Rocker control light

Rocker control 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 use Python programming to change the color of the RGB light with the joystick.

2. Building blocks

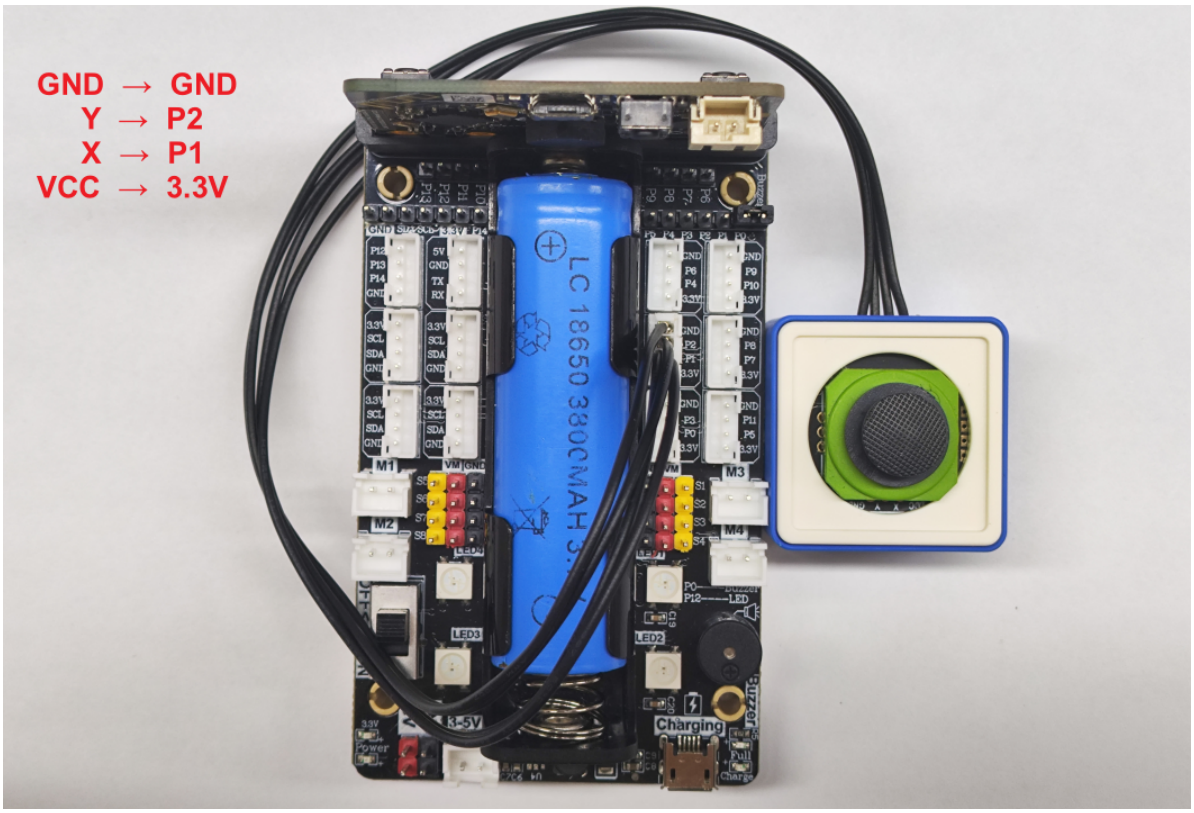
For the building blocks steps, please refer to the installation drawings of [Assembly Course]--[Rokcer color changing light] or the building blocks installation brochure.

3. Sensor wiring

Insert the building blocks servo wiring into the S1 interface of the Super:bit expansion board, and the orange servo wiring into the yellow pin of S1.

The color recognition module is connected to the I2C (SDA, SCL) interface.

The joystick module is connected to the P1P2 interface.



4. Code analysis

For the program of this course, please refer to the **Rocker-control-light.py** file.

```
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.

```
Red = (255, 0, 0)
Green = (0, 255, 0)
Blue = (0, 0, 255)
White = (255, 255, 255)

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

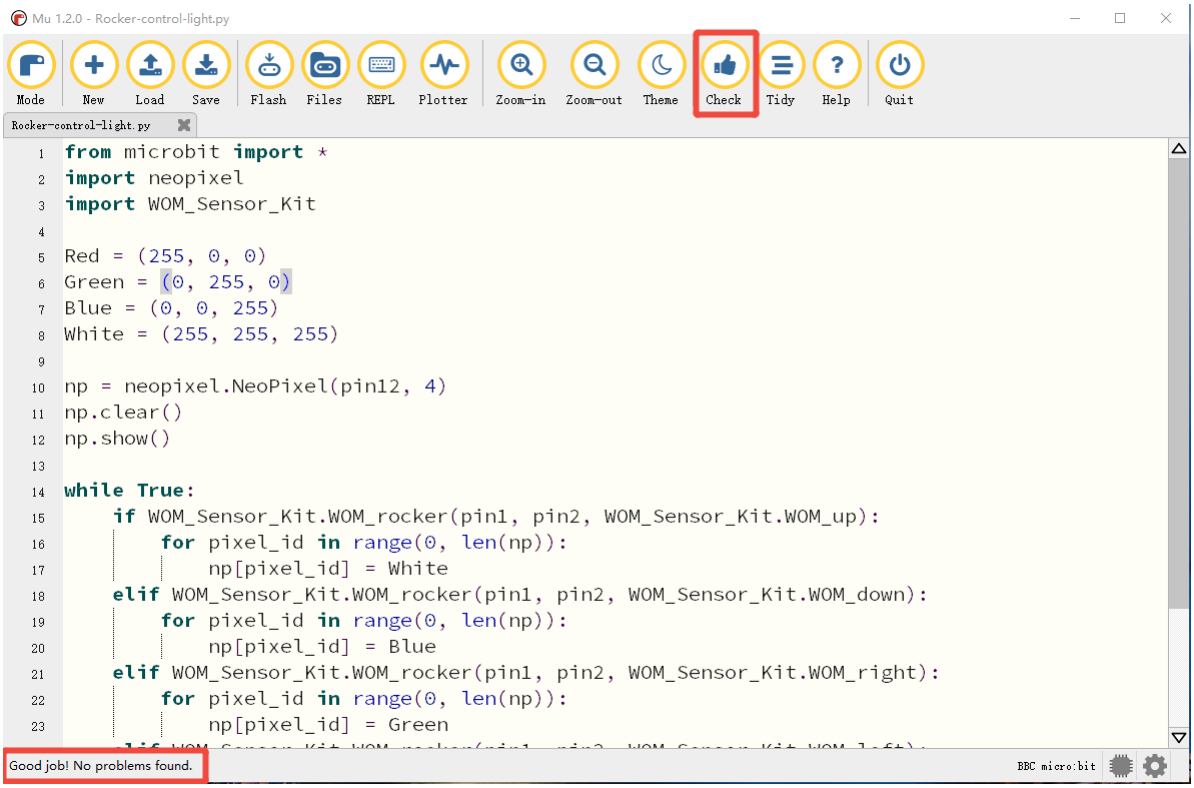
Define the RGB values of the four basic colors (red, green, blue, white), create a colored light `np` connected to `pin12`, initially clear the GBD light and refresh the display to ensure that all lights are off

```
while True:
if WOM_Sensor_Kit.WOM_rocker(pin1, pin2, WOM_Sensor_Kit.WOM_up):
for pixel_id in range(0, len(np)):
np[pixel_id] = White
elif WOM_Sensor_Kit.WOM_rocker(pin1, pin2, WOM_Sensor_Kit.WOM_down):
for pixel_id in range(0, len(np)):
np[pixel_id] = Blue
elif WOM_Sensor_Kit.WOM_rocker(pin1, pin2, WOM_Sensor_Kit.WOM_right):
for pixel_id in range(0, len(np)):
np[pixel_id] = Green
elif WOM_Sensor_Kit.WOM_rocker(pin1, pin2, WOM_Sensor_Kit.WOM_left):
for pixel_id in range(0, len(np)):
np[pixel_id] = Red
np.show()
sleep(5)
```

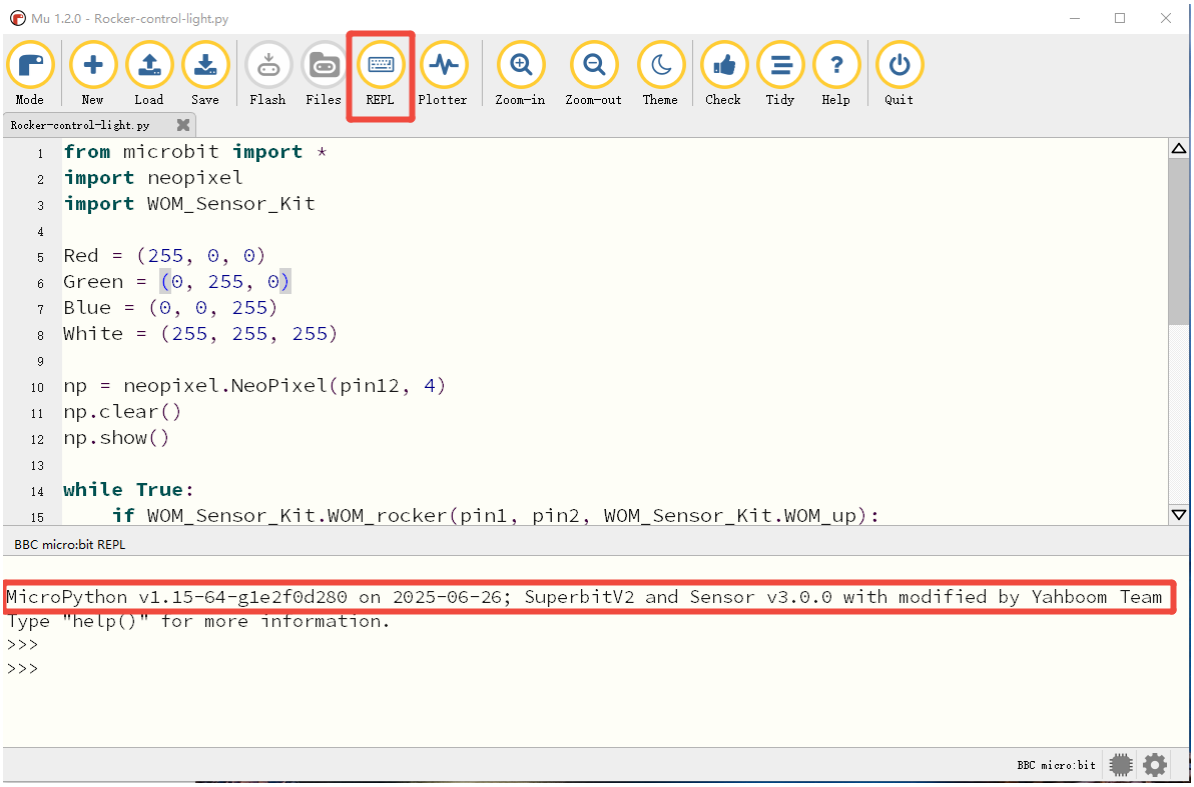
Listen to the joystick direction input (via `pin1` and `pin2`) in an infinite loop, and set all according to the joystick direction. The colors of the 4 lamp beads are white when pointing upward, blue when pointing downward, green when pointing right, and red when pointing left. Refresh the RGB lamp to show the color change.

5. Write and download the program

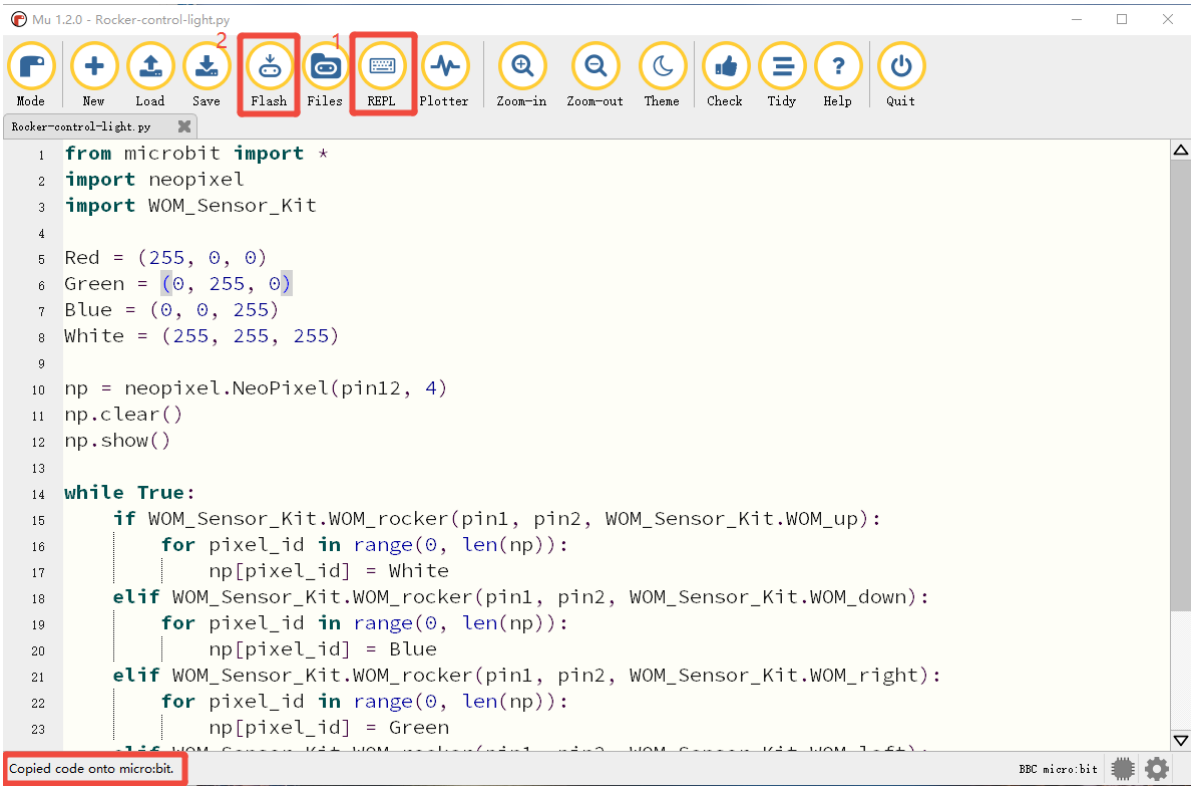
1. Open the Mu software and enter the code in the editing window. **Note! All English and symbols should be entered in English mode, use the Tab key (tab key) for indentation, and the last line ends with a blank program.**
2. Click the thumb 'Check' button to check if there are any errors in our code. 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 whether 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 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 connected to the computer normally via the microUSB data cable and the Superbit Python library has been imported.

6. Experimental phenomenon

After the program runs successfully, shake the joystick. If the joystick moves to the far left in the X direction, the RGB light turns red. If it moves to the far right, the RGB light turns green. If it moves to the bottom in the Y direction, the RGB light turns blue. If it moves to the top, the RGB light turns white. Rotating the joystick can achieve the effect of alternating colored lights.