

Marquee

Learning goals

In this lesson, we mainly learn how to realize marquee by micro:bit and Super:bit expansion board.

Code

```
1 from microbit import *
2 import superbit
3 import microbit
4 display.show(Image.HAPPY)
5
6 while True:
7     superbit.motor_control(superbit.M1, 0, 0)
8     superbit.motor_control(superbit.M3, 0, 0)
9     microbit.sleep(1000)
10    superbit.motor_control(superbit.M1, 125, 0)
11    superbit.motor_control(superbit.M3, 125, 0)
12    microbit.sleep(1000)
13    superbit.motor_control(superbit.M1, 255, 0)
14    superbit.motor_control(superbit.M3, 255, 0)
15    microbit.sleep(1000)
16    superbit.motor_control(superbit.M1, 125, 0)
17    superbit.motor_control(superbit.M3, 125, 0)
18    microbit.sleep(1000)
19
```

Assembly steps

Please refer to the [assembly steps](#) in the [Assembly instructions](#) folder for building blocks assembly steps.

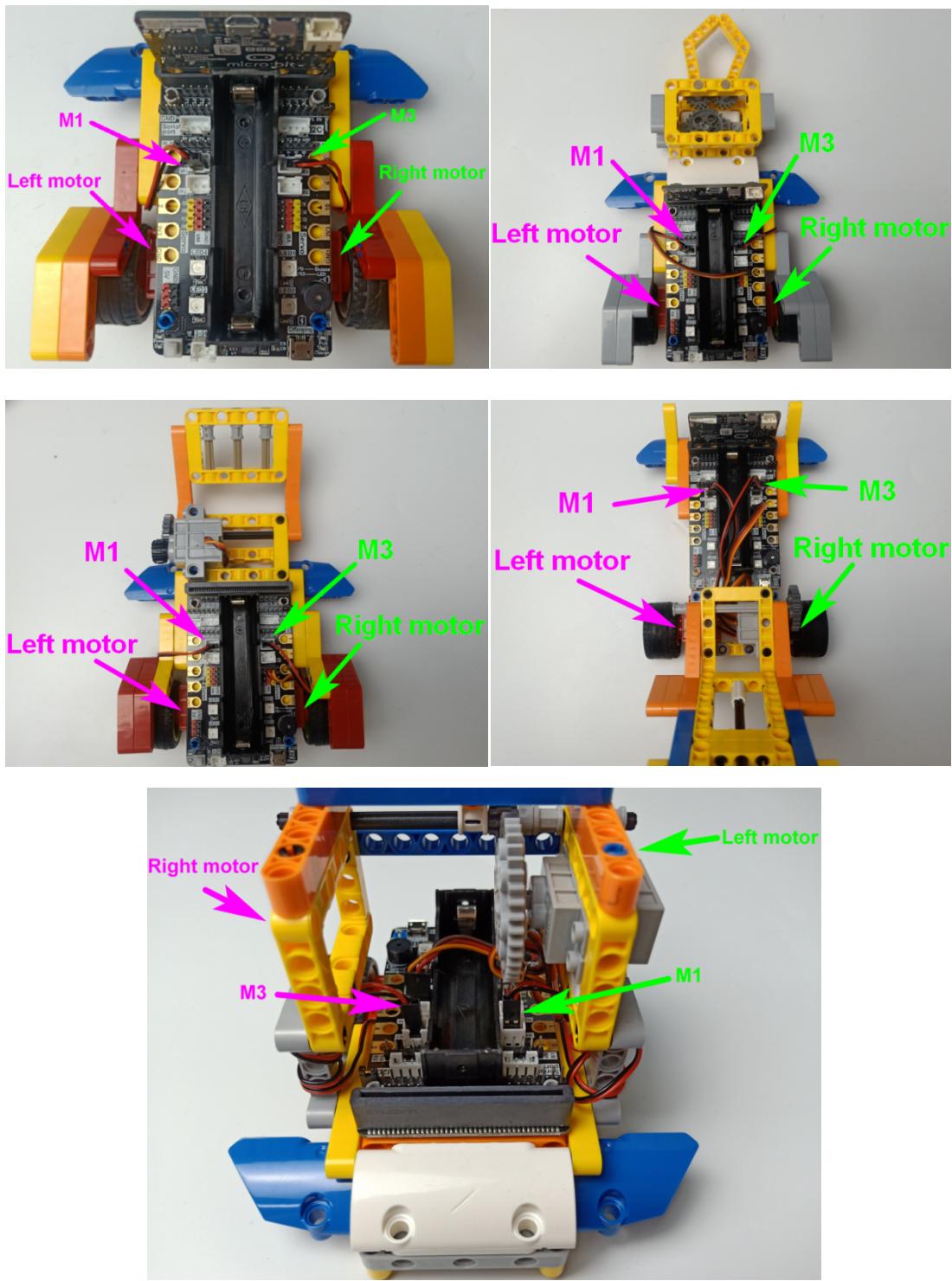
About wiring

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.



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.

The screenshot shows the Mu 1.0.3 Python editor interface. The toolbar at the top has icons for Mode, New, Load, Save, Flash, Files, REPL, Plotter, Zoom-in, Zoom-out, Theme, and Check. Below the toolbar is a tab labeled "Spider advance.py". The code in the editor is:

```

1 from microbit import *
2 import superbit
3
4 display.show(Image.HEART)
5
6
7 while True:
8     superbit.motor_control(superbit.M1, -255, 0)
9     superbit.motor_control(superbit.M3, -255, 0)
10
11

```

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.

The screenshot shows the Mu 1.0.3 Python editor interface. The toolbar at the top has icons for Mode, New, Load, Save, Flash, Files, REPL, Plotter, Zoom-in, Zoom-out, Theme, and Check. The "Check" button is highlighted with a red box. Below the toolbar is a tab labeled "Spider advance.py". The code in the editor is the same as in the previous screenshot. At the bottom of the screen, there is a message box with the text "Well done! No problems here." in green.

3. Click the 'REPL' button to check whether the super:bit library has been downloaded. If not, please refer to the [1.preparation before class] ---> [2.How to import Yahboom superbit library] import super:bit library tutorial.

Mu 1.0.3 - Spider advance.py

```

1 from microbit import *
2 import superbbit
3
4 display.show(Image.HEART)
5
6
7 while True:
8     superbbit.motor_control(superbit.M1, -255, 0)
9     superbbit.motor_control(superbit.M3, -255, 0)
10

```

BBC micro:bit REPL

```

MicroPython for Super:bit V1.1 modified by Yahboom Team
Type "help()" for more information.
>>>

```

4. After writing the code, please click the 'Flash' button to download the program to the micro:bit board.

Mu 1.0.3 - Spider advance.py

```

1 from microbit import *
2 import superbbit
3
4 display.show(Image.HEART)
5
6
7 while True:
8     superbbit.motor_control(superbit.M1, -255, 0)
9     superbbit.motor_control(superbit.M3, -255, 0)
10
11

```

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

For the specific method of adding library files, please refer to [【Preparation before class】](#) --- [【How to import Yahboom superbbit library】](#)

Experimental phenomena

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

We can see that the 4 RGB lights on the super:bit expansion board are lit up in turn with different color, which looks like Marquee.

The color of the RGB lights are changed every 1 seconds.