**Sunflower**

Note: Please use the double-headed data cable provided by us to connect the Micro:bit board and expansion board to the computer, otherwise it will not be possible to drive the servo due to insufficient power.

## 1. Learning target

In this lesson, we will use the micro:bit board, building blocks and sensor modules to build a sunflower. We can use the joystick to control the photosensitive module up, down, left, and right, or press the button to let the sunflower automatically track the brightest light.
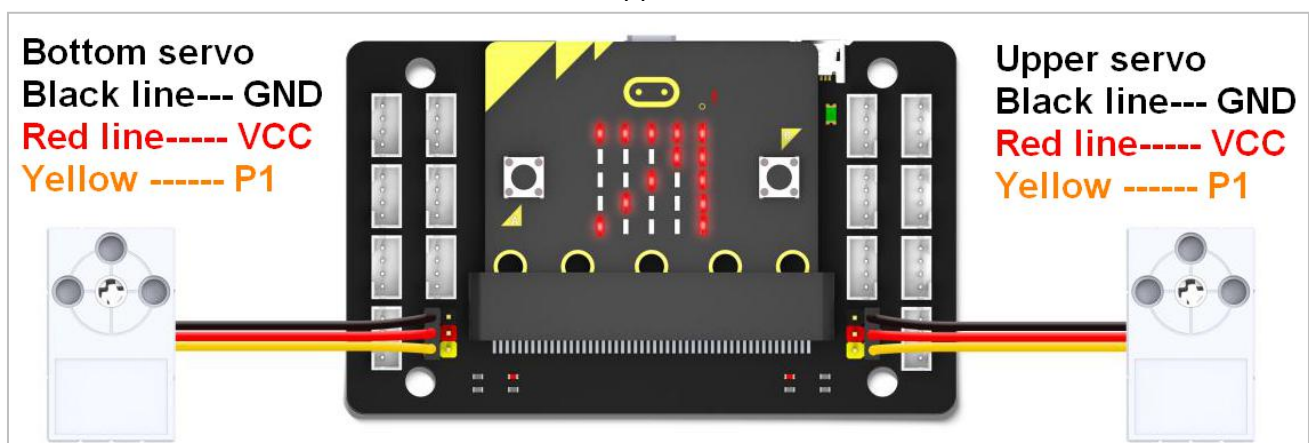
## 2. Servo calibration

Before assembling the building blocks, we need to use code to calibrate the servo to a fixed angle. If calibration is not calibrated before using, it is easy to jam the servo during use and cause the steering gear to stall and damage the servo.

Calibration method:

2.1 Connect the brown line of the servo to GND (black), the red line of the servo to VCC (red), and the yellow line of the servo to IO on expansion board.

The bottom servo is connected to P1, and the upper servo is connected to P10. As shown below.



Bottom servo
Black line--- GND
Red line----- VCC
Yellow ------ P1

Upper servo
Black line--- GND
Red line----- VCC
Yellow ------ P1

2.2 Then connect the computer to the Micro:bit board and expansion board through the double-head micro USB cable we provided.

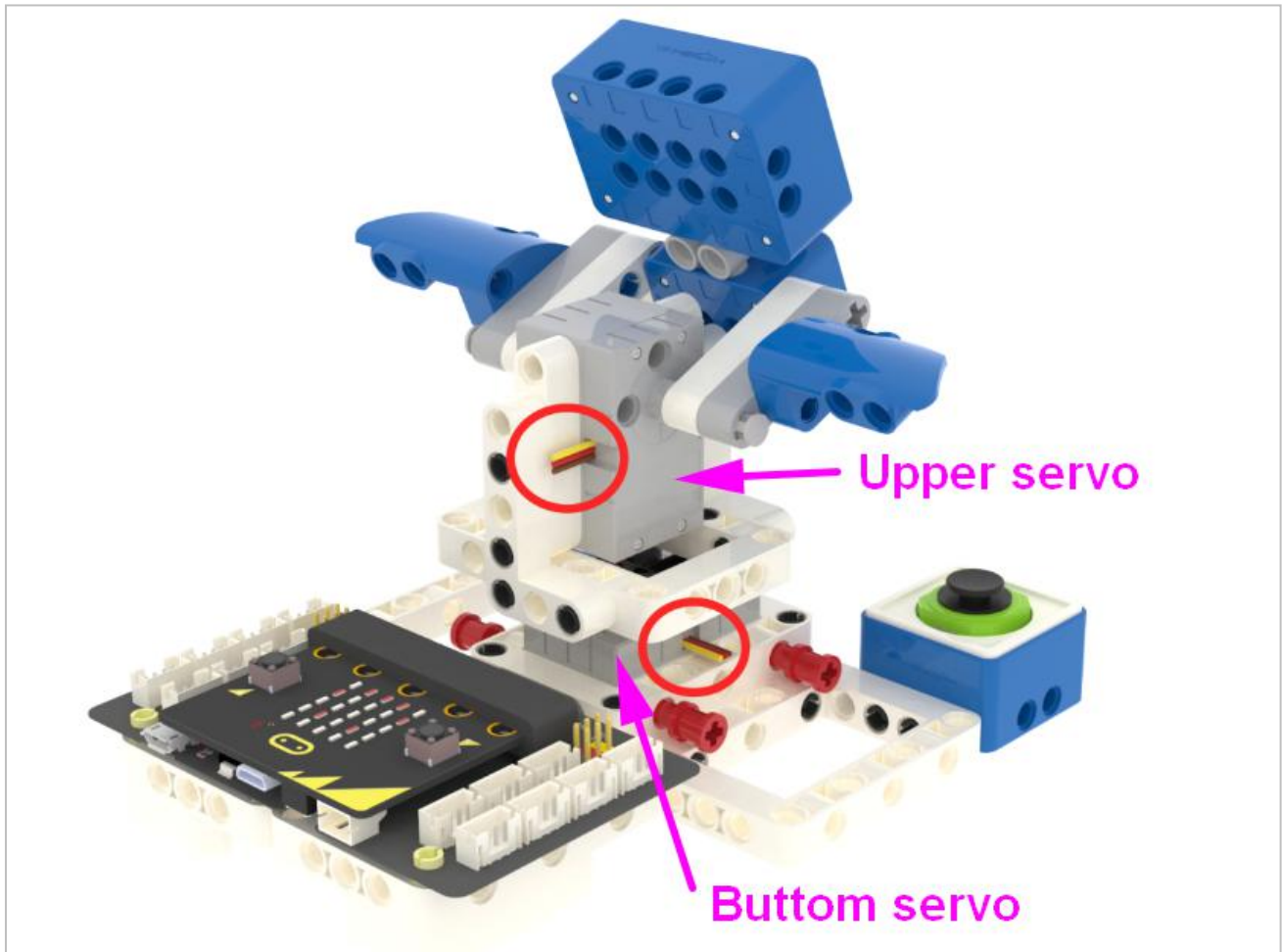2.3 Download the servo calibration code (**Servo-calibration-Sunflower.py**) to the micro:bit board.

2.4 When a "7" pattern is displayed on the dot matrix of the Micro:bit board, it means the servo be calibrated successfully.

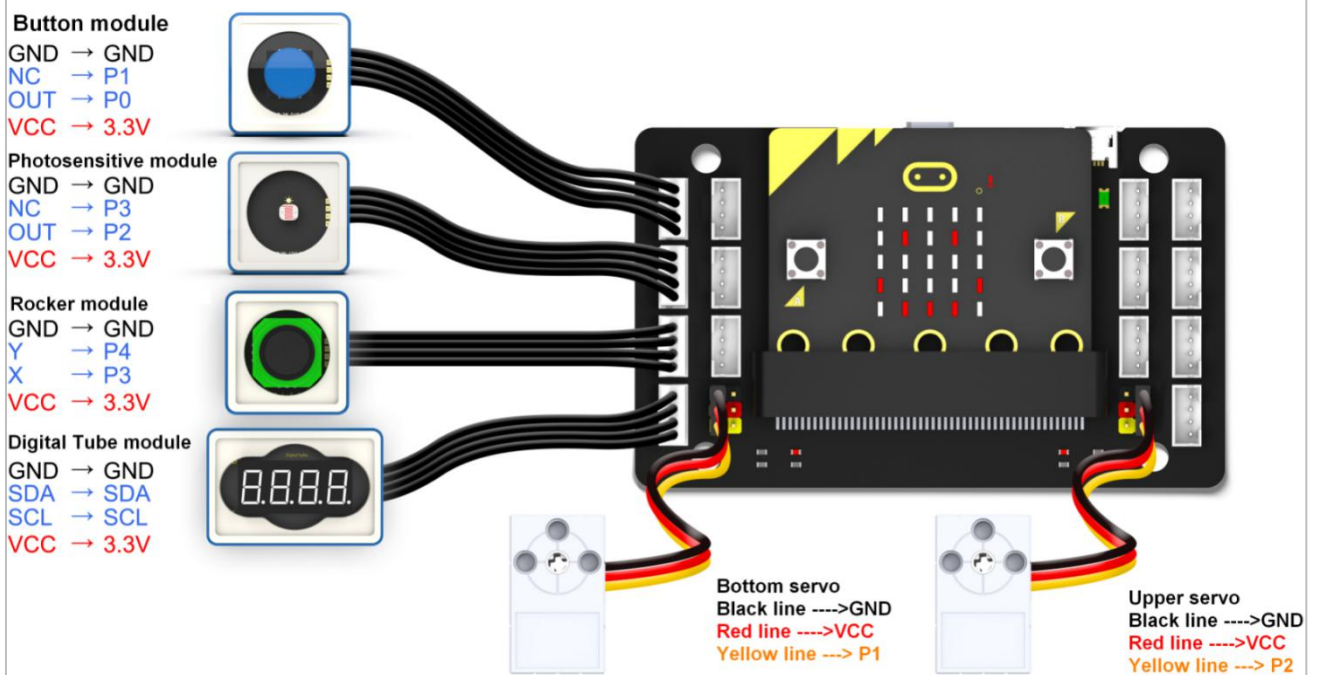## 3. Building blocks assembly

Please follow the steps we provide to assemble the building block models.

Pay attention to the installation direction of the servo when assembling, otherwise the servo will be damaged due to the wrong angle of the servo after running the program.

After the assembly is completed, please check the wiring of the servo as shown below.

Upper servo

Buttom servo

## 4. About wiring



**Button module**
GND → GND
NC → P1
OUT → P0
VCC → 3.3V

**Photosensitive module**
GND → GND
NC → P3
OUT → P2
VCC → 3.3V

**Rocker module**
GND → GND
Y → P4
X → P3
VCC → 3.3V

**Digital Tube module**
GND → GND
SDA → SDA
SCL → SCL
VCC → 3.3V

**Bottom servo**
Black line ---->GND
Red line ---->VCC
Yellow line ---> P1

**Upper servo**
Black line ---->GND
Red line ---->VCC
Yellow line ---> P2

## 5. About code

```python
# -*- coding: utf-8-*-# Encoding cookie added by Mu Editor
from microbit import *
import WOM_Sensor_Kit

display.off()
x = 200
y = 250
WOM_Sensor_Kit.WOM_init_display(7)
value = [0, 0, 0, 0, 0, 0]
minimum = 1023

while True:
    WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
    WOM_Sensor_Kit.WOM_servo360(pin1, 200)
    WOM_Sensor_Kit.WOM_servo360(pin10, 250)

    if WOM_Sensor_Kit.WOM_button(pin0) == 1:
        WOM_Sensor_Kit.WOM_servo360(pin1, 200)
        WOM_Sensor_Kit.WOM_servo360(pin10, 300)
        sleep(500)
        value[0] = WOM_Sensor_Kit.WOM_light(pin2)
        WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
        print(value[0])
        WOM_Sensor_Kit.WOM_servo360(pin1, 250)
        WOM_Sensor_Kit.WOM_servo360(pin10, 300)
        sleep(500)
        value[1] = WOM_Sensor_Kit.WOM_light(pin2)
        WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
        print(value[1])
        WOM_Sensor_Kit.WOM_servo360(pin1, 250)
        WOM_Sensor_Kit.WOM_servo360(pin10, 240)
        sleep(500)
        value[2] = WOM_Sensor_Kit.WOM_light(pin2)
        WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
        print(value[2])
        WOM_Sensor_Kit.WOM_servo360(pin1, 200)
        WOM_Sensor_Kit.WOM_servo360(pin10, 240)
        sleep(500)
        value[3] = WOM_Sensor_Kit.WOM_light(pin2)
        WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
        print(value[3])
        WOM_Sensor_Kit.WOM_servo360(pin1, 130)
```

```python
WOM_Sensor_Kit.WOM_servo360(pin10, 240)
sleep(500)
value[4] = WOM_Sensor_Kit.WOM_light(pin2)
WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
print(value[4])
WOM_Sensor_Kit.WOM_servo360(pin1, 130)
WOM_Sensor_Kit.WOM_servo360(pin10, 300)
sleep(500)
value[5] = WOM_Sensor_Kit.WOM_light(pin2)
WOM_Sensor_Kit.WOM_display(WOM_Sensor_Kit.WOM_light(pin2))
print(value[5])
WOM_Sensor_Kit.WOM_servo360(pin1, 200)
WOM_Sensor_Kit.WOM_servo360(pin10, 300)
sleep(500)
for i in range(6):
    if value[i] < minimum:
        minimum = value[i]
        print(minimum)
if value[0] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 200)
    WOM_Sensor_Kit.WOM_servo360(pin10, 300)
    sleep(5000)
if value[1] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 250)
    WOM_Sensor_Kit.WOM_servo360(pin10, 300)
    sleep(5000)
if value[2] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 250)
    WOM_Sensor_Kit.WOM_servo360(pin10, 240)
    sleep(5000)
if value[3] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 200)
    WOM_Sensor_Kit.WOM_servo360(pin10, 240)
    sleep(5000)
if value[4] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 130)
    WOM_Sensor_Kit.WOM_servo360(pin10, 240)
    sleep(5000)
if value[5] == minimum:
    WOM_Sensor_Kit.WOM_servo360(pin1, 130)
    WOM_Sensor_Kit.WOM_servo360(pin10, 300)
    sleep(5000)
minimum = 1023
```
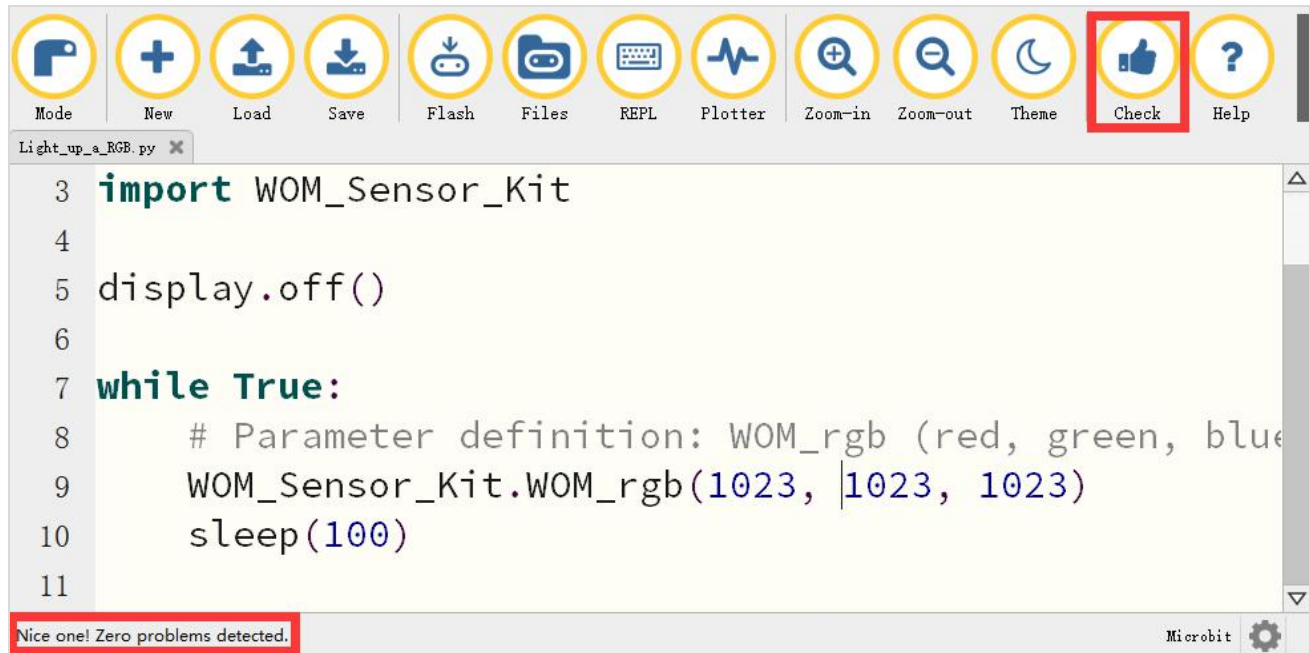
## 6.Writing and download code

6.1 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, use the Tab key (tab key) to indent and the last line must be a space.**

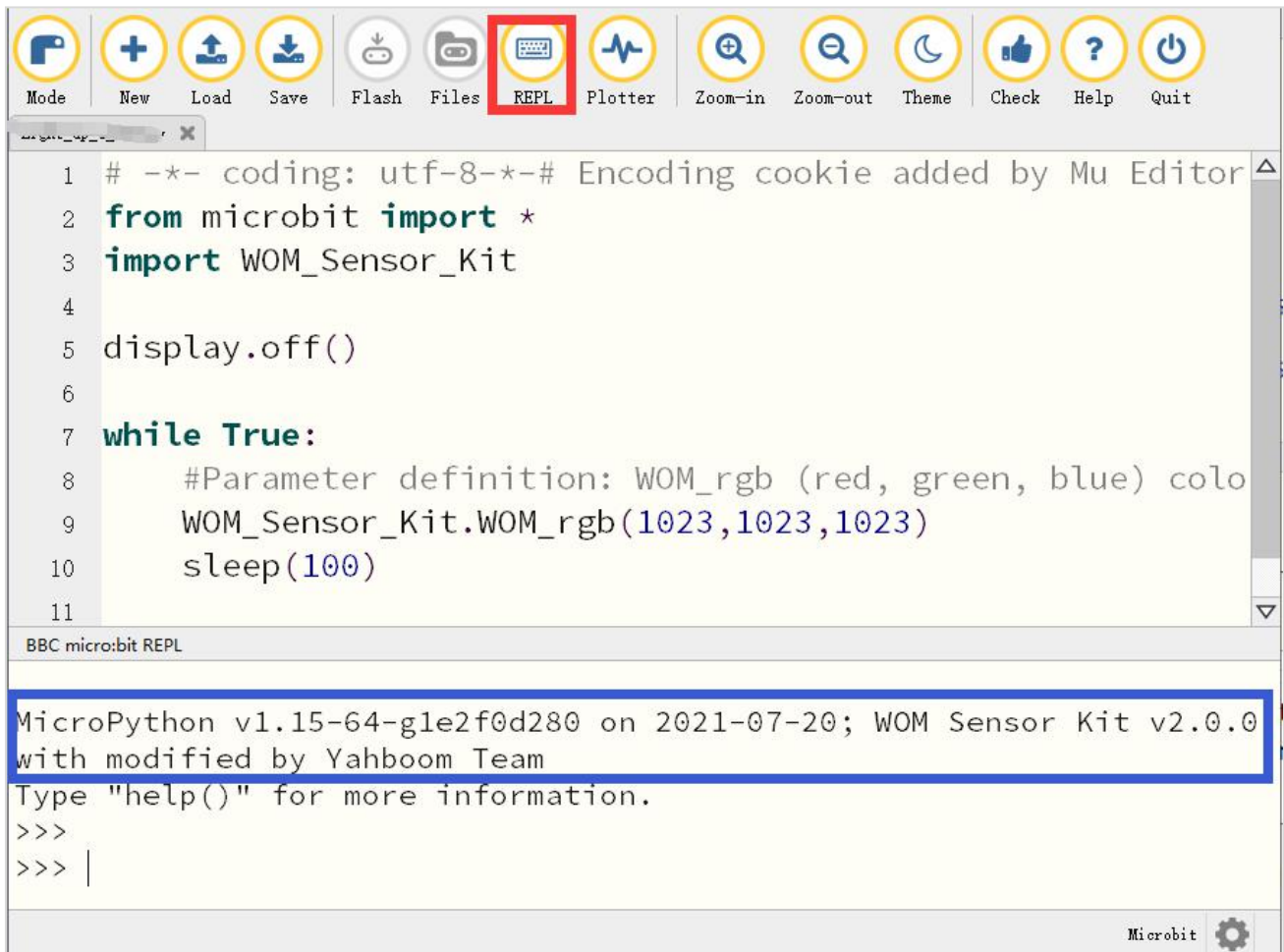6.2 You can click the "Check" button to check if our code has an error.

If a cursor or underline appears on a line, it indicates a syntax error, please check and modify. If there is no error in the program, the bottom left of the interface will prompt that there is no problem in detection.



```
3   import WOM_Sensor_Kit
4
5   display.off()
6
7   while True:
8       # Parameter definition: WOM_rgb (red, green, blue
9       WOM_Sensor_Kit.WOM_rgb(1023, 1023, 1023)
10      sleep(100)
11
```
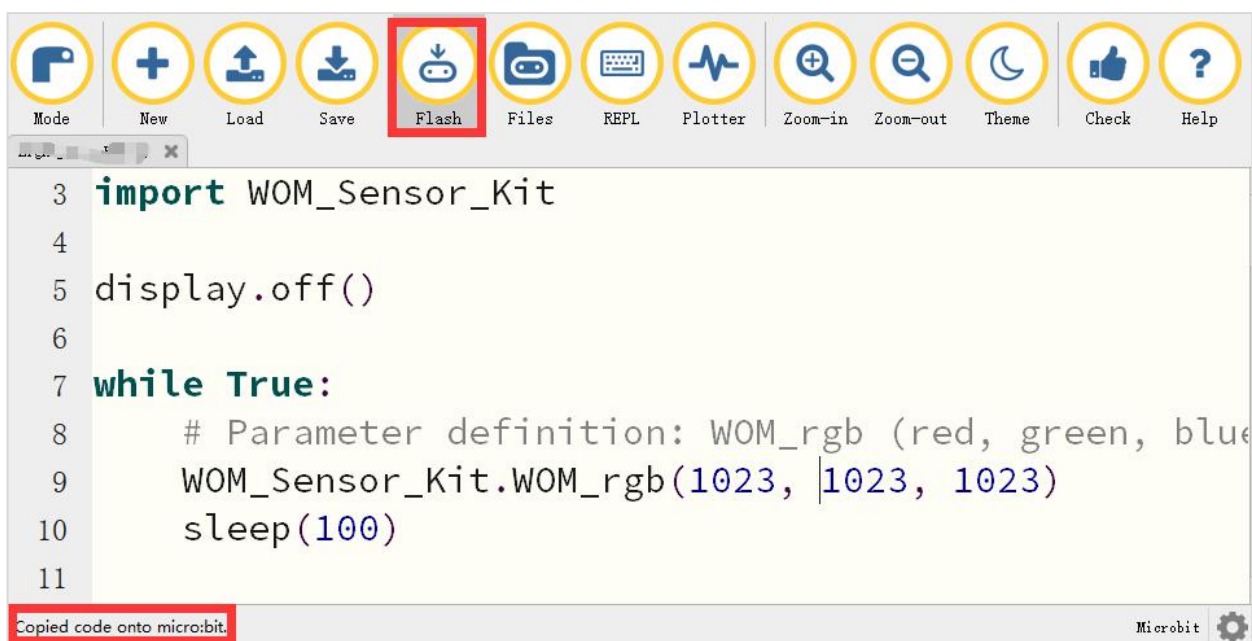
Nice one! Zero problems detected.

6.3 Click the 'REPL' button to check whether the WOM_Sensor_Kit Python library has been downloaded.

If not, please refer to [Preparation before class] --> [Python Programming Guide] .

```
1   # -*- coding: utf-8-*-# Encoding cookie added by Mu Editor
2   from microbit import *
3   import WOM_Sensor_Kit
4
5   display.off()
6
7   while True:
8       #Parameter definition: WOM_rgb (red, green, blue) colo
9       WOM_Sensor_Kit.WOM_rgb(1023,1023,1023)
10      sleep(100)
11
```

BBC micro:bit REPL

MicroPython v1.15-64-g1e2f0d280 on 2021-07-20; WOM Sensor Kit v2.0.0
with modified by Yahboom Team
Type "help()" for more information.
>>>
>>> |

6.4 After the program is written, use a micro USB cable to connect the computer and the micro:bit board. Please click the 'Flash' button to download the program to the micro:bit motherboard (You need to click the 'REPL' button again to close the function of importing library files before you download the program).



```
3   import WOM_Sensor_Kit
4
5   display.off()
6
7   while True:
8       # Parameter definition: WOM_rgb (red, green, blue
9       WOM_Sensor_Kit.WOM_rgb(1023, |1023, 1023)
10      sleep(100)
11
```

Copied code onto micro:bit.

6.5 If the download failed, please confirm whether the micro:bit is connected to the computer through the micro USB data cable, and confirm whether the **WOM_Sensor_Kit Python library** has been imported.

## 7. Experimental phenomena

After the program is downloaded successfully. We can use the joystick to control the photosensitive module up, down, left, and right, or press the button to let the sunflower automatically track the brightest light.