Music carousel

! Note: Due to the building block structure, the carousel only supports clockwise rotation. That is, the speed needs to be set to -255 during programming, and cannot be set to a positive number.

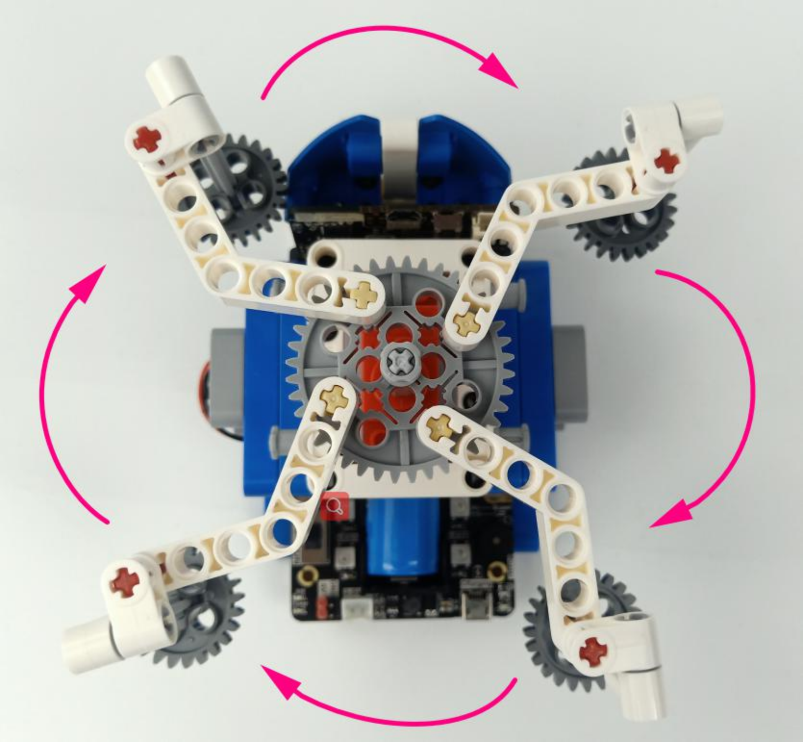
1.Learning goals

In this course, we mainly learn how to use the Python programming to realize the carousel rotation and “singing”, that is, the motor and buzzer work simultaneously.

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]-[Carousel].

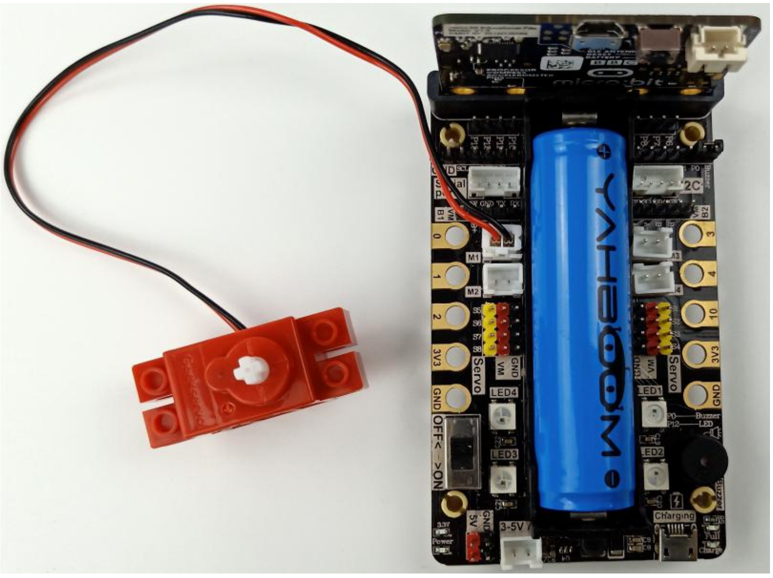
Please make sure that the direction of the L-shaped hole, as shown below.



3.Wiring of motor

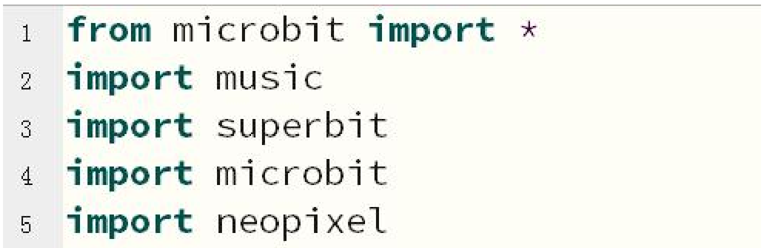
The motor wiring is inserted into the M1 interface of the Super:bit expansion board, and the black wire is close to the battery side;

As shown below.



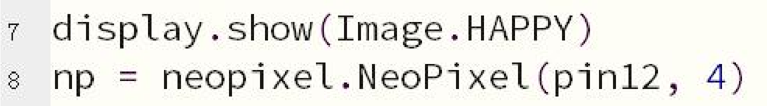
4.Code and analysis

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



```
from microbit import *
import music
import superbit
import microbit
import neopixel
```

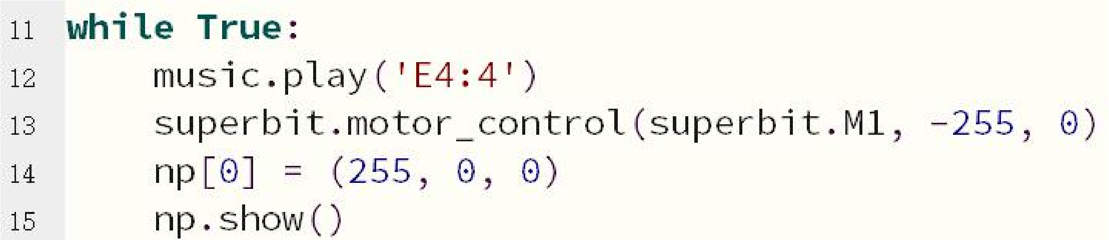
First, we need to import the library needed for this lesson from micro:bit, superbit library is dedicated to super:bit expansion board; music library is used to play music; neopixel is used to control RGB lights.



```
display.show(Image.HAPPY)
np = neopixel.NeoPixel(pin12, 4)
```

display.show(Image.HAPPY): Display smile pattern on micro:bit matrix.

np = neopixel.NeoPixel(pin12, 4)： RGB lamp initialization settings, a total of 4 RGB lamps, connected to the P12 pin of the micro:bit board (you can check the hardware interface manual);



```
while True:
    music.play('E4:4')
    superbit.motor_control(superbit.M1, -255, 0)
    np[0] = (255, 0, 0)
    np.show()
```

while True: In an infinite loop

music.play('E4:4')： The buzzer plays the tone. Parameter1 E4 represents the pitch, parameter2 4 represents the beat.

superbit.motor_control(superbit.M1, -255, 0)： The motor connected to the M1 interface reserve rotates with speed 255;

np[0] = (255, 0, 0)
np.show()： The first RGB light is red.
.....
In a loop

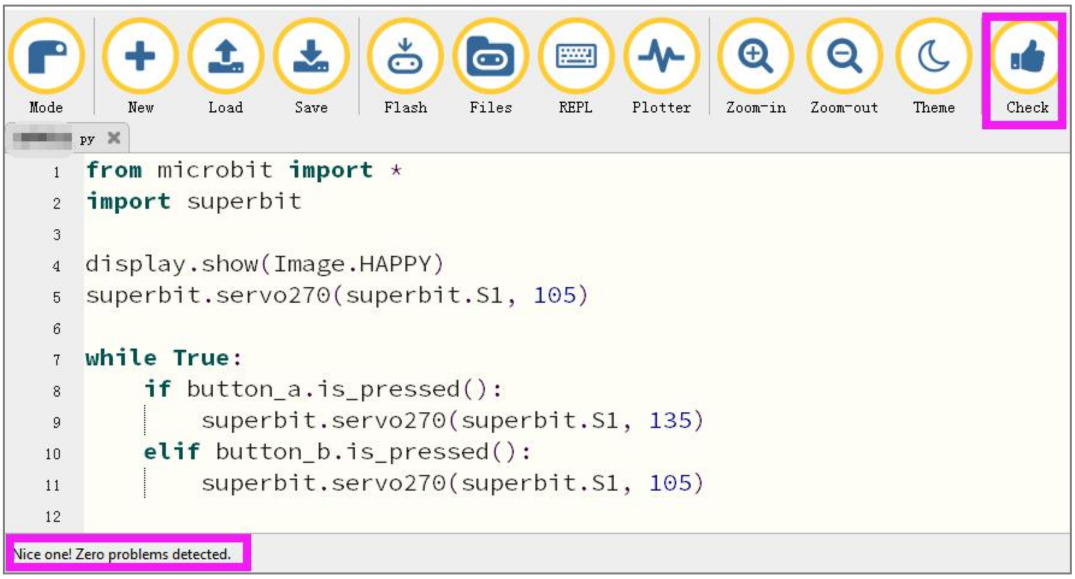
5. Writing and download code

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.

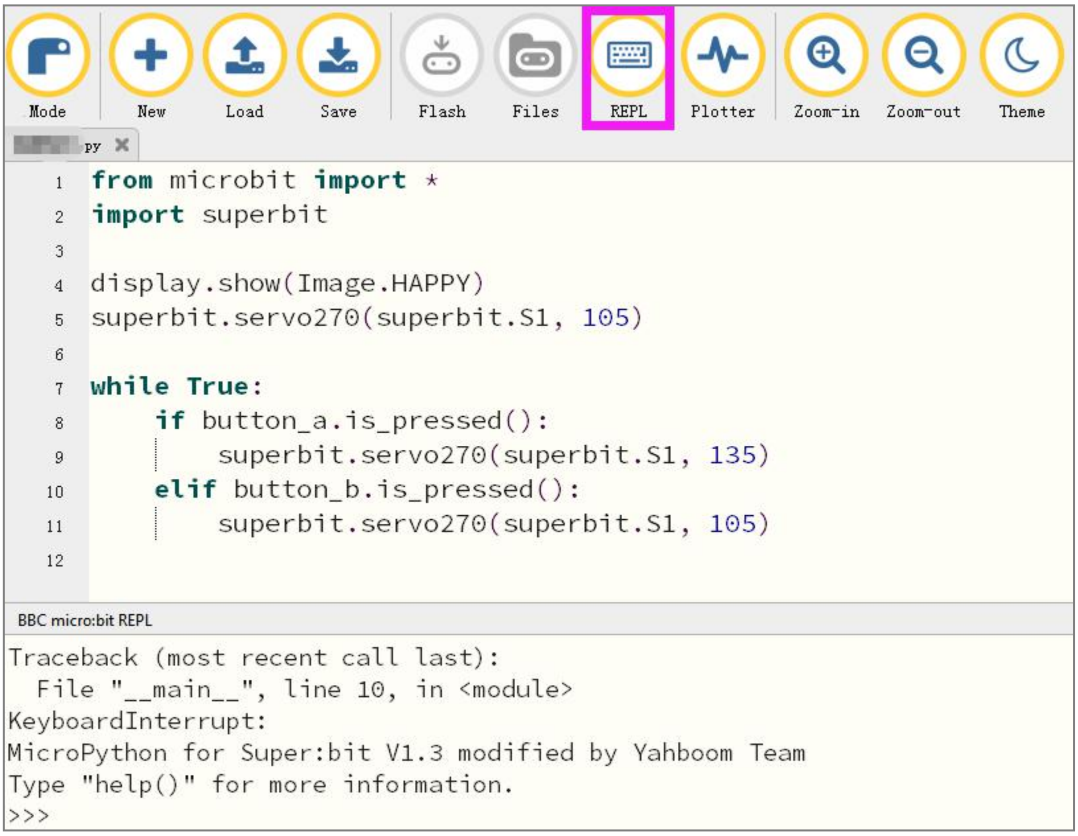


```
from microbit import *
import superbit

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] .

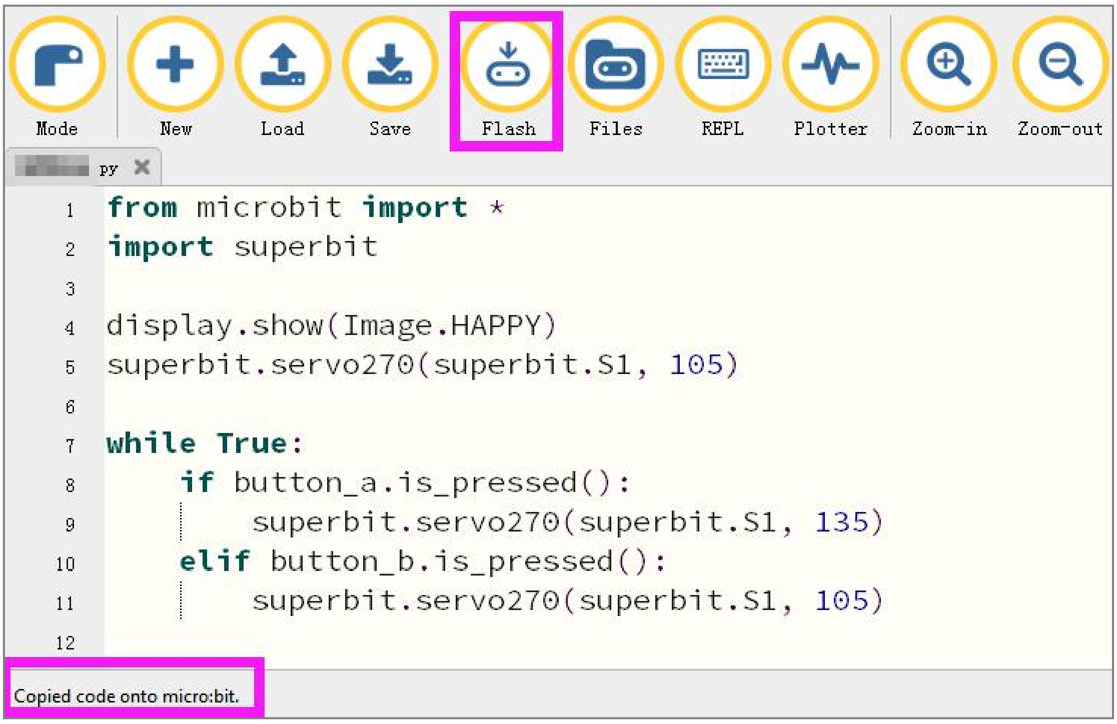


```
from microbit import *
import superbit

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)
```

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).



```
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)
```

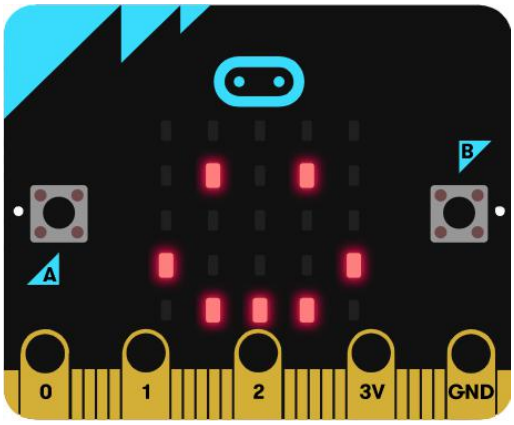
Copied code onto micro:bit.

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 display the smile pattern, as shown below. Then, the carousel rotates clockwise, the buzzer starts to play the music "Ode, RGB will also switch different colors.



If you need to restart, press the reset button on the back of the micro:bit board.