

Basic course12 -- IR avoid sensor

Learning goals:

This lesson we will learn how to use infrared avoid sensor of building:bit by Python programming.

Code :

```
# -*- coding: utf-8-*# Encoding cookie added by Mu Editor
from microbit import display
import buildingbit
import music

display.off()
avoid = False

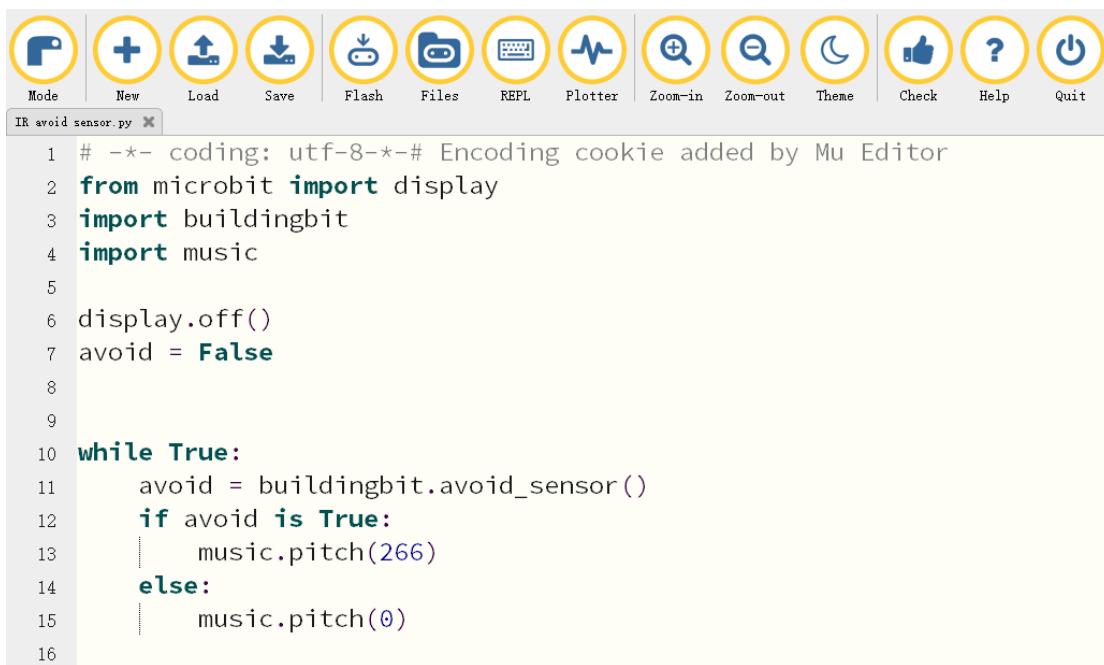
while True:
    avoid = buildingbit.avoid_sensor()
    if avoid is True:
        music.pitch(266)
    else:
        music.pitch(0)
```

- 1) First, we need to import Yahboom buildingbit library: **import buildingbit** and others library we need to use.
- 2) **display.off()** make micro:bit dot matrix off. Because the pins used by the infrared sensor are multiplexed with the pins used by the LED dot matrix screen, the LED dot matrix screen needs to be turned off before using the infrared sensor, otherwise an error will occur.
- 3) The avoid variable is used to save the value returned by the sensor. If an obstacle is detected, it returns True; if no obstacle is detected, it returns False. If an obstacle is detected, the buzzer plays sound; if no obstacle is detected, no sound is played.

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, Tab key for indentation, and the last line must be a space.

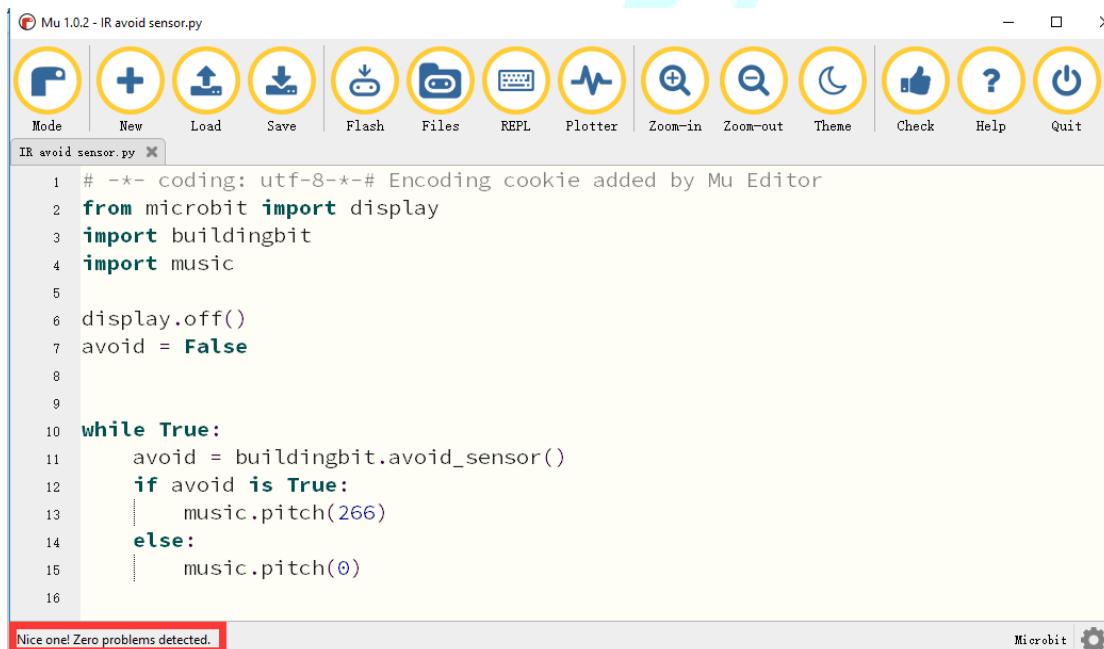


```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display
3 import buildingbit
4 import music
5
6 display.off()
7 avoid = False
8
9
10 while True:
11     avoid = buildingbit.avoid_sensor()
12     if avoid is True:
13         music.pitch(266)
14     else:
15         music.pitch(0)
16

```

2. You need to 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. If there is no cursor or underline, it means that the code is correct, and the bottom left will prompt that the check is OK.



```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display
3 import buildingbit
4 import music
5
6 display.off()
7 avoid = False
8
9
10 while True:
11     avoid = buildingbit.avoid_sensor()
12     if avoid is True:
13         music.pitch(266)
14     else:
15         music.pitch(0)
16

```

Nice one! Zero problems detected.

3. You need to connect the micro data cable to micro:bit and the computer and **download buildingbit library into micro:bit**. Then, click “REPL” button to import “Yahboom buildingbit library”. As shown below.

Mu 1.1.0.alpha.2 - 红外避障传感器.py

```

1  # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2  from microbit import display
3  import buildingbit
4  import music
5
6  display.off()
7  avoid = False
8
BBC micro:bit REPL
File "__main__", line 11, in <module>
KeyboardInterrupt:
MicroPython for Building:bit V1.0 modified by Yahboom Team
Type "help()" for more information.
>>>
MicroPython for Building:bit V1.0 modified by Yahboom Team
Type "help()" for more information.
>>> |

```

4. Click “Flash” to download program to micro:bit board.

```

1  # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2  from microbit import display
3  import buildingbit
4  import music
5
6  display.off()
7  avoid = False
8
9
10 while True:
11     avoid = buildingbit.avoid_sensor()
12     if avoid is True:
13         music.pitch(266)
14     else:
15         music.pitch(0)
16

```

Experimental phenomena

After download is complete. If an obstacle is detected, it returns True; if no obstacle is detected, it returns False. If an obstacle is detected, the buzzer plays sound; if no obstacle is detected, no sound is played.