

Python Basic course14 --- Tracking sensor

Learning goals:

This lesson we will learn how to use tracking sensor by Python programming.

Code:

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

display.show(Image.NO)

while True:
    trakeL = buildingbit.traking_sensor_L()
    trakeR = buildingbit.traking_sensor_R()
    if trakeL and trakeR:
        display.show(Image.HAPPY)
    elif trakeL is True:
        display.show("L")
    elif trakeR is True:
        display.show("R")
    else:
        display.show(Image.NO)
```

- 1) First, we need to import Yahboom buildingbit library: **import buildingbit** and others library we need to use.
- 2) **display.show(Image.NO)** make micro:bit display a “X”.
- 3) **trakeL = buildingbit.traking_sensor_L()** read left tracking sensor value and save it to trakeL variable. When the left tracking sensor detects a black line, trakeL is True; when no black line is detected, trakeL is False;
- 4) **trakeR = buildingbit.traking_sensor_R()** read right tracking sensor value and save it to trakeR variable. When the right tracking sensor detects a black line, trakeR is True; when no black line is detected, trakeR is False.

When trakeL and trakeR are True at the same time, that is, black lines are detected on both sides, a smile face is displayed; if black lines are detected on the left side, L is displayed; if black lines are detected on the right side, R is displayed; neither side is detected When there is a black line, the character X is displayed.

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.

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

display.show(Image.NO)

while True:
    trakeL = buildingbit.traking_sensor_L()
    trakeR = buildingbit.traking_sensor_R()
    if trakeL and trakeR:
        display.show(Image.HAPPY)
    elif trakeL is True:
        display.show("L")
    elif trakeR is True:
        display.show("R")
    else:
        display.show(Image.NO)
```

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.

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

display.show(Image.NO)

while True:
    trakeL = buildingbit.traking_sensor_L()
    trakeR = buildingbit.traking_sensor_R()
    if trakeL and trakeR:
        display.show(Image.HAPPY)
    elif trakeL is True:
        display.show("L")
    elif trakeR is True:
        display.show("R")
    else:
        display.show(Image.NO)
```

Good job! No problems found.

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
Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit
红外避障传感器.py x
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.

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit
红外避障传感器.py x
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. When left and right sensor detected black line, micro:bit will display a **smile**; if black lines are detected on the left side, micro:bit will display a “**L**”; if black lines are detected on the right side, micro:bit will display a “**R**”; left and right sensor no detected black line, micro:bit will display a **X**;