

## Control RGB color

### 1.Learning goals

In this lesson, learn to control the RGB light colors on the superbit expansion board using Python programming.

#### Code:

```

1  from microbit import *
2  import neopixel
3
4  Red = (255, 0, 0)
5  Orange = (255, 165, 0)
6  Yellow = (255, 255, 0)
7  Green = (0, 255, 0)
8  Blue = (0, 0, 255)
9  Dark_Violet = (148, 0, 211)
10 White = (255, 255, 255)
11
12 color = (Red, Orange, Yellow, Green, Blue, Dark_Violet, White)
13
14 display.show(Image.HAPPY)
15
16 np = neopixel.NeoPixel(pin12, 1)
17 i = 0
18 while True:
19     np.clear()
20     np[0] = color[i]
21     np.show()
22     sleep(1000)
23     i = i+1
24     if i > 6:
25         i = 0
26

```

Import micro:bit library, RGB light library neopixel;

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.

Define a tuple that contains the previously defined color RGB tuples, and then extract the pair colors by index.

`display.show(Image.HAPPY)`: display smile;

`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;

`np[0] = color[i]`: Number 0 means the first RGB light.

`np.show()`: Refresh the colorful lights display. If this function is not run, the above setting colors will not work;

### Sleep(500): delay

`i = i + 1`, `i` is used as the index pair of color tuple to switch between different colors, but since there are only 7 colors, `i` should be less than 7. When greater than or equal to 7, reset `i` to 0 and switch to the first color .

RGB Pin information can be found in the hardware interface manual provided by us,

Category	Function	Number	Drive	The number of Drive pin	The number of connected to the controller	micro:bit
	DRV8833	A11	PCA96850PYY	LED11	I2C_SCL	P19
		A12		LED10		
		B11		LED8		
		B12		LED9		
		B13		PINB(15)		
	DRV8833	B14		PINB(14)		
		A13		PINB(12)		
		A14		PINB(13)		
		AO1	DRV8833	AOUT1	AI1	
		AO2		AOUT2	AI2	
Stepper motor	Stepper motorB1	BO1		BOUT1	BI1	
		BO2		BOUT2	BI2	
	Stepper motorB2	AO3	DRV8833	AOUT1	BI3	
		AO4		AOUT2	BI4	
		BO3		BOUT1	AI3	
		BO4		BOUT2	AI4	
Buzzer	Buzzer	Buzzer		Buzzer		P0
RGB	RGB	RGB		DIN	P12	
Serial port	Serial port	RX	Micro:bit drive directly	RX	P16	
		TX		TX	P15	
I2C port	I2C	I2C_SDA		I2C_SDA	P20	
		I2C_SCL		I2C_SCL	P19	

### 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.**

```

Mu 1.0.3 - Play music.py
Mode New Load Save Flash Files REPL Plotter Zoom-in
Play music.py ×
1 from microbit import *
2 import music
3
4 display.show(Image.MUSIC_QUAVER)
5 music.play(music.BIRTHDAY)
6

```

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.

```

1 from microbit import *
2 import neopixel
3
4 Red = (255, 0, 0)
5 Orange = (255, 165, 0)
6 Yellow = (255, 255, 0)
7 Green = (0, 255, 0)
8 Blue = (0, 0, 255)
9 Dark_Violet = (148, 0, 211)
10 White = (255, 255, 255)
11
12 color = (Red, Orange, Yellow, Green, Blue, Dark_Violet, White)
13

```

Good job! No problems found.

3. Click the 'REPL' button to check whether the super:bit library has been downloaded. If not, please refer to the [preparation before class] ---> [2.How to import Yahboom superbit library] import super:bit library tutorial.

```

1 from microbit import *
2 import neopixel
3
4 Red = (255, 0, 0)
5 Orange = (255, 165, 0)
6 Yellow = (255, 255, 0)
7 Green = (0, 255, 0)
8 Blue = (0, 0, 255)
9 Dark_Violet = (148, 0, 211)
10 White = (255, 255, 255)
11

```

BBC micro:bit REPL

MicroPython for Super:bit V1.1 modified by Yahboom Team  
Type "help()" for more information.  
>>>  
>>>

4. After writing the code, please click the 'Flash' button to download the program to the micro:bit board.

The screenshot shows the micro:bit Superbit software interface. At the top, there are several icons: Mode, New, Load, Save, Flash (which is highlighted with a red box), Files, REPL, Plotter, and Zoom-in. Below the toolbar, a file tab labeled "microbit-superbit\_RGB\_one.py" is open. The code in the editor is as follows:

```
1 from microbit import *
2 import neopixel
3
4 Red = (255, 0, 0)
5 Orange = (255, 165, 0)
6 Yellow = (255, 255, 0)
7 Green = (0, 255, 0)
8 Blue = (0, 0, 255)
9 Dark_Violet = (148, 0, 211)
```

If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Buildingbit libraryhex 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】

6. After downloading the program, the smile symbol is displayed on the micro:bit dot matrix. RGB will light up to red, orange, yellow, green, blue, purple, white, in turn, with a time interval of 1 second.