

RGB Light

Learning goals

In this lesson, we mainly learn how to control RGB Light on the Super:bit expansion board.

Code

```
1 from microbit import *
2 import neopixel
3
4 np = neopixel.NeoPixel(pin12, 4)
5 display.show(Image.HAPPY)
6
7
8 def RGB_Scintillation(first, num):
9     global np
10    np.clear()
11    while first <= num:
12        np.clear()
13        np[0] = (255, 0, 0)
14        np[1] = (255, 0, 0)
15        np[2] = (255, 0, 0)
16        np[3] = (255, 0, 0)
17        np.show()
18        sleep(500)
19        np.clear()
20        np.show()
21        sleep(500)
22        first += 1
23    else:
24        first = 0
25
26 np.show()
```

```
27
28 def RGB_Waterlamp(first, num):
29     global np
30     np.clear()
31     while first <= num :
32         np[0] = (255, 0, 0)
33         np.show()
34         sleep(200)
35         np.clear()
36         np[1] = (255, 0, 0)
37         np.show()
38         sleep(200)
39         np.clear()
40         np[2] = (255, 0, 0)
41         np.show()
42         sleep(200)
43         np.clear()
44         np[3] = (255, 0, 0)
45         np.show()
46         sleep(200)
47         first += 1
```

```

48
49
50 def RGB_Breathinglamp(first, num):
51     global np
52     np.clear()
53     a = 0
54     while first <= num :
55         while a > 0 and a < 255:
56             a = a + 1
57             np[0] = (a, 0, 0)
58             np.show()
59             sleep(5)
60             np.clear()
61             first += 1
62
63 RGB_Breathinglamp(1, 3)
64 RGB_Waterlamp(1, 3)
65 RGB_Scintillation(1, 3)
66

```

Import the neopixel library for RGB lights and the microbit library.

The corresponding RGB values of each color, such as red, orange, yellow, green, blue, purple, and white, can be found online to define a tuple representing the RGB value of the color.

`display.show (Image.HEART)`: Micro:bit dot matrix display “heart” pattern;

`np = neopixel.NeoPixel (pin12, 4)`: Initialize the RGB programming lamp library.

The first parameter is the pins of the RGB lamp, and the second parameter is the number of RGB lamps.

`np.clear ()`: Clear RGB light display;

`np [i] = (R, G, B)`: This statement is to sort the i-th lamp, that is, from the LED1 on the superbit board, to the rear, RGB is the basic value of the three-color lamp;

`np.show ()`: Refresh the colorful lights display, if you do not run this function, the above setting color will not work;

`microbit.sleep (1000)`: Delay 1000 milliseconds.

`RGB_Breathinglamp (1, 3)`: Marquee function, loop three times;

`RGB_Waterlamp (1, 3)`: Water light function, loop three times;

`RGB_Scintillation (1, 3)`: Breathing light function, loop three times.

Assembly steps

Please refer to the **1.Omnibit installation steps** in the **1.Assembly steps** folder for building blocks assembly steps.

About wiring

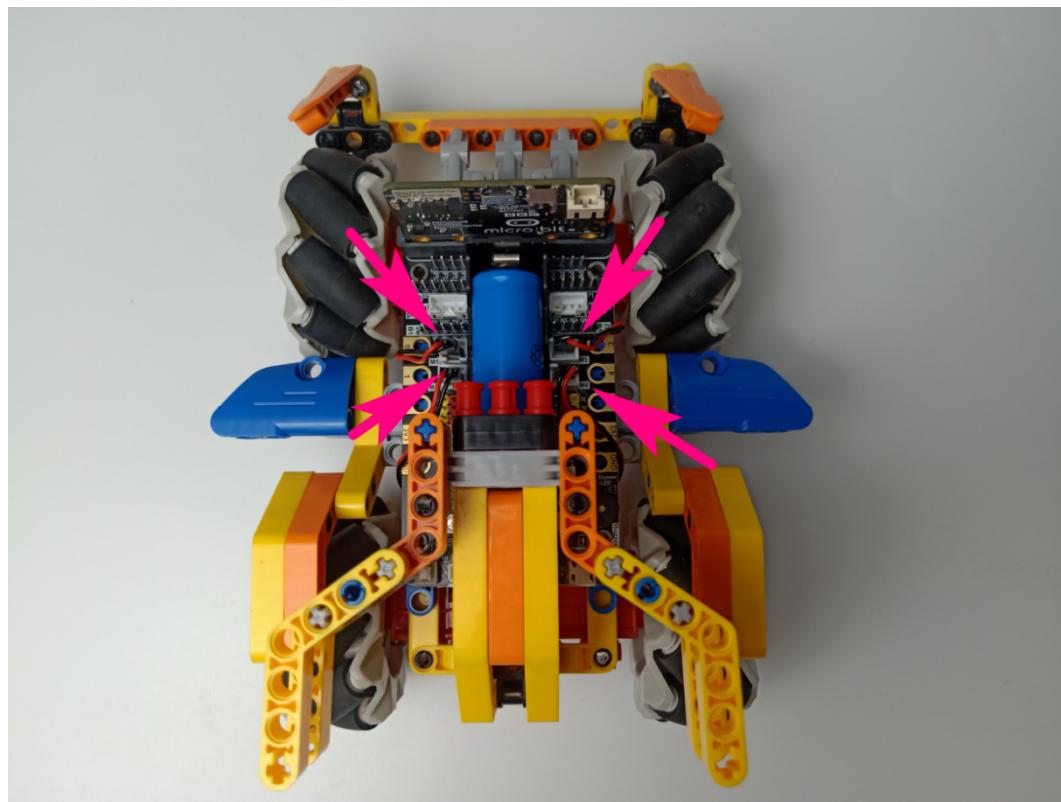
The left front motor is connected to the M1 interface of the Super:bit expansion board. The black line is on the battery side;

The left rear motor is connected to the M2 interface of the Super:bit expansion board, The black line is on the battery side;

The right front motor is connected to the M3 interface of the Super:bit expansion board, The black line is on the battery side;

The right rear motor is connected to the M4 interface of the Super:bit expansion board, The black line is on the battery side.

As shown below.



Programming and downloading

1. You should open the Mu software, and enter the code in the edit window, , as shown below.

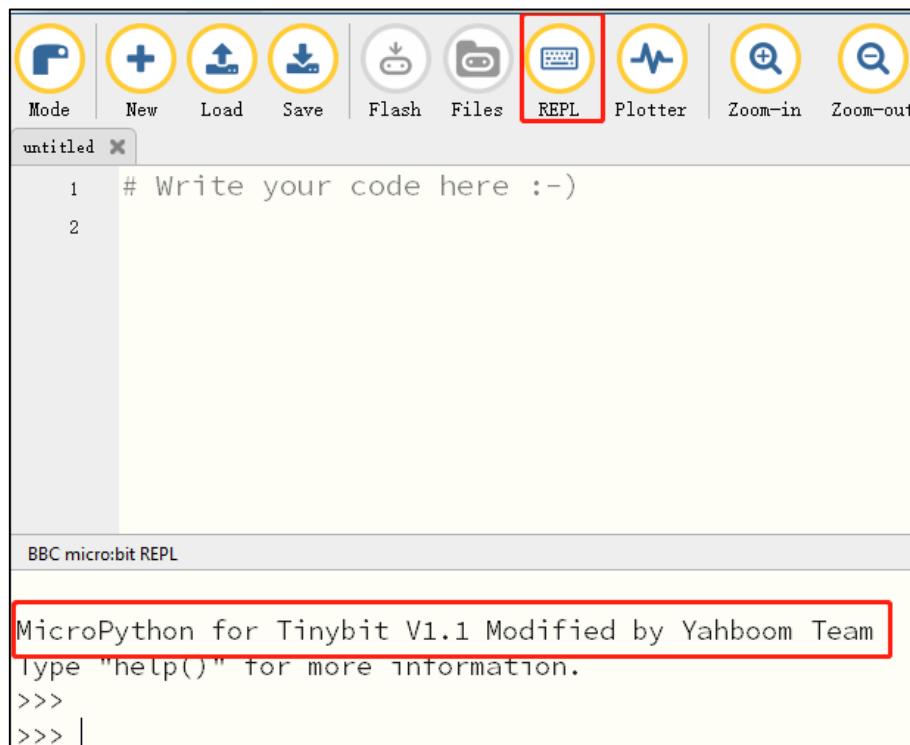
Note! All English and symbols should be entered in English, and the last line must be a space.

```
Voice control light.py
6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0
```

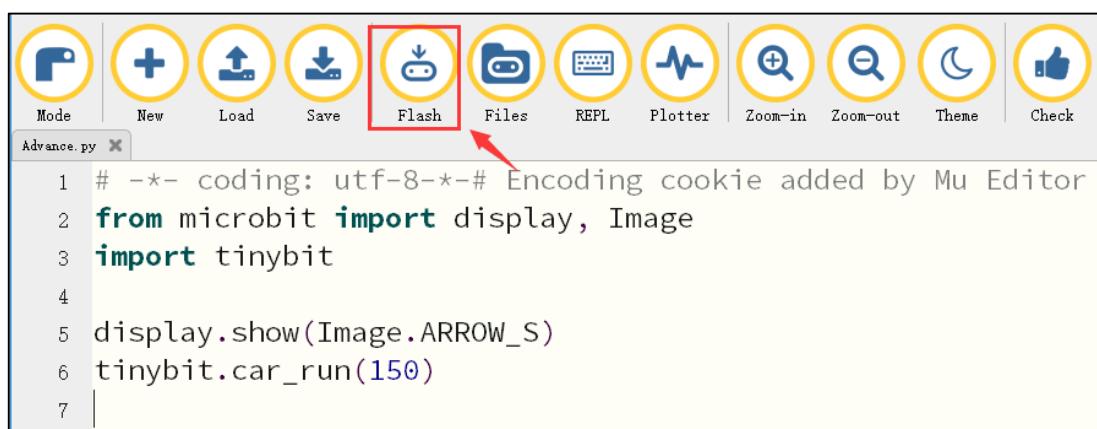
2. You can click the “**Check**” button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.

```
Voice control light.py
6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0
13
14
15 while True:
16     voice = tinybit.getVoicedata()
17     if voice > 100:
```

3. Click “**REPL**” button, check whether the tinybit library has been downloaded. If not, please refer to the [preparation before class]---> [Python programming]



4. Click the “Flash” button to download the program to micro:bit board.



If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Superbit library hex file we provided to the micro: bit board.

For the specific method of adding library files, please refer to 【1.Preparation before class】---【Python programming】

Experimental phenomena

After the program is successfully downloaded. The micro:bit dot matrix will display the smile and turn off all RGB light. Then, red RGB light will flash 3 times, the water light 3 times, the marquee 3 times, the breathing light 3 times, And keep the loop in such a state.