

## Proficient Carrier Dance battle

### Learning goals

In this lesson, we mainly learn how to control building block motor and RGB lights by micro:bit and Super:bit expansion board.

### Code

```
1 from microbit import *
2 import music
3 import superbit
4 import microbit
5 import neopixel
6
7 display.show(Image.HAPPY)
8 np = neopixel.NeoPixel(pin12, 4)
9 superbit.servo270(superbit.S1, 120)
10
11 while True:
12     music.play('E4:4')
13     superbit.servo270(superbit.S1, 60)
14     superbit.motor_control(superbit.M1, 255, 0)
15     superbit.motor_control(superbit.M3, 255, 0)
16     np[0] = (255, 0, 0)
17     np.show()
18     music.play('E4:4')
19     superbit.servo270(superbit.S1, 120)
20     superbit.motor_control(superbit.M1, 255, 0)
21     superbit.motor_control(superbit.M3, 255, 0)
22     np[1] = (0, 255, 0)
23     np.show()
24     music.play('F4:4')
25     superbit.servo270(superbit.S1, 180)
26     superbit.motor_control(superbit.M1, -255, 0)
27     superbit.motor_control(superbit.M3, -255, 0)
28     np[2] = (0, 0, 255)
29     np.show()
30     music.play('G4:4')
31     superbit.servo270(superbit.S1, 120)
32     superbit.motor_control(superbit.M1, -255, 0)
33     superbit.motor_control(superbit.M3, -255, 0)
```

```
34     np[3] = (255, 255, 0)
35     np.show()
36     music.play('G4:4')
37     superbit.servo270(superbit.S1, 60)
38     superbit.motor_control(superbit.M1, 255, 0)
39     superbit.motor_control(superbit.M2, -255, 0)
40     np[0] = (0, 255, 255)
41     np.show()
42     music.play('F4:4')
43     superbit.servo270(superbit.S1, 120)
44     superbit.motor_control(superbit.M1, 255, 0)
45     superbit.motor_control(superbit.M2, -255, 0)
46     np[1] = (255, 0, 255)
47     np.show()
48     music.play('E4:4')
49     superbit.servo270(superbit.S1, 180)
50     superbit.motor_control(superbit.M1, -255, 0)
51     superbit.motor_control(superbit.M2, 255, 0)
52     np[2] = (255, 255, 255)
53     np.show()
54     music.play('D4:4')
55     superbit.servo270(superbit.S1, 120)
56     superbit.motor_control(superbit.M1, -255, 0)
57     superbit.motor_control(superbit.M2, 255, 0)
58     np[3] = (255, 0, 0)
59     np.show()
60     music.play('C4:4')
61     superbit.servo270(superbit.S1, 60)
62     superbit.motor_control(superbit.M1, -255, 0)
63     superbit.motor_control(superbit.M2, -255, 0)
64     np[0] = (0, 0, 255)
65     np.show()
66     music.play('C4:4')
```

```

67     superbit.servo270(superbit.S1, 120)
68     superbit.motor_control(superbit.M1, -255, 0)
69     superbit.motor_control(superbit.M2, -255, 0)
70     np[1] = (255, 255, 255)
71     np.show()
72     music.play('D4:4')
73     superbit.servo270(superbit.S1, 180)
74     superbit.motor_control(superbit.M1, 255, 0)
75     superbit.motor_control(superbit.M2, 255, 0)
76     np[2] = (255, 255, 0)
77     np.show()
78     music.play('E4:4')
79     superbit.motor_control(superbit.M1, 255, 0)
80     superbit.motor_control(superbit.M2, 255, 0)
81     music.play('E4:6')
82     superbit.servo270(superbit.S1, 60)
83     np[3] = (0, 255, 0)
84     np.show()
85     superbit.motor_control(superbit.M1, 255, 0)
86     superbit.motor_control(superbit.M2, -255, 0)
87     music.play('D4:2')
88     superbit.servo270(superbit.S1, 120)
89     np[0] = (255, 0, 255)
90     np.show()
91     superbit.motor_control(superbit.M1, 255, 0)
92     superbit.motor_control(superbit.M2, -255, 0)
93     music.play('D4:2')
94     superbit.servo270(superbit.S1, 180)
95     np[1] = (0, 0, 255)
96     np.show()
97     superbit.motor_control(superbit.M1, -255, 0)
98     superbit.motor_control(superbit.M2, 255, 0)
99     microbit.sleep(500)

```

## Assembly steps

Please refer to the **Proficient Carrier assembly steps in the Assembly instructions** folder for building blocks assembly steps.

## About wiring

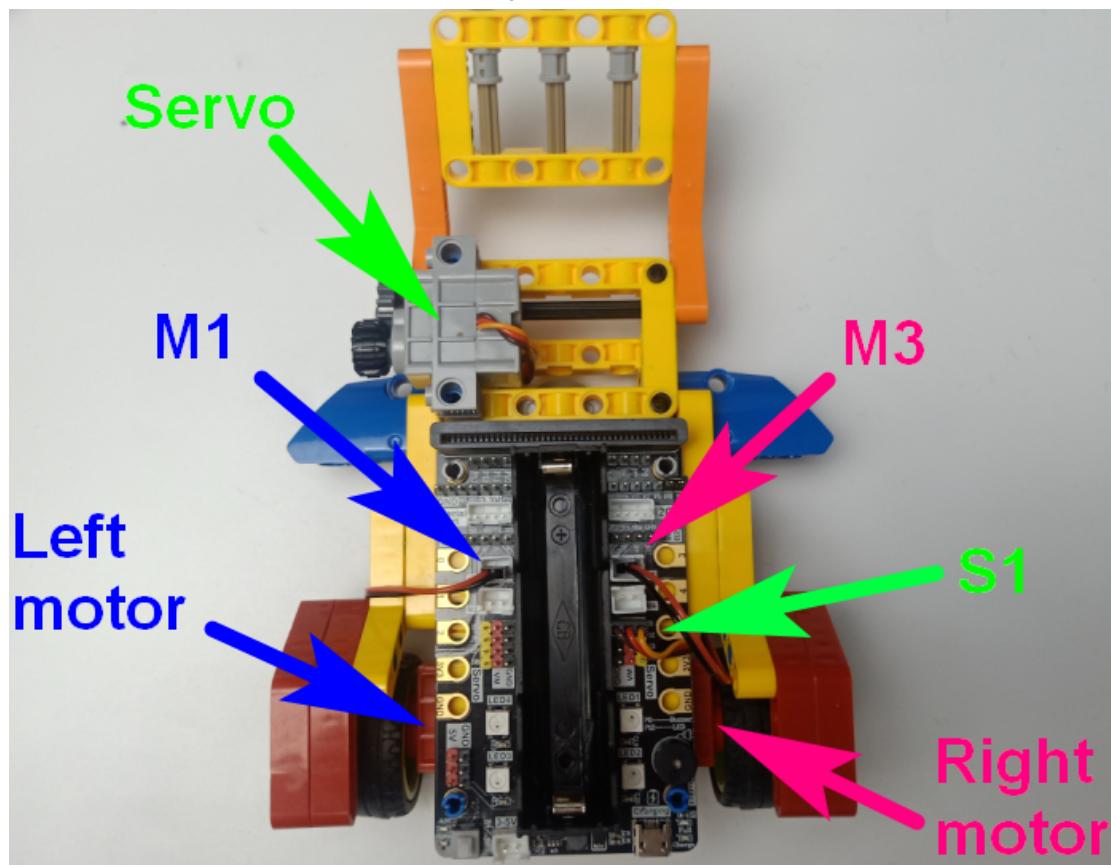
As shown below,

[Left motor connect to M1 interface of super:bit.](#)

Right motor connect to M3 interface of super:bit.

The black wiring of the motor is near the battery side.

Servo connect to S1 interface of super:bit.



**!!!Note:When using the program related to the building block servo for the first time.**

First, we need to remove the gear on the building block servo. Then, download the program to micro:bit. Next, turn on the power of the Super:bit expansion board and wait for the building block servo to turn to the initial position and turn off the power. Finally, we need to open the trolley clip to the widest point, and then install the gear on the building block servo.

#### Programming and downloading :

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, and the last line must be a space.**

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Voice control light.py ×
6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0

```

2. You can click the “**Check**” button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Voice control light.py ×
6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0
13
14
15 while True:
16     voice = tinybit.getVoicedata()
17     if voice > 100:

```

3. Click “**REPL**” button, check whether the tinybit library has been downloaded. If not, please refer to the [preparation before class]---> [Python programming]

```

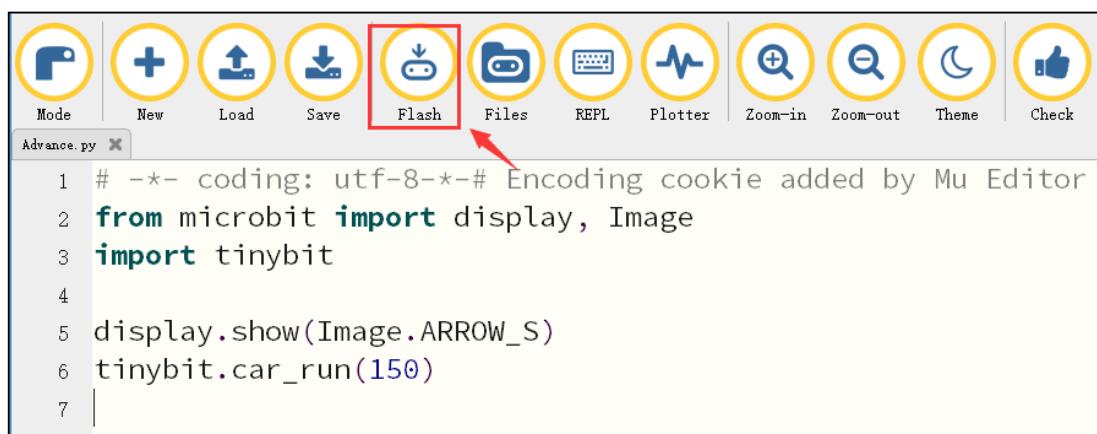
Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out
untitled ×
1 # Write your code here :-)
2

BBC micro:bit REPL

MicroPython for Tinybit V1.1 Modified by Yahboom Team
Type "help()" for more information.
>>>
>>> |

```

4.Click the “Flash” button to download the program to micro:bit board.



If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Superbit library hex file we provided to the micro:bit board.

For the specific method of adding library files, please refer to 【1.Preparation before class】---【Python programming】

### Experimental phenomena

After the program is successfully downloaded, the micro:bit dot matrix will display the smile pattern and play “ODE”. The car will advance-->back-->spin left-->spin right.

At the same time, RGB lights will change different colors. At the same time, the shovel will flat ---> unload ---> flat ---> lift.

Keep this state into the loop.

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