Dance battle

1.Learning goals

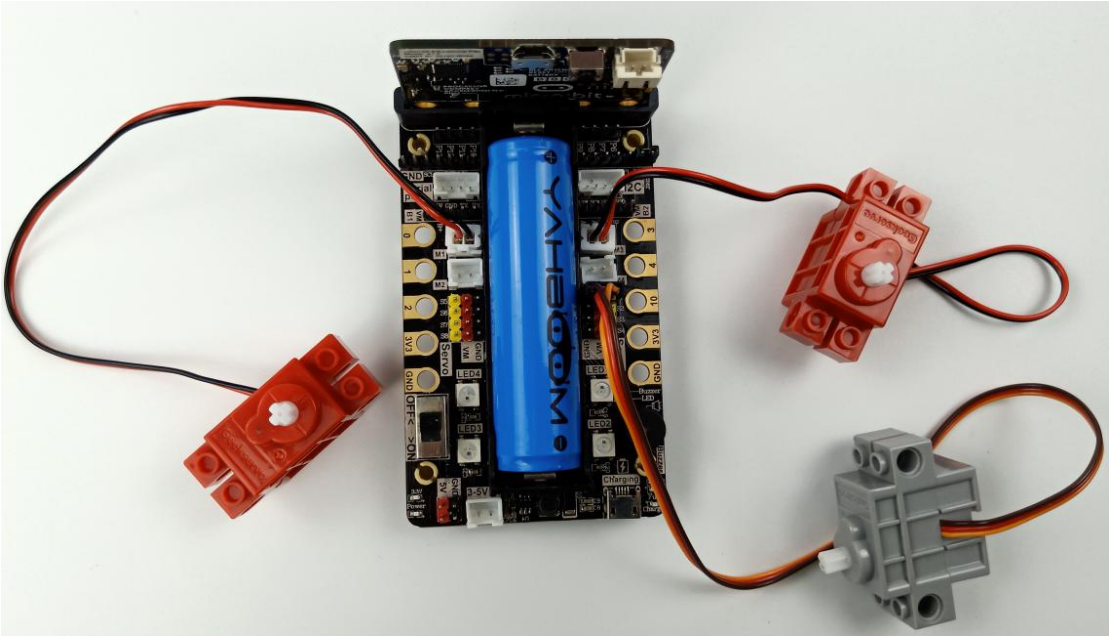
In this course, we mainly learn how to use the Python programming to realize the "singing" and "dancing" of the Skip car, that is, the servo, motor, buzzer and RGB lights work simultaneously.

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]-[Skip car].

3.Wiring of motor and servo

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;
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 loading platform of the car to keep it parallel to the ground. Finally, install the servo. (If you have used programs related to mobile shooter before, you can skip this step)

4.Code and analysis

The car 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; neopixel is used to control RGB lights.

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

display.show(Image.HAPPY)：Display smile pattern on micro:bit matrix.
np = neopixel.NeoPixel(pin12, 4)： RGB light initialization settings, a total of 4 RGB lights, (In this course, we only control one light) connected to the P12 pin of the micro:bit board (you can check the hardware interface manual).
superbit.servo270(superbit.S1, 120)：Initialize the servo to 120°;

```
while True:
    music.play('E4:4')
    superbit.servo270(superbit.S1, 240)
    superbit.motor_control(superbit.M1, 255, 0)
    superbit.motor_control(superbit.M3, 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.servo270(superbit.S1, 240)：Initialize the servo to 240°;
superbit.motor_control(superbit.M1, 255, 0)：The motor connected to the M1 interface rotates forward with speed 255;
superbit.motor_control(superbit.M3, 255, 0)：The motor connected to the M3 interface rotates forward with speed 255;
np[0] = (255, 0, 0)
np.show()：The first RGB light is red.
.....
In a loop

About tone:

In this course, we play “ODE”. Everyone can check the music notation of the song on some website, and then write the corresponding program according to the music notation. For example: music.play('E4:4'), music.play('F4:8') ...

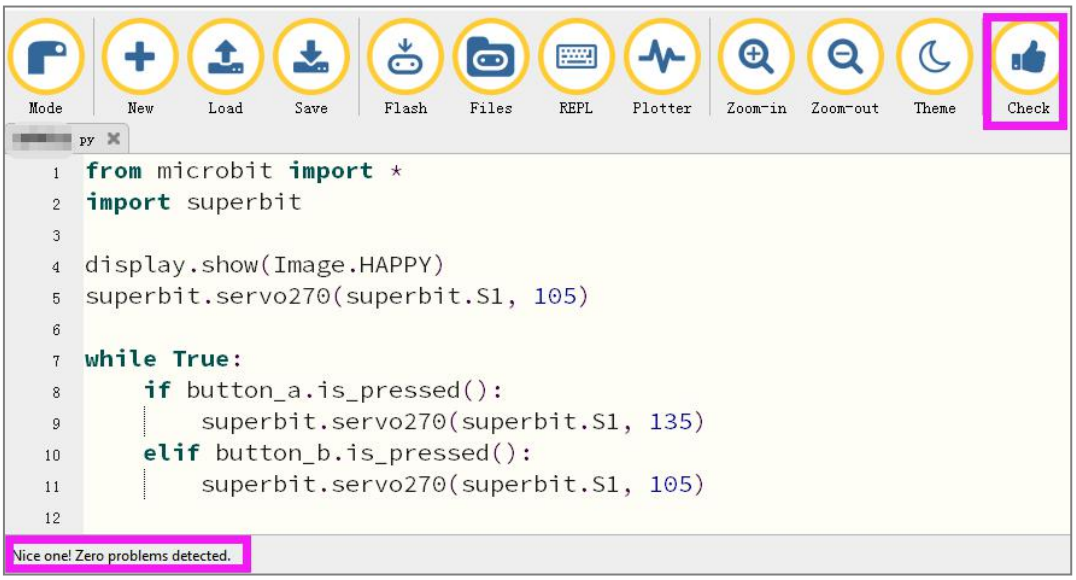
5.Writing and download code

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.

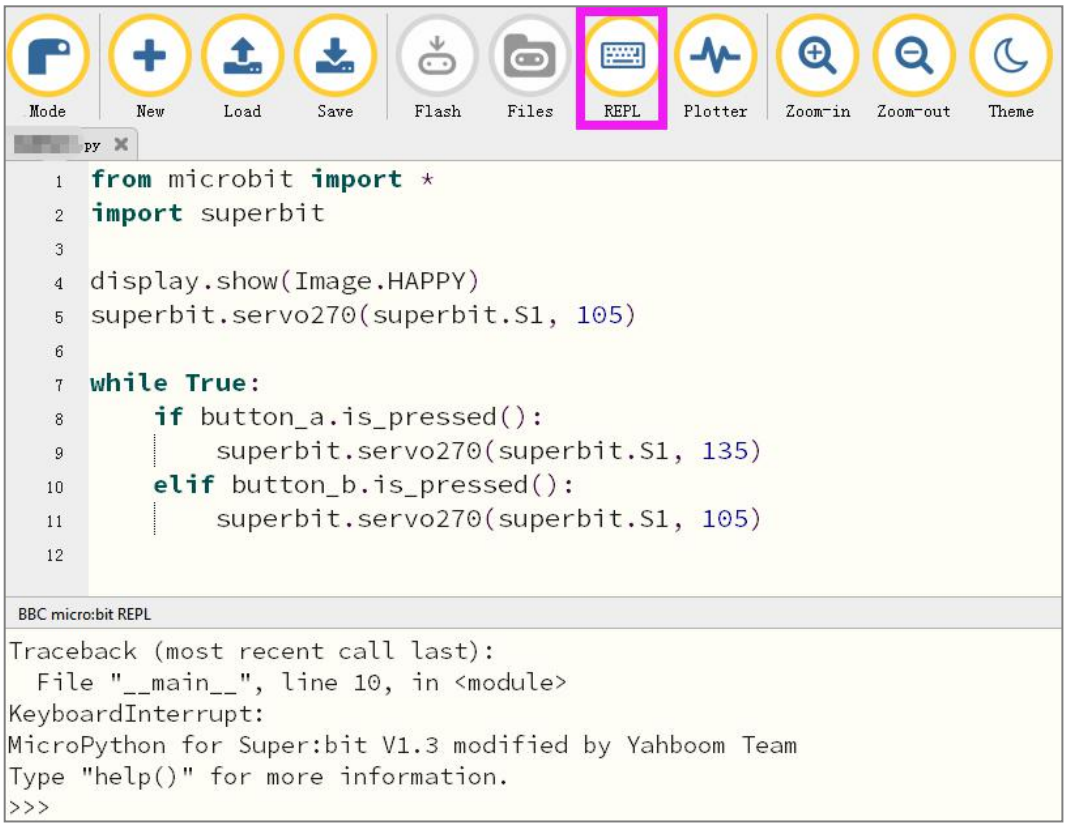
2.You can click the “Check” button to check if our code has an error.

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.

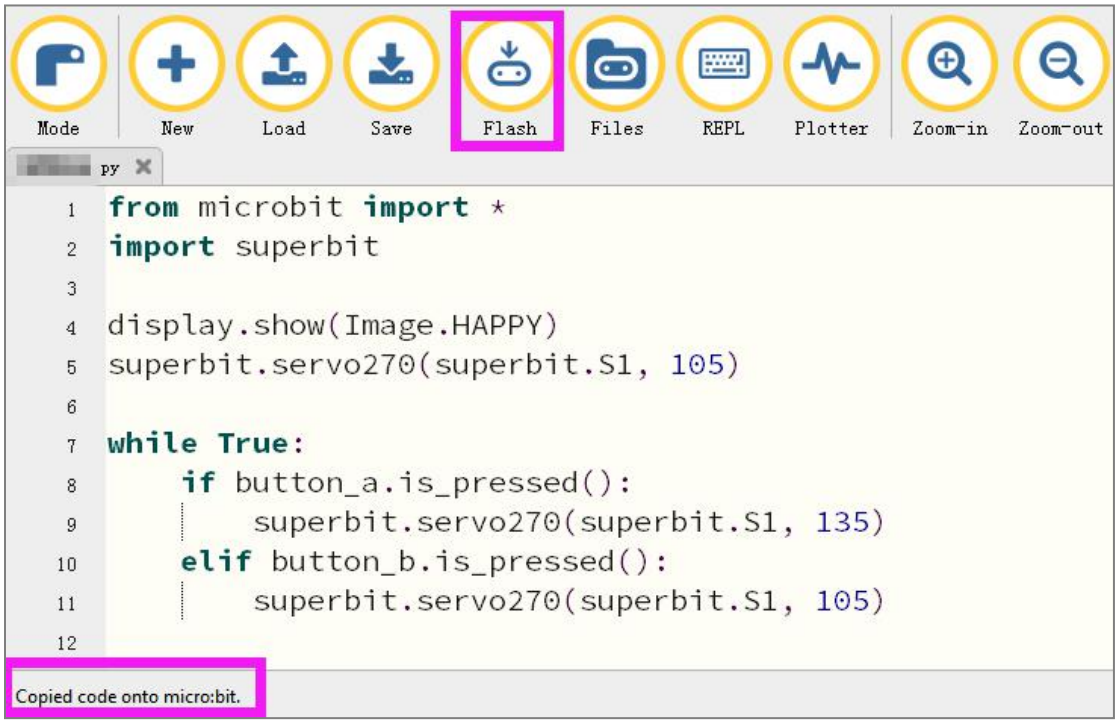


3.Click the ‘REPL’ button to check whether the Superbit library has been downloaded.

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).



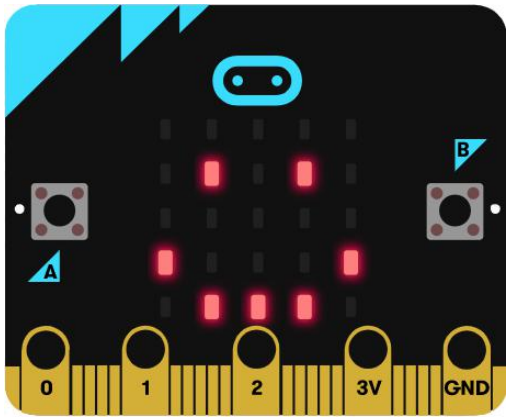
5.If the download failed, please confirm whether the micro:bit is connected to the computer through the micro USB data cable, and confirm whether the Super:bit

Python library has been imported.

6.Experimental phenomena

After the program is successfully downloaded, the micro: bit dot matrix will display the smile pattern, as shown below.

Open the power switch, the car will play the music "Ode", and it will go forward-> backward-> spin left-> spin right. RGB lights will switch different colors, shovel will raise->drop constantly.



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