

Control Motor

Learning goals

In this lesson, we mainly learn how to control motor on the Super:bit expansion board and make Omni:bit take the lightning shape route.

Code

```
1 from microbit import *
2 import superbit
3
4
5 display.show(Image.HAPPY)
6
7 def lightning():
8     superbit.motor_control_dual(superbit.M1, superbit.M4, 255, 255, 0)
9     sleep(1000)
10
11    superbit.motor_control_dual(superbit.M1, superbit.M4, 0, 0, 0)
12    superbit.motor_control_dual(superbit.M2, superbit.M3, 0, 0, 0)
13    superbit.motor_control_dual(superbit.M2, superbit.M3, 255, 255, 0)
14    sleep(1000)
15
16    superbit.motor_control_dual(superbit.M1, superbit.M4, 0, 0, 0)
17    superbit.motor_control_dual(superbit.M2, superbit.M3, 0, 0, 0)
18    superbit.motor_control_dual(superbit.M1, superbit.M4, 255, 255, 0)
19    sleep(1000)
20
21    superbit.motor_control_dual(superbit.M1, superbit.M4, 0, 0, 0)
22    superbit.motor_control_dual(superbit.M2, superbit.M3, 0, 0, 0)
23
24 lightning()
```

Import superbit library, import microbit library.

`display.show (Image.HAPPY)`: Microbit dot matrix display smiley face;

`superbit.motor_control_dual (superbit.M1, superbit.M3, 255, 255, 0)`: Controls the motors of M1 and M3 ports, all of which rotate at the speed of PWM 255.

`microbit.sleep (1000)`: Delay 1000 milliseconds.

`lightning ()`: Create a function that adjusts the steering of the motor to make the car move like lightning.

Assembly steps

Please refer to the **1.Omnibit installation steps** in the **1.Assembly steps** folder for building blocks assembly steps.

About wiring

The left front motor is connected to the M1 interface of the Super:bit expansion board. The black line is on the battery side;

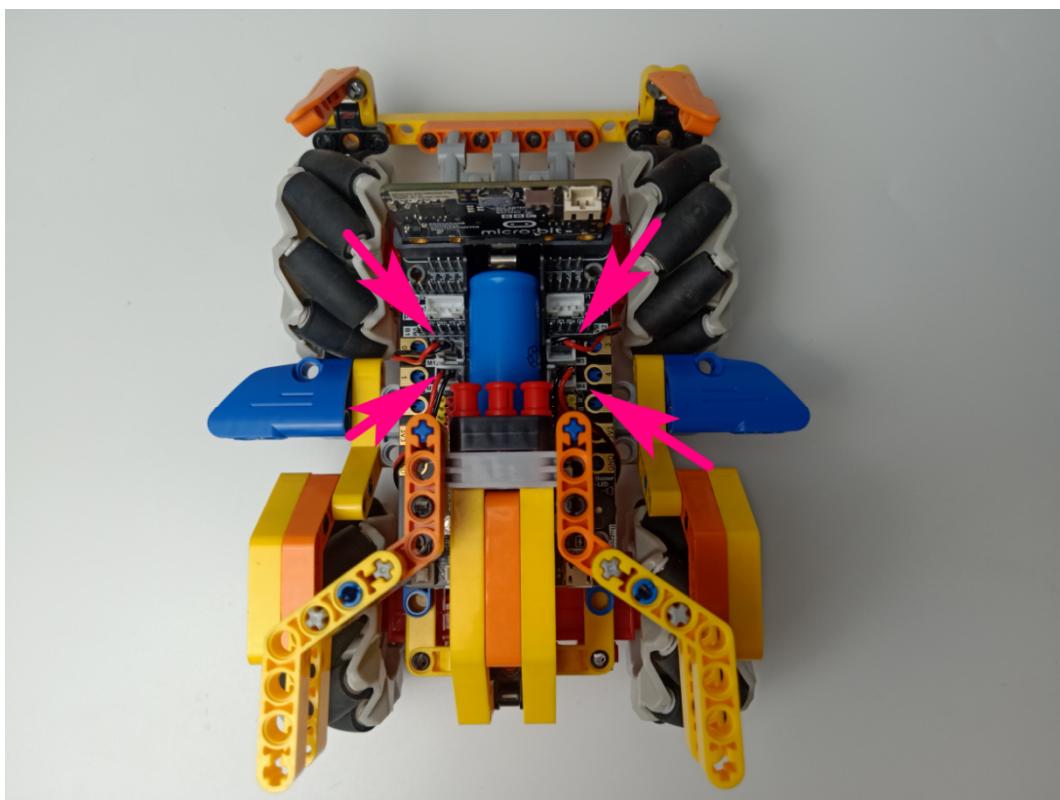
The left rear motor is connected to the M2 interface of the Super:bit expansion board, The black line is on the battery side;

The right front motor is connected to the M3 interface of the Super:bit expansion board, The black line is on the battery side;

The right rear motor is connected to the M4 interface of the Super:bit

expansion board, The black line is on the battery side.

As shown below.



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.

```

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]

```

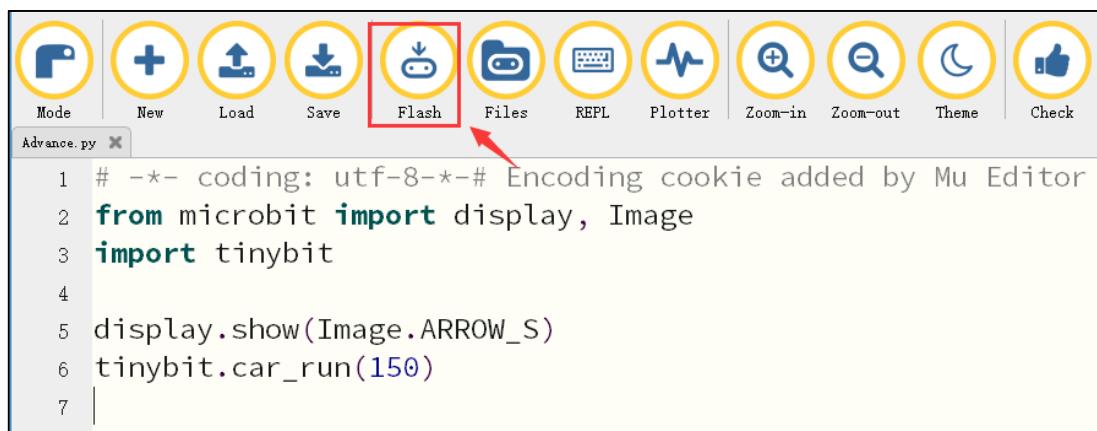
# Write your code here :-)

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 show a smile and take a lightning line, and keep the loop in such a state.

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