Cannonball shooting

1.Learning goals

In this course, we mainly learn how to realize cannonball shooting function by through Python programming.

When the A button on the micro:bit board is pressed, the "cannonball" will be shooted; when the B button on the micro:bit board is pressed, the throwing rod will reset.

2.Building block assembly steps

For the building block construction steps, please refer to the installation manual or building block installation picture of [Assembly course]-[Mobile shooter].

3.Wiring of motor and servo

The motor wiring on the left side of the car is inserted into the M1 interface of the Super:bit expansion board, and the black wire is close to the battery side;

The motor wiring on the right side of the car is inserted into the M3 interface of the Super:bit expansion board, and the black wire is close to the battery side;

Building block servo insert into the Super: bit expansion board S1 interface, and the orange wiring connect the yellow pin of S1.

As shown below:
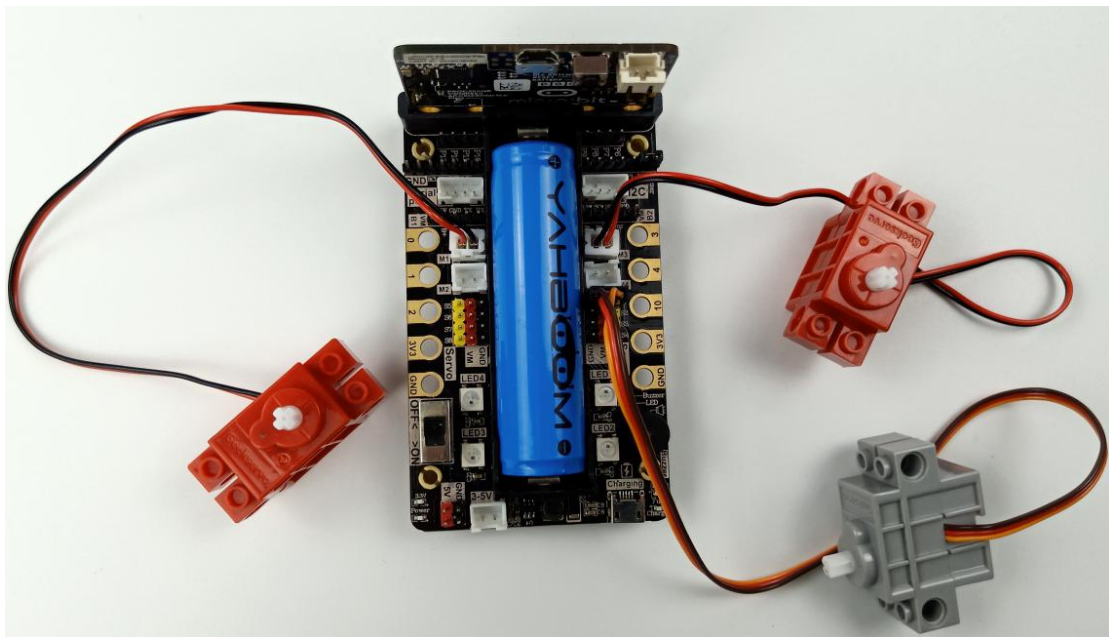
Note:

For the first course related to building block servo, we need to remove the gear on the servo and upload the program of this course to micro: bit. Then, turn on the power switch of the Super:bit expansion board and wait for the building block servo turn to the initial position. Next, we can turn off the power, and adjust the throwing rod of the car to keep it parallel to the ground. Finally, install the servo. (If you have used programs related to mobile shooter before, you can skip this step)

4.Code and analysis

The program for this course, please view .py file.

```python
from microbit import *
import superbit

display.show(Image.HAPPY)
superbit.servo270(superbit.S1, 105)

while True:
    if button_a.is_pressed():
        superbit.servo270(superbit.S1, 135)
    elif button_b.is_pressed():
        superbit.servo270(superbit.S1, 105)
```

First, we need to import the library needed for this lesson from micro:bit, superbit library is dedicated to super:bit expansion board;

display.show(Image.HAPPY)： Display smile pattern on micro:bit matrix.
superbit.servo270(superbit.S1, 105)： Initialize the servo to 105°;
while True: In an infinite loop
    if button_a.is_pressed():
        superbit.servo270(superbit.S1, 135)
    elif button_b.is_pressed():
        superbit.servo270(superbit.S1, 105)

In an infinite loop, determine whether the A and B buttons on the micro:bit board are pressed.
If the A button is pressed, the servo rotates to 135° ("cannonball" will be shooted); if the B button is pressed, the steering gear rotates to 105°(throwing rod will reset).
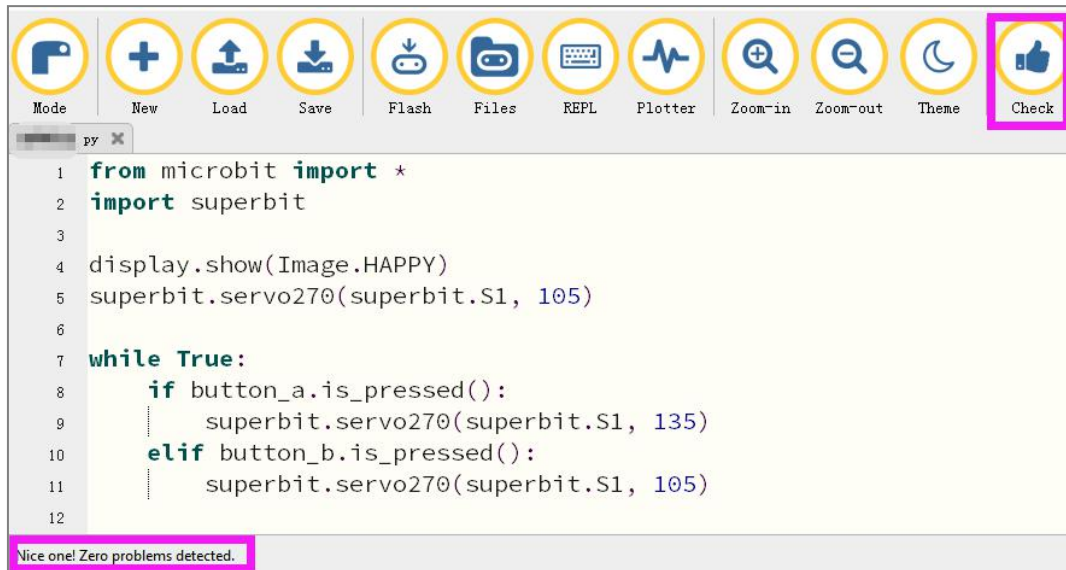
5.Writing and download code

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, use the Tab key (tab key) to indent and the last line must be a space.
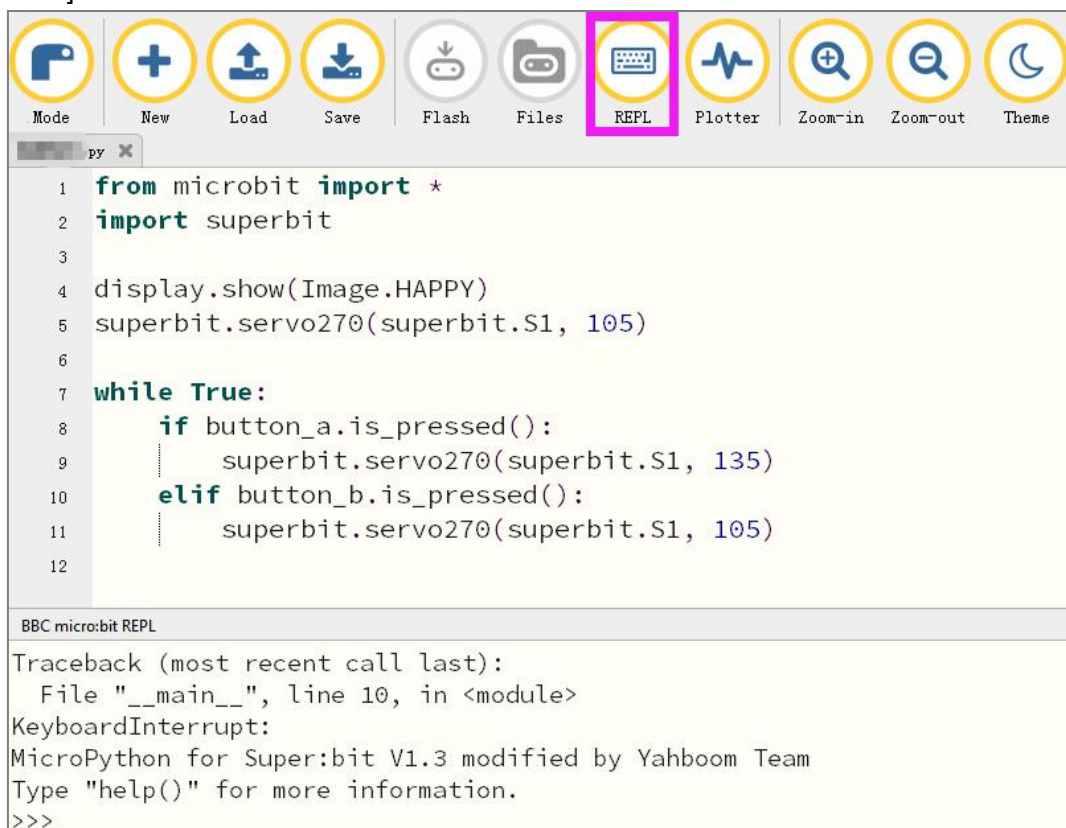2.You can click the "Check" button to check if our code has an error.
If a cursor or underline appears on a line, it indicates a syntax error, please check and modify. If there is no error in the program, the bottom left of the interface will prompt that there is no problem in detection.

```
1  from microbit import *
2  import superbit
3
4  display.show(Image.HAPPY)
5  superbit.servo270(superbit.S1, 105)
6
7  while True:
8      if button_a.is_pressed():
9          superbit.servo270(superbit.S1, 135)
10     elif button_b.is_pressed():
11         superbit.servo270(superbit.S1, 105)
12
```
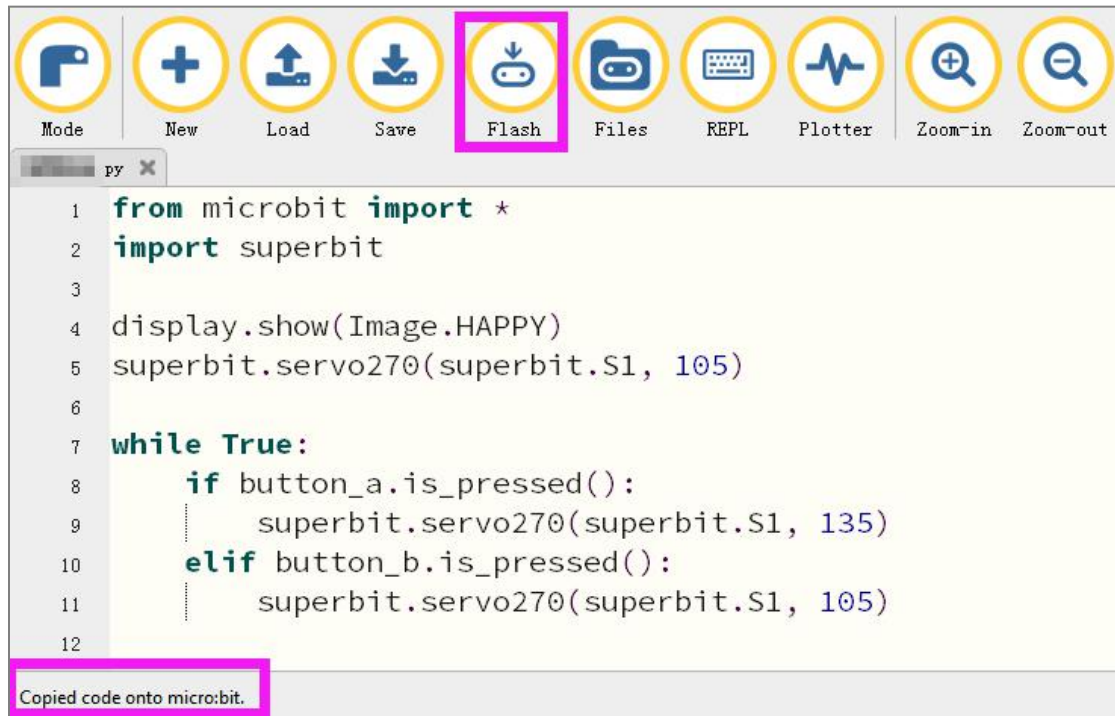
Nice one! Zero problems detected.

3. Click the 'REPL' button to check whether the Superbit library has been downloaded.

If not, please refer to [Preparation before class] --> [2.4 Python Programming Guide] .


```
1  from microbit import *
2  import superbit
3
4  display.show(Image.HAPPY)
5  superbit.servo270(superbit.S1, 105)
6
7  while True:
8      if button_a.is_pressed():
9          superbit.servo270(superbit.S1, 135)
10     elif button_b.is_pressed():
11         superbit.servo270(superbit.S1, 105)
12
```

BBC micro:bit REPL

```
Traceback (most recent call last):
  File "__main__", line 10, in <module>
KeyboardInterrupt:
MicroPython for Super:bit V1.3 modified by Yahboom Team
Type "help()" for more information.
>>>
```

4. After the program is written, use a micro USB cable to connect the computer and the micro:bit board. Please click the 'Flash' button to download the program to the micro:bit motherboard (You need to click the 'REPL' button again to close the function of importing library files before you download the program).

```
1  from microbit import *
2  import superbit
3
4  display.show(Image.HAPPY)
5  superbit.servo270(superbit.S1, 105)
6
7  while True:
8      if button_a.is_pressed():
9          superbit.servo270(superbit.S1, 135)
10     elif button_b.is_pressed():
11         superbit.servo270(superbit.S1, 105)
12
```

Copied code onto micro:bit.

5.If the download failed, please confirm whether the micro:bit is connected to the computer through the micro USB data cable, and confirm whether the Super:bit Python library has been imported.

6.Experimental phenomena
After the program is successfully downloaded, the micro: bit dot matrix will display the smile pattern.
Open the power switch, the servo will initialize to 105 ° (throwing rod reset). When we press the micro:bit A button, the throwing rod will throw a "cannonball"; when we press the B button, the throwing rod will reset.

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