

Small windmill

Learning goals

In this lesson, we mainly learn how to control small windmill by micro:bit python programming.

Code:

```
from microbit import *
import superbit

a = 135

display.show(Image.HEART)
superbit.servo270(superbit.S1, 135)
superbit.motor_control(superbit.M1, 255, 0)

while True:
    if button_a.is_pressed():
        a = a - 1
        if a < 0 :
            a = 0
        superbit.servo270(superbit.S1, a)
    elif button_b.is_pressed():
        a = a + 1
        if a > 270 :
            a = 270
        superbit.servo270(superbit.S1, a)
```

`from microbit import` means that we will use the functions in the micro:bit library.
`from microbit import *` is to import everything from the micro:bit library. Every program that uses micro:bit must import this library. We also need to import super:bit library;

`display.show (Image.HAPPY)`: Show heart pattern;

`superbit.servo270 (superbit.S1, 135)`: Rotate the 270 ° servo on S1 port to 135 °

`superbit.motor_control (superbit.M1, 255, 0)`: Control the speed of the motor on the M1 port with a duty cycle of 255

`button_a.is_pressed ()`: Determine whether button A on the micro:bit board is pressed

`button_b.is_pressed ()`: Determine whether button B on the micro:bit board is pressed

The overall idea is to judge whether the A and B buttons are pressed in the cycle. Pressing the A button causes the servo angle to rotate in the direction of decreasing angle, and pressing the B button causes the servo angle to rotate in the direction of increasing angle.

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 editor interface with the following details:

- Toolbar:** Includes icons for Mode, New, Load, Save, Flash, Files, REPL, Plotter, Zoom-in, Zoom-out, Theme, and Check.
- Code Editor:** A file named "microbit-servo270servo_one.py" is open. The code is as follows:

```

1 from microbit import *
2 import microbit
3 import superbit
4
5 display.show(Image.HEART)
6 superbit.servo270(superbit.S1, 0)
7 microbit.sleep(1000)
8
9 while True:
10     superbit.servo270(superbit.S1, 0)
11     microbit.sleep(1000)
12     superbit.servo270(superbit.S1, 90)
13     microbit.sleep(1000)
14     superbit.servo270(superbit.S1, 180)
15     microbit.sleep(1000)
16     superbit.servo270(superbit.S1, 270)
17     microbit.sleep(1000)
18     superbit.servo270(superbit.S1, 180)
19     microbit.sleep(1000)
20     superbit.servo270(superbit.S1, 90)
21     microbit.sleep(1000)

```

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.

microbit-superbit_270servo_one.py

```

1 from microbit import *
2 import microbit
3 import superbit
4
5 display.show(Image.HEART)
6 superbit.servo270(superbit.S1, 0)
7 microbit.sleep(1000)
8
9 while True:
10     superbit.servo270(superbit.S1, 0)
11     microbit.sleep(1000)
12     superbit.servo270(superbit.S1, 90)
13     microbit.sleep(1000)

```

Hurrah! Checker turned up no problems.

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

microbit-superbit_270servo_one.py

```

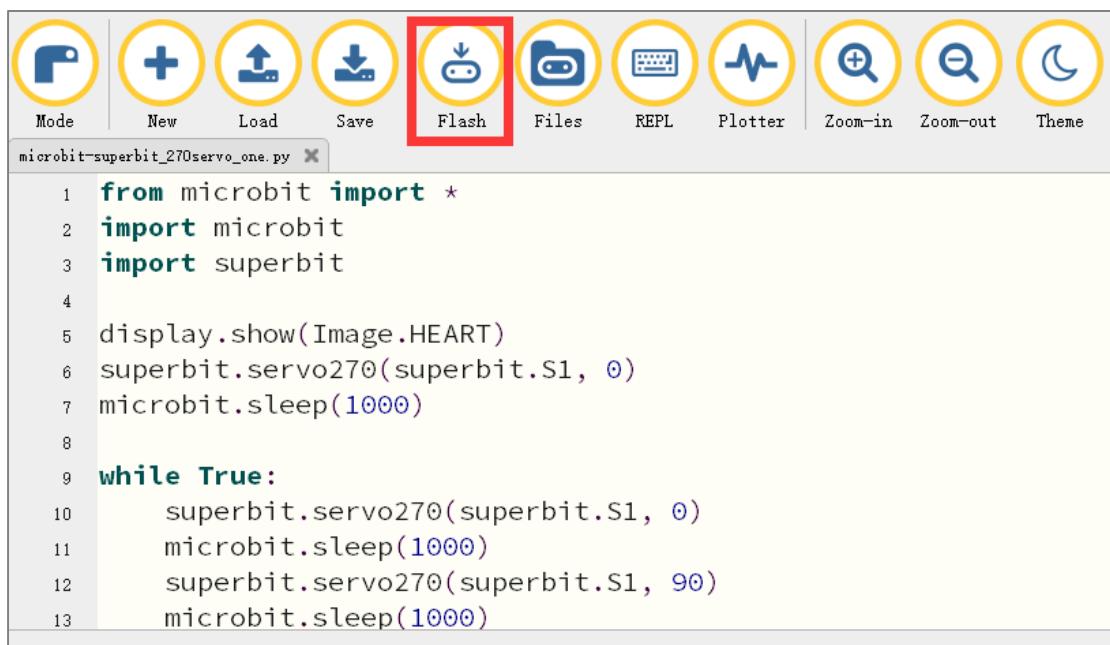
1 from microbit import *
2 import microbit
3 import superbit
4
5 display.show(Image.HEART)
6 superbit.servo270(superbit.S1, 0)
7 microbit.sleep(1000)

```

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.



```

1 from microbit import *
2 import microbit
3 import superbit
4
5 display.show(Image.HEART)
6 superbit.servo270(superbit.S1, 0)
7 microbit.sleep(1000)
8
9 while True:
10     superbit.servo270(superbit.S1, 0)
11     microbit.sleep(1000)
12     superbit.servo270(superbit.S1, 90)
13     microbit.sleep(1000)

```

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】---【How to import Yahboom superbit library】

Hardware connection

The 270° block servo connect to the S1 interface of the Super:bit expansion board. The orange wire of the 270° block servo is connected to the yellow pin of S1, the red wire of the 270° block servo is connected to the red pin of S1, and the brown wire of the 270° block servo is connected to the black pin of S1.

Experimental phenomena

After the download is completed, a heart is displayed on the micro:bit dot matrix, the servo rotates to 135 °, and the windmill connected to M1 port will rotate. Press A and B keys to switch the direction of fan rotation.