

Robot course6 --- Ultrasonic avoid

Learning goal:

In this lesson, we will learn how to obtain ultrasonic distance. According to the detected distance data, the direction of Tiny:bit operation is changed to achieve obstacle avoidance.

Code:

```

1  # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2  from microbit import sleep, display, Image, pin12
3  import tinybit
4  import music
5  import neopixel
6
7  np = neopixel.NeoPixel(pin12, 2)
8  np.clear()
9  tinybit.car_HeadRGB(0, 0, 0)
10
11
12 while True:
13     distance = tinybit.ultrasonic()
14     if distance < 20:
15         sleep(10)
16         distance = tinybit.ultrasonic()
17         if distance < 20:
18             tinybit.car_stop()
19             display.show(Image.NO)
20             tinybit.car_HeadRGB(255, 0, 0)
21             tinybit.car_back(100)
22             music.pitch(523)
23             sleep(500)
24             tinybit.car_spinright(100)
25             music.pitch(0)
26             sleep(500)
27     else:
28         tinybit.car_HeadRGB(0, 255, 0)
29         tinybit.car_run(100)
30         music.pitch(0)
31         display.show(Image.ARROW_S)
32

```

- 1) Import the library needed for this lesson from micro:bit, display is used for dot matrix display, Image calls the built-in image, pin12 is the pin of the body colorful lights, neopixel drives the body colorful lights, and tinybit controls the car.

display.show (Image.HAPPY): Display a smile on micro:bit dot matrix.

np = neopixel.NeoPixel (pin12, 2): Initialize the body colorful lights. The first parameter is the pins connected to the lights, and the second parameter is the total number of RGB lights.

np.clear (): Clear the body colorful lights.

tinybit.car_HeadRGB (0, 0, 0): Clear the head RGB searchlight.

2) The distance of the ultrasonic wave is read cyclically and stored in the distance variable.
tinybit.ultrasonic () returns the current distance data of the ultrasonic wave, the unit is centimeter (cm);

3) Judging the distance, if it is less than 20cm, confirm whether it is less than 20cm with a delay of 10 milliseconds to avoid the impact of data fluctuations;

After confirming that the distance is less than 20cm, stop the car and display an 'X' character on the micro:bit dot matrix. The headlight will display a red light, buzzer will sound, then the car back and spin the right to achieve ultrasonic avoidance.

If the distance is greater than 20cm, front RGB searchlight become green, the dot matrix shows an arrow pointing forward, and car will keep going forward.

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.

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Advance.py ×
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4
5 display.show(Image.ARROW_S)
6 tinybit.car_run(150)
7

```

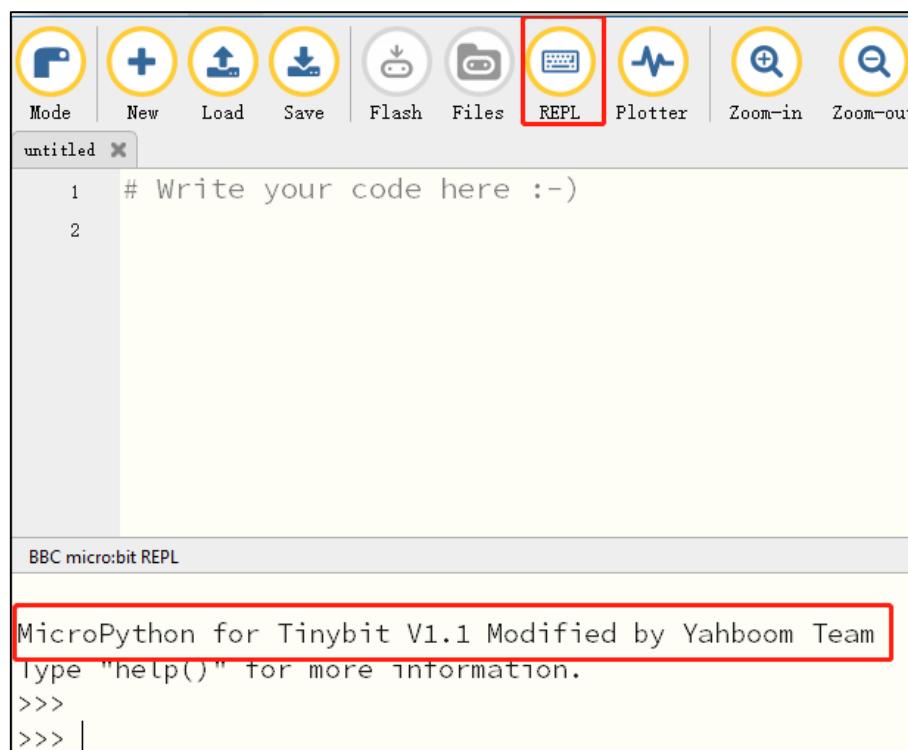
2.You can click the “Check” button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Advance.py ×
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4
5 display.show(Image.ARROW_S)
6 tinybit.car_run(150)
7

```

3.Click “REPL” button,check whether the tinybit library has been downloaded. If not, please refer to the [preparation before class]---> [Python programming]



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



If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Buildingbit libraryhex file we provided to the micro: bit board.

For the specific method of adding library files, please refer to [【1.Preparation before class】](#) --- [【Python programming】](#)

Experimental phenomena

If the distance is greater than 20cm, RGB headlight become green, the dot matrix shows an arrow pointing forward, and car will keep going forward. If it is less than 20cm, stop the car and display an 'X' character on the micro:bit dot matrix. The headlight will display a red light, buzzer will sound, then the car back and spin the right to achieve ultrasonic avoidance.