Soldiers on duty

1.Learning goals

In this course, we mainly learn how to use Python programming to realize following function.

Function: When the micro:bit A button is pressed, the dot matrix shows the direction of the door movement (to the left), the buzzer sounds, and the automatic door closes.

When the micro:bit B button is pressed, the dot matrix shows the direction of the door movement (to the right), the buzzer sounds shortly, and the automatic door opens.
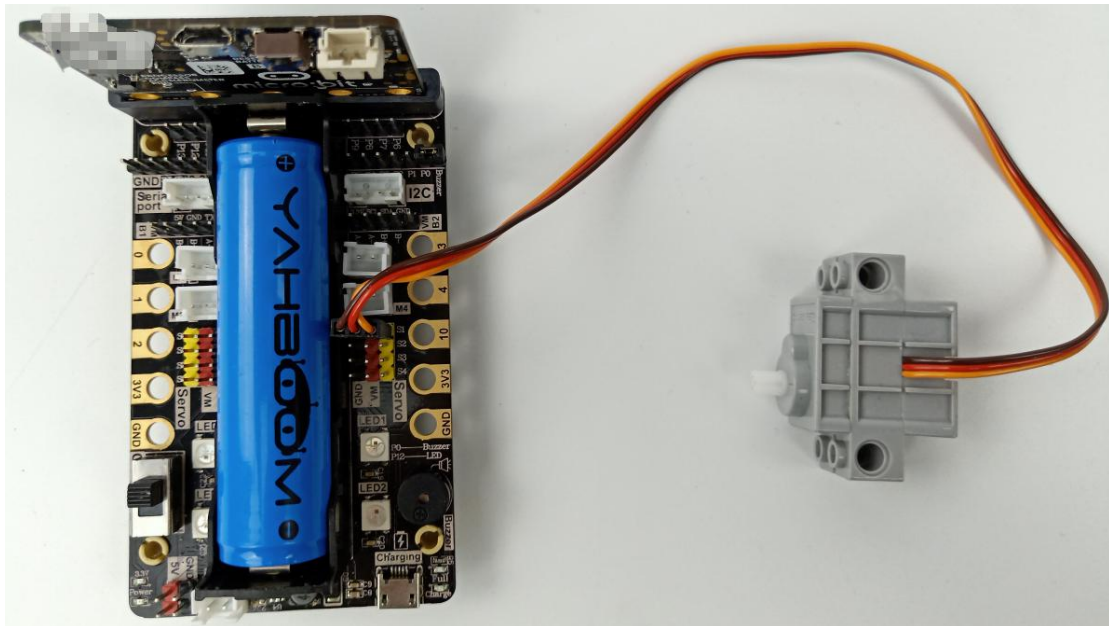
2.Building block assembly steps

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

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 automatic door is opened. Finally, install the servo. (If you have used programs related to clip robot before, you can skip this step)
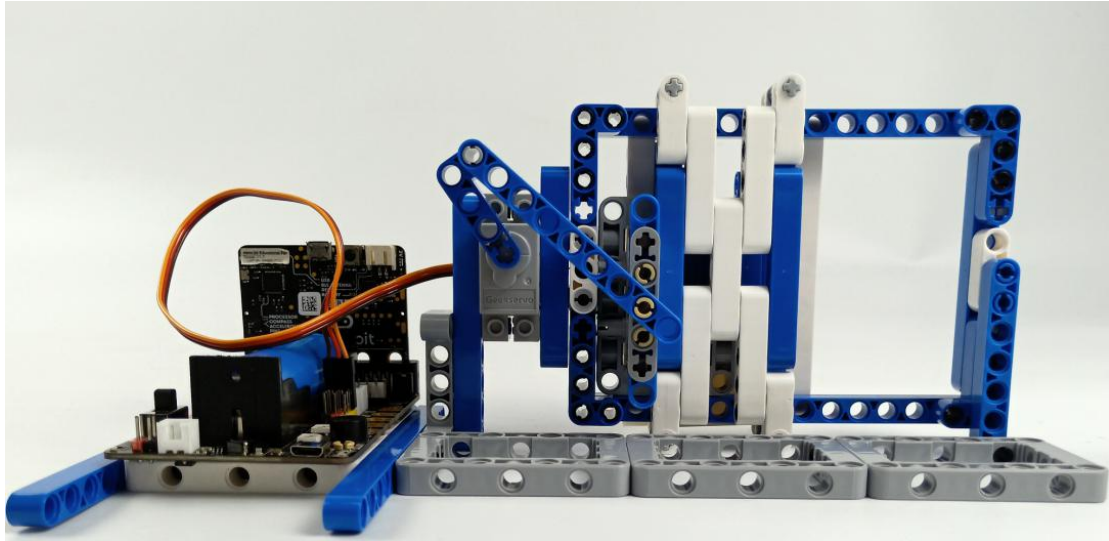
4.Code and analysis

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

```python
1  from microbit import *
2  import superbit
3  import music
4
5  superbit.servo270(superbit.S1, 90)
6  while True:
7      display.show(Image.HEART)
8      if button_a.is_pressed() is True and button_b.is_pressed() is False:
9          display.show(Image.ARROW_W)
10         music.play('B6:8')
11         superbit.servo270(superbit.S1, 0)
12     elif button_a.is_pressed() is False and button_b.is_pressed() is True:
13         display.show(Image.ARROW_E)
14         music.play('C3:4')
15         superbit.servo270(superbit.S1, 90)
```

First, we need to import the library needed for this lesson from micro:bit, superbit library is dedicated to super:bit expansion board; music library be used to play tone. superbit.servo270(superbit.S1, 90): Initialization makes the servo rotate to about 90°, that is open the door;

while True: Main loop
 display.show(Image.HEART)   Display heart pattern on micro:bit matrix.
 if button_a.is_pressed() is True and button_b.is_pressed() is False:
      display.show(Image.ARROW_W)
      music.play('B6:8')
      superbit.servo270(superbit.S1, 0)
   elif button_a.is_pressed() is False and button_b.is_pressed() is True:
      display.show(Image.ARROW_E)
      music.play('C3:4')
      superbit.servo270(superbit.S1, 90)

Determine whether the A and B buttons on the micro:bit board are pressed in an infinite loop.

When the micro:bit A button is pressed, the dot matrix shows the direction of the door movement (to the left), the buzzer sounds, and the automatic door closes;

When the micro:bit B button is pressed, the dot matrix shows the direction of the door movement (to the right), the buzzer sounds shortly, and the automatic door opens.
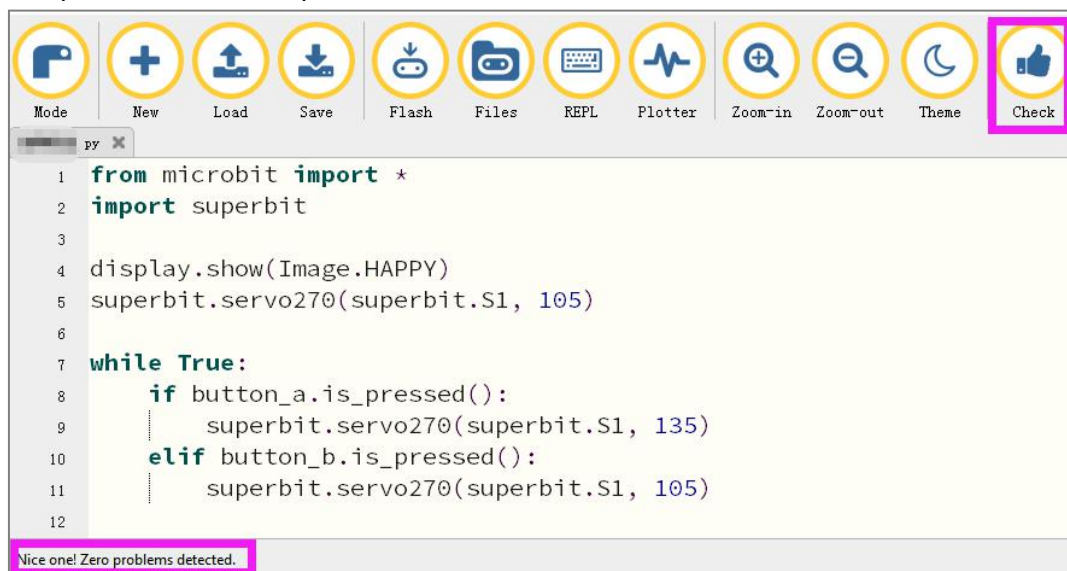
5.Writing and download code

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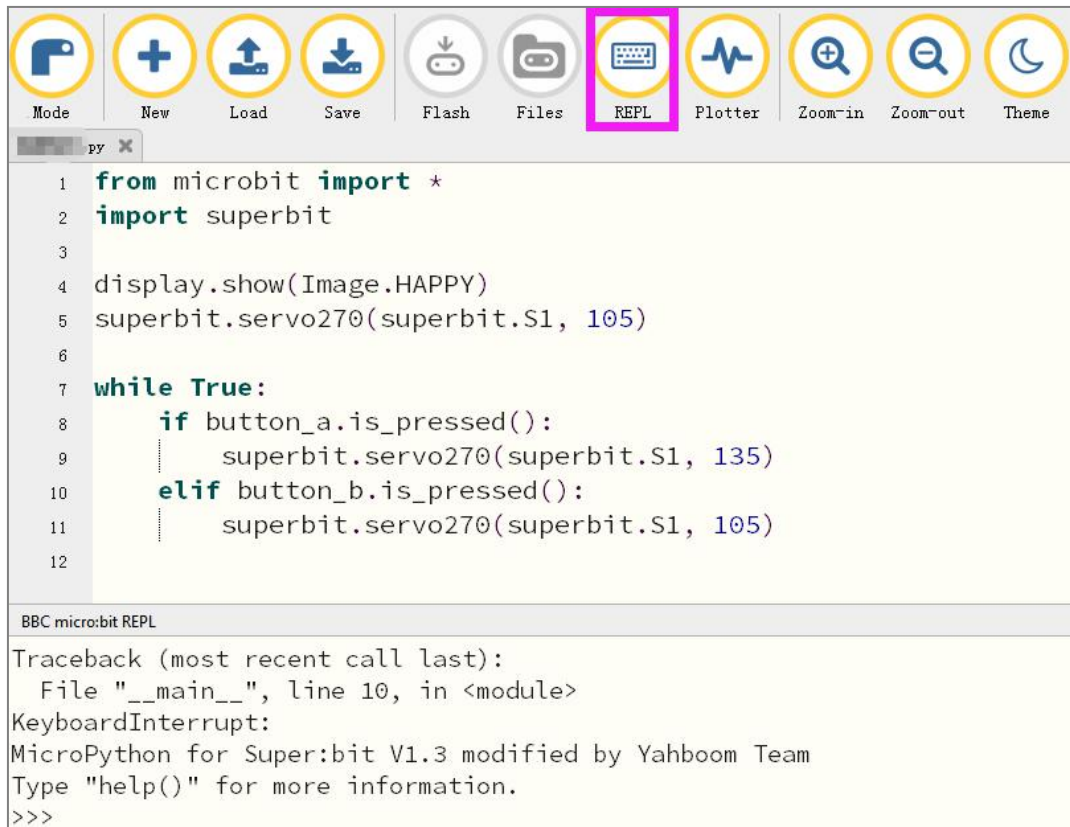
2.You can click the "Check" button to check if our code has an error.

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.

```
from microbit import *
import superbit

display.show(Image.HAPPY)
superbit.servo270(superbit.S1, 105)

while True:
    if button_a.is_pressed():
        superbit.servo270(superbit.S1, 135)
    elif button_b.is_pressed():
        superbit.servo270(superbit.S1, 105)
```
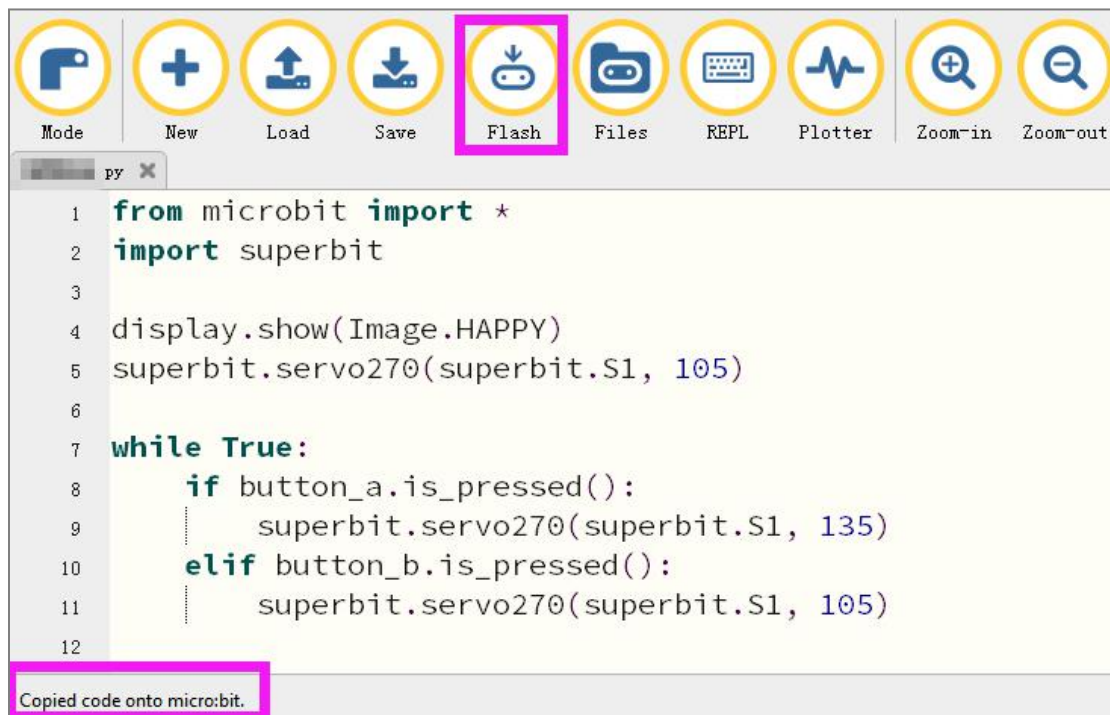
Nice one! Zero problems detected.

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] .

```
1  from microbit import *
2  import superbit
3
4  display.show(Image.HAPPY)
5  superbit.servo270(superbit.S1, 105)
6
7  while True:
8      if button_a.is_pressed():
9          superbit.servo270(superbit.S1, 135)
10     elif button_b.is_pressed():
11         superbit.servo270(superbit.S1, 105)
12
```

BBC micro:bit REPL

```
Traceback (most recent call last):
  File "__main__", line 10, in <module>
KeyboardInterrupt:
MicroPython for Super:bit V1.3 modified by Yahboom Team
Type "help()" for more information.
>>>
```

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).


```
1  from microbit import *
2  import superbit
3
4  display.show(Image.HAPPY)
5  superbit.servo270(superbit.S1, 105)
6
7  while True:
8      if button_a.is_pressed():
9          superbit.servo270(superbit.S1, 135)
10     elif button_b.is_pressed():
11         superbit.servo270(superbit.S1, 105)
12
```

Copied code onto micro:bit.

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, the micro: bit dot matrix will show a heart pattern.

When the micro:bit A button is pressed, the dot matrix shows the direction of the door movement (to the left), the buzzer sounds, and the automatic door closes.

When the micro:bit B button is pressed, the dot matrix shows the direction of the door movement (to the right), the buzzer sounds shortly, and the automatic door opens.