Vertical pointer

Learning goals

In this lesson, we will learn to program in Python using the super:bit expansion board to make a vertical pointer.

Code

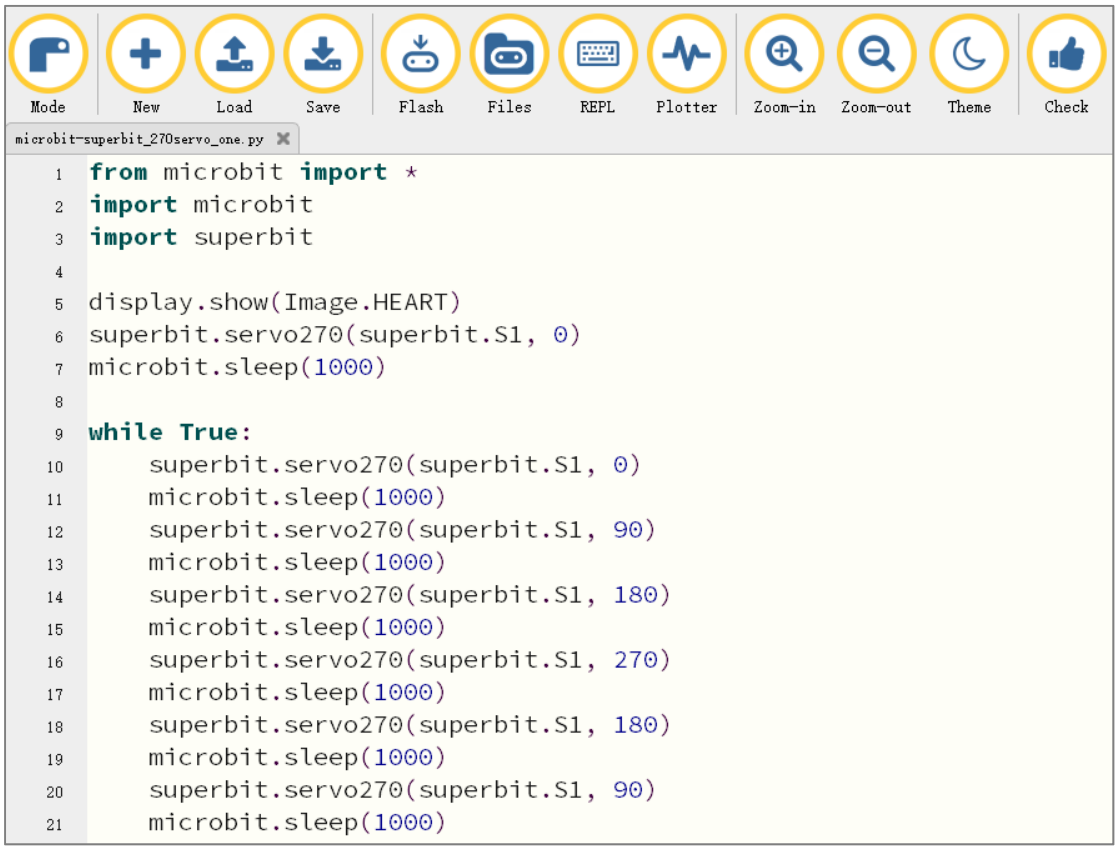
```
from microbit import *
import microbit
import superbit

display.show(Image.HEART)

while True:
    x = accelerometer.get_x()
    x1 = ((x+1024)*(225-45)/2048)+45
    x1 = int(270 - x1)
    superbit.servo270(superbit.S1, x1)
    microbit.sleep(20)

```

from microbit import * is to import everything from the microbit library. Every program that uses micro:bit must import this library. We also need to import superbit library;
display.show (Image.HAPPY): show love;
accelerometer.get_x (): measures the acceleration of the x axis and generates a corresponding positive or negative integer according to the change in direction. The returned value ranges from -1024 to 1024.
x1 = ((x + 1024) * (225-45) / 2048) +45: convert the range of values returned by the accelerometer obtained by the previous x from -1024 to 1024 to the corresponding range between 45 and 225 The value is then assigned to the variable x1.
x1 = int (270-x1): Since the previously calculated x1 may be calculated as a floating-point variable, x1 is coerced to an integer.
superbit.servo270 (superbit.S1, x1): Rotate the 270 ° servo on S1 port to the degree of x1

Programming and downloading：

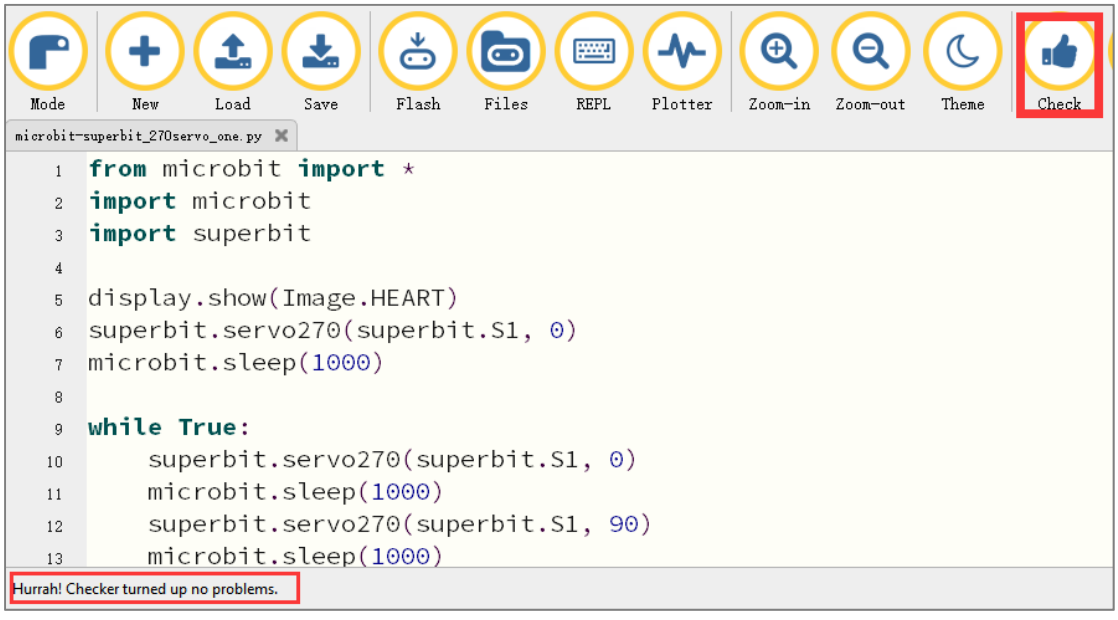
1.You should open the Mu software, and enter the code in the edit window, , as shown below.

Note! All English and symbols should be entered in English, and the last line must be a space.



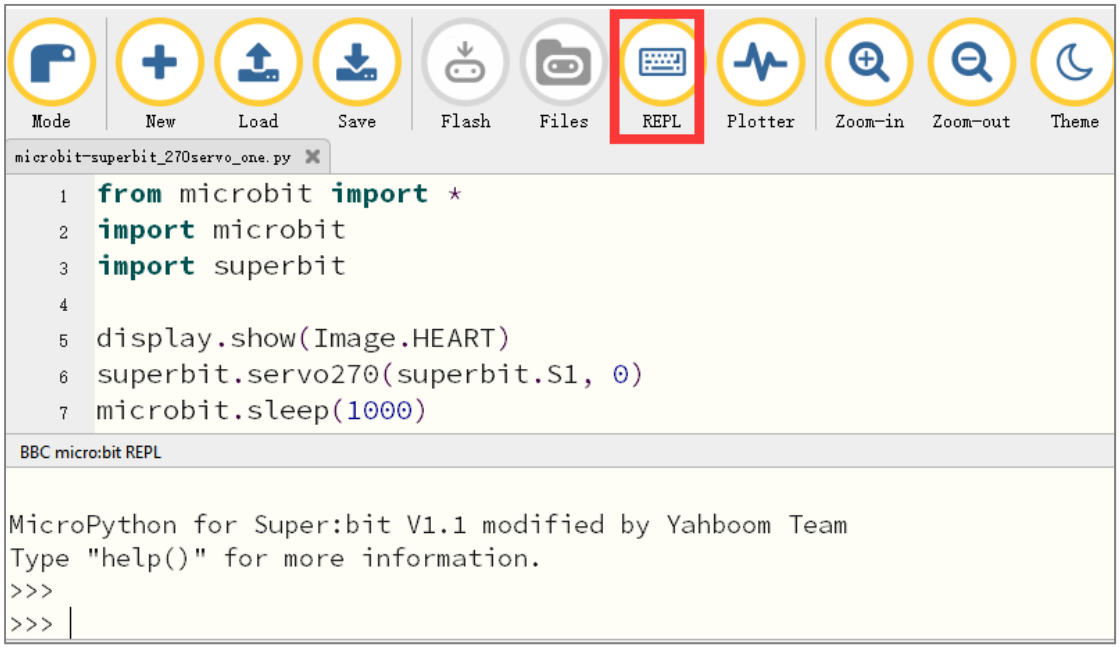

```
from microbit import *
import microbit
import superbit

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.

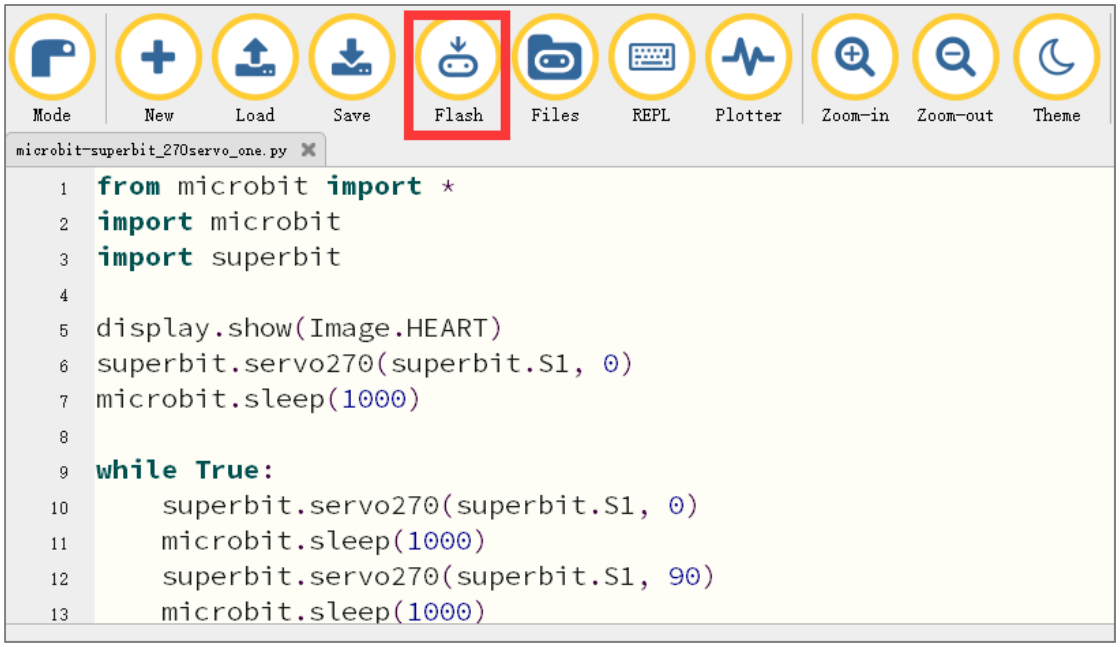



```
from microbit import *
import microbit
import superbit

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)
```

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.



4.After writing the code, please click the 'Flash' button to download the program to the 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】---【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, the micro:bit dot matrix will display the music pattern. The 270° block servo turns according to the data of the micro:bit acceleration sensor, keeping the pointer vertically upwards.

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