

Tumble:bit Handle control

Learning goals

In this lesson, we will learn how to control tumble:bit car by micro:bit Handle.

Wireless communication principles

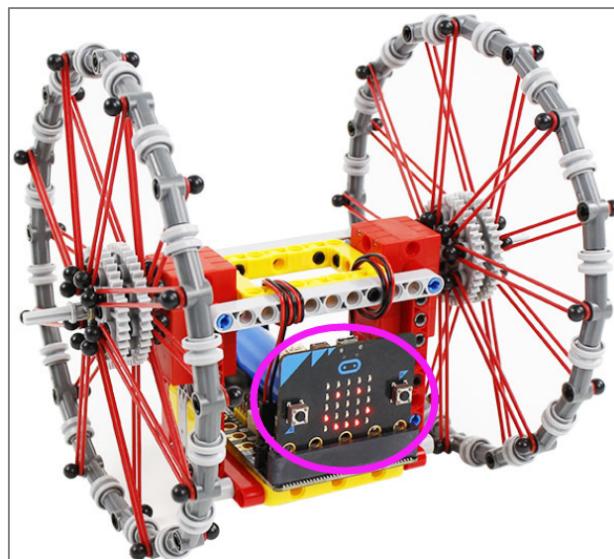
With the micro:bit radio module, different devices can work together through a simple wireless network. When the radio function is turned on for micro:bit, a simple wireless local area network is generated. The micro:bit board with radio function turned on can set parameters within the effective range.

Wireless communication is divided into sending and receiving two program blocks. Set the wireless group of radio to the same group, and the two micro:bit boards can communicate.

Code

Tumble:bit code

This code need to be downloaded into micro:bit board of tumble:bit car.



```

1  from microbit import *
2  import superbit
3  import radio
4  import neopixel
5  Red = (255, 0, 0)
6  Orange = (255, 165, 0)
7  Yellow = (255, 255, 0)
8  Green = (0, 255, 0)
9  Blue = (0, 0, 255)
10 Violet = (148, 0, 211)
11 White = (255, 255, 255)
12 color_lib = {'Red': Red, 'Orange': Orange, 'Yellow': Yellow, 'Green': Green,
13 'Blue': Blue, 'Violet': Violet, 'White': White}
14 def RGBLight_more_show(first, num, color):
15     global np
16
17     np.clear()
18     for i in range(first, first + num):
19         np[i] = color_lib[color]
20     np.show()
21 np = neopixel.NeoPixel(pin12, 4)
22 display.show(Image.HAPPY)
23 radio.on()
24 radio.config(group=1)
25
26 while True:
27     incoming = radio.receive()
28     if incoming == 'up':
29         superbit.motor_control(superbit.M1, 255, 0)
30         superbit.motor_control(superbit.M3, 255, 0)
31     elif incoming == 'down':
32         superbit.motor_control(superbit.M1, -255, 0)
33         superbit.motor_control(superbit.M3, -255, 0)
34     elif incoming == 'left':
35         superbit.motor_control(superbit.M1, -100, 0)
36         superbit.motor_control(superbit.M3, 100, 0)
37     elif incoming == 'right':
38         superbit.motor_control(superbit.M1, 100, 0)
39         superbit.motor_control(superbit.M3, -100, 0)
40     elif incoming == 'stop':
41         superbit.motor_control(superbit.M1, 0, 0)
42         superbit.motor_control(superbit.M3, 0, 0)
43     elif incoming == 'R':
44         RGBLight_more_show(0, 4, 'Red')
45     elif incoming == 'G':
46         RGBLight_more_show(0, 4, 'Green')
47     elif incoming == 'B':
48         RGBLight_more_show(0, 4, 'Blue')
49     elif incoming == 'Y':
50         RGBLight_more_show(0, 4, 'Yellow')
51

```

`from microbit import *` is to import everything from the micro:bit library. Every program using microbit must import this library. We also need to import Superbit library and import the RGB light library neopixel, radio wireless communication function library;

`np = neopixel.NeoPixel (pin12,4)`: Initialize the RGB programming light library. The first parameter is the pins of the RGB light, and the second parameter is the number of RGB lights.

`display.show (Image.HAPPY)`: Make a smile icon appear on the micro:bit dot matrix;

`radio.on ()`: Turn on the wireless function. Because the wireless function consumes more power and occupies memory, it is turned off by default. You can also use `radio.off ()` to turn off the wireless function;

`radio.config (group = 1)`: configure wireless group=1, so that other micro:bit devices with wireless group=1 can communicate with each other, the default is 0, the select-able group is 0 ~ 255. Handle and car need to be set in same group.

Control the car forward and backward and turn left and right.

`incoming = radio.receive ()`: Receives the data transmitted wirelessly and saves it to the incoming variable;

if incoming is 'up', the car moves forward,
 if incoming is 'up', the car moves backward,
 if incoming is 'left', the car will turn left,
 if incoming is 'right', the car will turn right,
 if incoming is 'stop', the car will stop.

if incoming is 'R', the body RGB lights up red,
 if incoming is 'G', the body RGB lights up green,
 if incoming is 'B', the body RGB lights up blue,
 if incoming is 'Y', the body RGB lights up yellow.

Note: The value of incoming should correspond to the value sent by the handle. Only the same value can receive and execute the command.

Handle code

This code need to be downloaded into micro:bit board of tumble:bit car.



Case --- 1

```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import ghandle
4 import radio
5
6 display.show(0)
7 radio.on()
8 radio.config(group=1)

```

Import display、sleep、Image、ghandle and radio libray.

`radio.on()`: Open wirelessly function

`radio.config(group=1)`: Set the wireless group = 1 to keep the same with the group of the car;

Case --- 2

```

10 while True:
11
12     if ghandle.rocker(ghandle.up):
13         radio.send('up')
14         display.show(Image.ARROW_N)
15     elif ghandle.rocker(ghandle.down):
16         radio.send('down')
17         display.show(Image.ARROW_S)
18     elif ghandle.rocker(ghandle.left):
19         radio.send('left')
20         display.show(Image.ARROW_W)
21     elif ghandle.rocker(ghandle.right):
22         radio.send('right')
23         display.show(Image.ARROW_E)
24     elif ghandle.rocker(ghandle.pressed):
25         radio.send('turn_off')
26         display.show(Image.NO)
27     else:
28         radio.send('stop')
29         display.clear()

```

- 1) If it detects that `ghandle.rocker (ghandle.up)` is True, it means that the rocker of the handle is pushed up, and the 'up' command is sent wirelessly, and an upward icon is displayed on LED dot matrix.
- 2) If it detects that `ghandle.rocker(ghandle.down)` is True, it means that the rocker of the handle is pushed down, and the 'down' command is sent wirelessly, and an down icon is displayed on LED dot matrix.
- 3) If it detects that `ghandle.rocker(ghandle.left)` is True, it means that the rocker of the handle is pushed left, and the 'left' command is sent wirelessly,

and an left icon is displayed on LED dot matrix.

- 4) If it detects that `ghandle.rocker(ghandle.right)` is True, it means that the rocker of the handle is pushed right, and the 'right' command is sent wirelessly, and an right icon is displayed on LED dot matrix.
- 5) If it detects that `ghandle.rocker(ghandle.pressed)` is True, it means that the rocker of the handle is pressed, and the 'pressed' command is sent wirelessly, and an "X" icon is displayed on LED dot matrix.
- 6) If it does not operate to send 'stop' and clear the display.

Case --- 3

```

30
31     if ghandle.B1_is_pressed():
32         radio.send('R')
33         display.show("R")
34     if ghandle.B2_is_pressed():
35         radio.send('G')
36         display.show("G")
37     if ghandle.B3_is_pressed():
38         radio.send('B')
39         display.show("B")
40     if ghandle.B4_is_pressed():
41         radio.send('Y')
42         display.show("Y")

```

Determine whether the button is pressed. The commands 'R', 'G', 'B', 'Y' are sent for B1 (red), B2 (green), B3 (blue), and B4 (yellow).

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.

```

6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0

```

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.

```

6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0
13
14
15 while True:
16     voice = tinybit.getVoicedata()
17     if voice > 100:

```

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



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 of handle We can see handle micro:bit dot matrix of car will display a smile pattern. Then, we can open the power switch of car, they will pair automatically.

The button on the right side of the handle can control headlights. Press the rocker down to turn off the headlights. Push the rocker forward, backward, left and right to control the car movement.