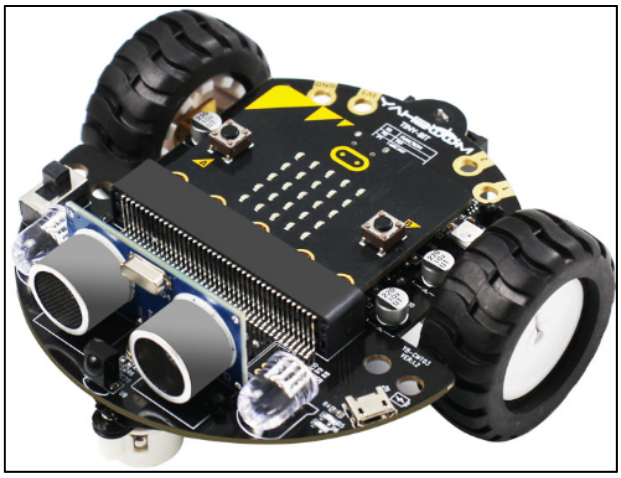
Robot course11 --- Handle control



Learning goal：

In this lesson, we will learn how to use micro:bit wireless transfer data,and control the Tiny:bit car with handle.

2.Wireless communication principles:

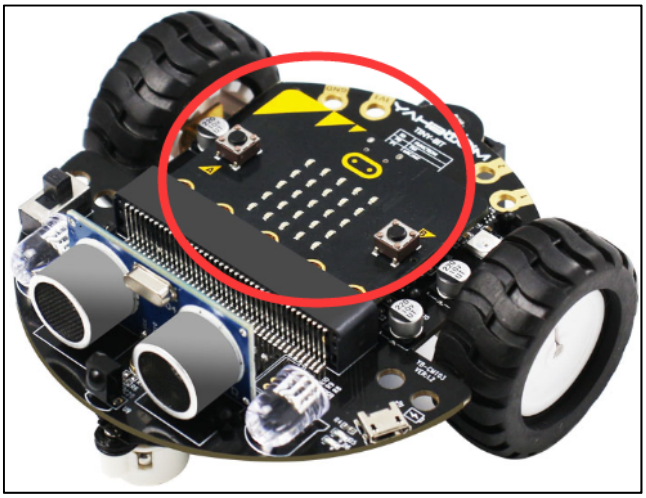
With the micro:bit radio module, different devices can work together through a simple wireless network. When the radio function is turned on for micro:bit, a simple wireless local area network is generated. The micro:bit board with radio function turned on can set parameters within the effective range.

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:

Tiny:bit car code:

Please use the MU software to open the **tinybit-car-code.py** file we provided.



1) 1)Import the libraries display, Image, tinybit and radio to be used.
   display.show (Image.HEART): Show a love icon on the microbit matrix;
   radio.on (): Turn on the wireless function. Because the wireless function consumes

more power and occupies memory, it is disabled 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, Range of group is 0 ~ 255.

Note: the set group value It needs to be consistent with the handle setting, otherwise normal communication cannot be performed.

Code as shown below:

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

display.show(Image.HEART)
radio.on()
radio.config(group=1)
```

2) Control the car advance, back, spin left and spin right functions:

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

if incoming is 'up', the car advance; 'down' the car back; 'left' the car spin left; 'right' makes the car spin right; And 'stop' makes the car stop.

Code as shown below:

```
while True:
    incoming = radio.receive()
    if incoming == 'up':
        tinybit.car_run(100, 100)
    elif incoming == 'down':
        tinybit.car_back(100, 100)
    elif incoming == 'left':
        tinybit.car_spinleft(100, 100)
    elif incoming == 'right':
        tinybit.car_spinright(100, 100)
    elif incoming == 'stop':
        tinybit.car_stop()
```

3) If incoming is ‘R’, the car headlights are red, ‘G’ is the car headlights are green, ‘B’ is the car headlights are blue, and ‘Y’ is the car headlights are yellow.

Code as shown below:

```
    if incoming == 'R':
        tinybit.car_HeadRGB(255, 0, 0)
    elif incoming == 'G':
        tinybit.car_HeadRGB(0, 255, 0)
    elif incoming == 'B':
        tinybit.car_HeadRGB(0, 0, 255)
    elif incoming == 'Y':
        tinybit.car_HeadRGB(255, 255, 0)
    elif incoming == 'turn_off':
        tinybit.car_HeadRGB(0, 0, 0)
```

!Note:

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

Handle control code:

Please use the MU software to open the **ghandle-code.py** file we provided.

1) Import the libraries display, sleep, Image, ghandle, and radio that you need to use.
radio.on (): Turn on the wireless function.
radio.config (group = 1): set wireless group = 1, which is consistent with the group of the car.

Code as shown below:

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

display.show(0)
radio.on()
radio.config(group=1)
```

2) 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.

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.

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.

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.

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.

If it does not operate to send 'stop' and clear the display.

Code as shown below:

```
while True:

    if ghandle.rocker(ghandle.up):
        radio.send('up')
        display.show(Image.ARROW_N)
    elif ghandle.rocker(ghandle.down):
        radio.send('down')
        display.show(Image.ARROW_S)
    elif ghandle.rocker(ghandle.left):
        radio.send('left')
        display.show(Image.ARROW_W)
    elif ghandle.rocker(ghandle.right):
        radio.send('right')
        display.show(Image.ARROW_E)
    elif ghandle.rocker(ghandle.pressed):
        radio.send('turn_off')
        display.show(Image.NO)
    else:
        radio.send('stop')
        display.clear()
```

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

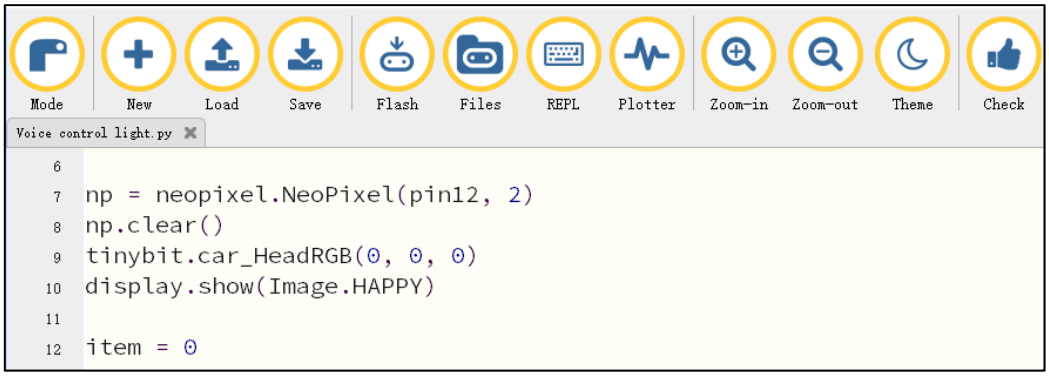
Code as shown below:

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

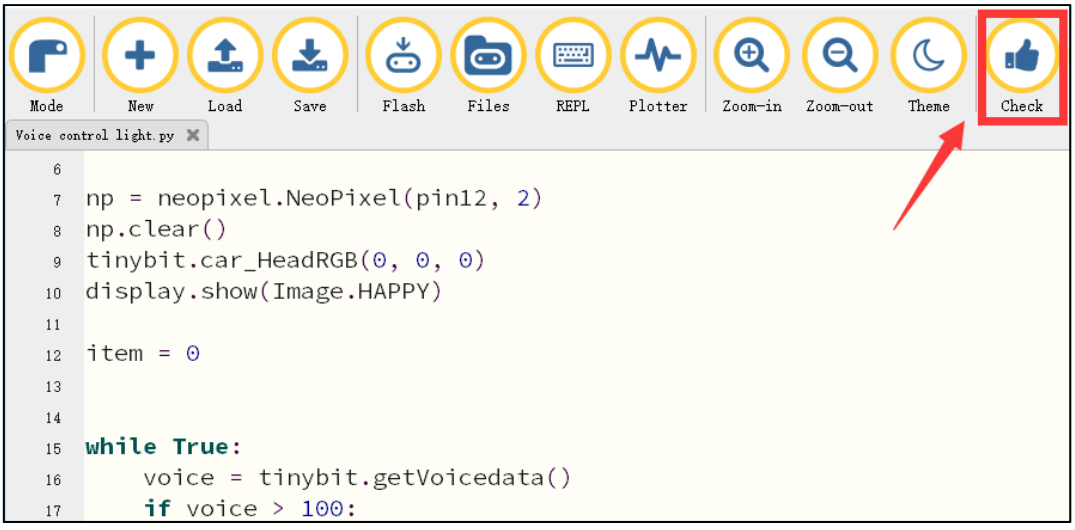
Programming and downloading：

1.You should open the Mu software, and enter the code in the edit window, , as shown below.

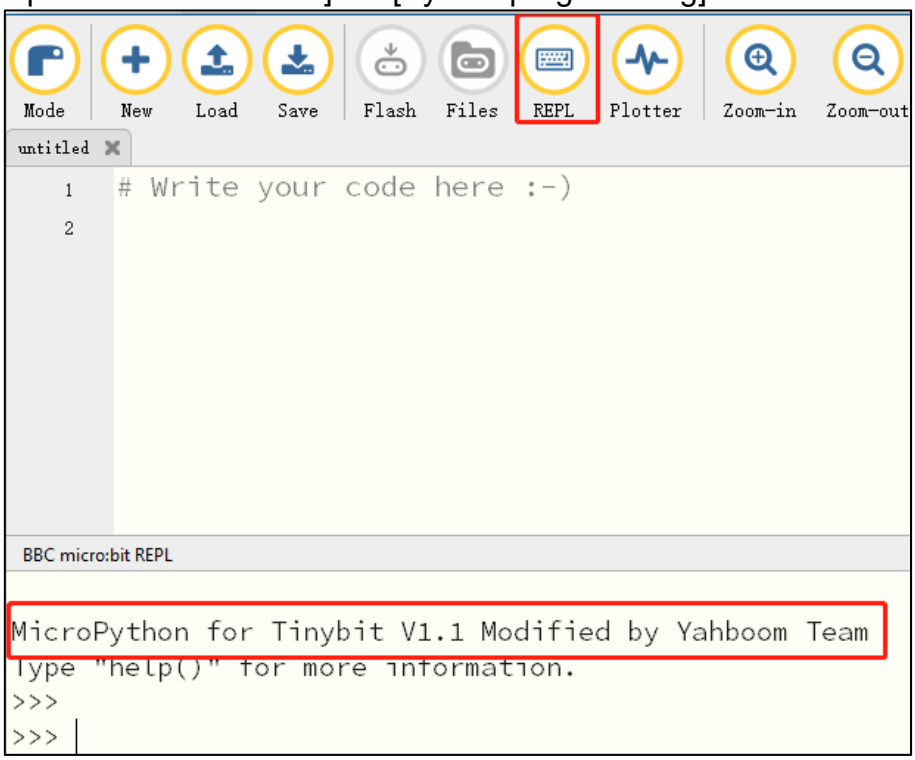
Note! All English and symbols should be entered in English, and the last line must be a space.



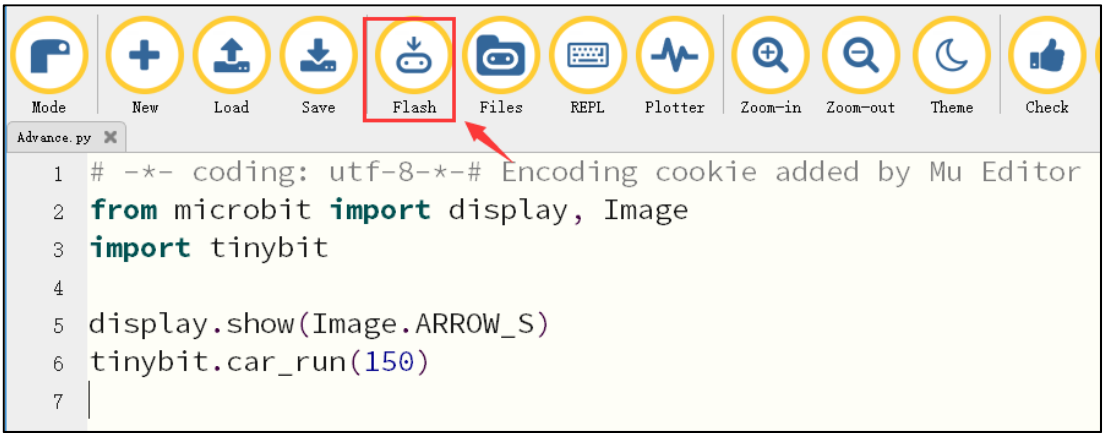
2.You can click the **"Check"** button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.



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

After download is complete, open the power switch of car. We can see micro:bit dot matrix of car will display a Heart pattern. Then, we can open the power switch of handle, 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.