

## Basic course10---RGB Search Light

### Learning goals:

This lesson we will learn how to drive RGB Search Light by Python programming.

### Code :

```
# -*- coding: utf-8-*# Encoding cookie added by Mu Editor
from microbit import display, Image, sleep
import buildingbit

display.show(Image.HAPPY)

while True:
    buildingbit.car_HeadRGB(255, 0, 0)
    sleep(500)
    buildingbit.car_HeadRGB(0, 255, 0)
    sleep(500)
    buildingbit.car_HeadRGB(0, 0, 255)
    sleep(500)
    buildingbit.car_HeadRGB(255, 255, 255)
    sleep(500)
    buildingbit.car_HeadRGB(0, 0, 0)
    sleep(500)
```

- 1) First, we need to import Yahboom buildingbit library: **import buildingbit** and others library we need to use.
- 2) **display.show(Image.HAPPY)** make micro:bit display a smile.
- 3) **buildingbit.car\_HeadRGB(255, 0, 0)** make RGB search light display red.  
Parameter 1: red value range:0~255  
Parameter 2: green value range:0~255  
Parameter 3: blue value range:0~255
- 4) **sleep(500)** can modify time value (ms)

### 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, Tab key for indentation, and the last line must be a space.**

```

车头探照灯.py
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, sleep
3 import buildingbit
4
5 display.show(Image.HAPPY)
6
7 while True:
8     buildingbit.car_HeadRGB(255, 0, 0)
9     sleep(500)
10    buildingbit.car_HeadRGB(0, 255, 0)
11    sleep(500)
12    buildingbit.car_HeadRGB(0, 0, 255)
13    sleep(500)
14    buildingbit.car_HeadRGB(255, 255, 255)
15    sleep(500)
16    buildingbit.car_HeadRGB(0, 0, 0)
17    sleep(500)
18

```

2. You need to 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. If there is no cursor or underline, it means that the code is correct, and the bottom left will prompt that the check is OK.

```

车头探照灯.py
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, sleep
3 import buildingbit
4
5 display.show(Image.HAPPY)
6
7 while True:
8     buildingbit.car_HeadRGB(255, 0, 0)
9     sleep(500)
10    buildingbit.car_HeadRGB(0, 255, 0)
11    sleep(500)
12    buildingbit.car_HeadRGB(0, 0, 255)
13    sleep(500)
14    buildingbit.car_HeadRGB(255, 255, 255)
15    sleep(500)
16    buildingbit.car_HeadRGB(0, 0, 0)
17    sleep(500)
18

```

Well done! No problems here.

3. You need to connect the micro data cable to micro:bit and the computer and **download buildingbit library into micro:bit**. Then, click “**REPL**” button to import “Yahboom buildingbit library”. As shown below.

```

carRun.py x 驅動機器人.py x
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, sleep
3 import buildingbit
4
5 display.show(Image.HAPPY)
6 buildingbit.car_run(255, 255, 1000)
7 buildingbit.car_stop()
BBC micro:bit REPL
>>> microbit.reset()
MicroPython for Building:bit V1.0 modified by Yahboom Team
Type "help()" for more information.
>>>
MicroPython for Building:bit V1.0 modified by Yahboom Team
Type "help()" for more information.
>>>
>>> |

```

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

```

carRun.py x 驅動機器人.py x
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, sleep
3 import buildingbit
4
5 display.show(Image.HAPPY)
6 buildingbit.car_run(255, 255, 1000)
7 buildingbit.car_stop()
8 sleep(1000)
9 buildingbit.car_back(255, 255, 1000)
10 buildingbit.car_stop()
11

```

### Experimental phenomena

After download is complete. We can see micro:bit dot matrix display a “smile”. Two RGB search light will become red 0.5s -> green 0.5s -> blue0.5s ->white 0.5s -> off 0.5s, And keep looping like this status.