

Changing face

1.Learning goals

In this course, we mainly learn how to use the Python programming to realize the "one-second face-changing" gameplay.

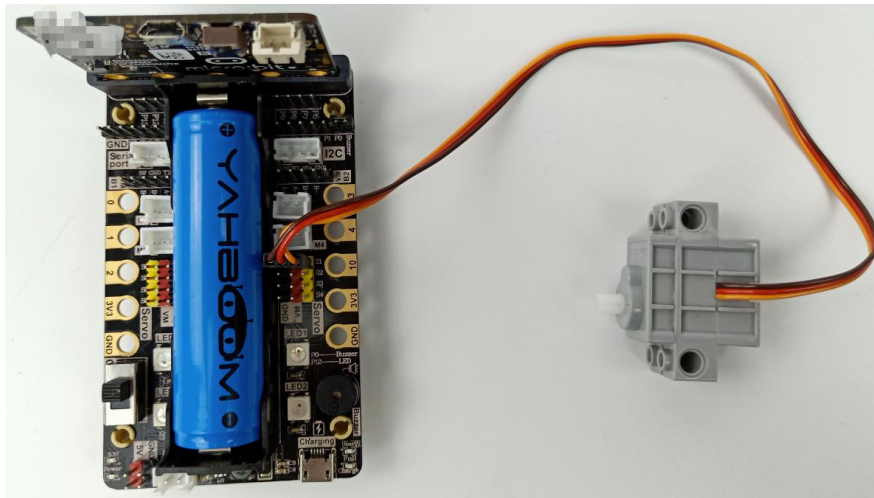
2.Building block assembly steps

For the building block construction steps, please refer to the installation manual or building block installation picture of [Assembly course]-[Changing face].

3.Wiring of servo

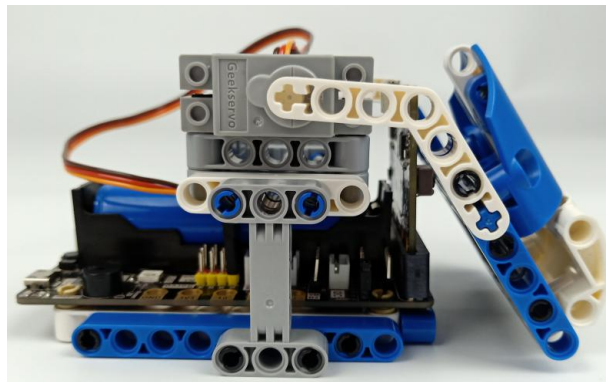
Building block servo insert into the Super: bit expansion board S1 interface, and the orange wiring connect the yellow pin of S1.

As shown below:



Note:

For the first course related to building block servo, we need to remove the gear on the servo and upload the program of this course to micro: bit. Then, turn on the power switch of the Super:bit expansion board and wait for the building block servo turn to the initial position. Next, we can turn off the power, and adjust the changing face mask to be off. Finally, install the servo. (If you have used programs related to clip robot before, you can skip this step)



4.Code and analysis

The program for this course, please view .py file.

```
1  from microbit import *
2  import superbitt
3  import microbit
4  import random
5
6  superbitt.servo270(superbitt.S1, 50)
7  microbit.sleep(500)
8  superbitt.servo270(superbitt.S1, 140)
9  display.scroll("Go!")
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)
24     elif a == 6:
25         display.show(Image.HEART)
26
27 while True:
28     superbitt.servo270(superbitt.S1, 50)
29     microbit.sleep(200)
30     Face_show()
31     microbit.sleep(200)
32     superbitt.servo270(superbitt.S1, 140)
33     microbit.sleep(1000)
```

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

superbitt.servo270(superbitt.S1, 50): Initialize the block servo to 50°;

`microbit.sleep(500)`: Delay 00 ms;
`superbit.servo270(superbit.S1, 140)`: Initialize the block servo to 140°;
`display.scroll("Go!")`: Scroll the string "Go!" on the dot matrix;

```
def Face_show():
    global a
    a = random.randint(1, 7): Random output 1~6, random.randint(a, b), the output
    random number is a~(b-1).
    if a == 1:
        display.show(Image.HAPPY)
    ...
Define a function to randomly display different patterns.
while True:
    superbit.servo270(superbit.S1, 50)
    microbit.sleep(200)
    Face_show()
    microbit.sleep(200)
    superbit.servo270(superbit.S1, 140)
    microbit.sleep(1000)
```

In an infinite loop, the servo turns to 50° (the hanging face mask is closed). After a delay of 200 ms, the random display pattern function `Face_show()` is called, after a further delay of 200 ms, the servo rotates to 140° (the changing face mask is opened), and we can see that the pattern on the dot matrix has randomly changed.

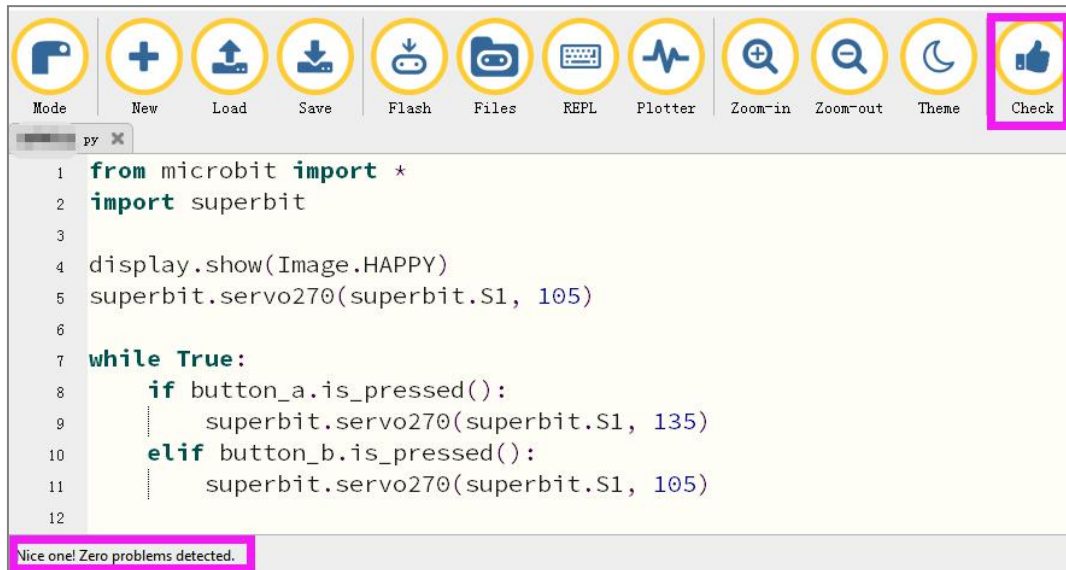
5. Writing and download code

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.

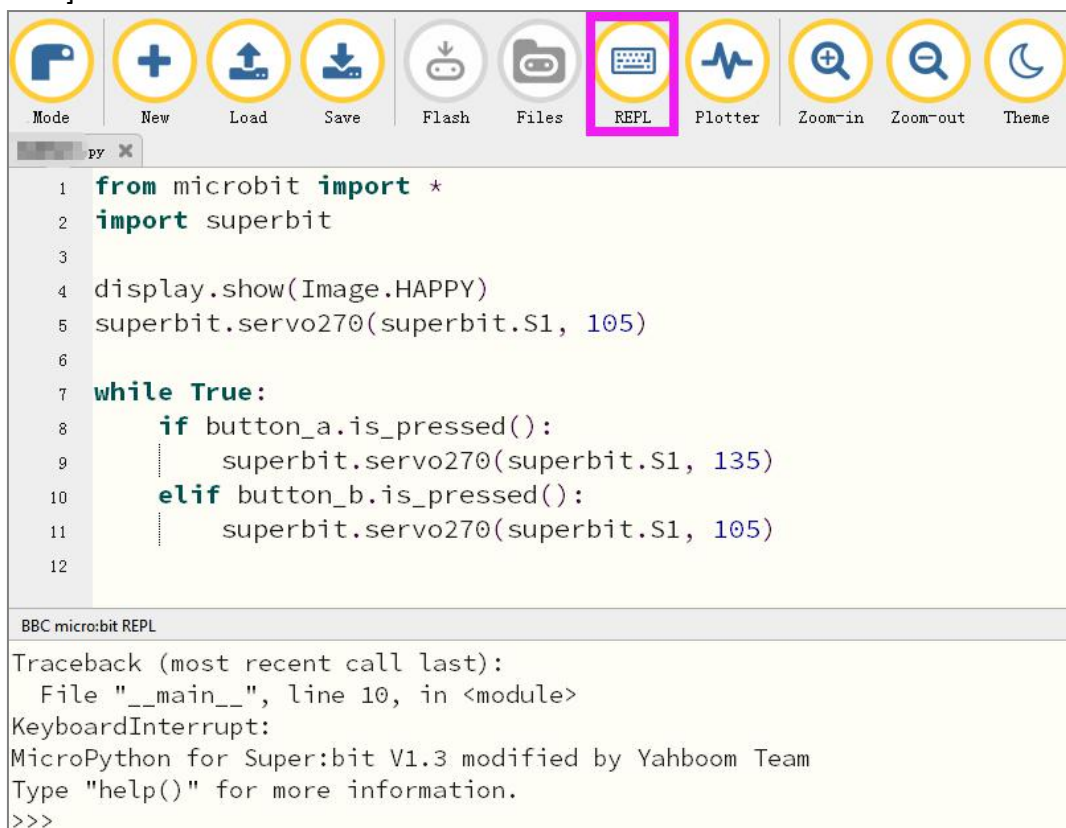
2. You can click the "Check" button to check if our code has an error.

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.

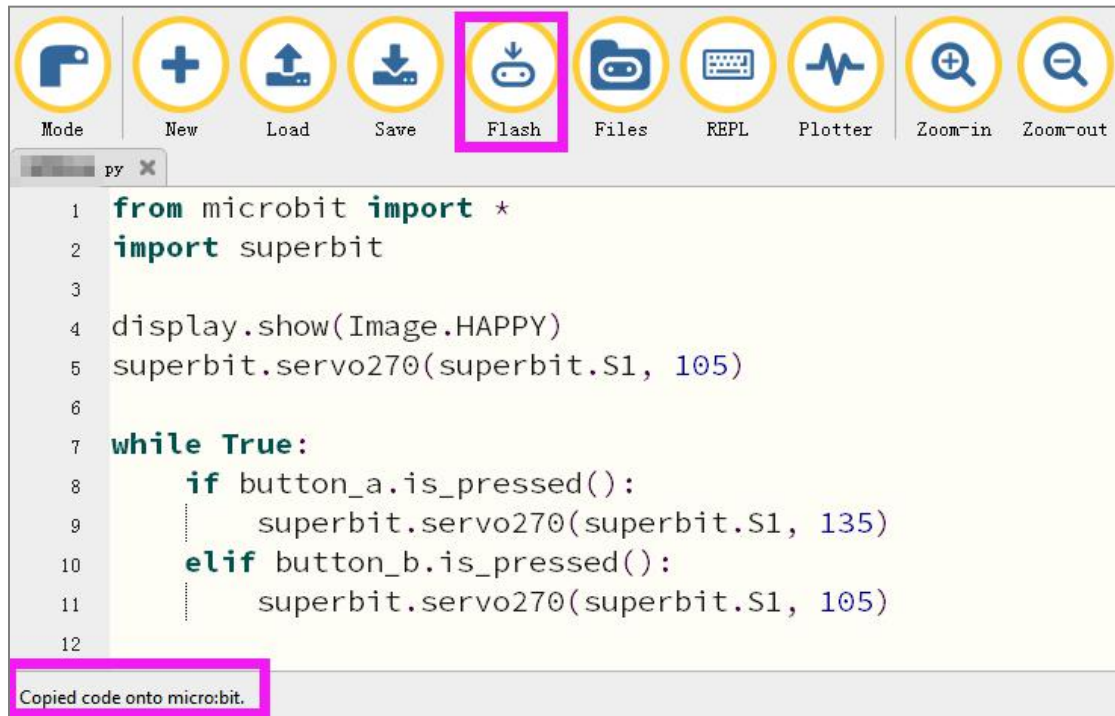


3. Click the 'REPL' button to check whether the Superbit library has been downloaded.

If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide] .



4. After the program is written, 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).



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 Super:bit Python library has been imported.

6.Experimental phenomena

After the program is successfully downloaded, turn on the power switch, and the servo will initialize to 50° (the changing face mask is closed). After 500 ms, the servo will turn to 140° (the changing face mask is on), and the dot matrix will display "GO!" Then, it began to change the face randomly, and keep a cycle in this state.

Tips: There are six patterns written in this program. You can modify the program yourself to add more patterns.