

Music Metronome

Learning goals

In this lesson, we will learn use Python programming to control 270° block servo.

Code:

```
1 from microbit import *
2 import music
3 import superbit
4
5
6 superbit.servo270(superbit.S1, 0)
7 music.play('g4:2')
8 superbit.servo270(superbit.S1, 90)
9 music.play('e4:2')
10 superbit.servo270(superbit.S1, 0)
11 music.play('g4:2')
12 superbit.servo270(superbit.S1, 90)
13 music.play('e4:2')
14 superbit.servo270(superbit.S1, 0)
15 music.play('g4:2')
16 superbit.servo270(superbit.S1, 90)
17 music.play('e4:2')
18 superbit.servo270(superbit.S1, 0)
19 music.play('c4:4')
20 superbit.servo270(superbit.S1, 90)
21 music.play('d4:2')
22 superbit.servo270(superbit.S1, 0)
23 music.play('f4:2')
24
25 superbit.servo270(superbit.S1, 90)
26 music.play('e4:2')
27 superbit.servo270(superbit.S1, 0)
28 music.play('d4:2')
29 superbit.servo270(superbit.S1, 90)
30 music.play('g4:4')
31 superbit.servo270(superbit.S1, 0)
32 music.play('e4:4')
33 superbit.servo270(superbit.S1, 90)
34 music.play('g4:2')
35 superbit.servo270(superbit.S1, 0)
36 music.play('e4:2')
37 superbit.servo270(superbit.S1, 90)
38 music.play('g4:2')
39 superbit.servo270(superbit.S1, 0)
40 music.play('e4:2')
41 superbit.servo270(superbit.S1, 90)
```

```

42 music.play('g4:2')
43
44 superbit.servo270(superbit.S1, 0)
45 music.play('e4:2')
46 superbit.servo270(superbit.S1, 90)
47 music.play('c4:4')
48 superbit.servo270(superbit.S1, 0)
49 music.play('d4:2')
50 superbit.servo270(superbit.S1, 90)
51 music.play('f4:2')
52 superbit.servo270(superbit.S1, 0)
53 music.play('e4:2')
54 superbit.servo270(superbit.S1, 90)
55 music.play('d4:2')
56 superbit.servo270(superbit.S1, 0)
57 music.play('c4:4')
58 superbit.servo270(superbit.S1, 90)
59 music.play('e4:4')
60 superbit.servo270(superbit.S1, 0)
61 music.play('d4:2')
62 |
63
64 superbit.servo270(superbit.S1, 90)
65 music.play('d4:2')
66 superbit.servo270(superbit.S1, 0)
67 music.play('f4:2')
68 superbit.servo270(superbit.S1, 90)
69 music.play('f4:2')
70 superbit.servo270(superbit.S1, 0)
71 music.play('e4:2')
72 superbit.servo270(superbit.S1, 90)
73 music.play('c4:2')
74 superbit.servo270(superbit.S1, 0)
75 music.play('g4:4')
76 superbit.servo270(superbit.S1, 90)
77 music.play('d4:2')
78 superbit.servo270(superbit.S1, 0)
79 music.play('f4:2')
80 superbit.servo270(superbit.S1, 90)
81 music.play('e4:2')

```

Import music library, And make a note list to create a melody. For example, G4: 2 in the list means that the G4 tone is played at 2 as the beat, C4: 4 means that C4 tone is played at the 4 as the beat...

In this lesson, we use <painter> song, we can search for the notation of this song from the internet

The numbers appearing in the score and the tones in our program have been mapped here one by one, as shown below..

1	2	3	4	5
C4	D4	E4	F4	G4

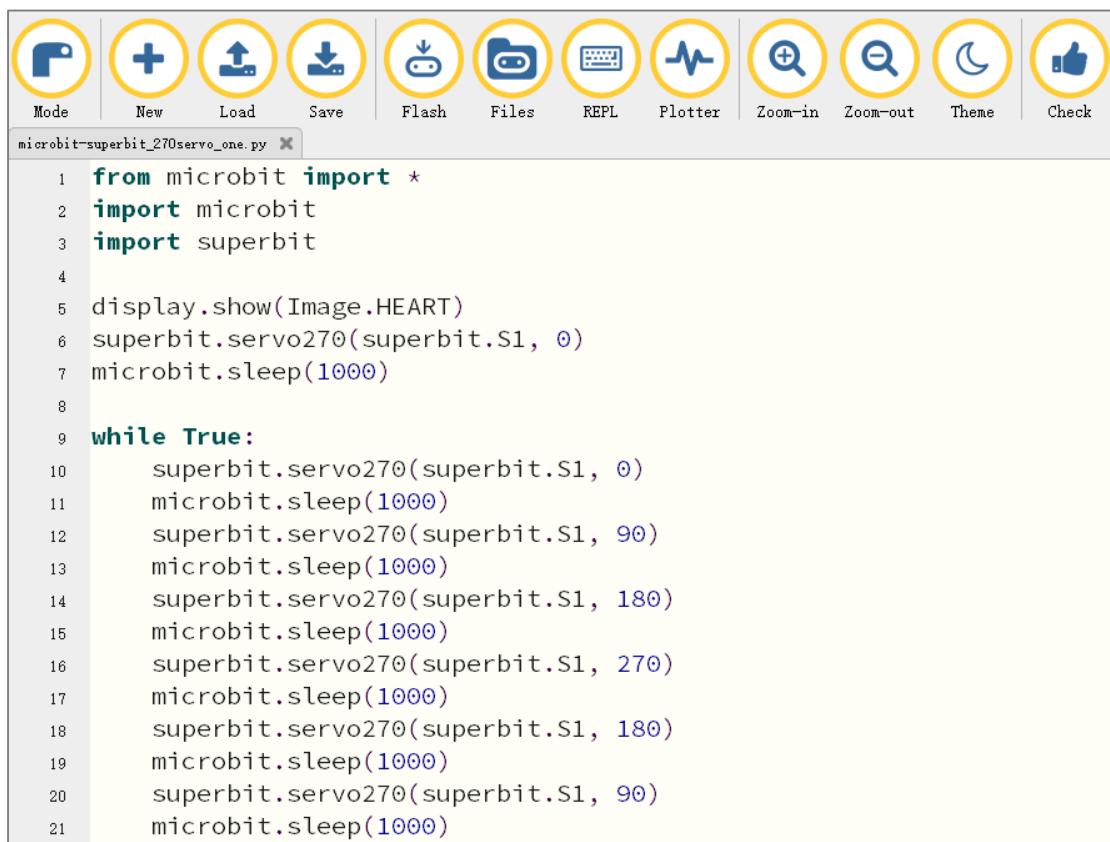
About beat. If the number is underlined, we set its beat to 2. If there is a horizontal bar after the number, we set its beat to 8. If it is a single number, the beat is 4.

If there is no pause in the song, we can also put an E1 tone in the middle, and set its beat to 4 as a pause.

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.



```

from microbit import *
import microbit
import superbit

display.show(Image.HEART)
superbit.servo270(superbit.S1, 0)
microbit.sleep(1000)

while True:
    superbit.servo270(superbit.S1, 0)
    microbit.sleep(1000)
    superbit.servo270(superbit.S1, 90)
    microbit.sleep(1000)
    superbit.servo270(superbit.S1, 180)
    microbit.sleep(1000)
    superbit.servo270(superbit.S1, 270)
    microbit.sleep(1000)
    superbit.servo270(superbit.S1, 180)
    microbit.sleep(1000)
    superbit.servo270(superbit.S1, 90)
    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

```

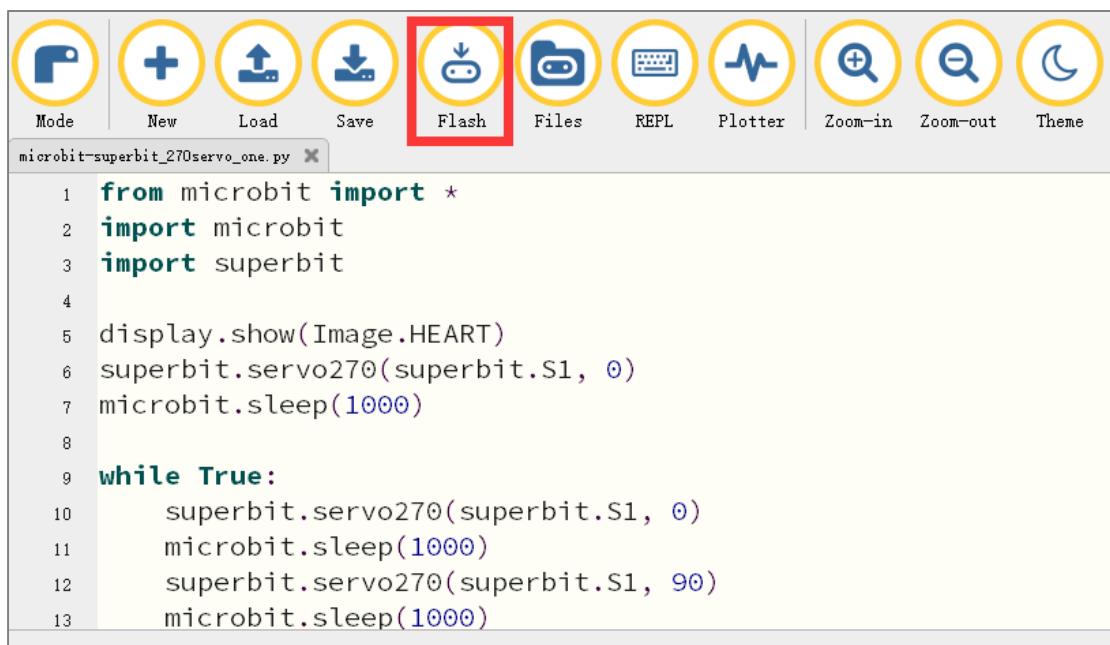
1 from microbit import *
2 import microbit
3 import superbit
4
5 display.show(Image.HEART)
6 superbit.servo270(superbit.S1, 0)
7 microbit.sleep(1000)

```

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 program is successfully downloaded, we can hear that song "Paint Painter", and the building block motor rotates according to the sound adjustment, the rotation angle is 90 °.

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