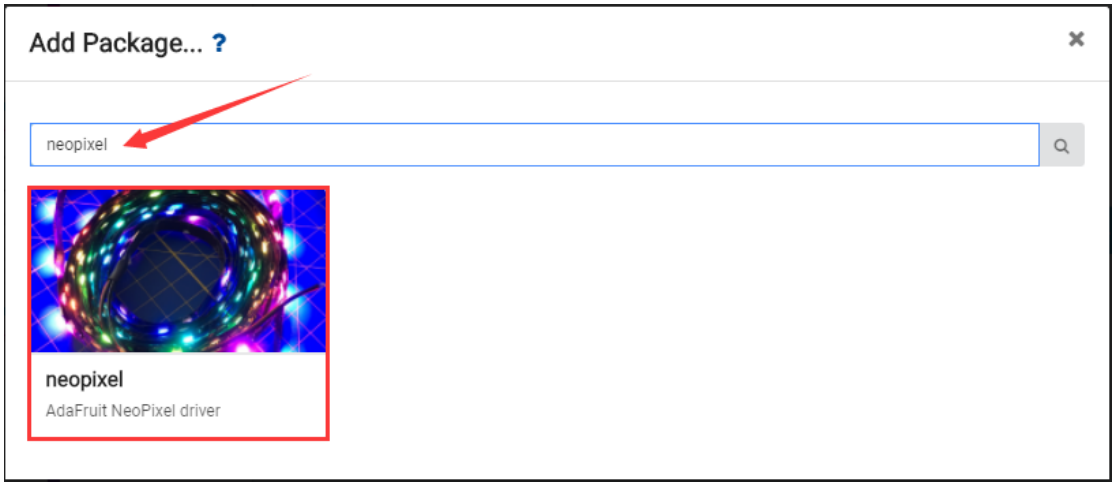
Now you can see the programming tool



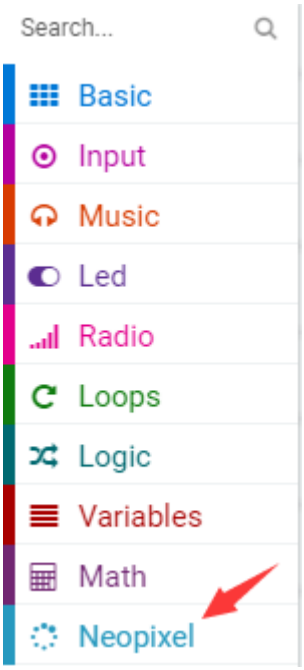
We need to manually add the library—neopixel

Click Advanced->Add Package

Fill in neopixel in the search bar -> click on neopixel

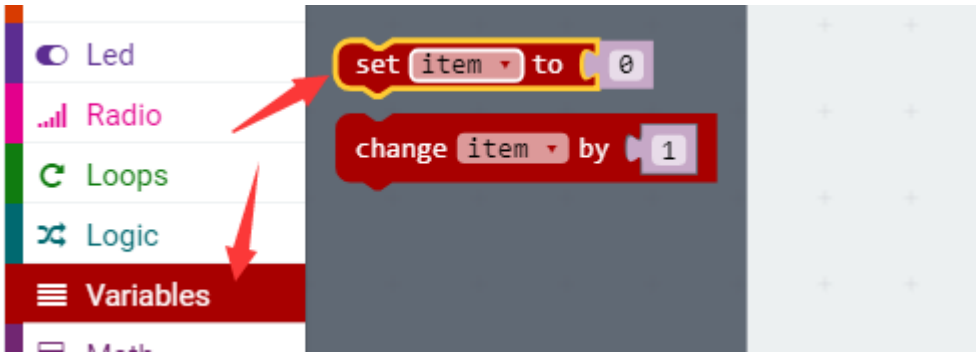

Add Package... ?
neopixel
neopixel
AdaFruit NeoPixel driver

You can see Neopixel below after adding it.



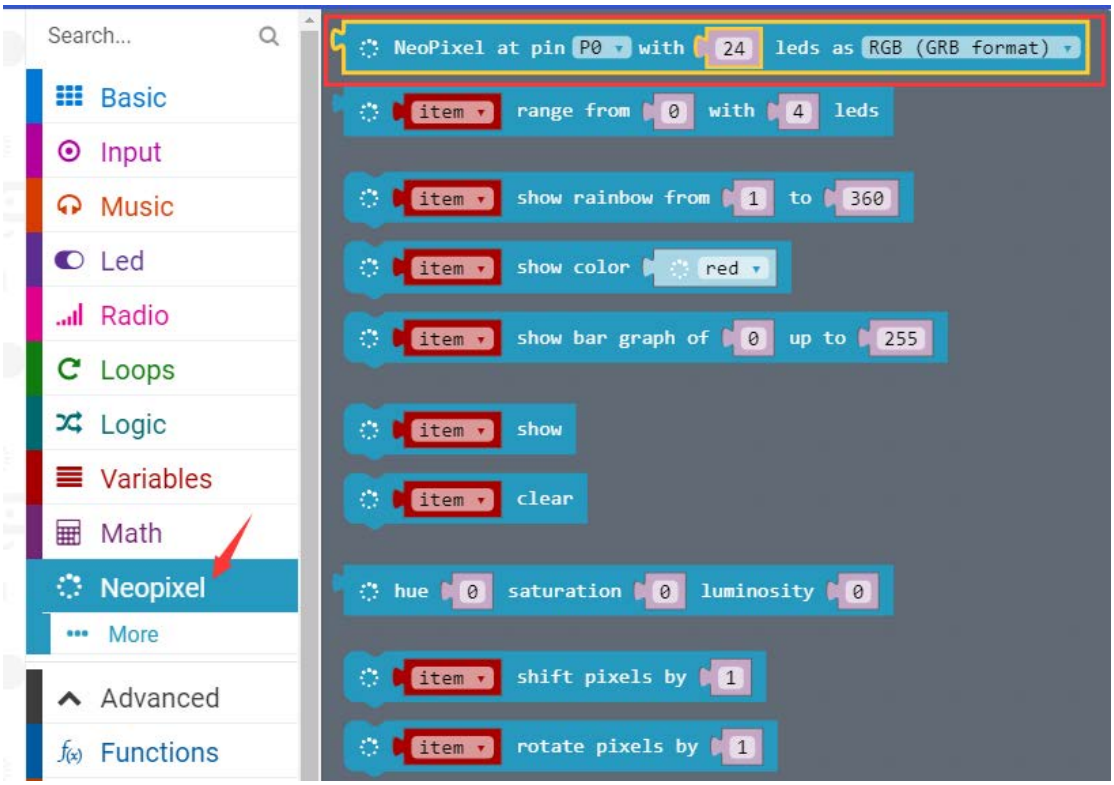
Note: You need to add related libraries to operate when doing other projects.

Coding process:



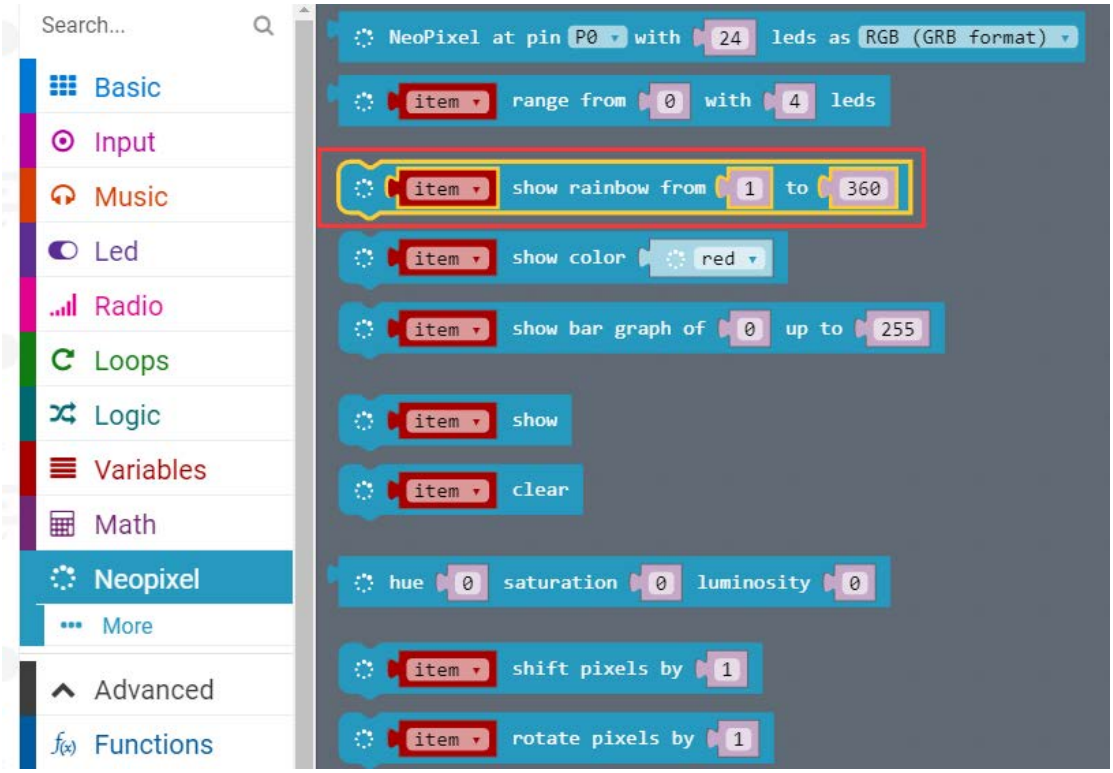
Rename item as Rainbow

Click Neopixel, choose the blocks as bellow

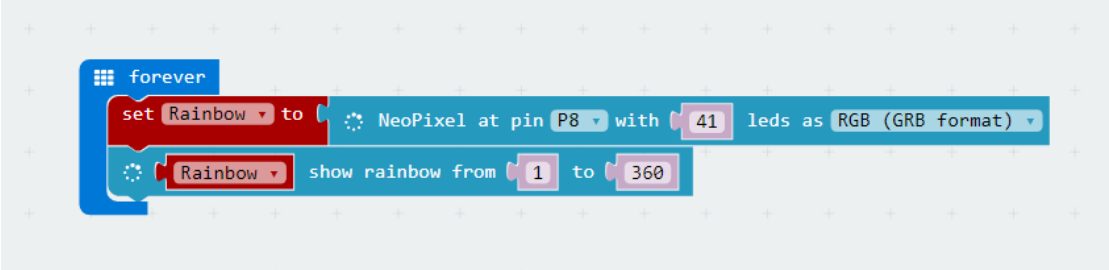


Using Pin8, the car Neopixel has a total of 40 LEDs. 41 LEDs were written here because in the test the last LED will not change color if 40 were written. The code is as follows:

Then set the color of Neopixel.



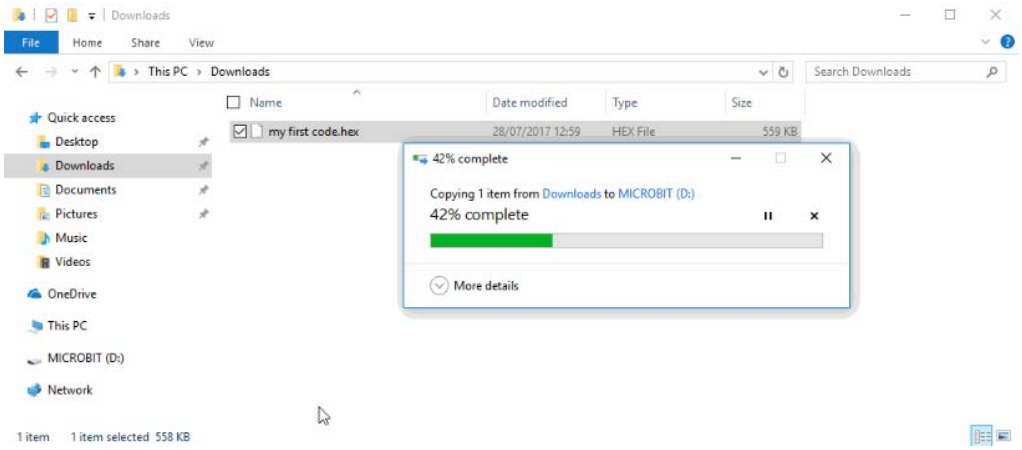
Rename item as Rainbow, as picture shows below:



The code is written.

Click the download button in the bottom right corner

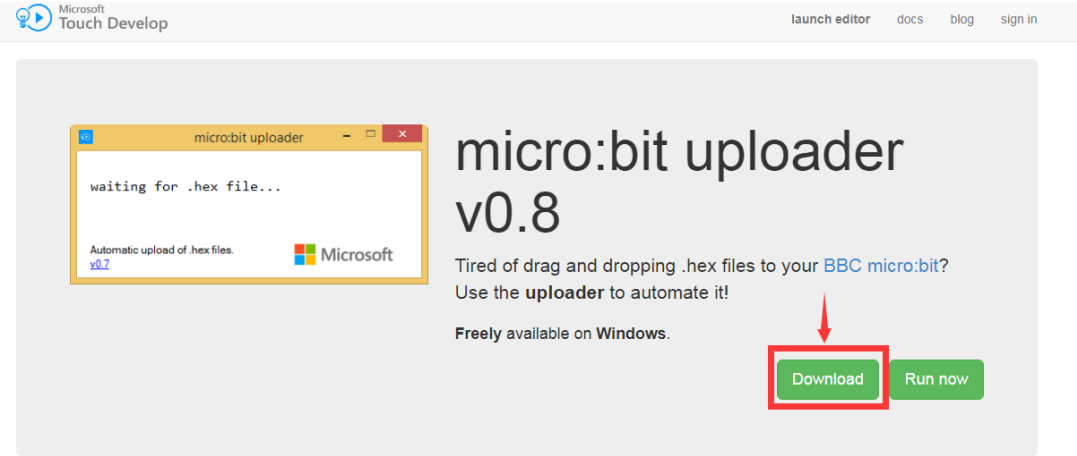
You can see that the download is complete in the lower right corner of the browser. But it hasn't been downloaded to the micro: bit. Open the location of the file, and then copy or cut it into the micro: bit disk.



Now you can see a rainbow light showing on the micro: bit.

If you want to save the trouble, you can download a plugin

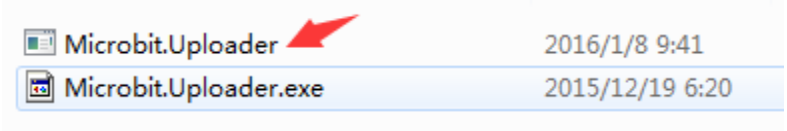
1. Plugin address：Enter：https://www.touchdevelop.com/microbituploader

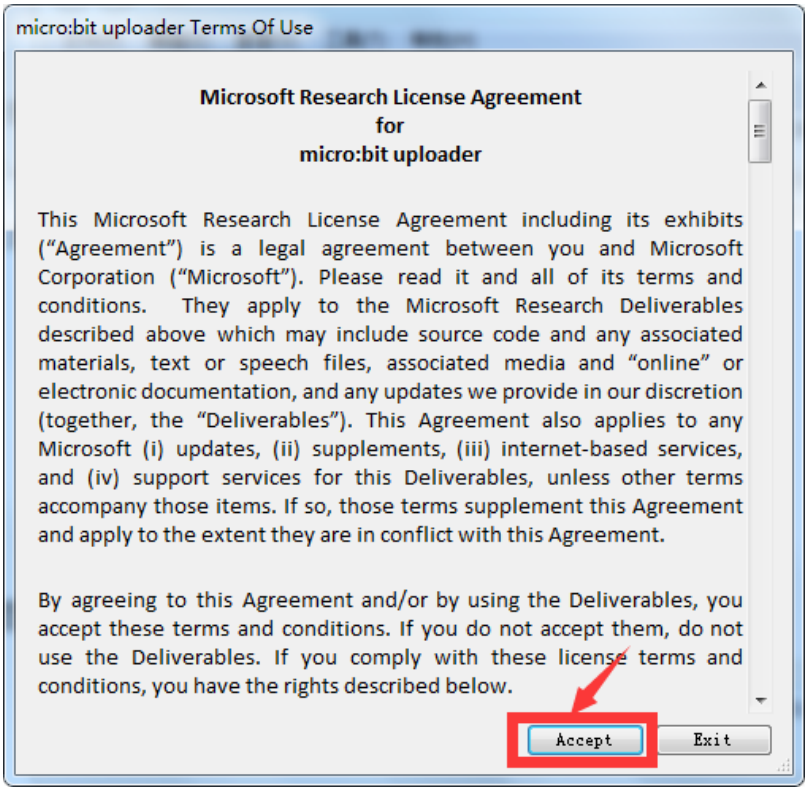


Click the button “Download”, when it is finished we only get ZIP compressed files.

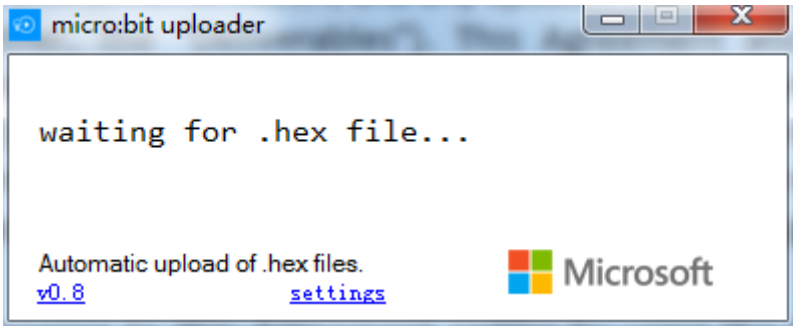
3. Next, extract the compressed files to any folder.

4. Then open the first file after uncompress.



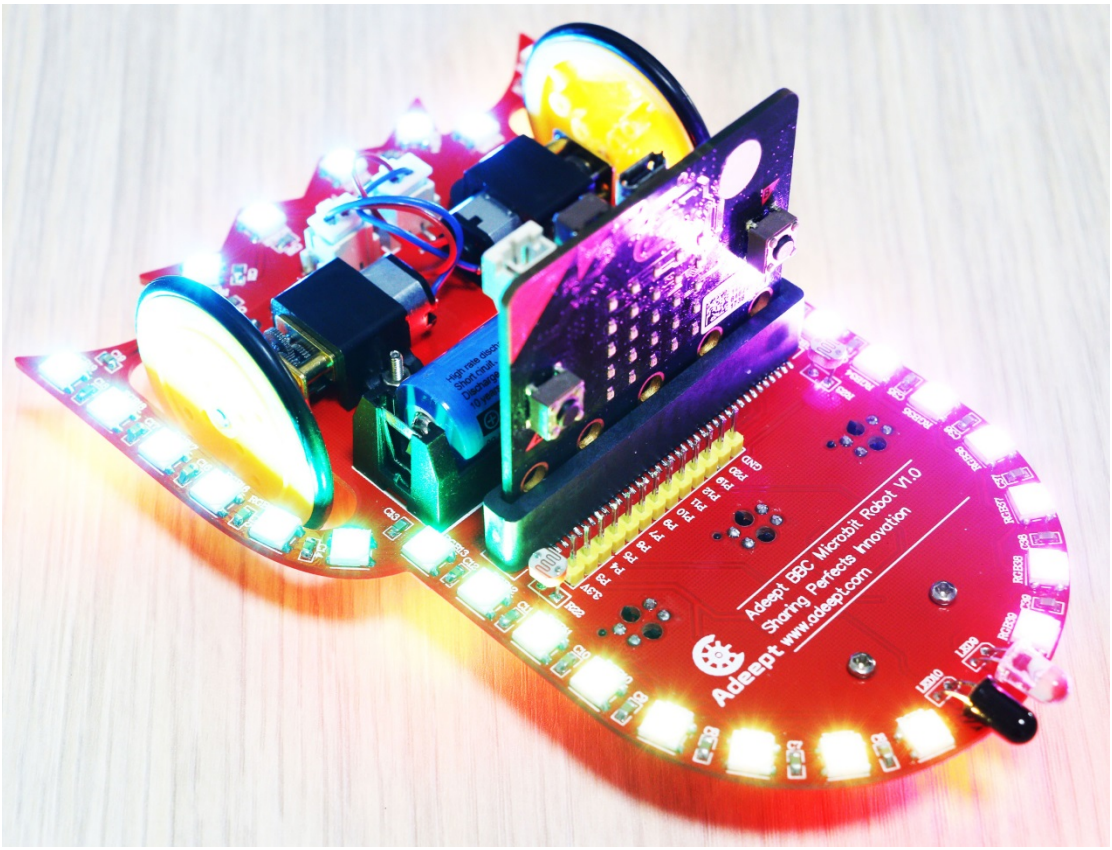


5. When it shows the icon below, it means waiting for downloading, we can minimize it but can't quit.



Click the button "Download" directly and we can download the program onto micro: bit.

Effect picture:



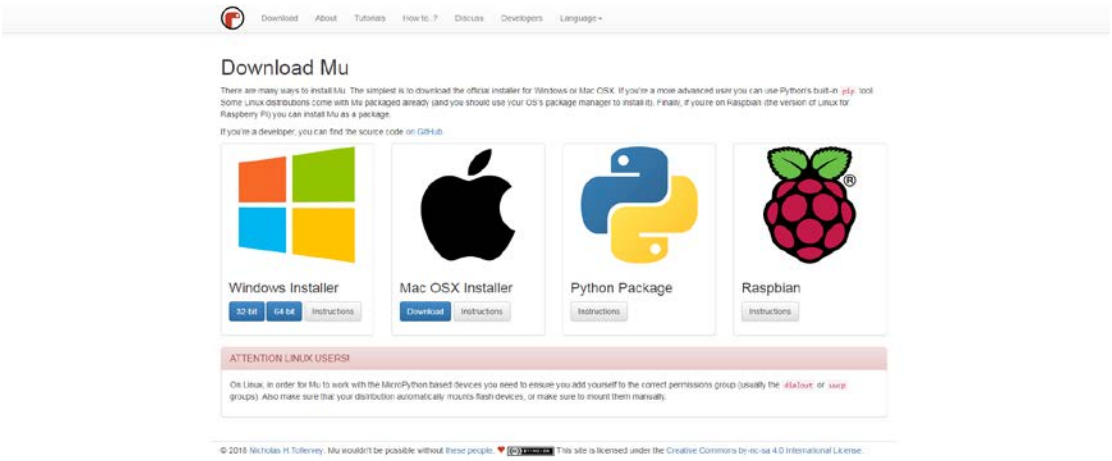
Note: If you start the car without pulling out the USB cable of the micro: bit after downloading the program, the R30-R39 light will be on even if you turn off the power supply of the car. This is because the micro: bit is co-powered with these 10 lights. This situation will not occur when the USB cable is unplugged.

MU microPython

Download and install steps :

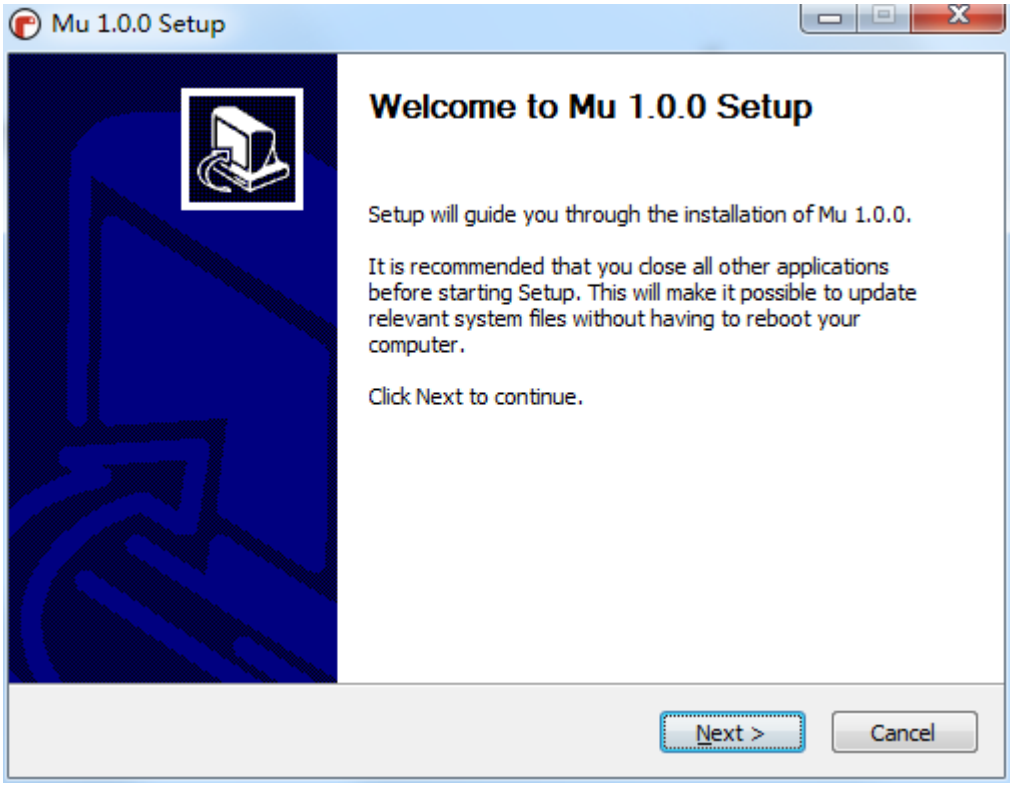
1. Enter https://codewith.mu/#download
2. Click “ Download” button

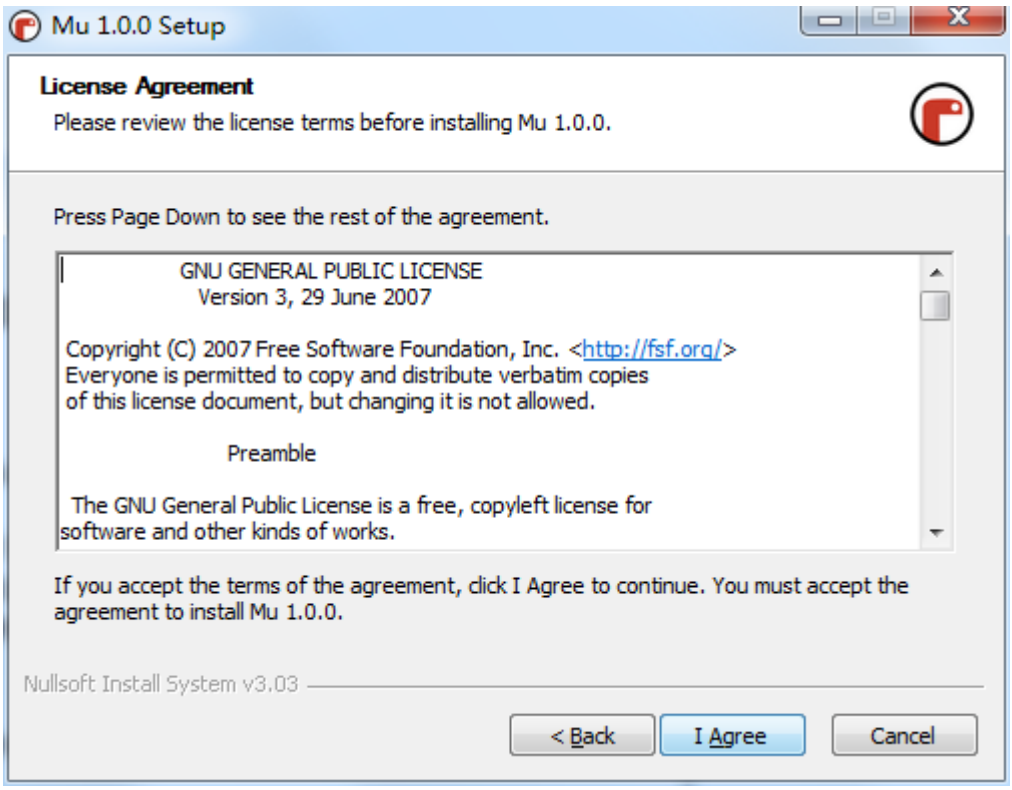
2. We can download it according to what operating system we need, here we choose Windows

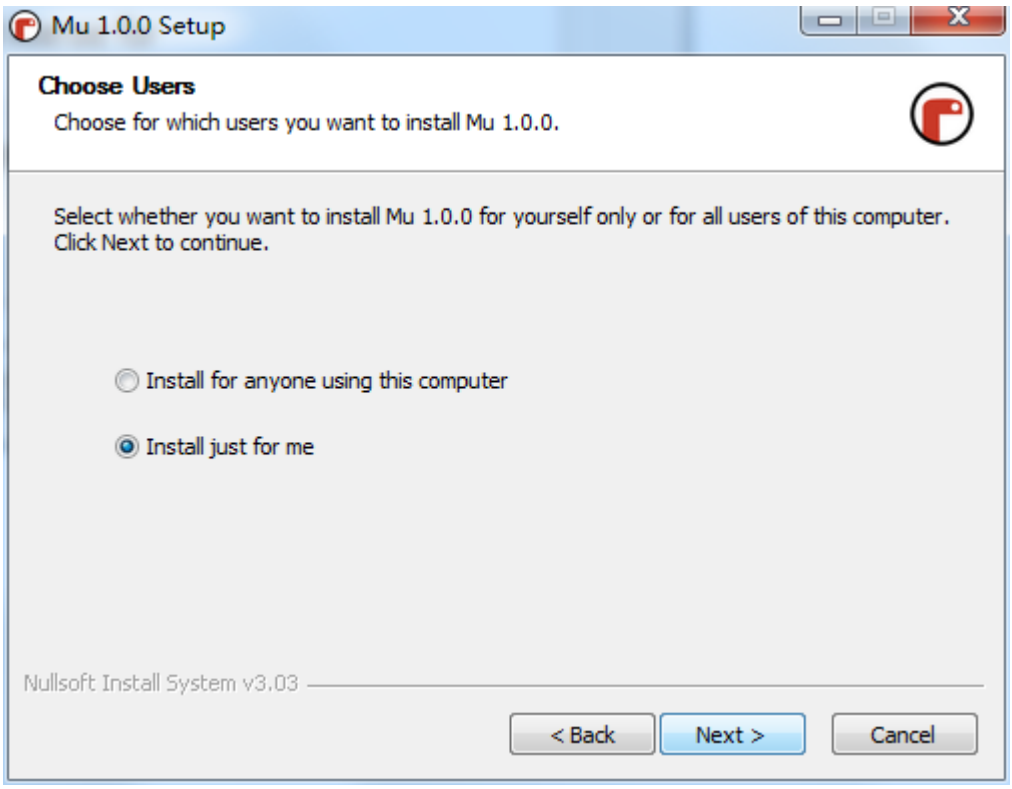


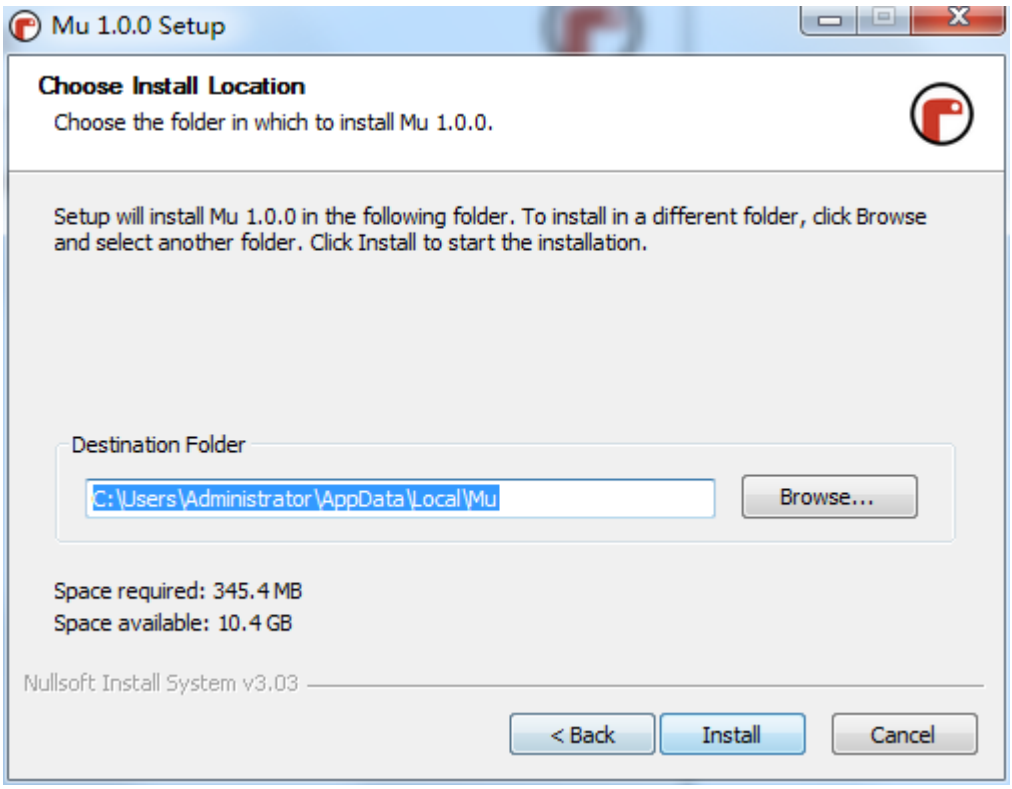
3. Start installation after the download is complete

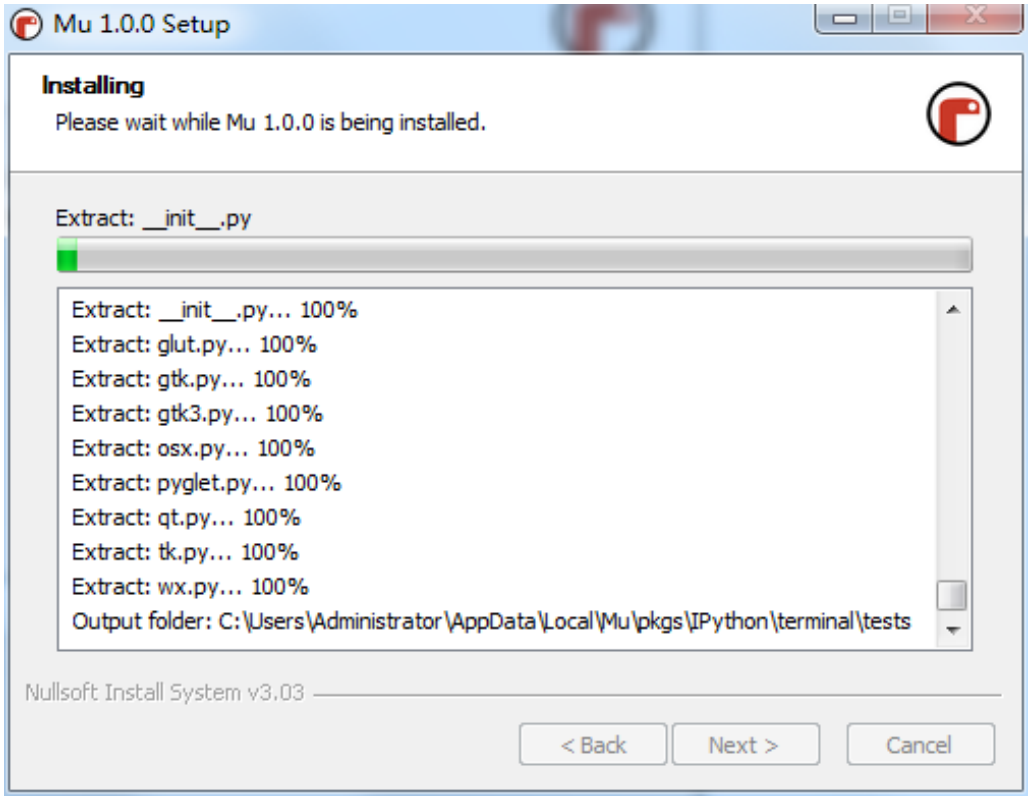
Software installation steps :

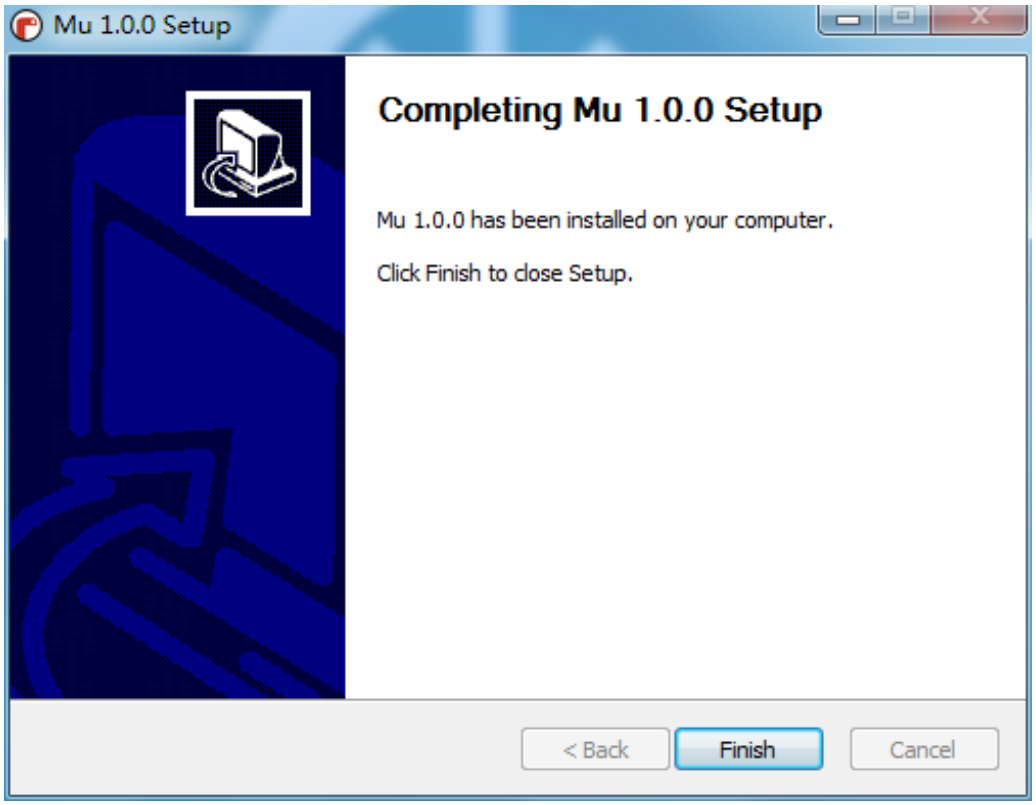




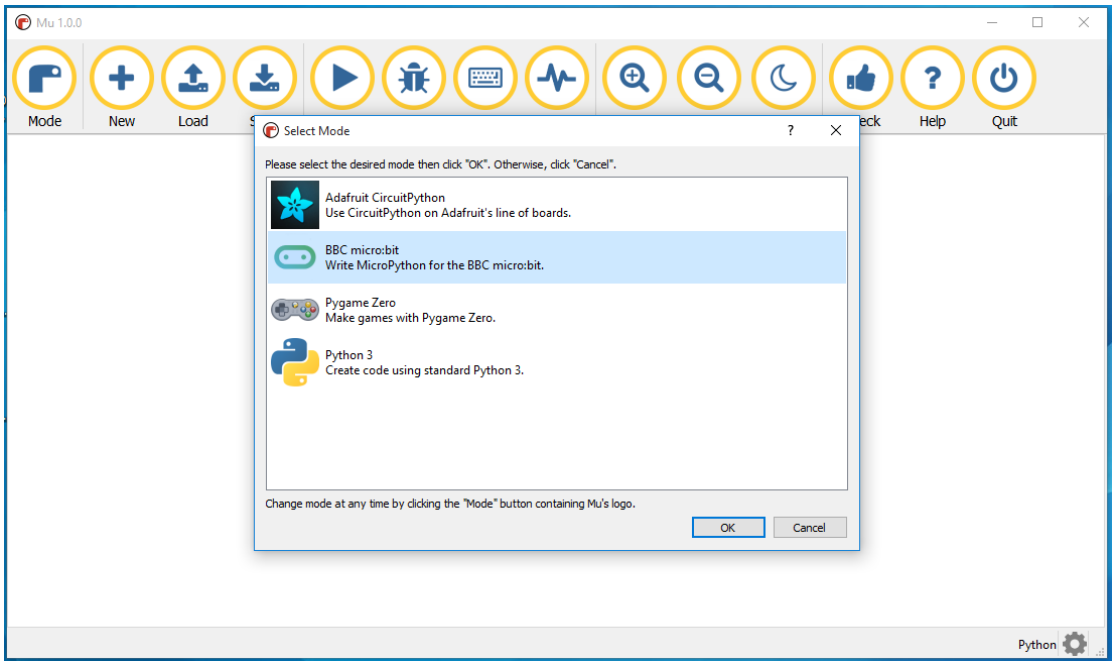

Mu 1.0.0 Setup
Choose Users
Choose for which users you want to install Mu 1.0.0.
Select whether you want to install Mu 1.0.0 for yourself only or for all users of this computer. Click Next to continue.
Install for anyone using this computer
Install just for me
Nullsoft Install System v3.03
< Back
Next >
Cancel


Mu 1.0.0 Setup
Choose Install Location
Choose the folder in which to install Mu 1.0.0.
Setup will install Mu 1.0.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Users\Administrator\AppData\Local\Mu
Browse...
Space required: 345.4 MB
Space available: 10.4 GB
Nullsoft Install System v3.03
< Back
Install
Cancel

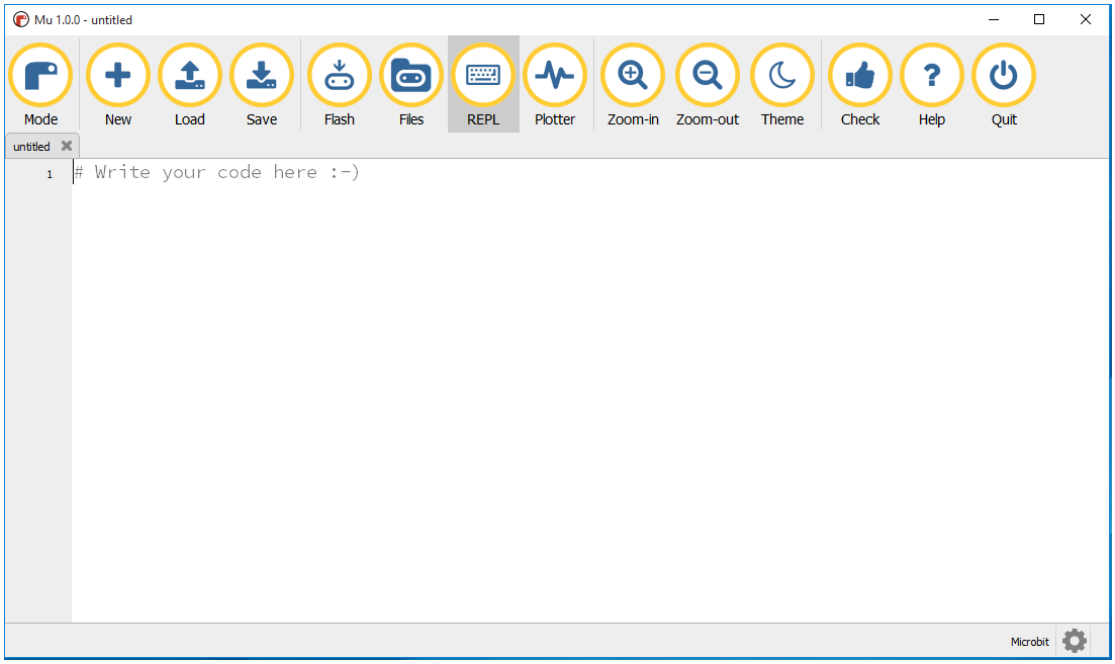

Mu 1.0.0 Setup
Installing
Please wait while Mu 1.0.0 is being installed.
Extract: __init__.py
Extract: __init__.py... 100%
Extract: glut.py... 100%
Extract: gtk.py... 100%
Extract: gtk3.py... 100%
Extract: osx.py... 100%
Extract: pyglet.py... 100%
Extract: qt.py... 100%
Extract: tk.py... 100%
Extract: wx.py... 100%
Output folder: C:\Users\Administrator\AppData\Local\Mu\pkgs\IPython\terminal\tests
Nullsoft Install System v3.03
< Back
Next >
Cancel


Mu 1.0.0 Setup
Completing Mu 1.0.0 Setup
Mu 1.0.0 has been installed on your computer.
Click Finish to close Setup.
< Back
Finish
Cancel

Open the mu, you can see the following interface



Select the second BBC micro: bit, click on ok and you will see the following programming interface.



Click Load, follow the path of your computer to select the first code and then open

5. Add another REPL plug-in we can display data through gorge line, the download link is

http://cdn.kittenbot.cn/mbedWinSerial_16466.exe

When finish downloading, we can use it directly

6. In the next lessons, we will use Python language to program, now we begin the journey of programming.

If you don't know the function of other buttons on this software, you can click help.

Click load -> select the path of the code to open the code in MU

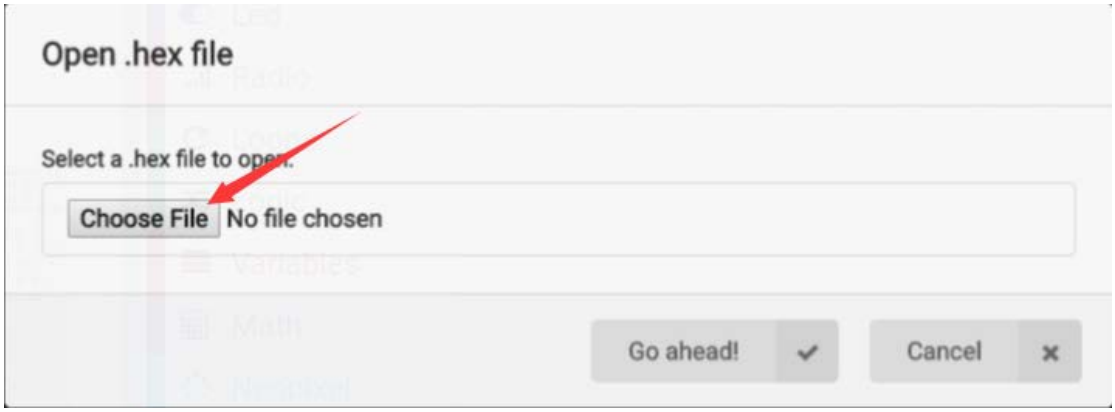
Click flash to download to micro: bit

At this time, you can see a rainbow light on the micro: bit.

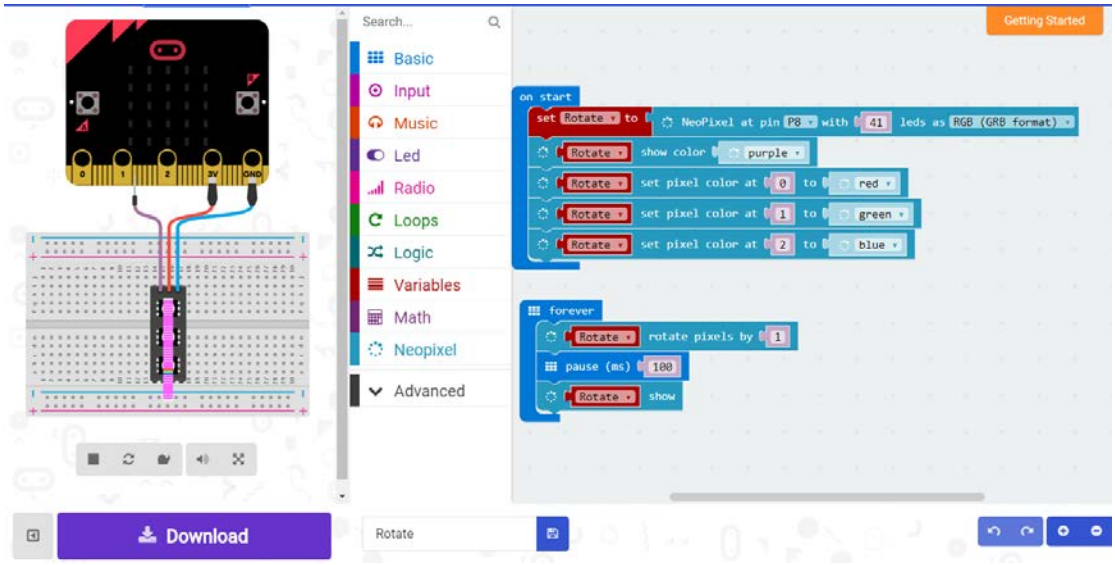
LED Rotate

Makecode

Open to the code editing area, we have provided a set of code for you, firstly click Projects –> Improt File->



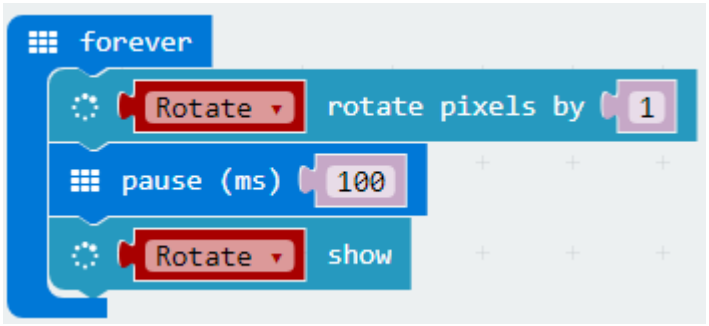
Select the file and click Go ahead; we can see the image as below.



code :

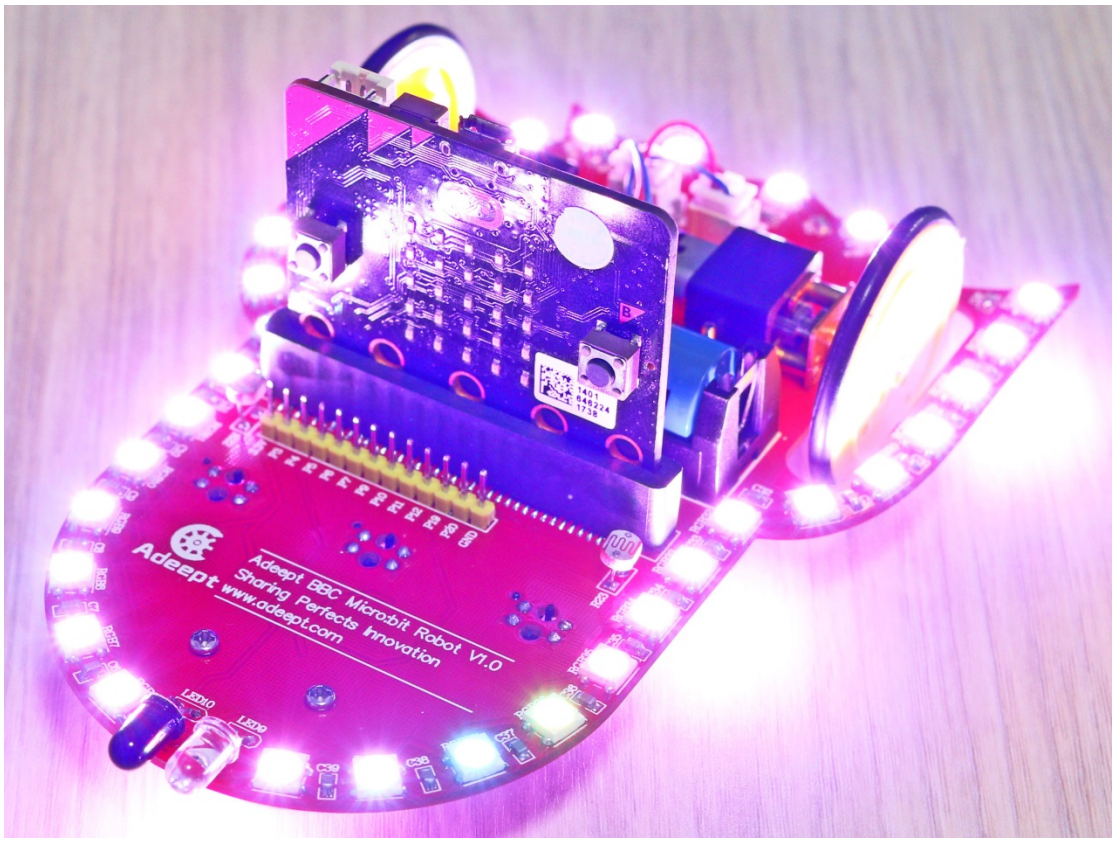
Set the io port of Neopixel's and control 41 LEDs when booting

Set all the LEDs to purple and the LED 0, 1, and 2 to red, green and blue respectively.



Set the shifting display in forever, rotate pixels by 1 and pause 100ms.

Download the program to the BBC micro: bit and you can see the beautiful LED blink.



Mu microPython:

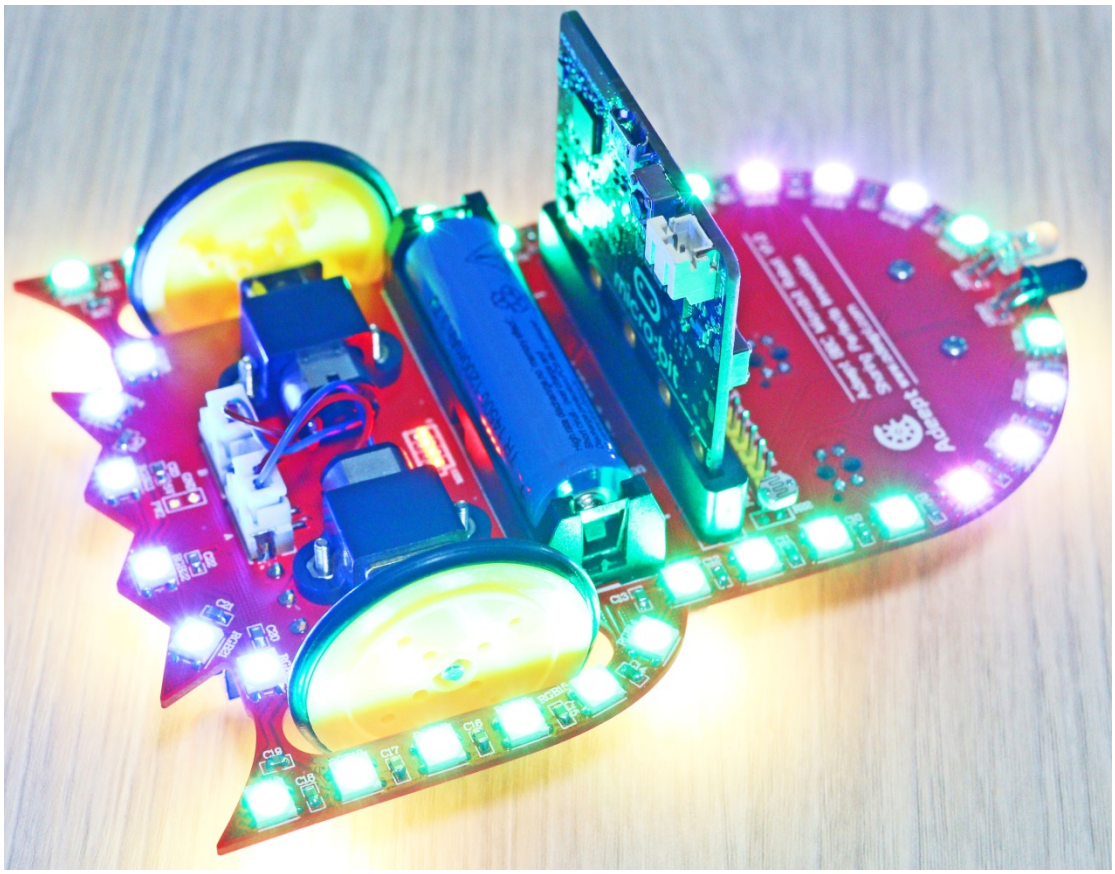
Code: Project 1 –> Neopixel -> lesson 2 -> Rotate

Neopixel

Makecode

Code: Project 1 Neopixel -> lesson 3-> microbit-BBC-NeNeopixel.hex

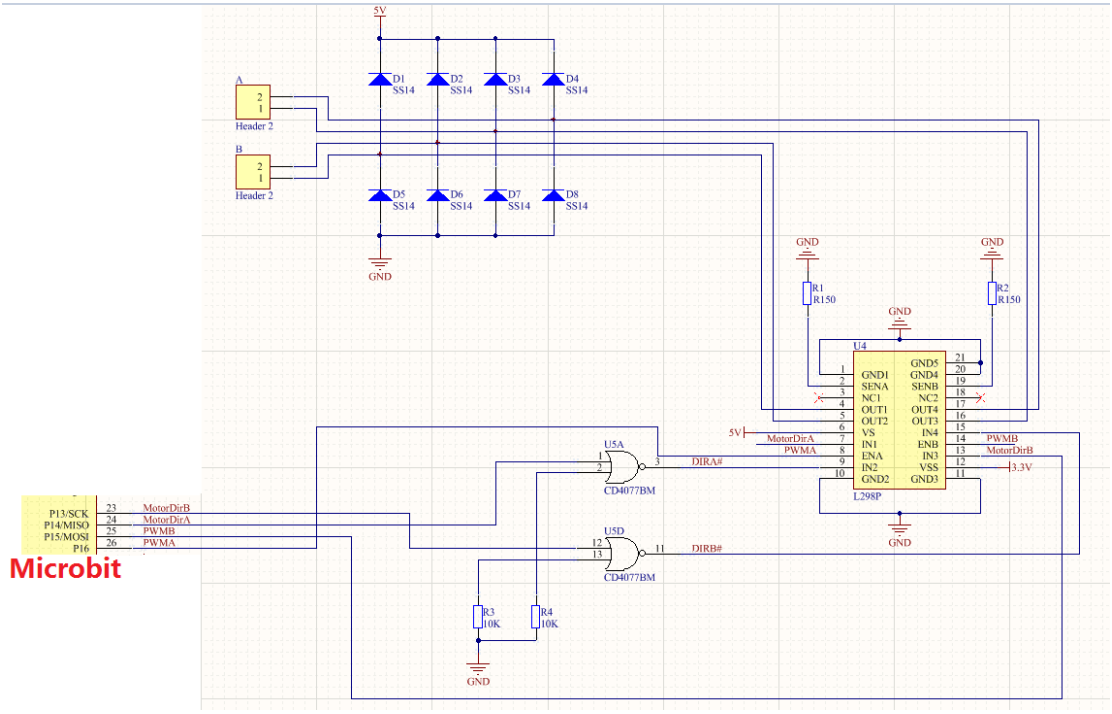
Effect picture of the car :



Mu microPython

Code: Project 1 Neopixel -> lesson 3 –> NeNeopixel.py

Project 2 Motor



Here we use the L298P motor control chip, as shown above, MotorDirA controls the rotation direction of motor B, and MotorDirB controls the rotation direction of motor A. PWMA controls the rotational speed of motor A, PWMB controls the rotational speed of motor A and B respectively, and the PWM input range is

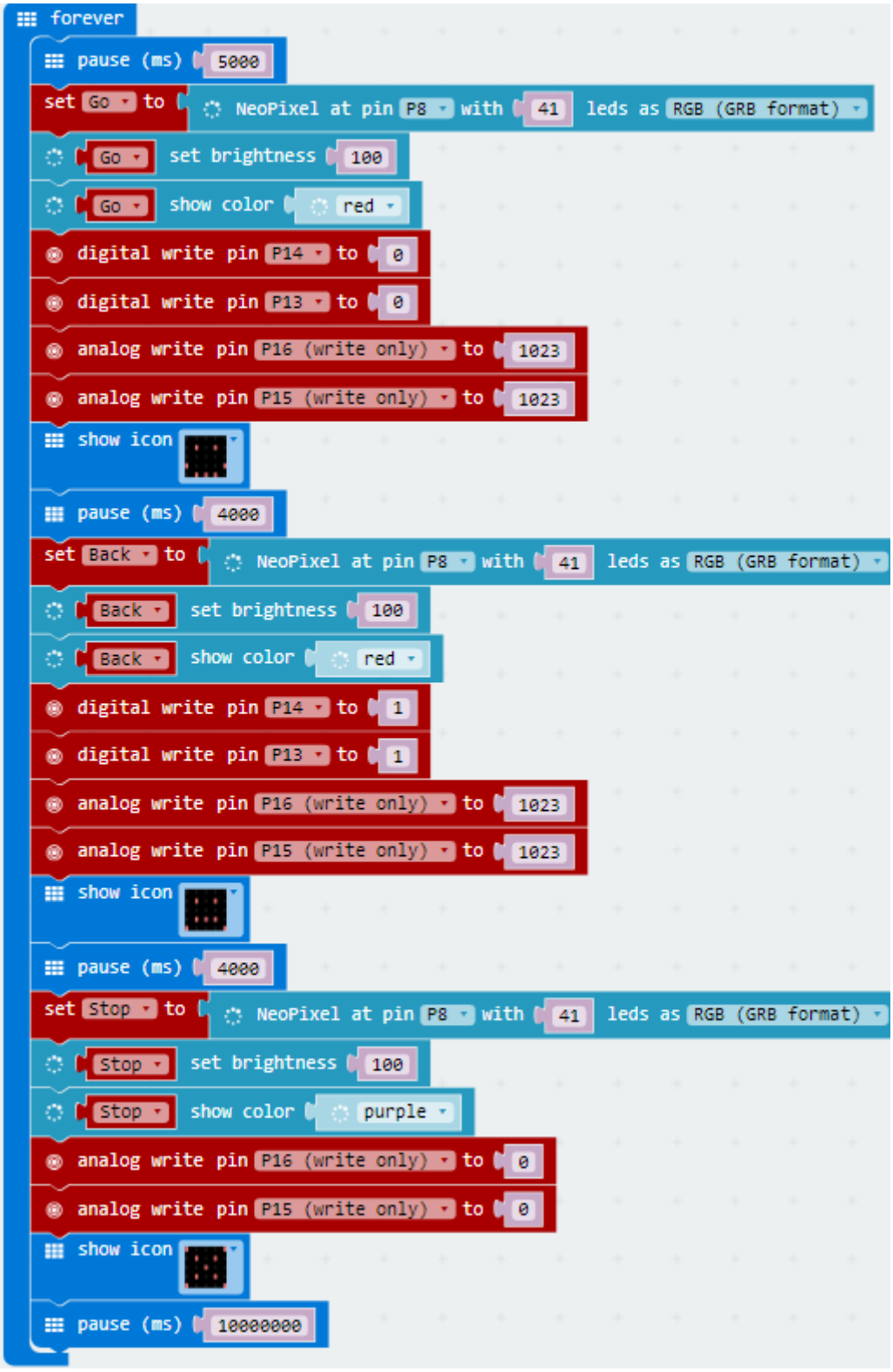
A DC motor is any of a class of electrical machines that converts direct current electrical power into mechanical power. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. Most types produce

rotary motion; a linear motor directly produces force and motion in a straight line.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight motor used for portable power tools and appliances.

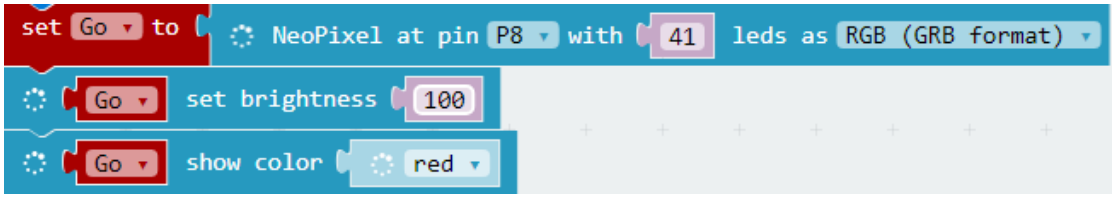
Move forward and backward

Makecode



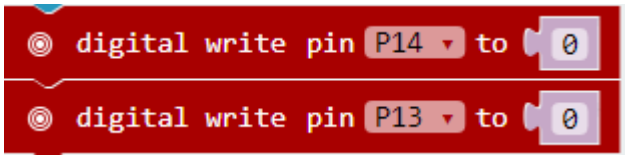
First delay 5000ms in forever to prevent the motor turning on immediately when the car is power- on.

Set Go to Neopixel , io port as pin P8 and with 41 LEDs, then the LEDs show green

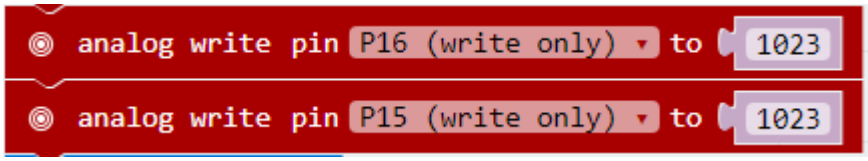


Set the io port of the left and right motor to pin14, pin13 and motor rotation direction to forward.

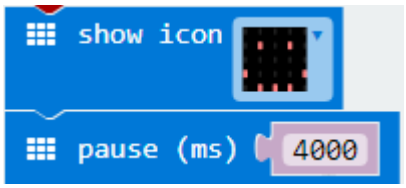
Note: Usually when is 0, the motor rotation direction is forward, and 1 is reverse, it depends on the actual installation of the motor. If the car is moving in the wrong way during the test, you can modify the code.



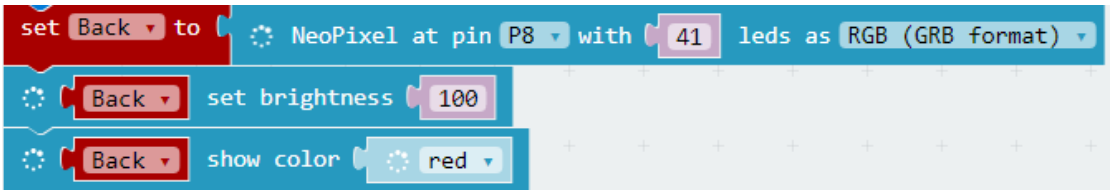
Set the motor to run at full speed (speed is adjustable from 0 to 1023)



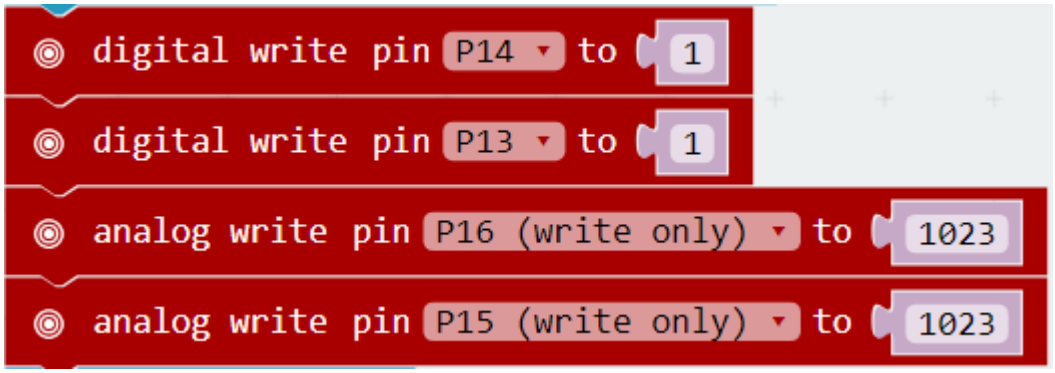
Show icon the smiley face, the program runs at 4000ms.



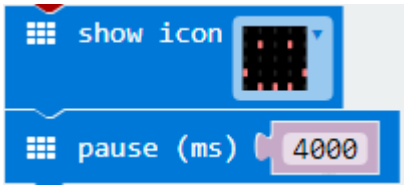
Set Back show color as red, brightness to 100.



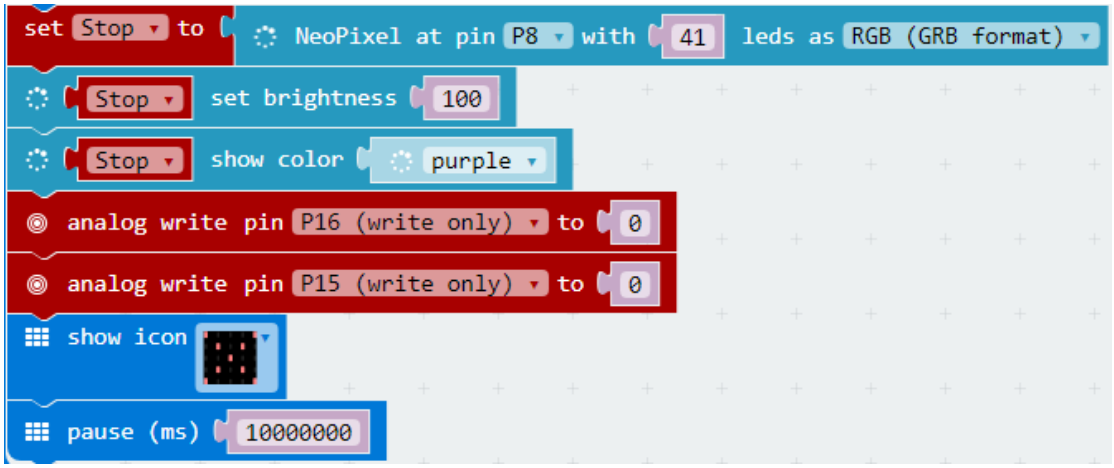
Motors rotate reverse and run at full speed.



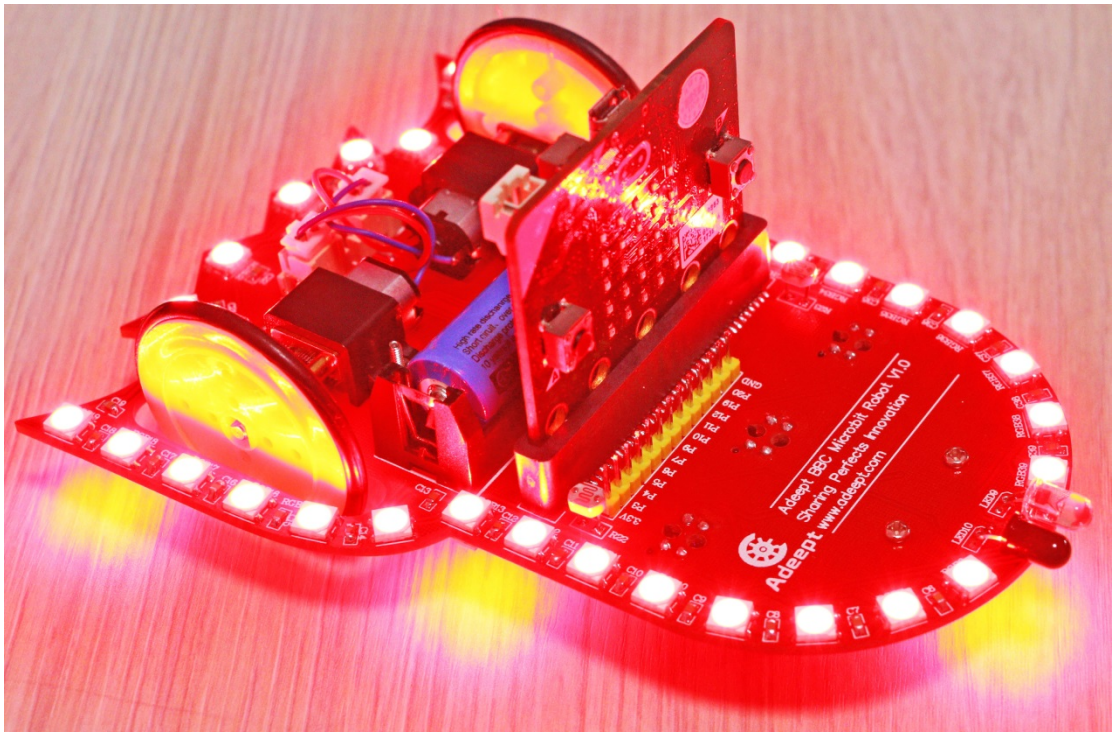
Show icon the crying face, the program runs at 4000ms



Set show color as purple, motors speed as 0, show icon as X when the car stops. And pause 10,000,000 ms is to make it stop for a long time.



Download the code to micro: bit.



Mu microPython

Open relevant code：

```
from microbit import *
import neopixel


# define pin8 control 41 Neopixel
npix = neopixel.NeoPixel(pin8, 41)

# define color
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
nocol = (0, 0, 0)


# define all light
def LightAll(col):
    for pix in range(0, len(npix)):
        npix[pix] = col
```

```
    npix.show()
    return


# define Drive
def Drive(lft, rgt):
    pin14.write_digital(0)
    pin13.write_digital(0)
    if lft < 0:
        pin14.write_digital(1)
        lft = 1023 + lft
    if rgt < 0:
        pin13.write_digital(1)
        rgt = 1023 + rgt
    pin16.write_analog(lft)
    pin15.write_analog(rgt)


while True:
    Drive(500, 500)
    LightAll(green)
    sleep(4000)
    LightAll(red)
    Drive(-500, -500)
    sleep(4000)
    Drive(0, 0)
    npix.clear()
    sleep(100000000)
```

Click flash, download the program to micro: bit directly.

Next you can see the status of the car.

Rotate around

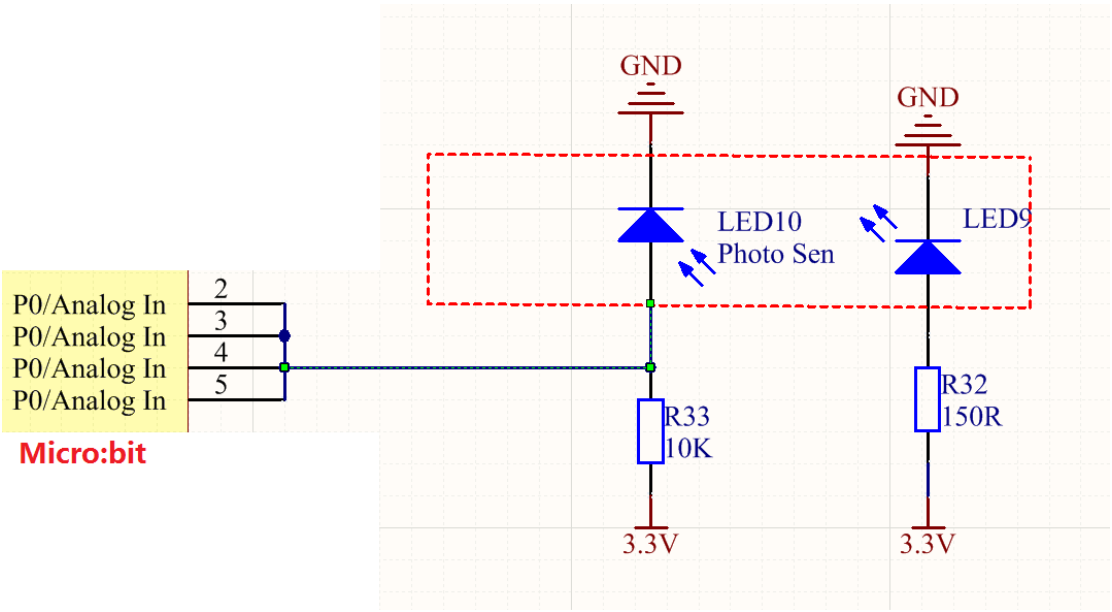
Makecode

Code: Project 2 Motor -> lesson 2-> microbit-L1-Motor.hex

Mu microPython

Code: Project 2 Motor -> lesson 2-> L1-Motor.py

Project 3 Avoid obstacle



As shown in the figure, the infrared sensor LED9 is the transmitting end, the LED10 is the receiving end, and the receiving end receives the signal transmitted by the transmitting end. The receiving end only reacts to the infrared light and will not be interfered by the visible light.

The principle of obstacle avoidance:

There is an infrared distance measuring sensor at the front of the car. We connect it to pin0 of micro: bit to read the analog quantity of the sensor and then determine the distance between the car and obstacle. When the infrared distance measuring sensor detects an obstacle ahead, the car will turn left and loop infinitely, which forms an effect of obstacle avoidance.

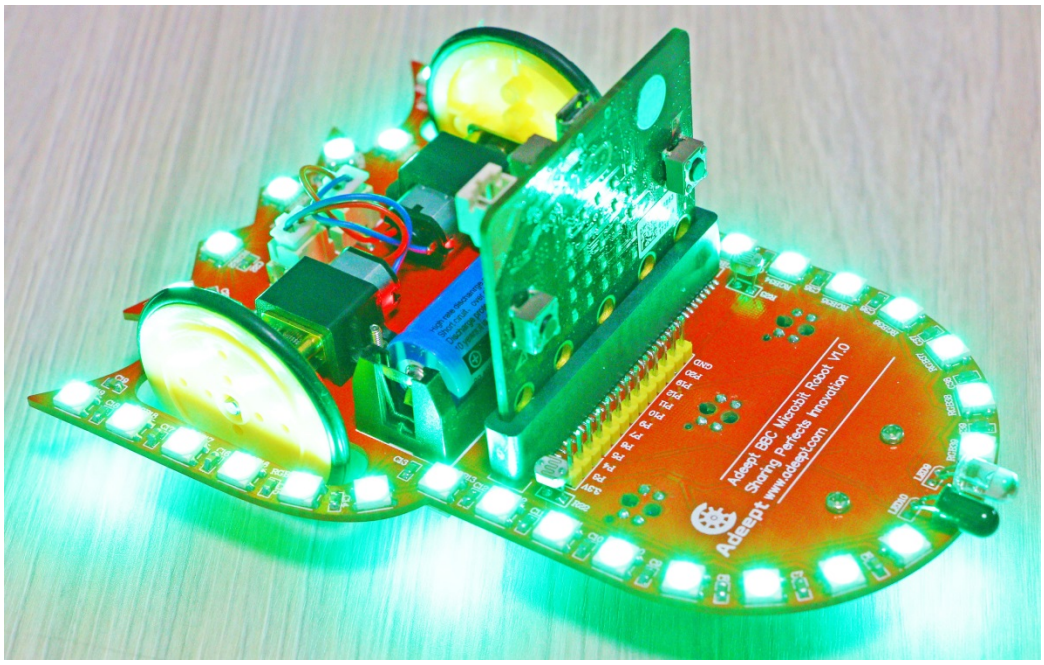
Makecode

Code: Project 3 Avoid_obstacle –> microbit-Avoid-obstacle.hex

Mu microPython

Code: Project 3 Avoid_obstacle –> Avoid-obstacle.py

Effect picture of the car :



Project 4 Follow

The principle of following:

The infrared sensor detects the distance. When the detected object is less than a certain distance away, the car retreats, and when it is greater than a certain distance away, it goes forward. If the sensor does not detect any item, it stops moving.

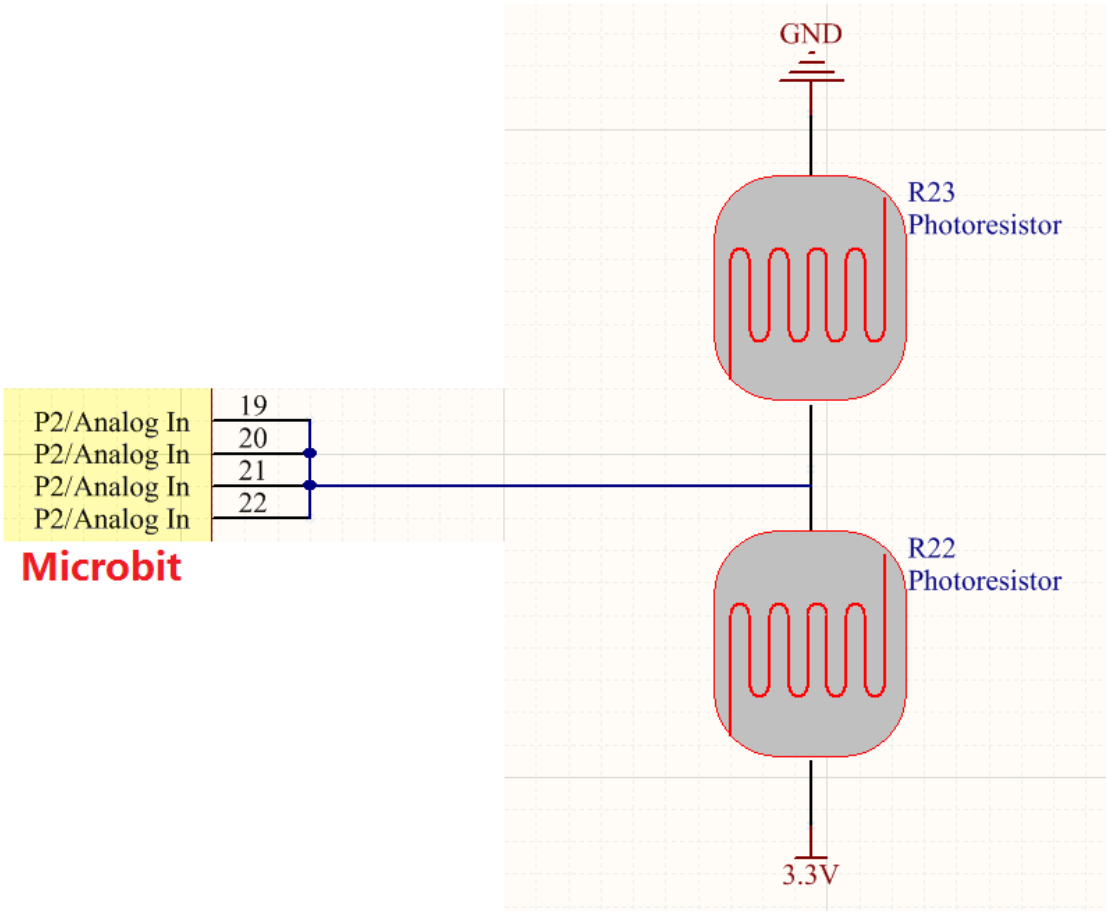
Makecode

Code : Project 4 Follow -> microbit-Follow.hex

Mu microPython

Code: Project 4 Follow -> Follow.py

Project 5 Follow light



A photoresistor is a light-controlled variable resistor. The resistance of a photoresistor decreases with the increasing incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits.

A photoresistor is made of a high resistance semiconductor. In the dark, a photoresistor can have a resistance as high as a few megohms (MΩ), while in the light, a photoresistor can have a resistance as low as a few hundred ohms. If incident light on a photoresistor exceeds a certain frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump

into the conduction band. The resulting free electrons (and their hole partners) conduct electricity, thereby lowering resistance. The resistance range and sensitivity of a photoresistor can substantially differ among dissimilar devices. Moreover, unique photoresistors may react substantially differently to photons within certain wavelength bands.

With the increase of the light intensity, the resistance of photoresistor will be decreased. The voltage of P1 port in the figure will rise.

The two photoresistors shown in the figure share a control port. When R22 shed by strong light, the value of P2/AnalogIn will decrease. When R23 shed by strong light, the value of P2/AnalogIn will rise.

The data read by the serial port refers to the following table:

Light shed on the left photoresistor	No light	Light shed on the right photoresistor
< 400	400 ~ 750	> 750

The principle of following light:

There are two photoresistors respectively on the left and right sides of the car. The flashlight flash on either side of the photoresistor and the difference in the value of the each photoresist will be read, which ensures the light intensity of both sides can be determined. And the car moves in the direction of strong light.

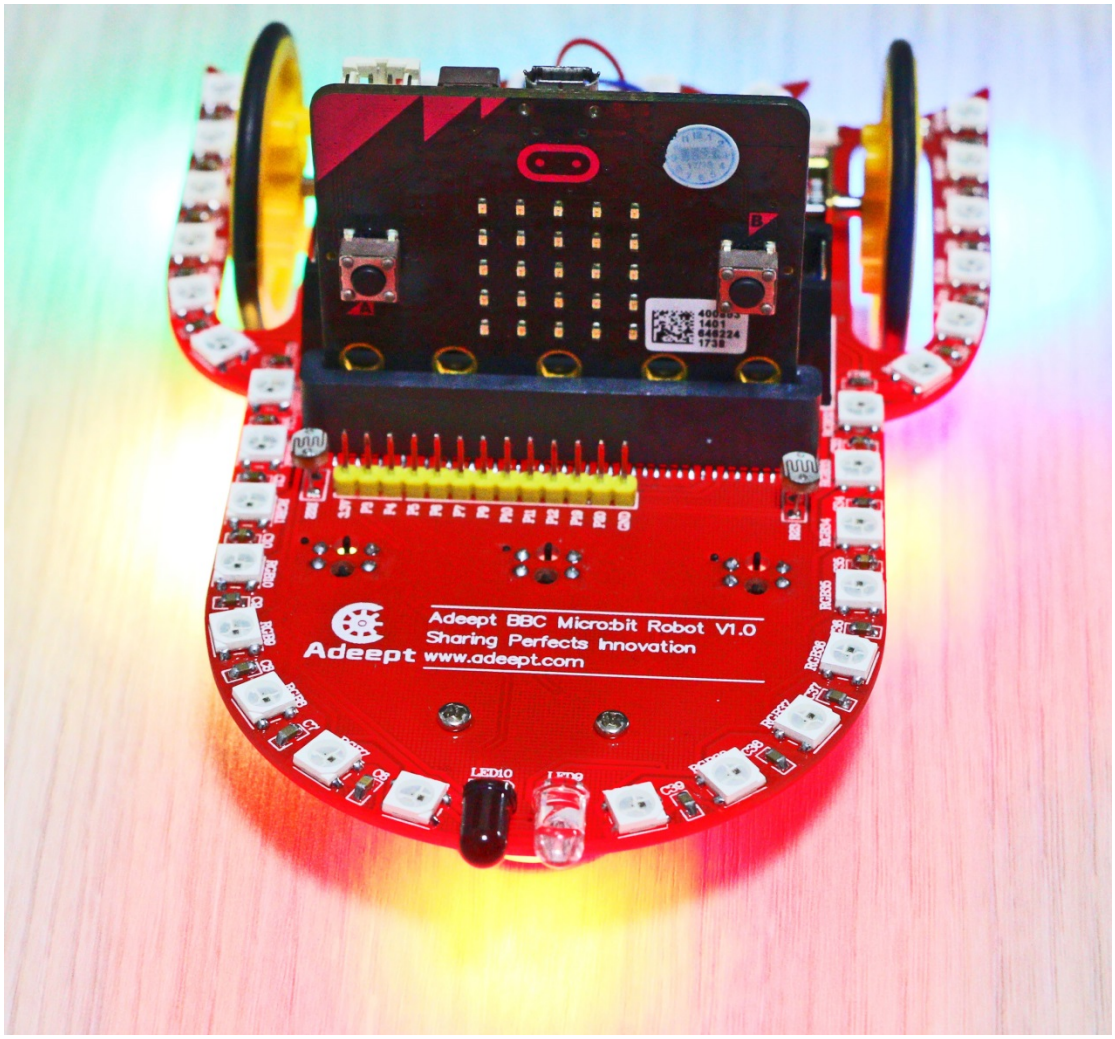
Makecode

Code:Project 5 Follow_Light -> microbit-Follow_Light.hex

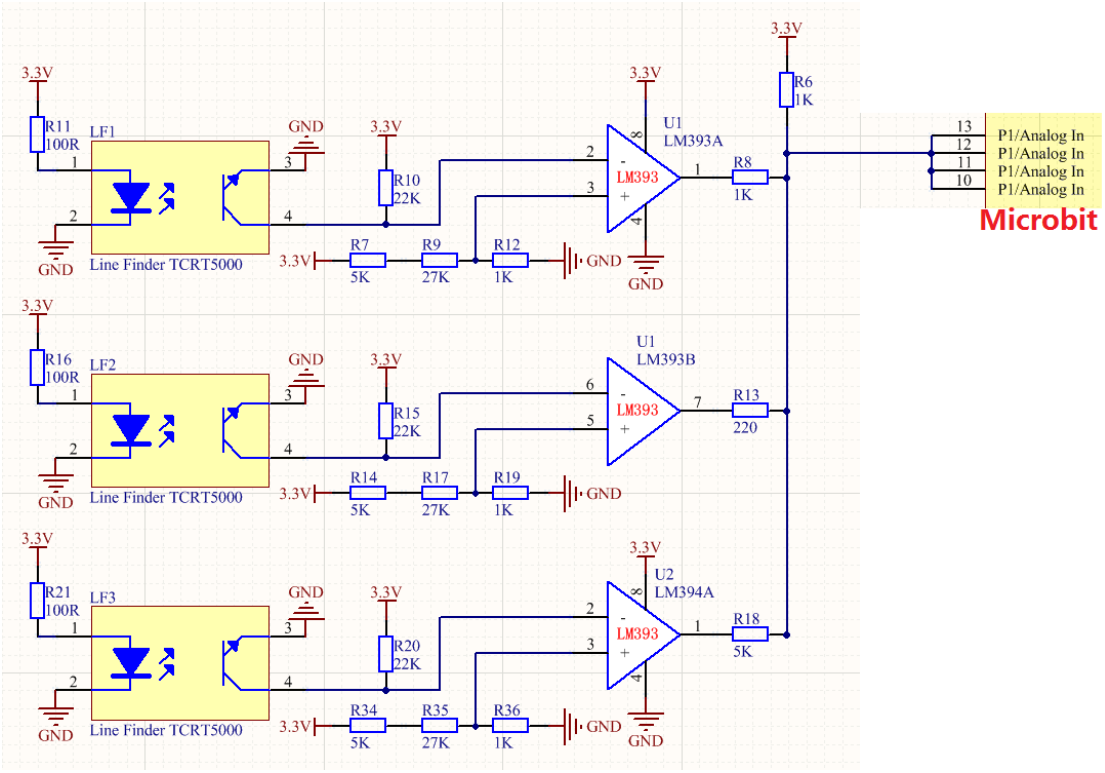
Mu microPython

Code: Project 5 Follow_Light -> Follow_Light.py

Effect picture of the car :



Project 6 Find line



As the circuit diagram, one control port of is connected with three tracking sensors. Each tracking sensor is connected with a comparator. When the tracking sensor touches the black line, the analog voltage value read by P1 / AnalogIn will change. The range of values is shown in the table below.

The principle of finding line:

Three tracking sensors that can recognize black tracks are installed on the car. The infrared signal emitted by the sensor is sent to the comparator after it is received. If the sensor recognizes the black track, the comparator will output a signal to the mcu. Here the way of reading analog quantity is applied.

The analog quantity read by tracking sensor and the motor direction of the car :

analog quantity	direction
800-899	Turn left
200-299	Go forward
500-599	Turn right

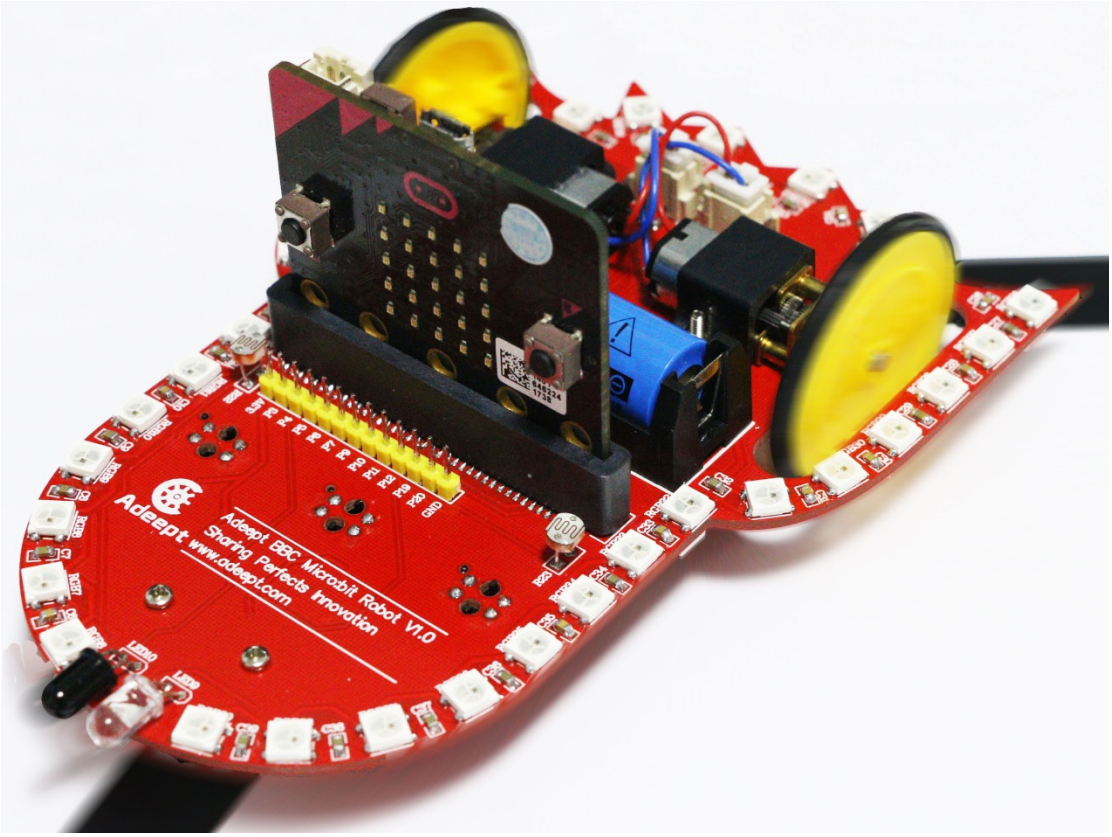
Makecode

Code: Project 6 Find_Line –> microbit-Find_Line.hex

Mu microPython

Code: Project 6 Find_Line –> Find_Line.py

Effect picture of the car :

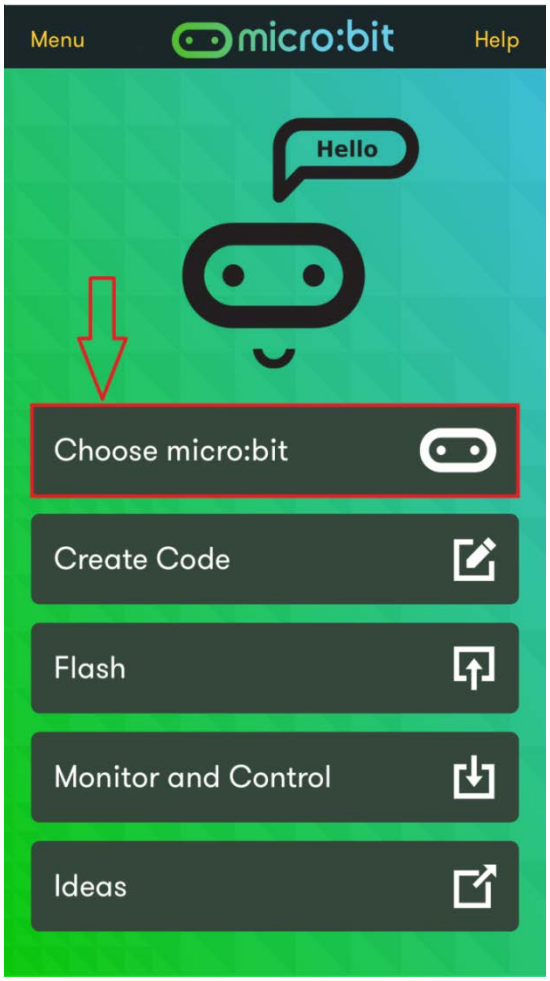


Project 7 Mobile Bluetooth control

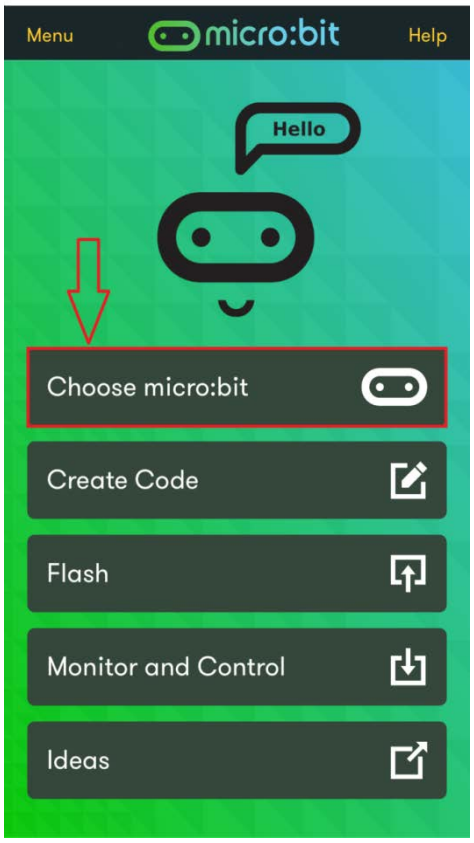
First import the microbit-IPhone-control.hex file into makecode and download the program to the BBC microbit

Next, connect the microbit to the mobile phone Bluetooth.

1. Search micro: bit in mobile phone store and download it onto mobile phone
2. When we start the application, we'll see the following interface.

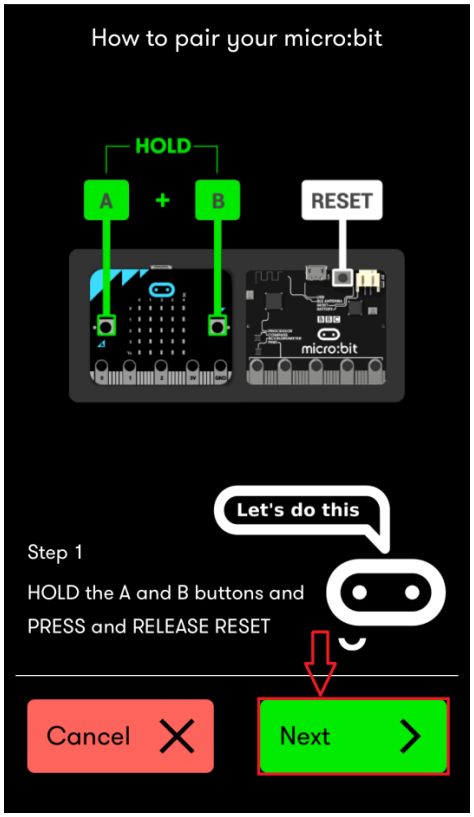


4. Connect micro: bit with the computer
5. Turn on the Bluetooth of the mobile phone, and go to the downloaded micro: bit page, then click “Choose micro: bit”

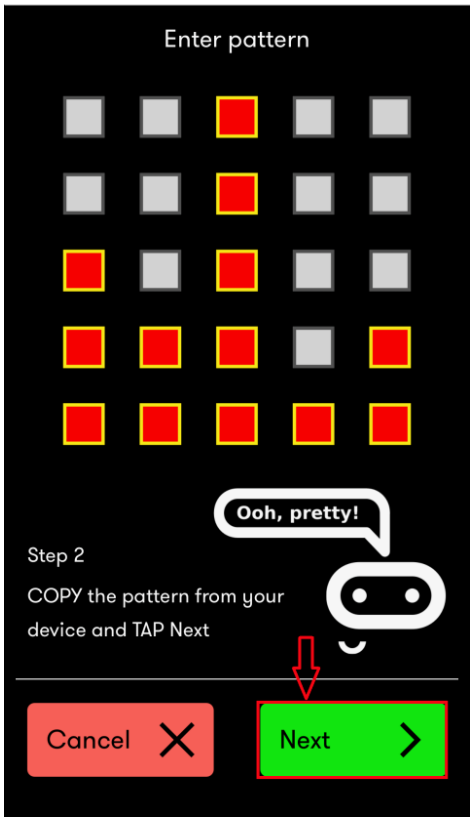


6. Click “pair a micro: bit”

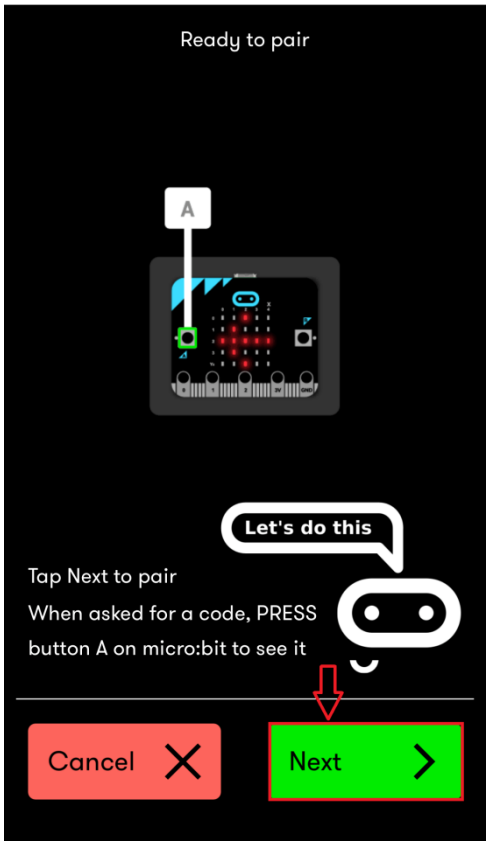
7. As the picture shows, we press the buttons A, B at the same time, and press the button “reset”, and then loosen button “reset”, when the LED matrix lamp displays “pairing mode”, please loosen button A and B, then go on.



8. Light up the lamps on mobile phone that have been lighten in matrix lamps, as the picture shows: (notice: each micro: bit shows different when matched), then click “next”.



9. Now we should press the button “A” on micro:bit

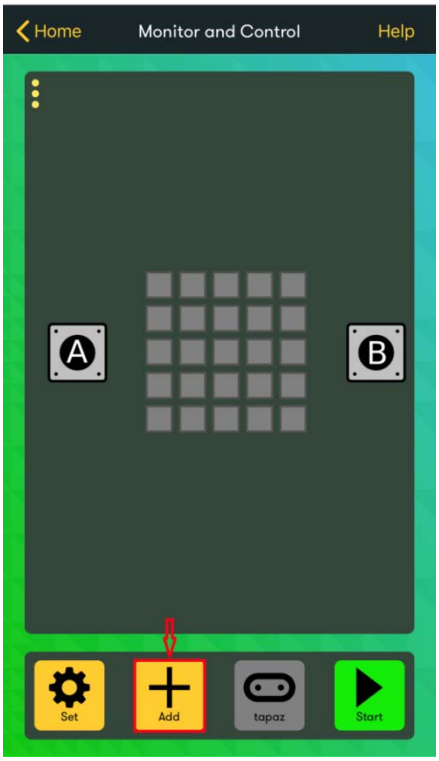


10. Then we can see that the mobile phone has been connected with the micro: bit successfully, (notice: before then IOS users run into the situations that unable to connect or unable to connect again after disconnection, if you meet this kind of situation, you can change Android to solve this problem.)

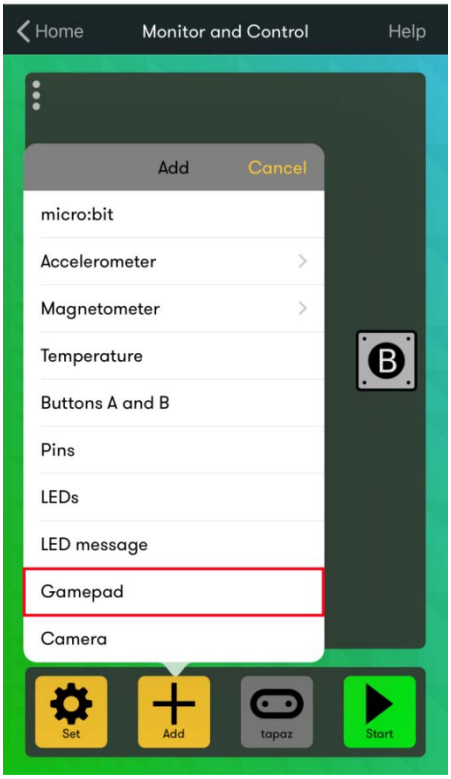
Now, we can use mobile phone to control the display of micro” bit lamp

Turn back to the homepage and click “monitor and control”

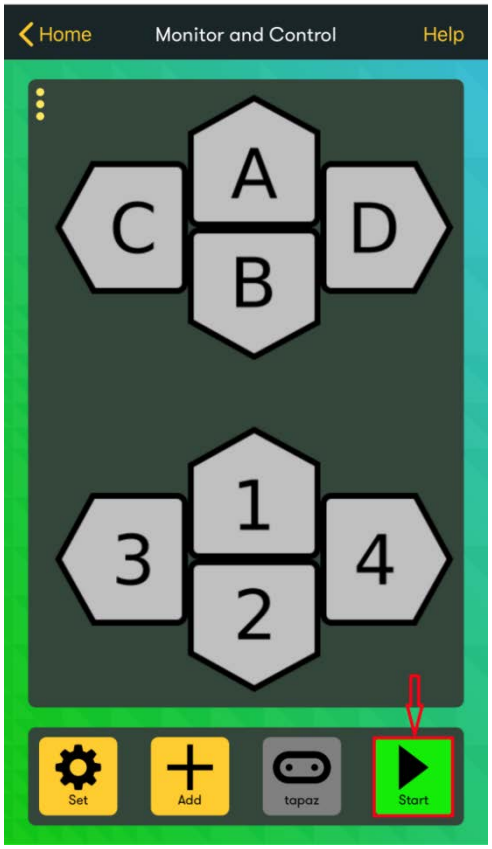
You can see the following interface, and then click “Add”


Home
Monitor and Control
Help
A
B
Set
Add
tapaz
Start

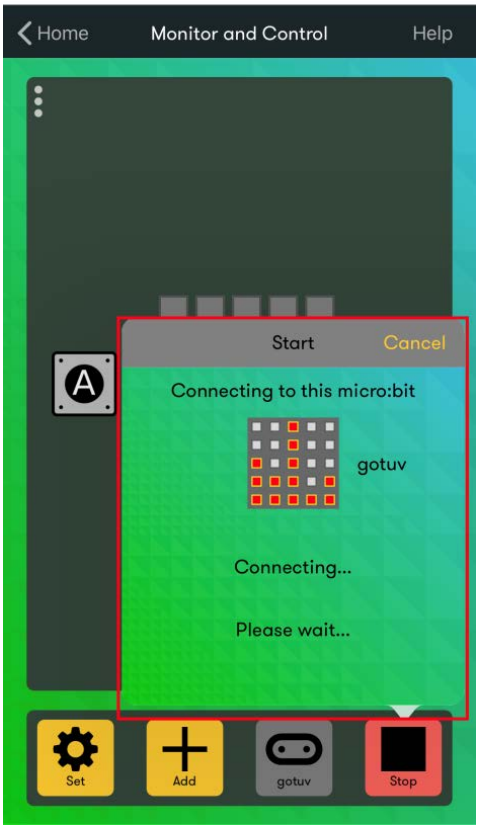
Add Gamepad


Home
Monitor and Control
Help
Add
Cancel
micro:bit
Accelerometer
Magnetometer
Temperature
Buttons A and B
Pins
LEDs
LED message
Gamepad
Camera
B
Set
Add
tapaz
Start

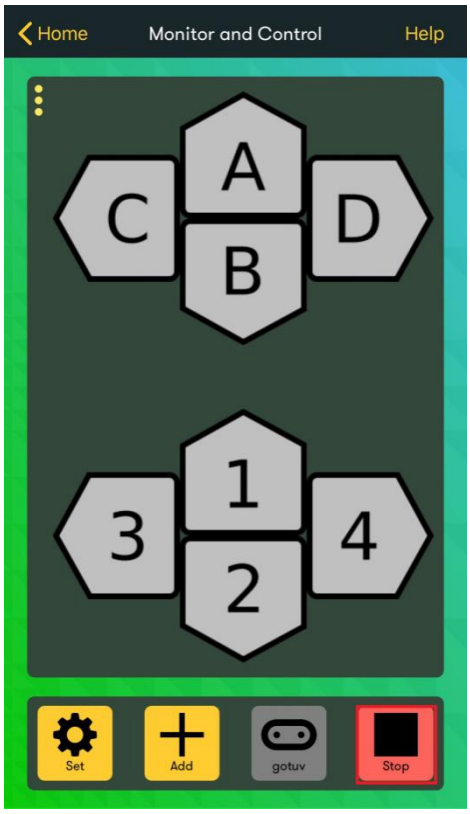
Now you can see the interface of a remote controller, click start to connect micro:bit.



You can see that it is now automatically connecting



Note: The following situation will occur in the first pairing most of the time. If that happen, click “connect” directly. The following interface will appear if connect successfully.



Now we can control the car

Introduction of function keys:

A	B	C	D
Forward	backward	Turn left	Turn right
1	2	3	4
Stop	Find line	Avoid obstacle	Find light

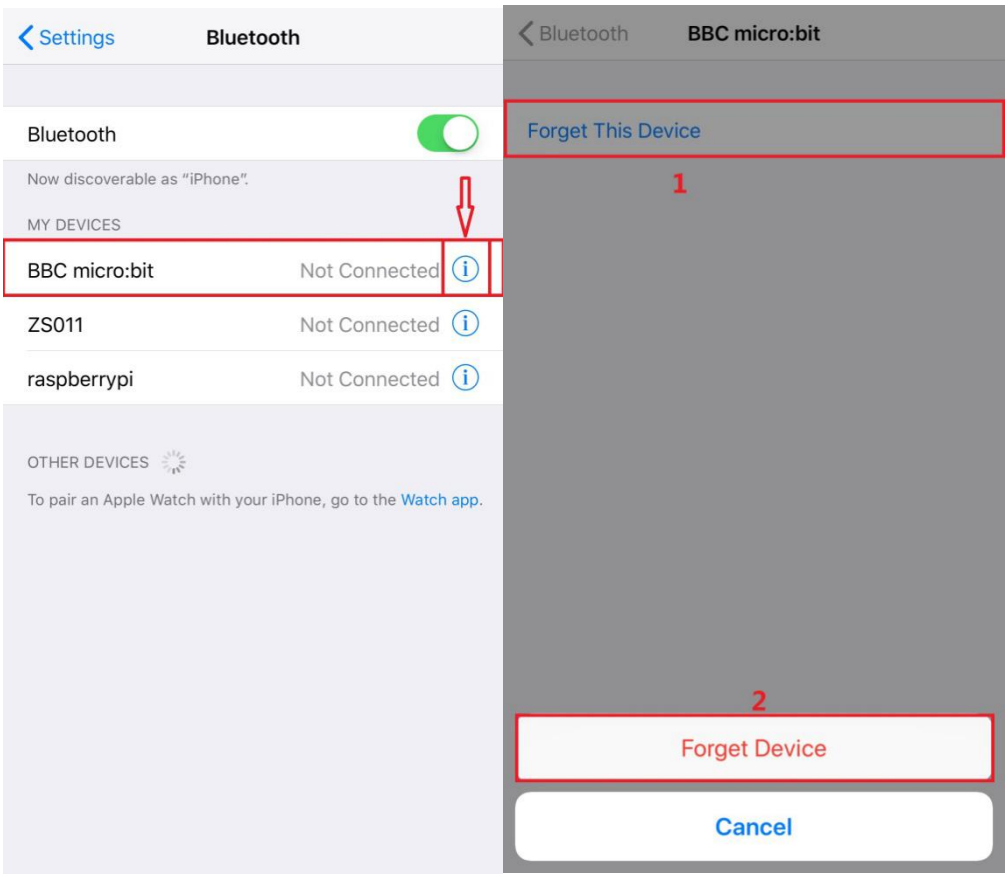
Code：Project 7 Mobile phone bluetooth –> microbit-IPhone-control.hex

Frequently asked question:

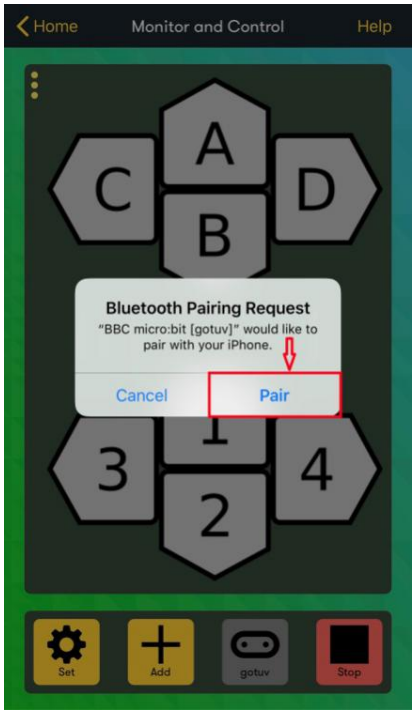
There’s no problem connecting to the phone for the first time after downloading the program to micro: bit, but download again, connections keep failing.

Solution:

Pone settings -> Bluetooth-BBC micro:bit –> Forget This Device



We also need to reconnect on the mobile application. Successfully paired with mocro:bit before, we just need to click “monitor and control” directly to enter the remote control interface and then click “start”. The following interface will appear. Click “pair”, and the following two “connect” to successful connection.



At present, only the ios mobile phones have the control interface, Android is unsupported.

Author	Version	Modified date
Jason	V1.0	2018/10/10

Adeept

Sharing Perfects Innovation

E-mail: support@adeept.com
website: www.adeept.com