

Dancing

Dancing

1. Learning Objectives
2. Building Blocks
3. Motor 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 achieve the "singing" and "dancing" of the transporter, that is, the motor, servo, buzzer, and RGB light work at the same time.

2. Building Blocks

For the steps of building blocks, please refer to the installation drawings or the building block installation album of [Assembly course]-[Proficient carrier] in the materials.

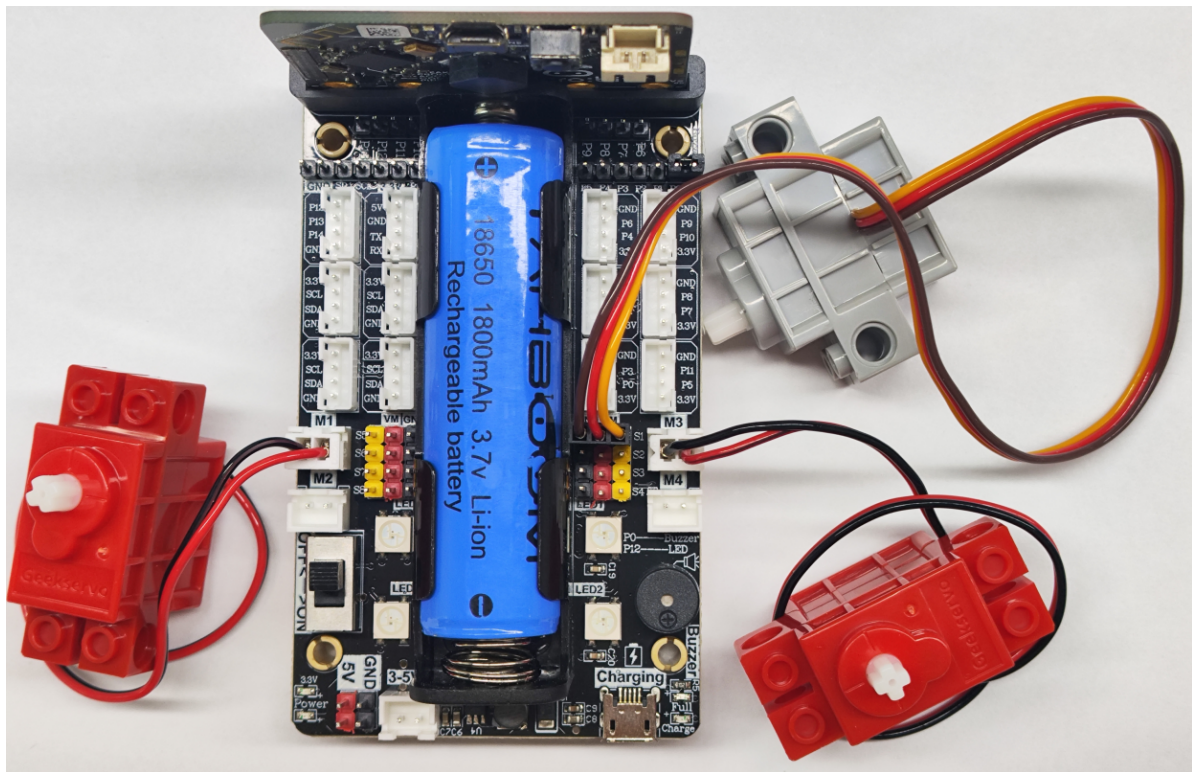
3. Motor Wiring

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

The motor wiring on the right side of the car is inserted into the M3 interface of the Super:bit expansion board, and the black wire is close to the battery side;

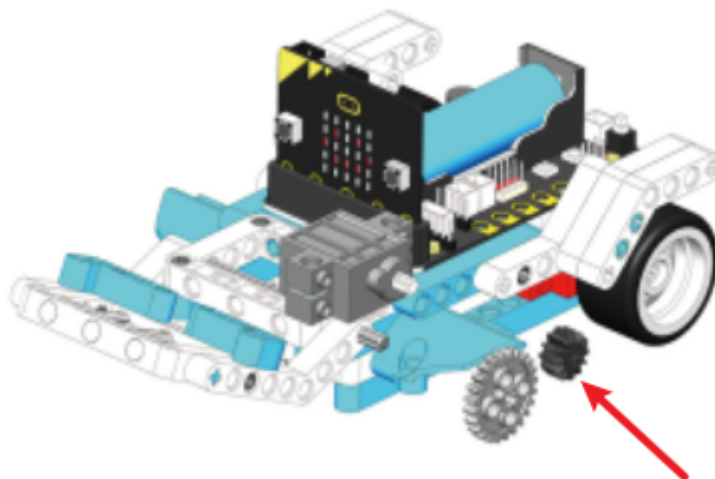
The building block servo wiring is inserted into the S1 interface of the Super:bit expansion board, and the orange servo wiring is inserted into the yellow pin of S1.

As shown in the figure below:



! Notes:

When taking a course related to building block servos for the first time, we need to remove the gears on the servos 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 servos to turn to the initial position; then, we can turn off the power, adjust the angle of the shovel of the car to be parallel to the ground, and then install the servo gear. (If you have used the program related to the transport master and servos before, you can skip this step)



4. Code analysis

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

```
from microbit import *
import music
import superbitt
import microbit
import neopixel
```

First, import the libraries needed for this lesson from microbit: the superbitt library is dedicated to the superbitt expansion board; the music library is used to play music; and the neopixel is used to control the RGB light.

```
display.show(Image.HAPPY)
np = neopixel.NeoPixel(pin12, 4)
superbitt.servo270(superbitt.S1, 120)
```

display.show(Image.HAPPY): Display a smiley face pattern on the microbit dot matrix;
np = neopixel.NeoPixel(pin12, 4): Initialize the RGB light. There are 4 RGB lights in total, connected to the P12 pin of the microbit motherboard (you can check the hardware interface manual);
superbitt.servo270(superbitt.S1, 105): Initialize the building block servo to rotate to about 105°;

```
while True:
    music.play('E4:4')
    superbitt.servo270(superbitt.S1, 60)
    superbitt.motor_control(superbitt.M1, 255, 0)
    superbitt.motor_control(superbitt.M3, 255, 0)
    np[0] = (255, 0, 0)
    np.show()
    ...
```

while True: infinite loop

music.play('E4:4'): The buzzer plays a tone. Parameter 1 E4 represents the tone, and parameter 2 4 represents the beat.

superbitt.servo270(superbitt.S1, 135): The building block servo rotates to 135°;

superbitt.motor_control(superbitt.M1, 255, 0): The motor connected to the M1 interface rotates forward at a speed of 255;

superbitt.motor_control(superbitt.M3, 255, 0): The motor connected to the M3 interface rotates forward at a speed of 255;

np[0] = (255, 0, 0)

np.show(): The first RGB light turns red.

Repeat this cycle.

About the tone:

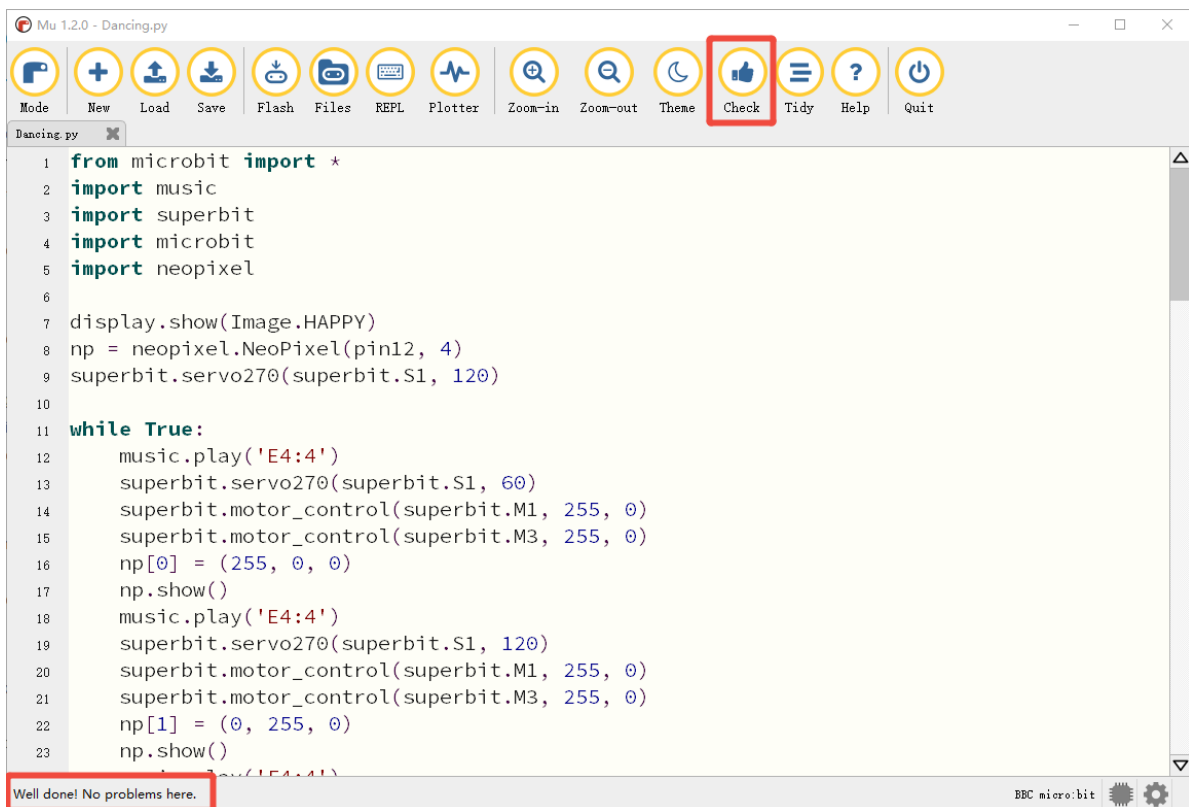
The song played here is "Ode to Joy". You can check the simple score of the song on the Internet and then write the corresponding program according to the simple score. For example:
music.play('E4:4'), music.play('F4:8') ...

5. Write and download the program

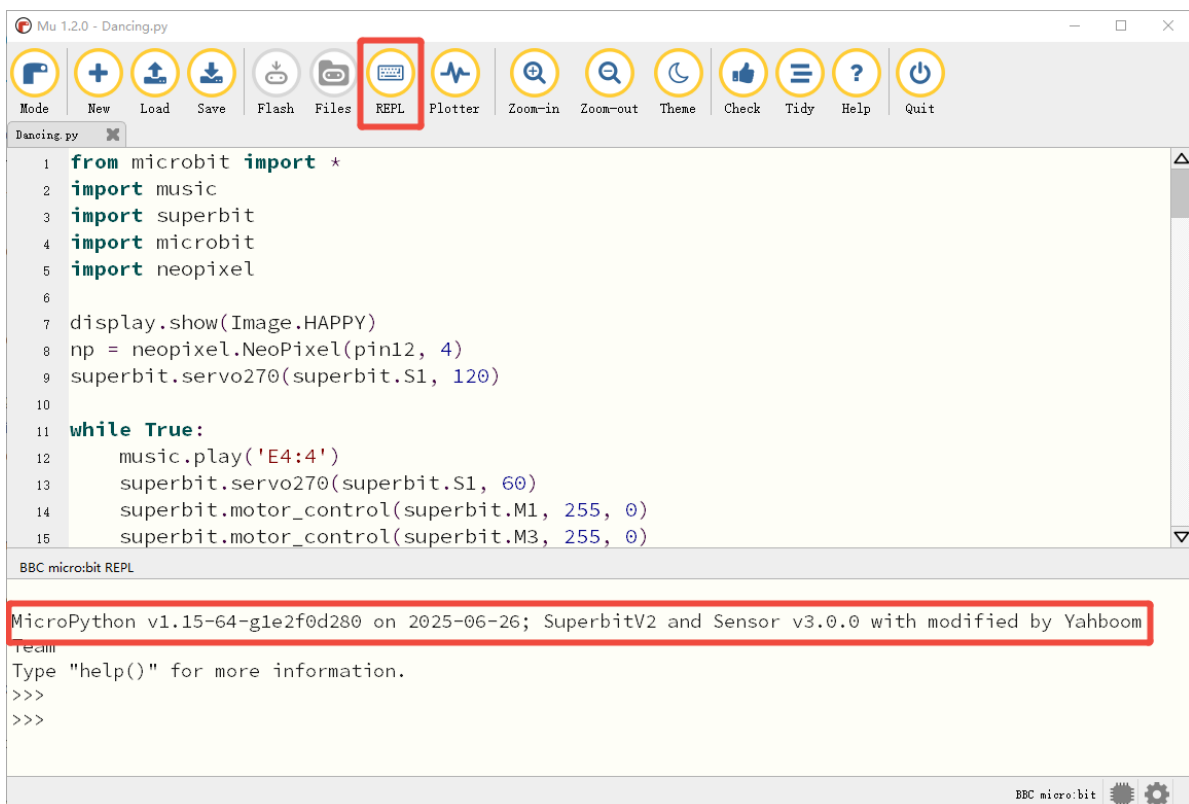
1. Open the Mu software and enter the code in the editing window. **Note! All English and symbols should be entered in English mode, use the Tab key for indentation, and the**

last line ends with a blank program.

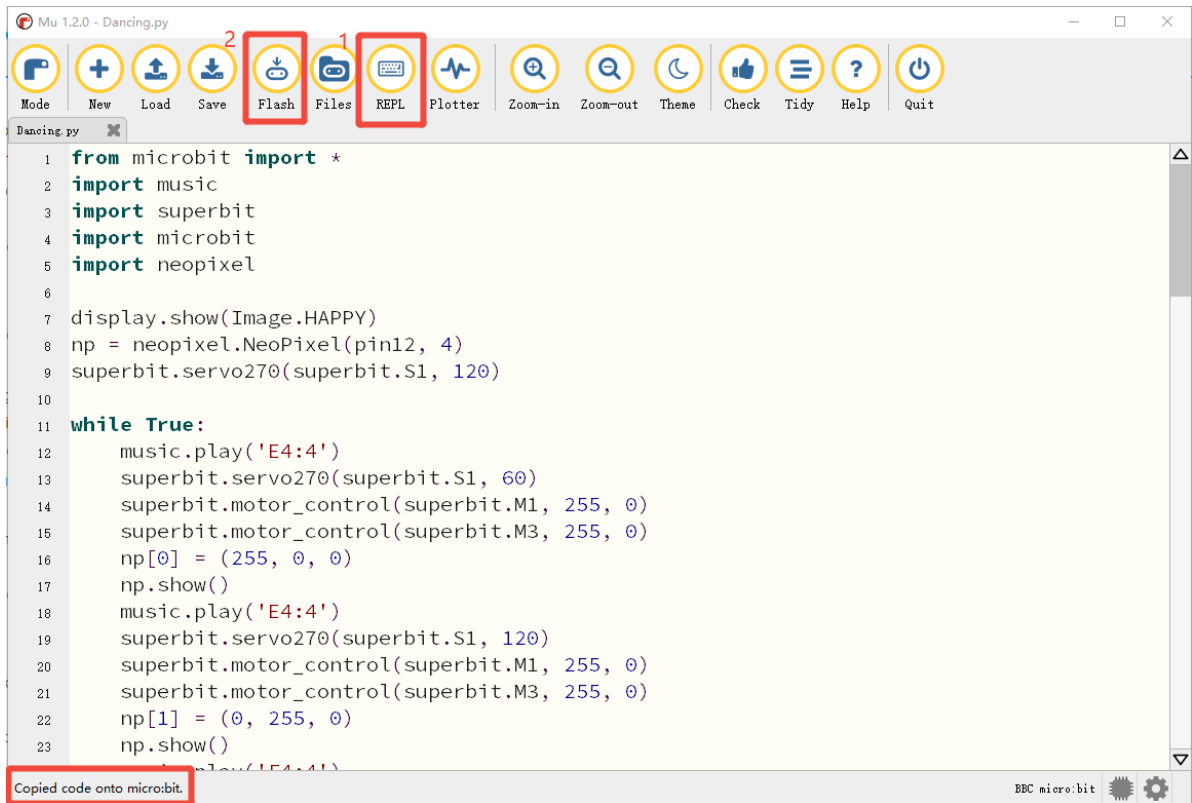
2. Click the thumb 'Check' button to check whether there are any errors in our code. If a cursor or underline appears in a line, it means a syntax error. Please check and modify it. If there is no error, the lower left corner will prompt that there is no problem with the detection.



3. Click the 'REPL' button to check whether the Superbit library has been downloaded. If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide].



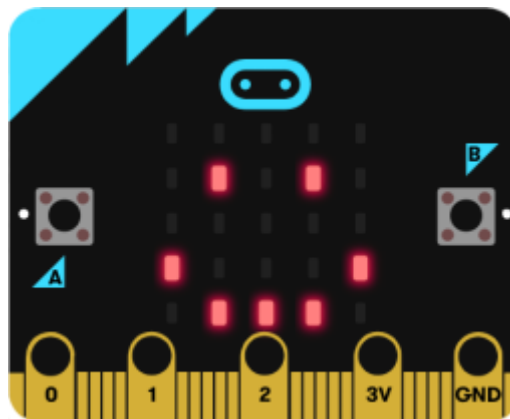
4. After the program is written, connect the computer and microbit mainboard with a microUSB data cable, click the 'Flash' button to download the program to the micro:bit mainboard. **(You need to click the 'REPL' button again to turn off the import library file function before you can download the program 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 is downloaded successfully, the micro:bit dot matrix will display a smiley face, as shown in the figure below. Turn on the power switch, the transporter will play the music "Ode to Joy", and will move forward-->backward-->rotate left-->rotate right-->turn left-->turn right, the RGB light will switch different colors, and the shovel will be continuously raised and lowered.



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