

Vibrating alarm clock

Vibrating alarm clock

1. Learning Objectives
2. Building Blocks
3. Sensor Wiring
4. Code Analysis
5. Write and download the program
6. Experimental phenomenon

1. Learning Objectives

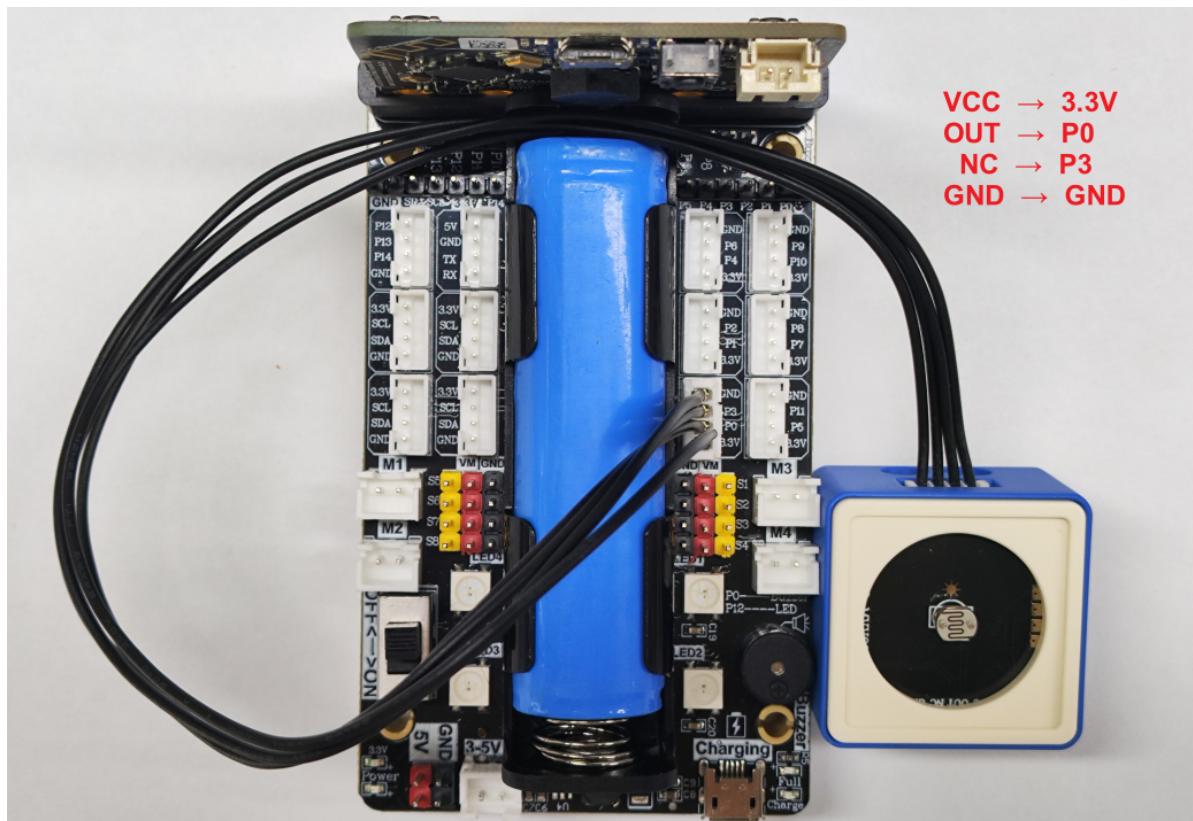
In this course, we mainly learn how to use Python programming to display the switch of lights according to the external light intensity.

2. Building Blocks

For the building block steps, please refer to the installation drawings of [Assembly Course]-- [Smart alarm clock] or the building block installation album in the materials.

3. Sensor Wiring

The photosensitive module is connected to the P0P3 interface.



4. Code Analysis

For the program of this course, please see the **Vibrating-alarm-clock.py** file.

```
from microbit import *  
import music  
import WOM_Sensor_Kit
```

First, import the libraries needed for this lesson from microbit: WOM_Sensor_Kit library is used for sensors; music library is used to play music.

```
a = 0  
  
display.clear()
```

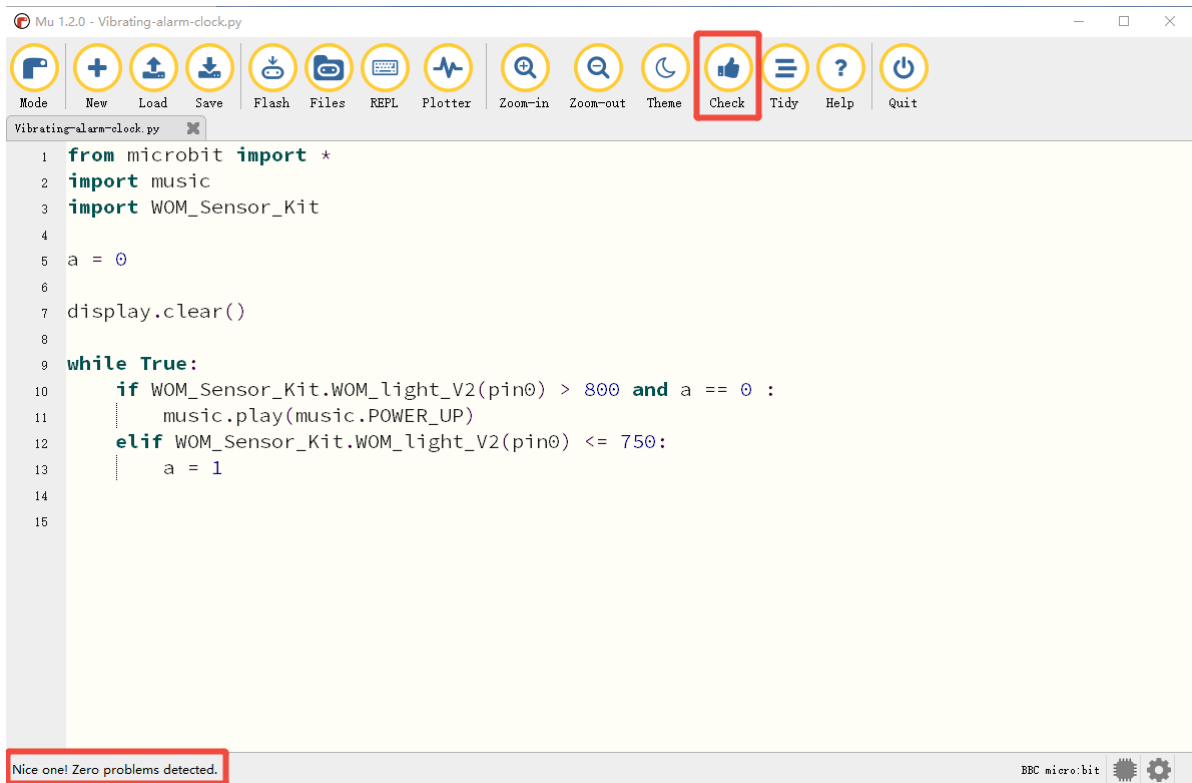
Define a flag variable `a` to control whether to play music to avoid repeated triggering; clear the default startup icon on the micro:bit screen;

```
while True:  
    if WOM_Sensor_Kit.WOM_light_v2(pin0) > 800 and a == 0 :  
        music.play(music.POWER_UP)  
    elif WOM_Sensor_Kit.WOM_light_v2(pin0) <= 750:  
        a = 1
```

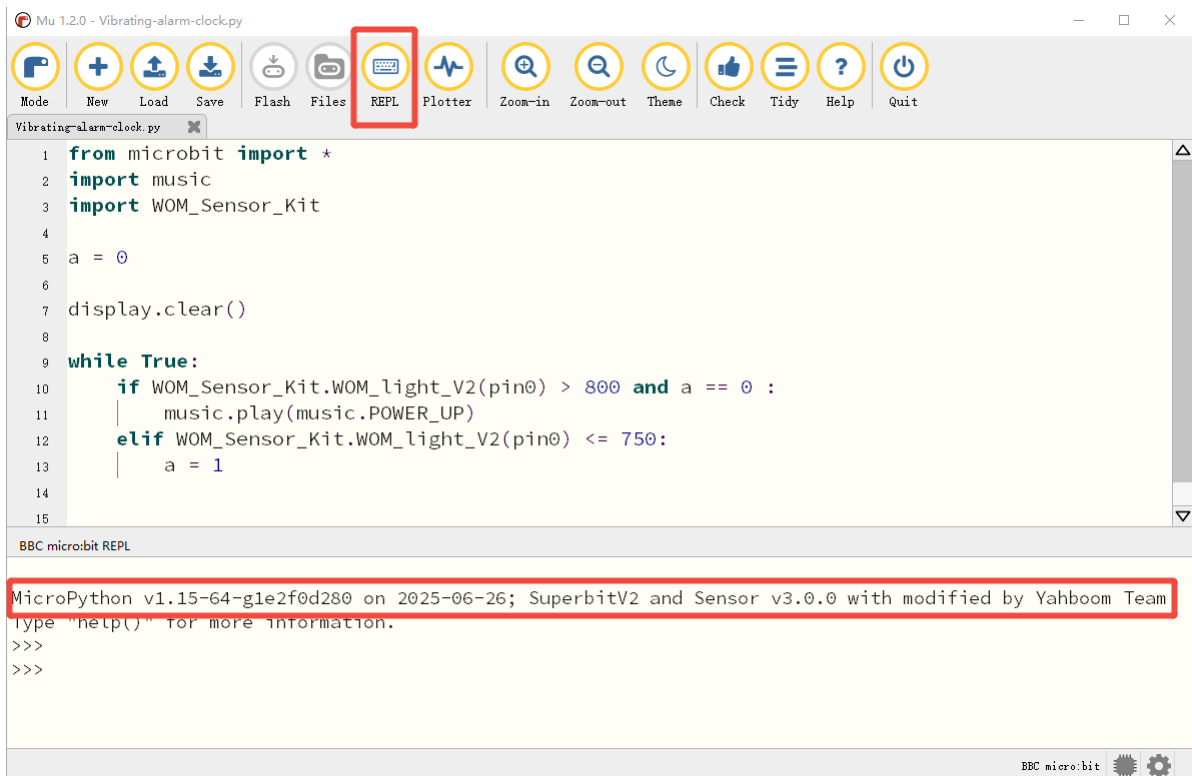
Read the value of the light sensor connected to pin0 in an infinite loop. If the ambient light is detected to be stronger (>800) and music has not been played (`a==0`), play the startup sound effect; if the light becomes darker (≤ 750) afterwards, set the variable `a` to 1, indicating that the music has been played to prevent re-triggering

5. Write and download the program

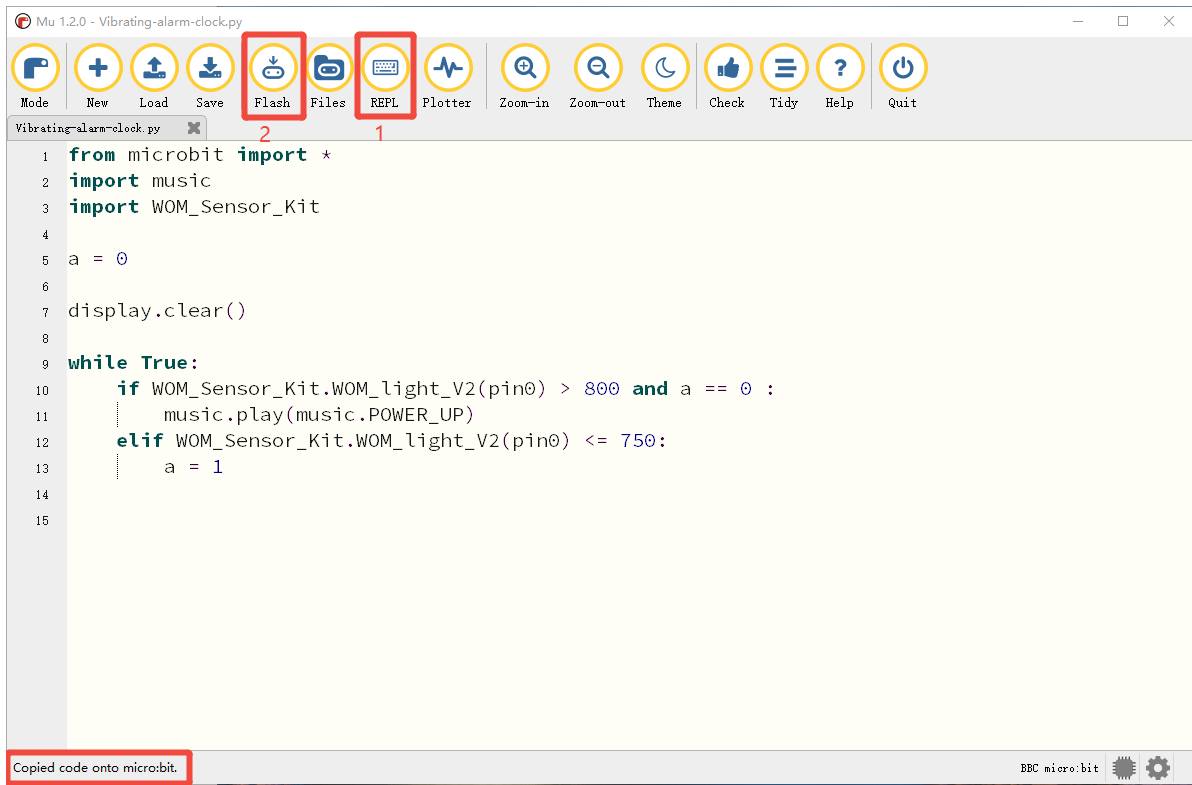
1. Open the Mu software and enter the code in the editing window. **Note! All English and symbols should be entered in English, use the Tab key (tab key) for indentation, and the last line ends with a blank program.**
2. Click the thumb 'Check' button to check whether our code has errors. If a cursor 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, 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, the alarm rings when there is light, and the alarm rings when there is no light.