

# Adjustable RGB light

---

## Adjustable RGB light

1. Learning objectives
2. Building blocks
3. Sensor wiring
4. Code analysis
5. Write and download the program
6. Experimental phenomenon

## 1. Learning objectives

---

In this course, we mainly learn how to adjust the brightness of RGB lights through Python programming.

## 2. Building blocks

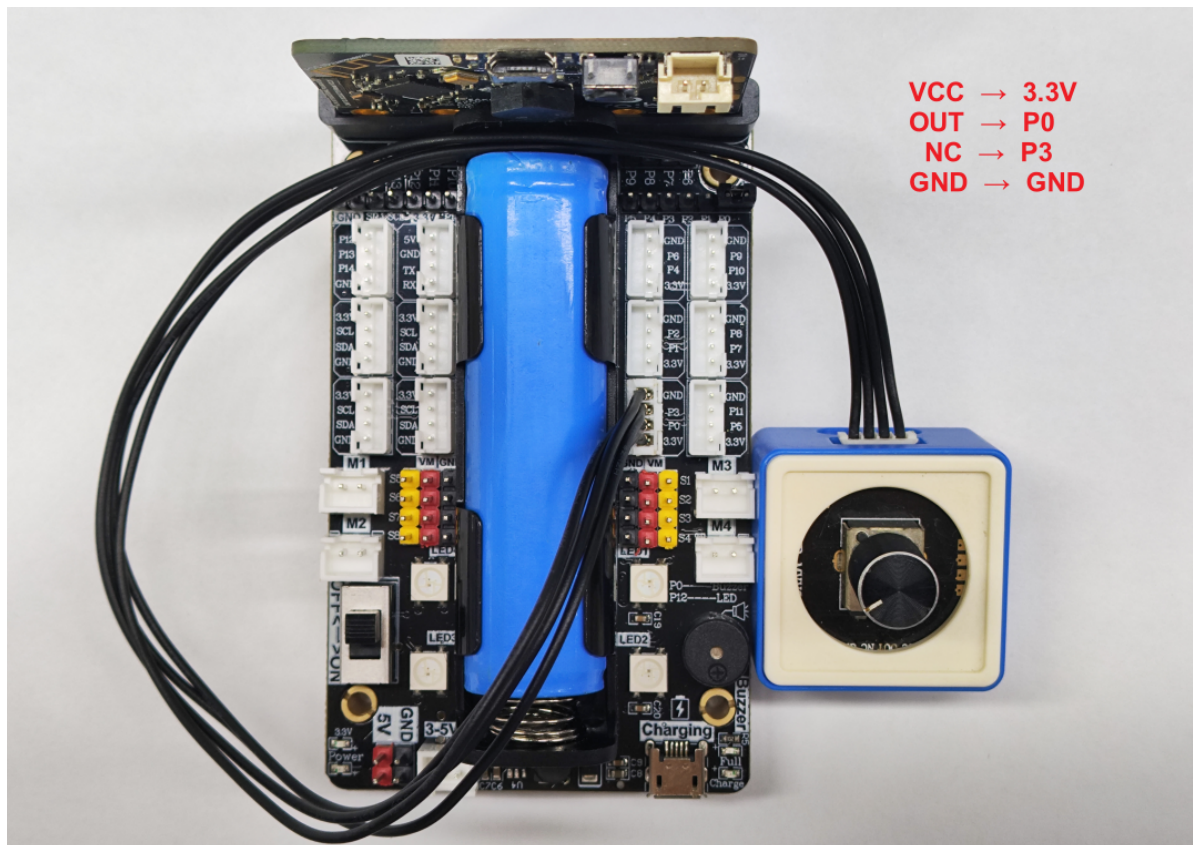
---

For the building blocks steps, please refer to the installation drawings of [Assembly Course]-- [Adjustable RGB light] or the building blocks installation brochure in the materials.

## 3. Sensor wiring

---

The potentiometer is connected to the P0P3 pin.



```
from microbit import *  
import neopixel  
import WOM_Sensor_Kit
```

First, import the libraries needed for this lesson from microbit: WOM\_Sensor\_Kit library is used for sensors; neopixel is used to control RGB lights.

```
np = neopixel.NeoPixel(pin12, 4)  
np.clear()  
np.show()
```

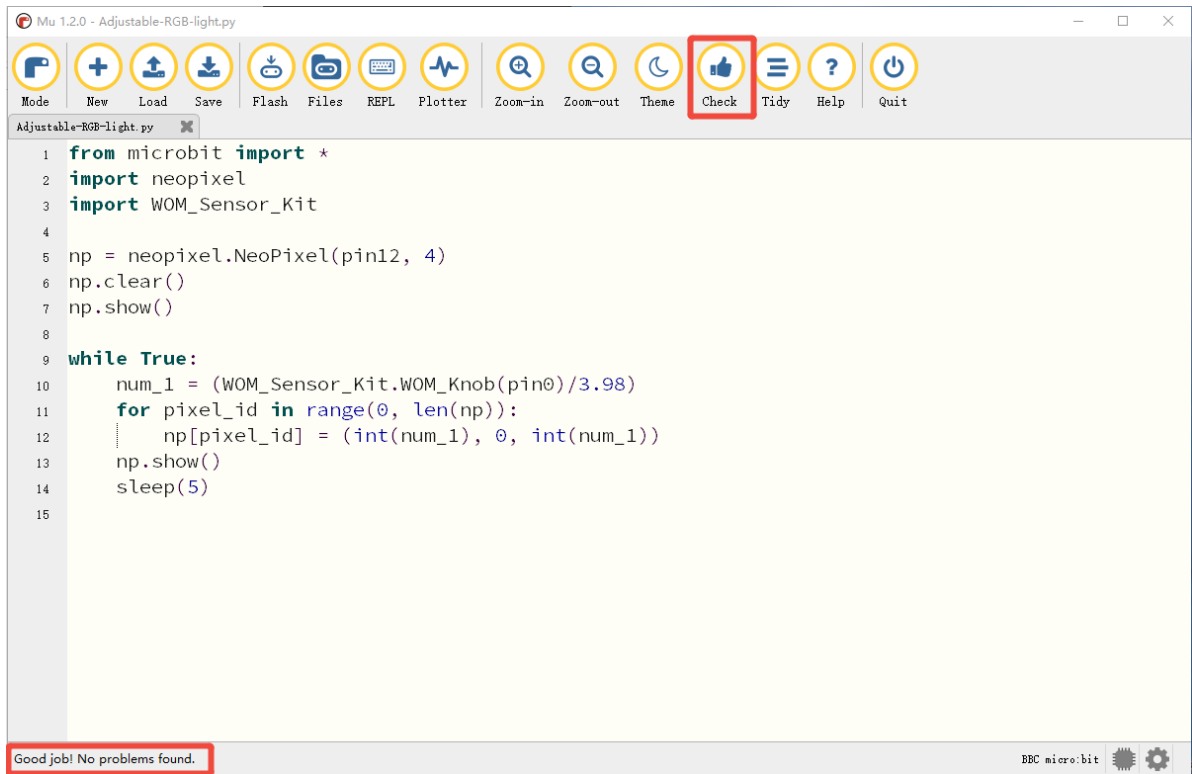
RGB light initialization settings, there are 4 RGB lights, connected to the P12 pin of the microbit motherboard (you can check the hardware interface manual); `np.clear()` is used to clear all the lamp beads (black), and `np.show()` will update the clearing operation to the actual light strip.

```
while True:  
    num_1 = (WOM_Sensor_Kit.WOM_Knob(pin0)/3.98)  
    for pixel_id in range(0, len(np)):  
        np[pixel_id] = (int(num_1), 0, int(num_1))  
    np.show()  
    sleep(5)
```

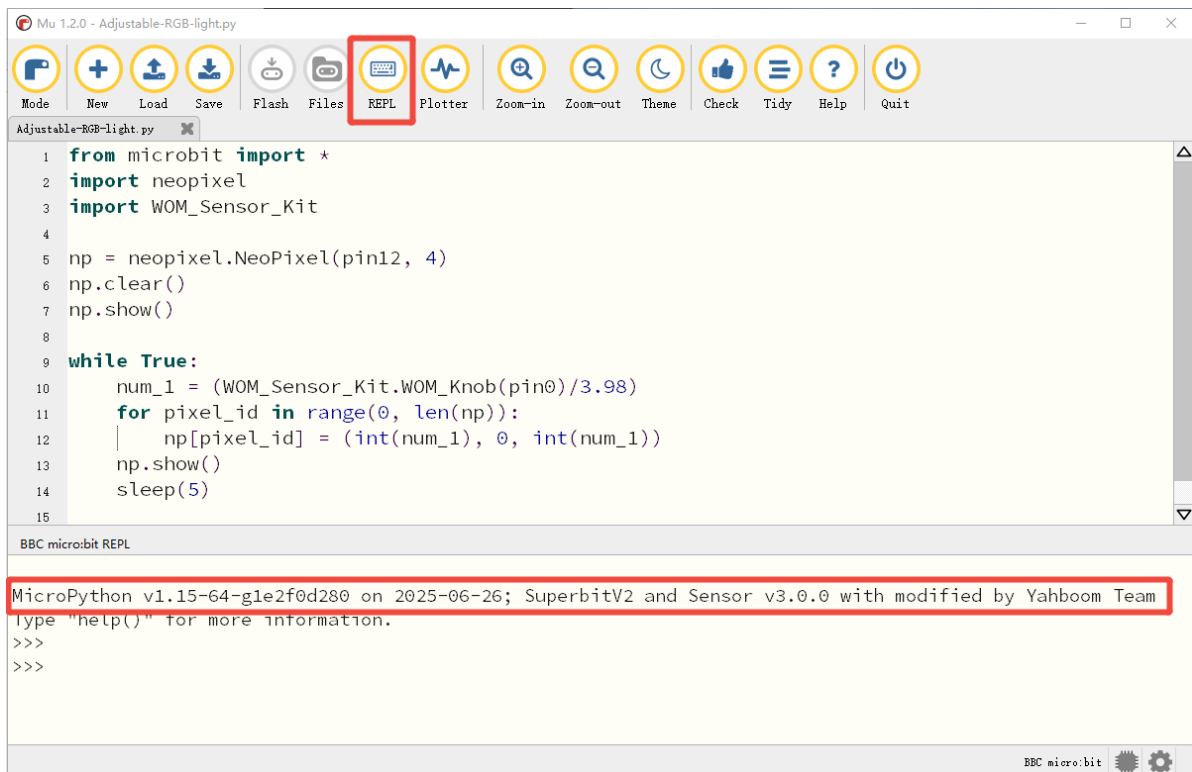
In an infinite loop, read the value of the knob on `pin0` and convert it to a range of 0–255; traverse each lamp bead and set its color to purple (red and blue components are `num_1`, green is 0); each light is displayed immediately and paused for 5 milliseconds.

## 5. Write and download the program

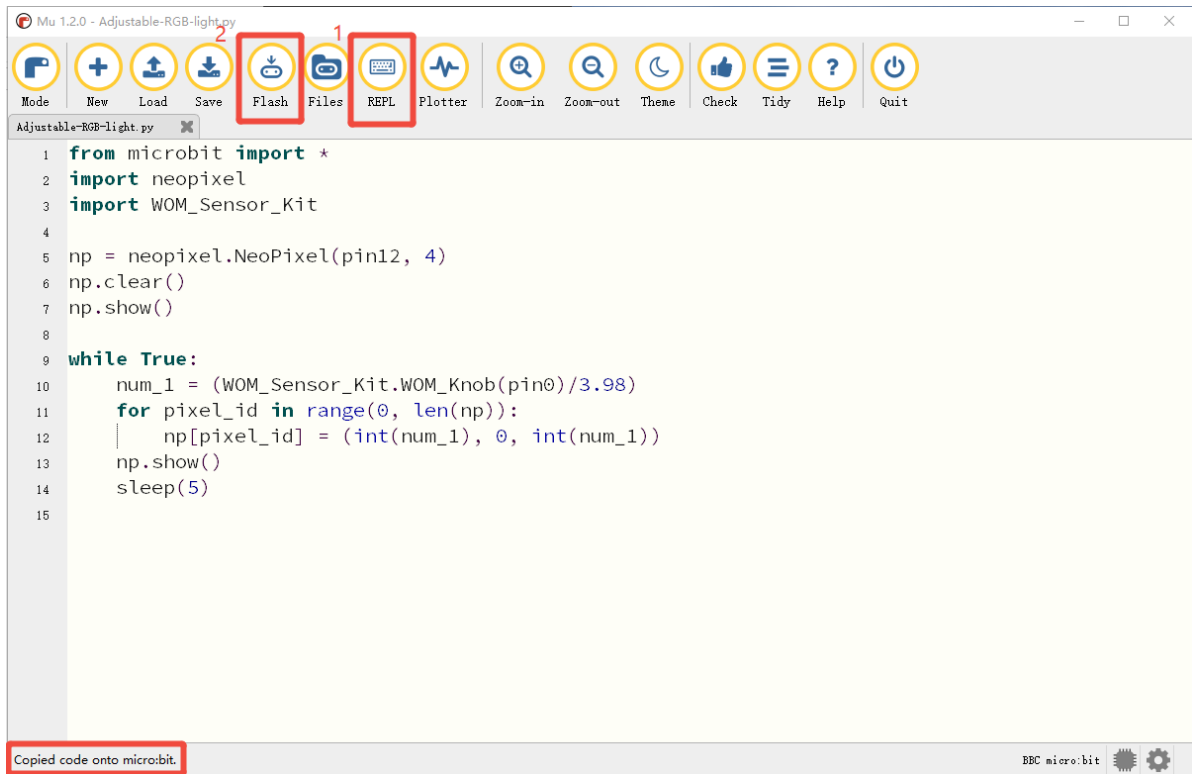
1. Open the Mu software and enter the code in the editing window. **Note! All English and symbols should be entered in English mode, use the Tab key for indentation, and the last line ends with a blank program.**
2. Click the thumb 'Check' button to check if there are any errors in our code. If a cursor or underline appears in a line, it means a syntax error. Please check and modify it. If there is no error, the lower left corner will prompt that there is no problem with the detection.



3. Click the 'REPL' button to check if the Superbit library has been downloaded. If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide].



4. After the program is written, connect the computer and the microbit mainboard with a microUSB data cable. Please click the 'Flash' button to download the program to the micro:bit mainboard. **(You need to click the 'REPL' button again to turn off the import library file function before you can download the program normally).**



5. If the download fails, please confirm whether the microbit is properly connected to the computer via the microUSB data cable and the Superbit Python library has been imported.

## 6. Experimental phenomenon

After the program runs successfully, twist the potentiometer and the brightness of the RGB light on the expansion board will change accordingly.