

## Robot course5 --- Distance control light

### Learning goal:

In this lesson, we will learn how to obtain ultrasonic distance, and use the detected distance data to switch the color of the lights through Python programming.

### Code:

Please use the MU software to open the **Distance control light.py** file we provided.

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.

Code as shown below:

```
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, pin12
3 import tinybit
4 import neopixel
5
6 display.show(Image.HAPPY)
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
```

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 0 and less than or equal to 4, let the front RGB searchlights and body colorful lights be red, and if the distance is greater than 4 and less than or equal to 8, let the front RGB searchlight and body colorful lights be green; analogy. Finally, when it is over 20cm, clear all lights.

```

11 while True:
12     distance = tinybit.ultrasonic()
13     if distance > 0 and distance <= 4:
14         tinybit.car_HeadRGB(255, 0, 0)
15         np[0] = (255, 0, 0)
16         np[1] = (255, 0, 0)
17         np.show()
18
19     if distance > 4 and distance <= 8:
20         tinybit.car_HeadRGB(0, 255, 0)
21         np[0] = (0, 255, 0)
22         np[1] = (0, 255, 0)
23         np.show()

```

...

```

43     if distance > 20:
44         tinybit.car_HeadRGB(0, 0, 0)
45         np.clear()
46

```

### 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.**

The image shows the Mu IDE interface. At the top is a toolbar with various icons: Mode, New, Load, Save, Flash, Files, REPL, Plotter, Zoom-in, Zoom-out, Theme, and Check. Below the toolbar is a tab labeled "Advance.py" with a close button. The main area is a code editor containing the following Python code:

```

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.

```

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]

```

# Write your code here :-(

BBC micro:bit REPL

MicroPython for Tinybit V1.1 Modified by Yahboom Team
Type "help()" for more information.
>>>
>>> |

```

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

```

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

```

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**

After download is complete, open the power switch. Micro:bit will display a smile on dot matrix, color of RGB light will change with distance different.