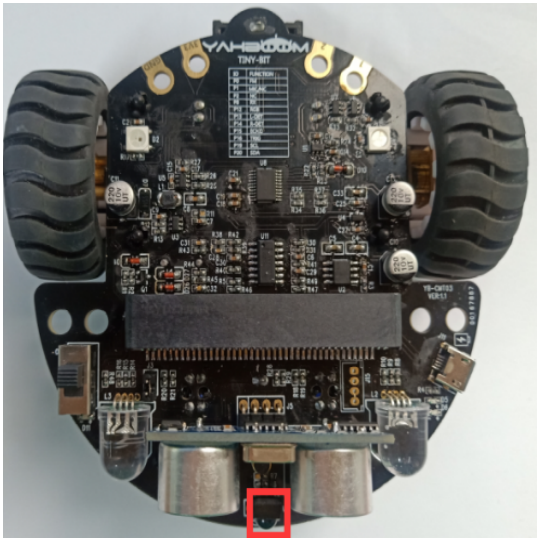
Robot course10 --- IR control



Note：

1. When performing infrared remote control, the remote controller should face the infrared receiver on the expansion board.

2. There is a plastic piece on the bottom of the infrared remote controller that needs to be taken down for normal use.

3. The infrared light emitted by the infrared remote controller and the infrared receiver is invisible to the human eye. It can be seen under the camera without filtering infrared light.

Learning goal：

In this lesson, we will learn how to read the data received by the Tiny:bit infrared receiver, so as to control the Tiny:bit car with the infrared remote controller.

Introduction of infrared remote communication principle:

Infrared remote control consists of two parts: sending and receiving. The transmitting end uses a single chip computer to code and modulate the binary signal to be transmitted into a series of pulse train signals, and transmits the infrared signal through an infrared transmitting tube. Infrared receiving is used to receive, amplify, detect, and shape infrared signals, and demodulate remote control code pulses. In order to reduce interference, a reliable integrated IR receiver IRM3638T (receiving infrared signal frequency: 38kHz) is used to receive infrared signals. It simultaneously amplifies, detects, and shapes the signals to obtain TTL-level coded signals, which are then sent to the microcontroller. The code value corresponding to each button is obtained by the single-chip decoding, and the relevant program is executed to control the car.

Code:

Please use the MU software to open the **IR control.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, sleep delay time, music is play music, pin8 is the pin of the infrared receiver.
Code as shown below:

```
# -*- coding: utf-8-*-# Encoding cookie added by Mu Editor
# from microbit import *
from microbit import display, sleep, pin8, pin12
import tinybit
import music
import neopixel
```

From the hardware interface manual, we can know that buzzer are directly driven by P10 of micro:bit board .

Category	Function	Number	Drive	The number of Drive pin	The number of connected to the controller	micro:bit
Buzzer	Buzzer	FM	Micro:bit drive directly		FM	P0
Voice sensor	Voice sensor	MIC			MIC	P1
LED light	Water light	LED-RGB			LED-RGB	P12
Tracking sensor	Left tracking	L-DET			L-DET	P13
	Right tracking	R-DET			R-DET	P14
Ultrasonic module	Echo pin	ECHO			ECHO	P15
	Trigger pin	TRIG			TRIG	P16
Infrared receiver	Infrared remote control	RX			RX	P8
I2C interface	I2C interface	SCL			SCL	P19
		SDA			SDA	P20
Motor	Left motor Forward	L-INA	STM8S	PC6/TIM1_CH1	SCL, SDA	P19, P20
	Left motor Reverse	L-INB		PC7/TIM1_CH2		
	Right motor Forward	R-INA		PC3/TIM1_CH3		
	Right motor Reverse	R-INB		PC4/TIM1_CH4		
RGB Searching light	Red	LED-R		PC5/TIM2_CH1		
	Green	LED-G		PD3/TIM2_CH2		
	Blue	LED-B		PD2/TIM2_CH3		

2) **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.

tinybit.init_IR (pin8): Initialize the infrared receiver. The parameter is the pin of the infrared sensor.

tinybit.car_stop (): stop the car by default.

display.off (): Turn off the micro:bit dot matrix screen. If it is not turned off, it will conflict with the infrared receiver and cause errors.

speed = 100: set the car speed.

a: Record the data, and solve the situation of occasionally receiving -0x01 when long-pressed.

music.pitch ('c'): buzzer whistle.

```
np = neopixel.NeoPixel(pin12, 2)
np.clear()
tinybit.car_HeadRGB(0, 0, 0)
tinybit.init_IR(pin8)
tinybit.car_stop()
display.off()

speed = 100
a = 0
music.play('c')
```

3) Read the data received by the infrared receiver, and only take the upper 8 bits. The key code value of the remote control is shown in the following figure:

```
while True:
    # print(hex(tinybit.get_IR(pin8)))
    value = tinybit.get_IR(pin8)
    value = value >> 8
```

4) First,we need to process two special values. By default, you will always receive the value-0x01. If you press and hold a key, you will receive 0xff. The variable a is to solve the problem that you may receive -0x01 when you press and hold the key. The code value of each key.

```
25      # default
26      if value == -0x01:
27          a = a + 1
28          if (a > 3):
29              tinybit.car_stop()
30              a = 0
31      # long pressed
32      elif value == 0xff:
33          a = 0
```

5) The code value of the red power button is 0x00, we set **tinybit.car_HeadRGB (0, 0, 0)** is to turn off the headlights of car.

```
35          # off
36          if value == 0x00:
37              tinybit.car_HeadRGB(0, 0, 0)
```

6) Press Up button, control car advance.

```
38          # up
39          elif value == 0x80:
40              tinybit.car_run(speed, speed)
```

7) Press light button, HeadRGB light become red.

```
41          # light
42          elif value == 0x40:
43              tinybit.car_HeadRGB(255, 255, 255)
```

8) Press horn button, car will whistle.

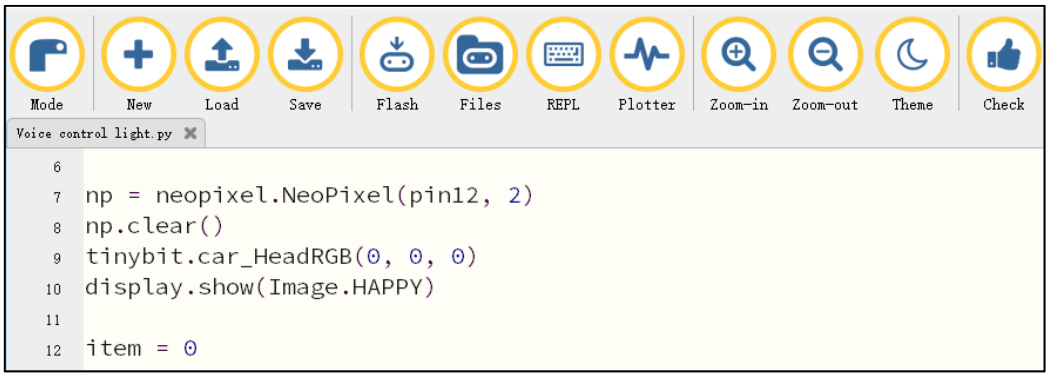
```
47          # buzzer
48          elif value == 0xa0:
49              music.pitch(698)
50              sleep(400)
51              music.pitch(0)
```

...

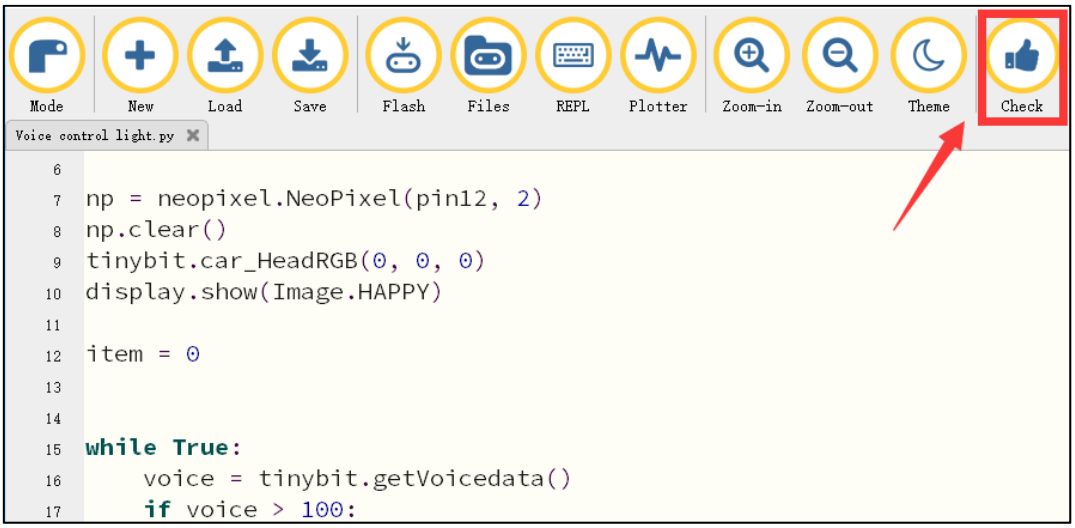
Programming and downloading：

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

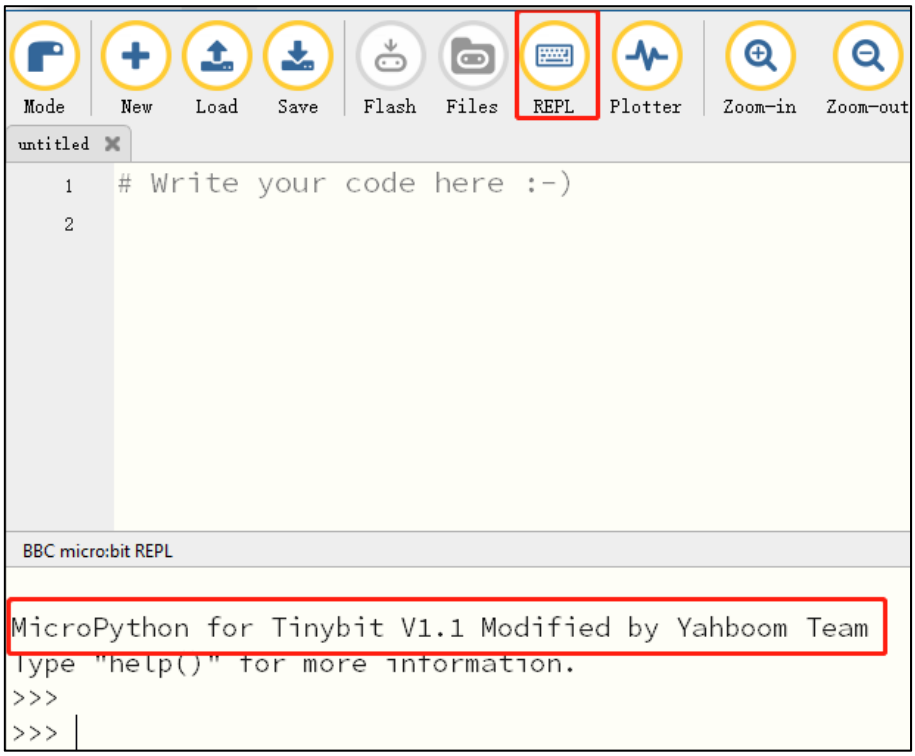
Note! All English and symbols should be entered in English, and the last line must be a space.



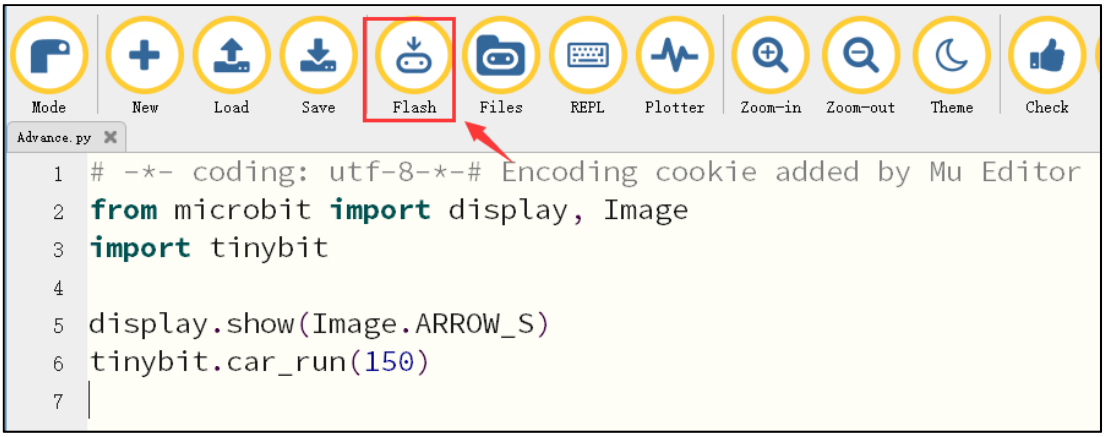
2.You can click the **"Check"** button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.



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.

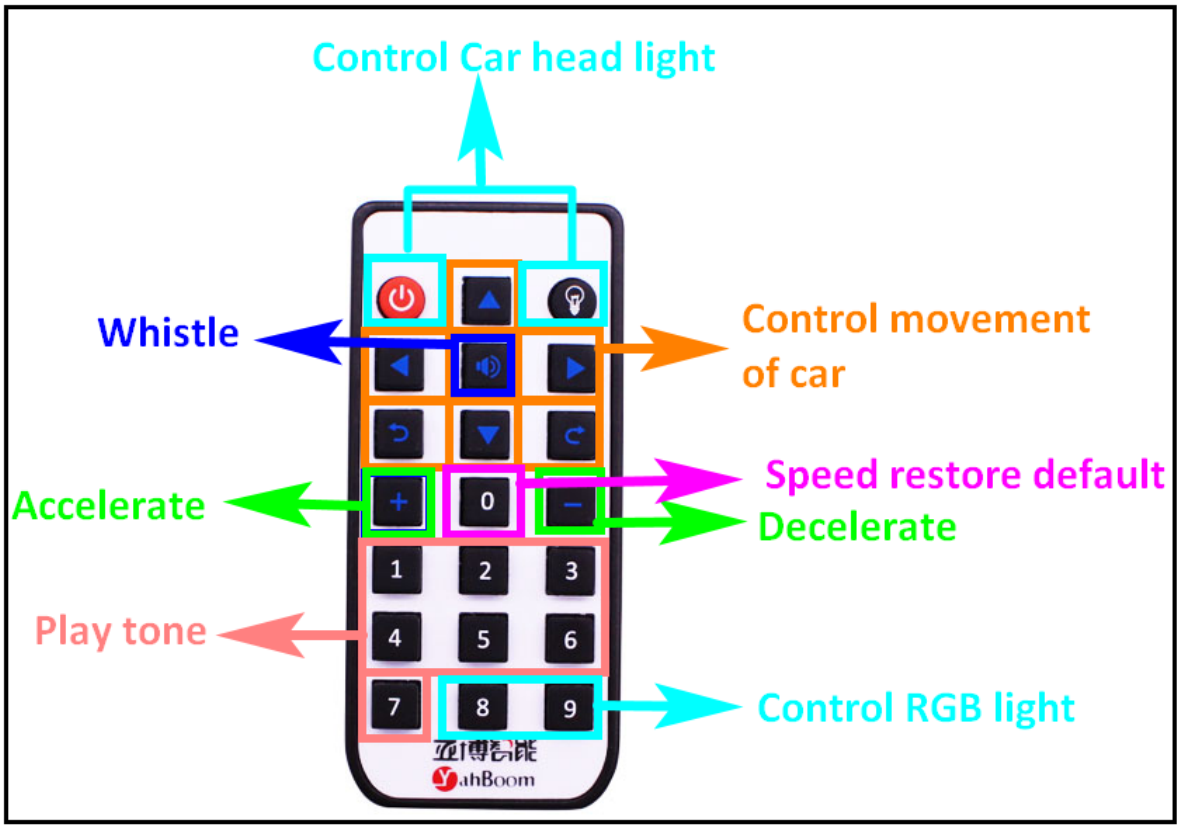


If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Buildingbit libraryhex file we provided to the micro: bit board.

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. The corresponding functions of all the keys on the infrared remote controller are shown below:


Control Car head light
Whistle
Control movement of car
Accelerate
Speed restore default
Decelerate
Play tone
Control RGB light
0
1
2
3
4
5
6
7
8
9