Ferris wheel Handle Control

Learning goals:

In this lesson we will learn to use the Handle to remotely control the building block Ferris wheel.

About wiring:

We need to connect two building block motors to the **M1** interfaces of the Super:bit expansion board.

The black wiring of the motor is near the battery side.

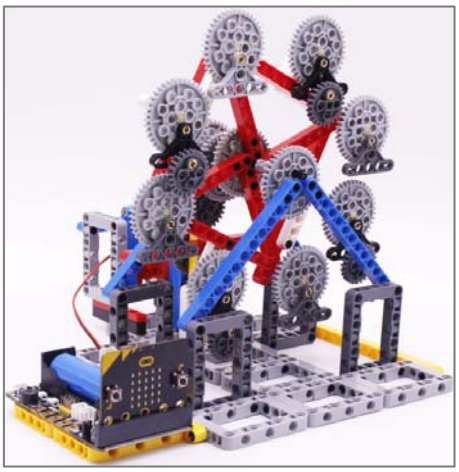
Wireless communication principles:

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.

About code:

Ferris wheel code

Please use the MU software to open the **Ferris_wheel_code.py** file we provided.



1) Import the libraries neopixel, super:bit and radio to be used.
display.show(Image.HEART): Show a “heart” on the micro:bit 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:

```
from microbit import *

import superbit
import radio

display.show(Image.HEART)
radio.on()
radio.config(group=1)
a = 0
flag = 0
```

2) Control the motor rotation:

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

If incoming is ‘R’, the motor forward, ‘G’ makes the arm rise, ‘B’ makes the arm drop, and ‘Y’ makes the clip close.

Code as shown below:

```
while True:

    incoming = radio.receive()
    if incoming == 'B' and flag == 0 :
        a = a - 50
        if a < 0:
            a = 0
        display.show(a)
        incoming = radio.receive()
        if incoming == 'B':
            flag = 1

    elif incoming == 'G' and flag == 0:
        a = a + 50
        if a > 250:
            a = 250
        display.show(a)
        incoming = radio.receive()
        if incoming == 'G':
            flag = 1

    elif incoming == 'R':
        superbit.motor_control(superbit.M1, a, 0)

    elif incoming == 'Y':
        superbit.motor_control(superbit.M1, -a, 0)

    elif incoming == 'T':
        flag = 0
```

Handle control code:

Please use the MU software to open the **Handle_code.py** file we provided.

1) Import the libraries microbit, ghandle, and radio that you need to use.
display.show(Image.HEART): Show a icon on the micro:bit matrix;
radio.on (): Turn on the wireless function;
radio.config (group = 1): set wireless group = 1, which is consistent with the group of the Arm:bit;
Code as shown below:

```
from microbit import *
import ghandle
import microbit
import radio

display.show(Image.HEART)
microbit.sleep(1000)
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.
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:

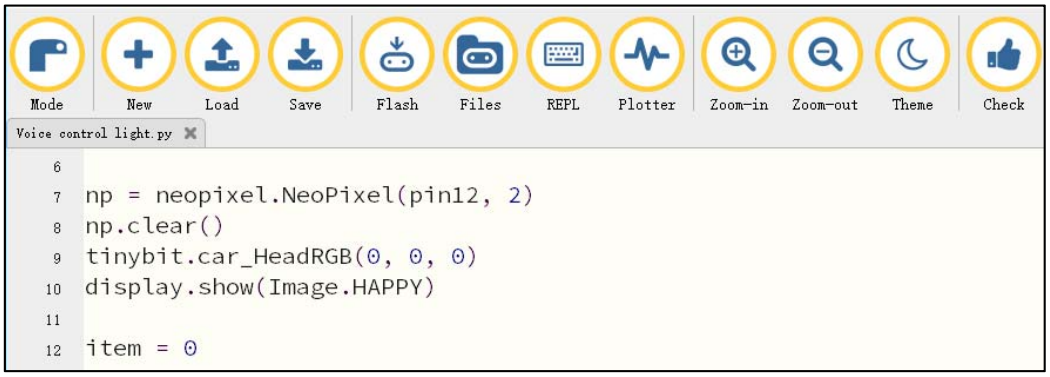
```
while True:

    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)
    elif ghandle.B1_is_pressed():
        radio.send("R")
        display.show("R")
    elif ghandle.B2_is_pressed():
        radio.send("G")
        display.show("G")
    elif ghandle.B3_is_pressed():
        radio.send("B")
        display.show("B")
    elif ghandle.B4_is_pressed():
        radio.send("Y")
        display.show("Y")
```

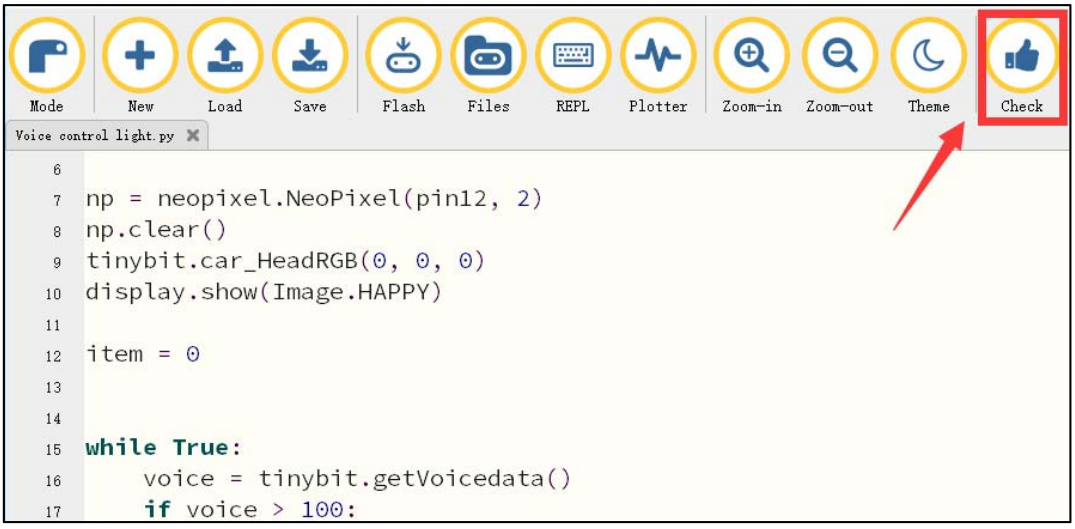
Programming and downloading：

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

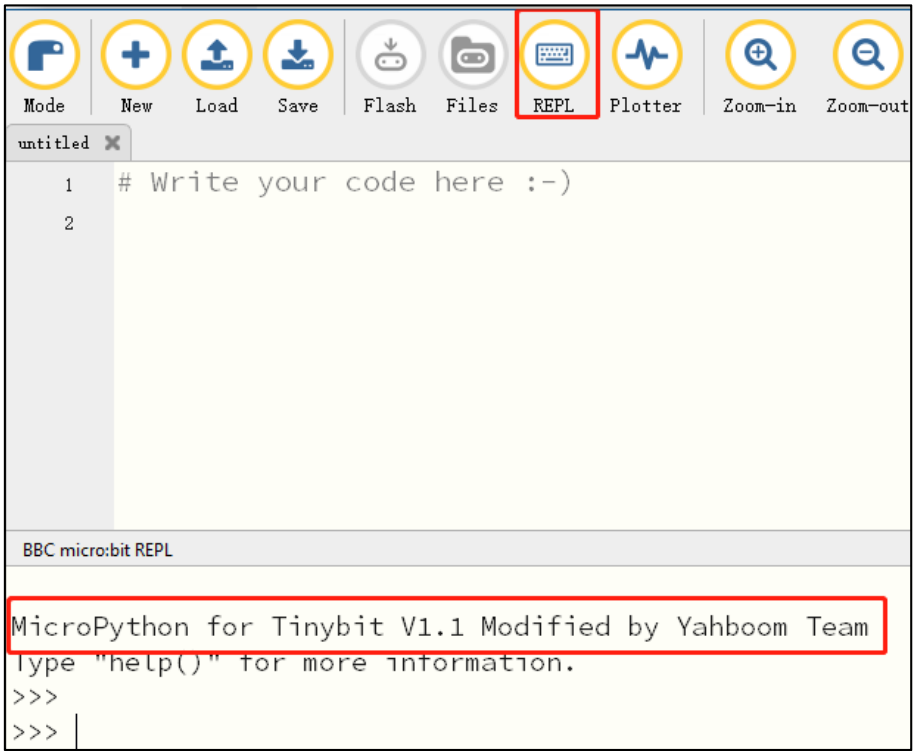
Note! All English and symbols should be entered in English, and the last line must be a space.



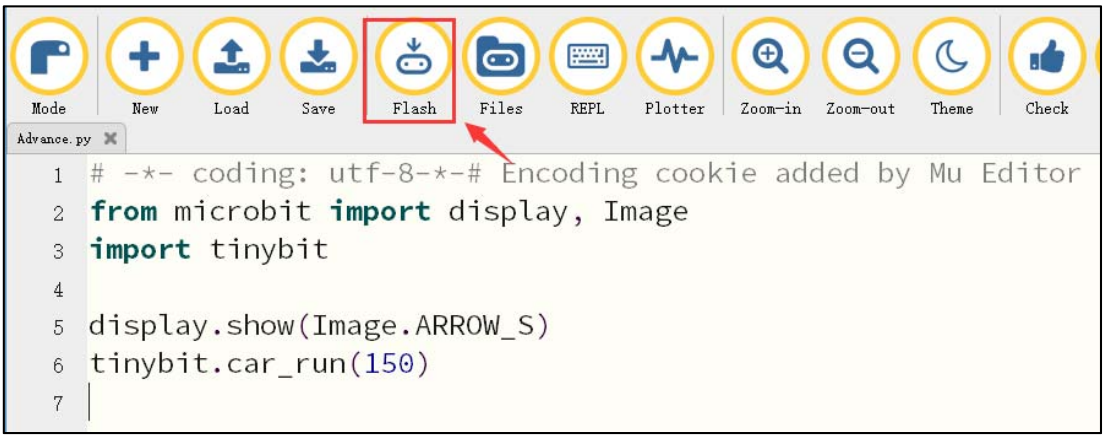
2.You can click the **"Check"** button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.



3.Click **"REPL"** button,check whether the tinybit library has been downloaded. If not, please refer to the [Preparation before class]---> [About 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 Superbit library hex file we provided to the micro: bit board.
For the specific method of adding library files, please refer to 【1.Preparation before class】---【About Python programming】

Experimental phenomena

After download is complete, you can see that the micro:bit dot matrix shows an "heart" as shown in Figure 1.1-1.2.

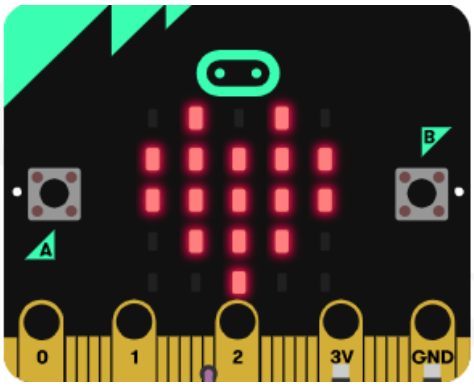


Figure 1.1

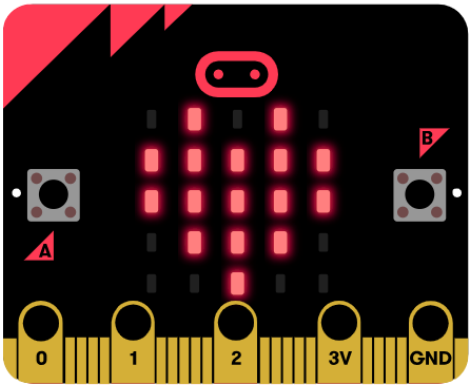


Figure 1.2

Then, open the micro:bit handle. After the handle is connected with the micro:bit building block Ferris wheel, you can use the button to control the Ferris wheel.

Red and yellow button can switch direction of Ferris wheel rotation,
Green and blue button can increase or decrease speed of Ferris wheel rotation.

Note: The speed of motor will become 0 when direction is switched.