Changing face

1. Learning Objectives

In this course, we mainly learn how to use Python programming to make the Face Changing King building blocks realize the gameplay of "face changing in one second".
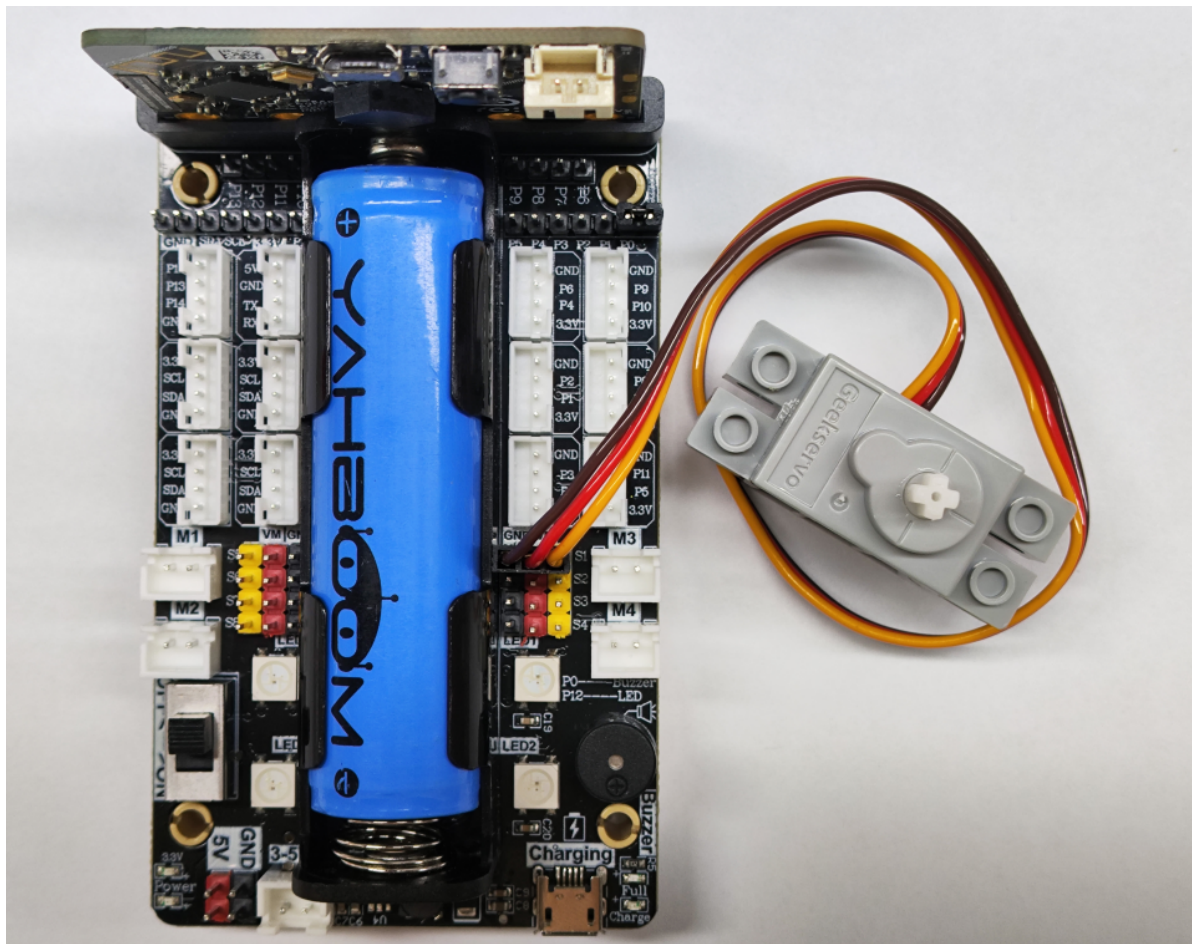
2. Building Blocks

For the building block construction steps, please refer to the installation drawings of [Assembly course]-[Changing face] or the building block installation album in the materials.

3. Motor Wiring

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

As shown in the figure below:

! Notes:

When taking the course related to the building block servo for the first time, we need to remove the white building blocks installed on the servo and upload the program of this course to the micro:bit; then turn on the power switch of the Super:bit expansion board and wait for the building block servo to turn to the initial position; then, we can turn off the power, adjust the mask of the face-changing king to the closed state, as shown in the figure below, and then install the building blocks. (If you have used the program related to the face-changing king and the servo before, you can skip this step)
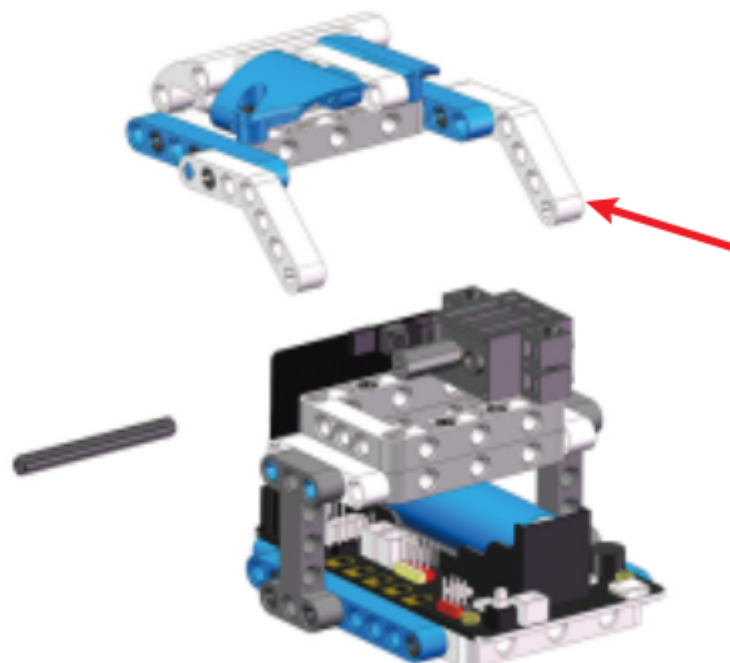
4. Code analysis

For the program of this course, please see the **Changing face.py** file.

```python
from microbit import *
import superbit
import microbit
import random
```

First, import the libraries needed for this lesson from microbit: the superbit library is dedicated to the superbit expansion board; the random library is dedicated to generating random numbers;

```python
superbit.servo270(superbit.S1, 50)
microbit.sleep(500)
superbit.servo270(superbit.S1, 140)
display.scroll("Go!")
```

superbit.servo270(superbit.S1, 50): Initialize the building block servo to rotate to 50°;

microbit.sleep(500): Delay 00 milliseconds;

superbit.servo270(superbit.S1, 140): Initialize the building block servo to rotate to 140°;

display.scroll("Go!"): Scroll the string Go! on the dot matrix;

```python
def Face_show():
    global a
    a = random.randint(1, 7)
    if a == 1:
        display.show(Image.HAPPY)
    ...
```

a = random.randint(1, 7): Randomly output 1~6. Here, please note the function random.randint(a, b). The output random number is a~(b-1).

Define a function to randomly display different patterns.

```python
while True:
    superbit.servo270(superbit.S1, 50)
    microbit.sleep(200)
    Face_show()
    microbit.sleep(200)
    superbit.servo270(superbit.S1, 140)
    microbit.sleep(1000)
```

Infinite loop, the servo turns to 50° (the face-changing mask is closed), and after a delay of 200 milliseconds, the random display pattern function Face_show() is called. After another delay of 200 milliseconds, the servo turns to 140° (the face-changing mask is open), and we can see that the pattern on the dot matrix has changed randomly.
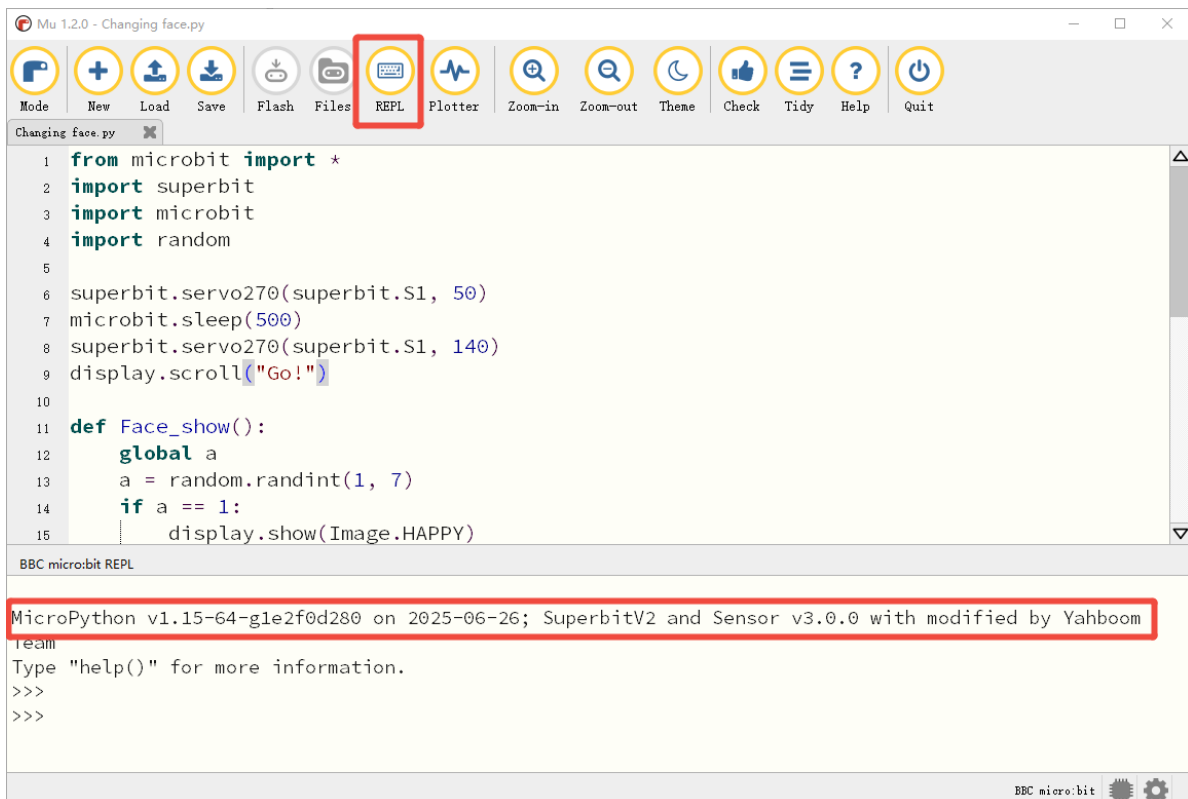
5. Write and download the program

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
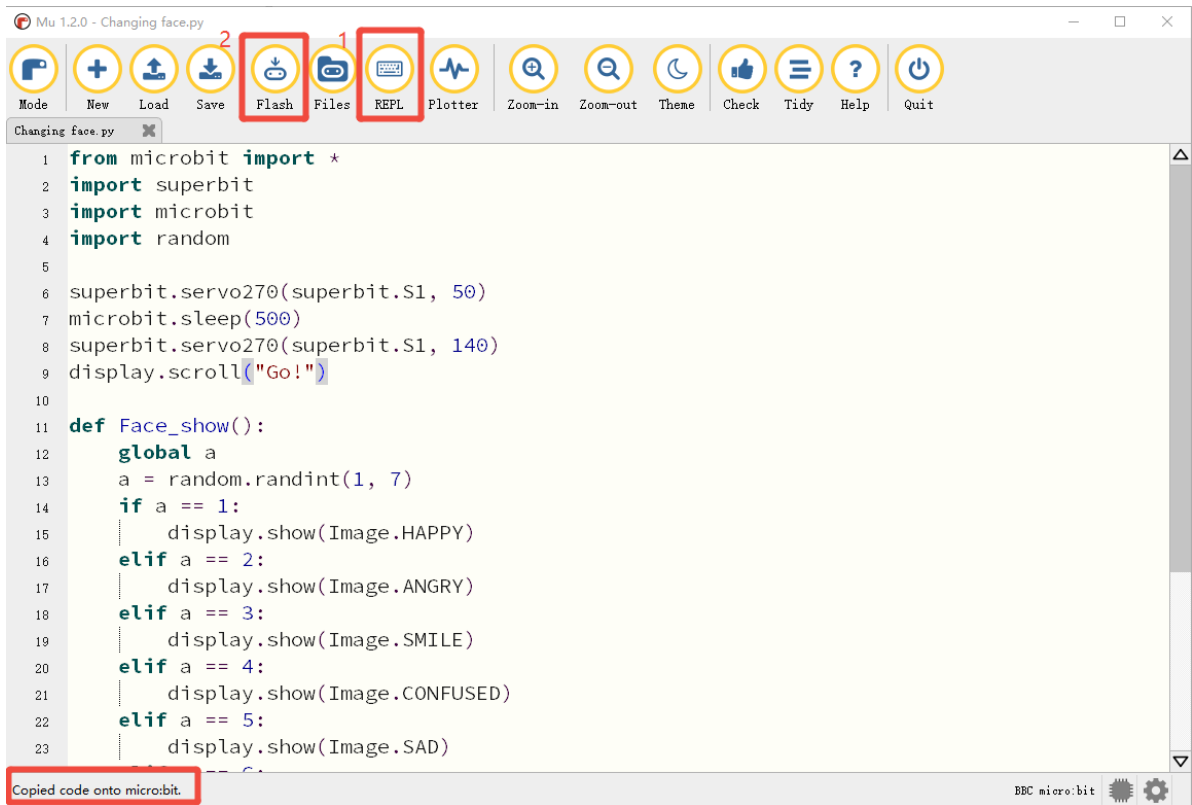
error, the lower left corner will prompt that there is no problem with the detection.
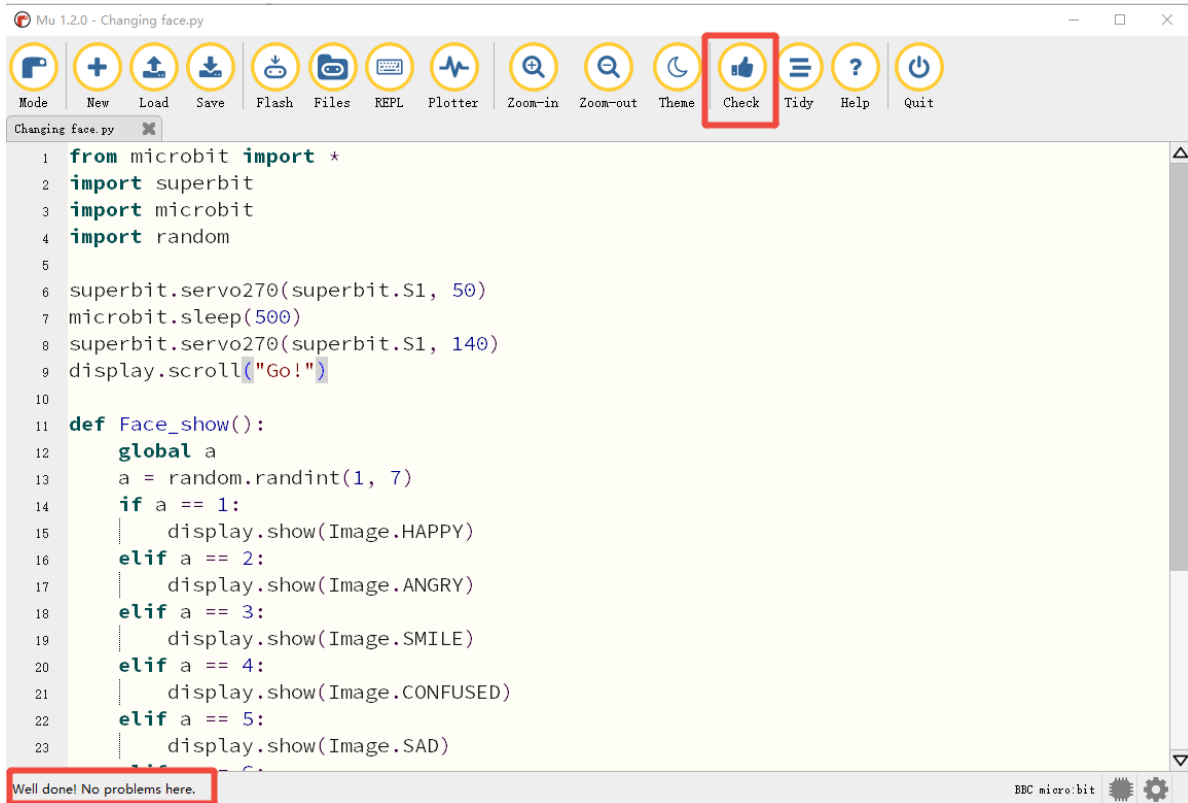
3. Click the 'REPL' button to check if the Superbit library has been downloaded. If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide].

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).**

```
1  from microbit import *
2  import superbit
3  import microbit
4  import random
5
6  superbit.servo270(superbit.S1, 50)
7  microbit.sleep(500)
8  superbit.servo270(superbit.S1, 140)
9  display.scroll("Go!")
10
11 def Face_show():
12     global a
13     a = random.randint(1, 7)
14     if a == 1:
15         display.show(Image.HAPPY)
16     elif a == 2:
17         display.show(Image.ANGRY)
18     elif a == 3:
19         display.show(Image.SMILE)
20     elif a == 4:
21         display.show(Image.CONFUSED)
22     elif a == 5:
23         display.show(Image.SAD)
```

Copied code onto micro:bit.

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 is successfully downloaded, turn on the power switch, the servo will initialize to 50° (the face-changing mask is closed), and after 500 milliseconds, the servo will rotate to 140° (the face-changing mask is open), and "GO!" will be displayed on the dot matrix. Then it will start to change faces randomly and keep looping in this state.

Six patterns are written in this program. You can modify the program to add more patterns.