Dancer

1. Learning Objectives

In this course, we mainly learn how to use Python programming to make the flying car "sing" and "dance" at the same time, that is, the motor, servo, buzzer, and RGB light work at the same time.

2. Building Blocks

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

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 line 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 line is close to the battery side;

As shown in the figure below:
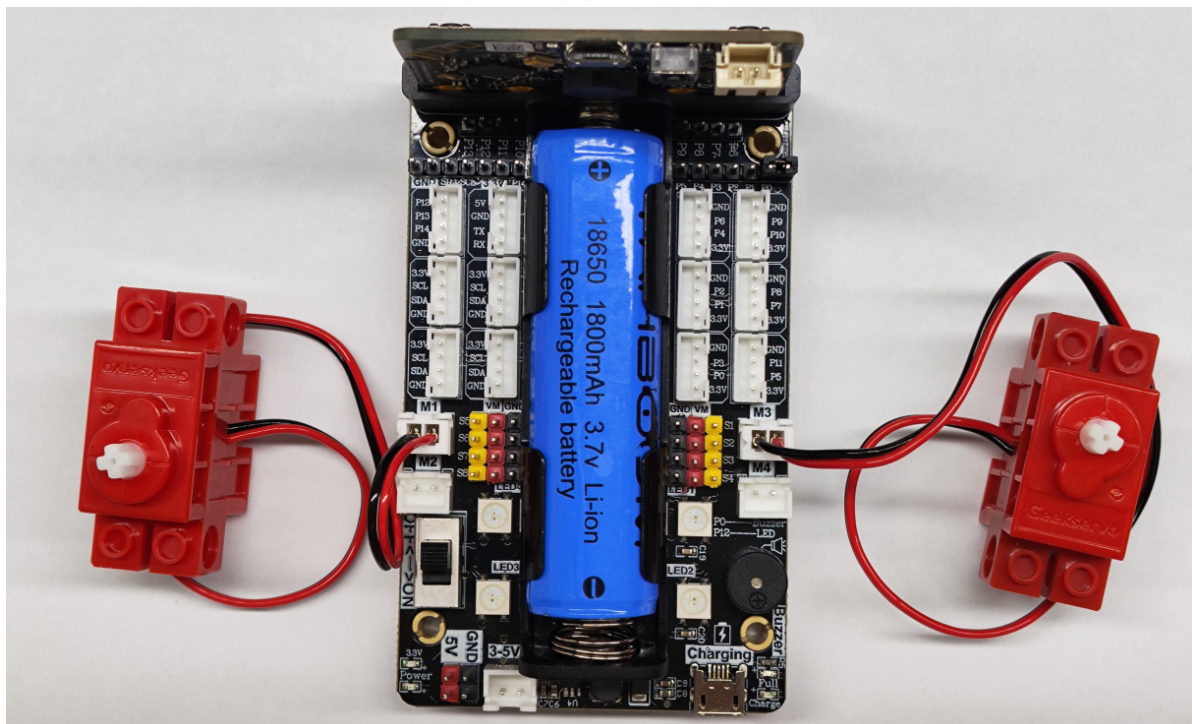
4. Code Analysis

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

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

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

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

display.show(Image.HAPPY): Display a smiley face pattern on the microbit dot matrix;

np = neopixel.NeoPixel(pin12, 4): Initialize the RGB light settings. There are 4 RGB lights in total, connected to the P12 pin of the microbit motherboard (you can check the hardware interface manual);

```python
while True:
    music.play('E4:4')
    superbit.motor_control(superbit.M1, 255, 0)
    superbit.motor_control(superbit.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 indicates the pitch, and parameter 2 4 indicates the beat.

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

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

superbit.motor_control(superbit.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.

This cycle repeats.

About the pitch:

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 based on the simple score. For example: music.play('E4:4'), music.play('F4:8') ...

5. Write and download programs

1. Open the Mu software and enter the code in the editing window. **Note! All English and symbols should be entered in English, use the Tab key for indentation, and the last line ends with a blank program.**

2. Click the thumb 'Check' button to check whether our code has errors. 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.
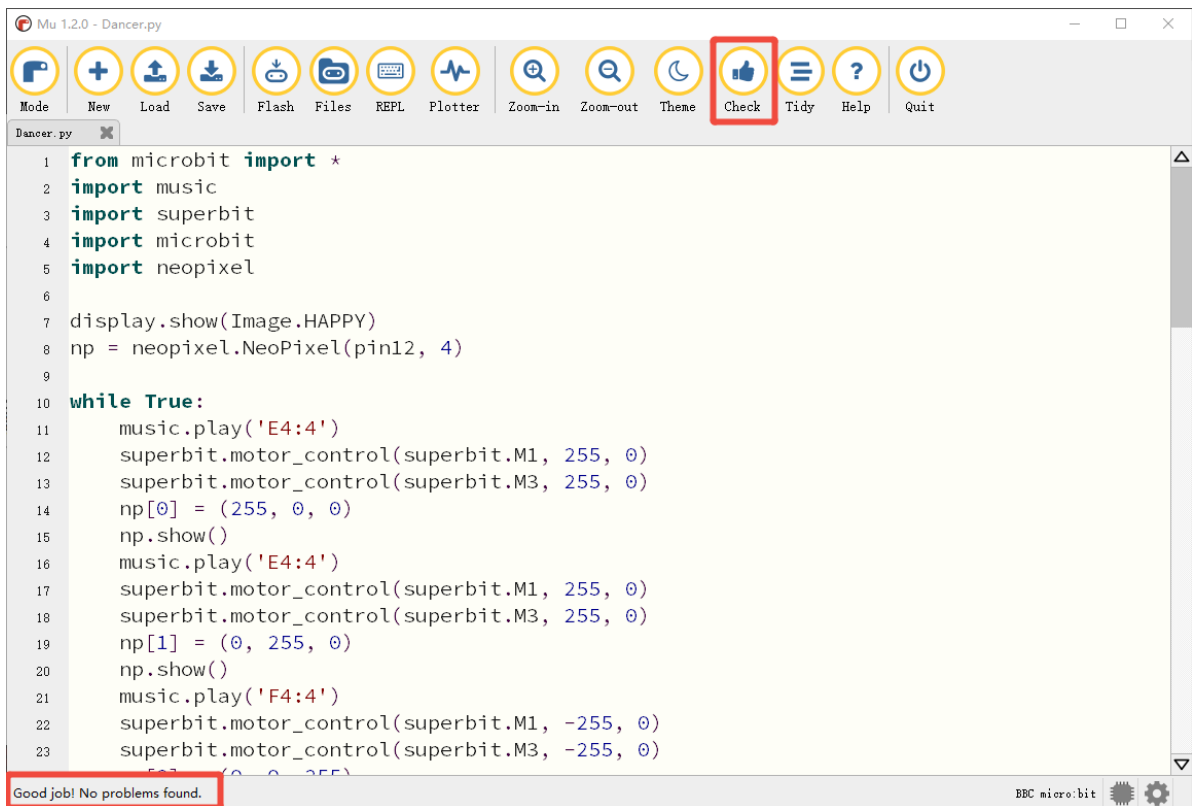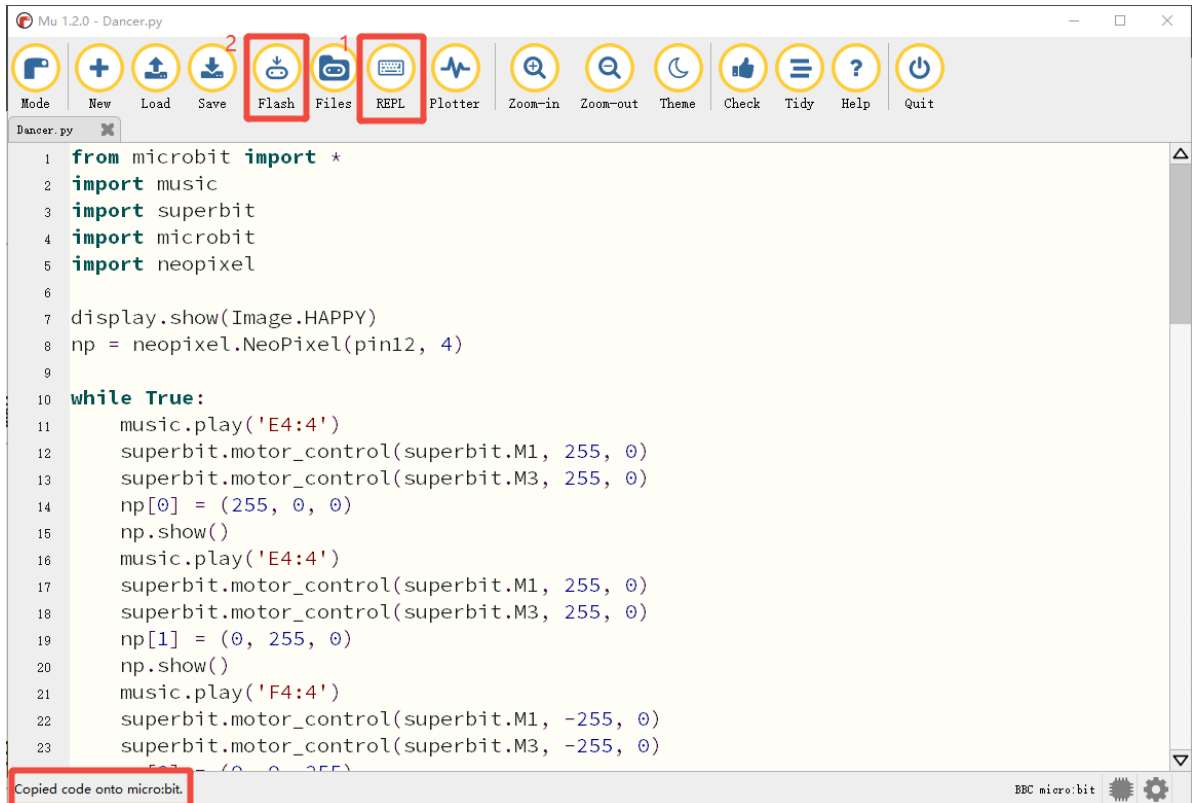

3. Click the 'REPL' button to check whether the Superbit library has been downloaded. If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide].
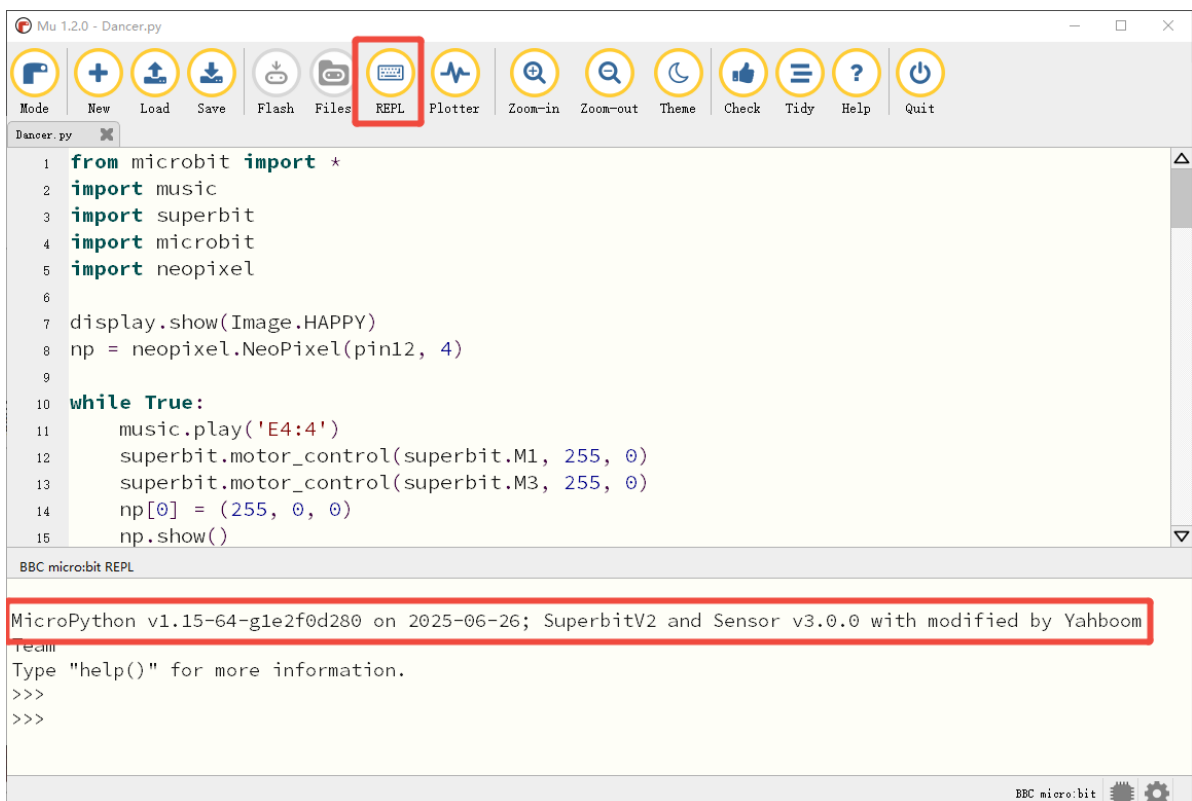
4. After the program is written, connect the computer and microbit mainboard with a microUSB data cable, click the 'Flash' button to download the program to the micro:bit

mainboard. **(You need to click the 'REPL' button again to turn off the import library file function before you can download the program normally).**


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

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

while True:
    music.play('E4:4')
    superbit.motor_control(superbit.M1, 255, 0)
    superbit.motor_control(superbit.M3, 255, 0)
    np[0] = (255, 0, 0)
    np.show()
    music.play('E4:4')
    superbit.motor_control(superbit.M1, 255, 0)
    superbit.motor_control(superbit.M3, 255, 0)
    np[1] = (0, 255, 0)
    np.show()
    music.play('F4:4')
    superbit.motor_control(superbit.M1, -255, 0)
    superbit.motor_control(superbit.M3, -255, 0)
```

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 car will play the music "Ode to Joy", and it will move forward-->backward-->left-->right to switch different motion states, and the RGB light will switch different colors.
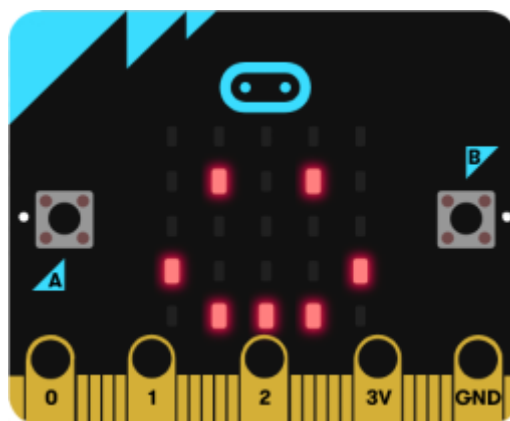
If you need to restart, please press the reset button on the back of the micro:bit motherboard.