Robot course 9 --- Listener

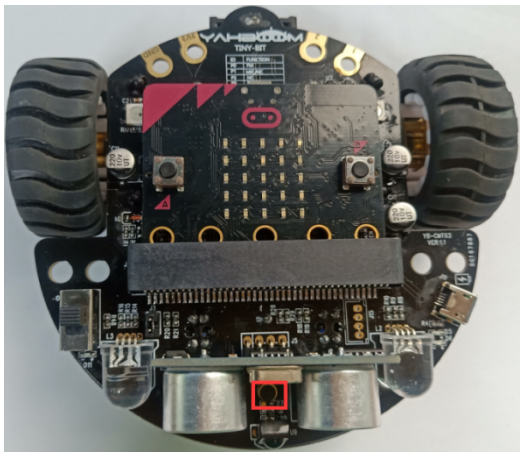
Learning goal：

In this lesson, we will learn how to read sound sensor data, and control motor and RGB light based on the detected data through Python programming.

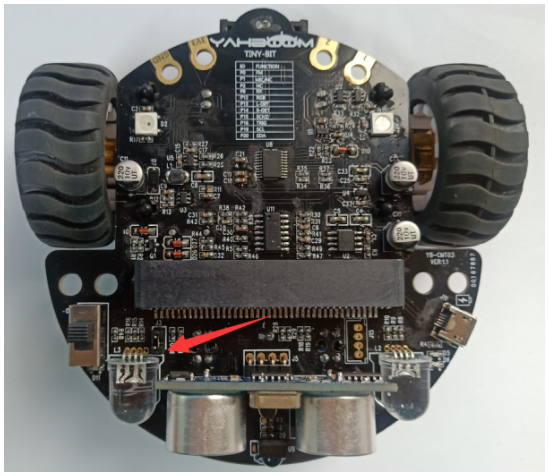
Introduction of Sound sensor principle

The role of the sound sensor is equivalent to a microphone, which can be used to receive sound wave vibration, so as to detect changes in the current ambient sound, but it cannot measure the intensity of noise. Built-in condenser cylindrical microphone that is sensitive to sound, sound waves can vibrate electret films in microphones, causing the capacitance to change, thereby generating a tiny voltage corresponding to it.

The position of the Voice sensor module in the robot.



！！！Note: In this experiment, we need to install the jumper cap in the position shown below.



Code:

Please use the MU software to open the **Listener.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.

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.

display.show (Image.HAPPY): Display a smile on micro:bit dot matrix.

delay = 80: Declare item as a global variable, and initialize it to 80.

Code as shown below:

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

np = neopixel.NeoPixel(pin12, 2)
np.clear()
tinybit.car_HeadRGB(0, 0, 0)
display.show(Image.HAPPY)
delay = 80
```

2) The custom micro:bit pattern is divided into nine levels in total, level_0 is all off, level_9 is all on, and the other represents different display heights.

```
level_0 = Image("00000:00000:00000:00000:00000")
level_1 = Image("00000:00000:00000:00000:09990")
level_2 = Image("00000:00000:00000:00900:99999")
level_3 = Image("00000:00000:00000:09990:99999")
level_4 = Image("00000:00000:00900:99999:99999")
level_5 = Image("00000:00000:99999:99999:99999")
level_6 = Image("00000:09990:99999:99999:99999")
level_7 = Image("00900:99999:99999:99999:99999")
level_8 = Image("99999:99999:99999:99999:99999")
```

3) **voice = tinybit.getVoicedata()**: Read the data from the sound sensor and save it to the voice variable;

If the variable voice<10, stop the Tiny:bit car and clear the headlights and micro:bit dot matrix.

In other cases, according to the data of the voice sensor, set the level of the dot matrix display, switch the brightness and color of the front RGB searchlights and the body colorful lights.

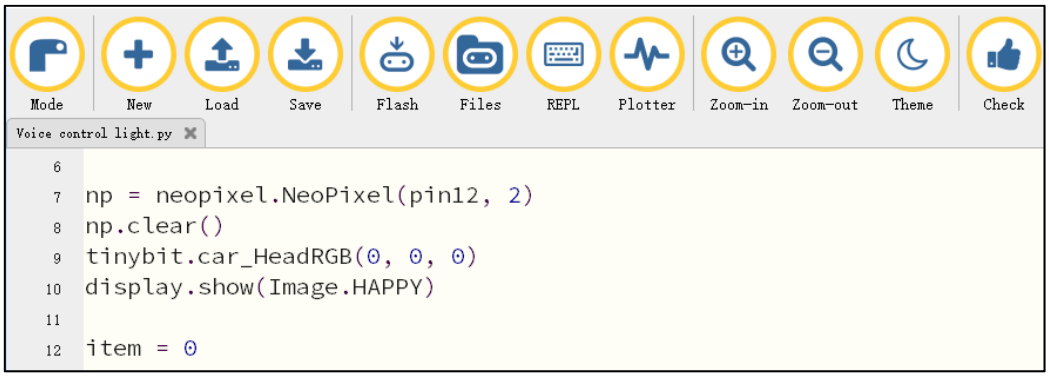
The larger the value,the larger the motor vibration amplitude.

```
while True:
    voice = tinybit.getVoicedata()
    if voice < 10:
        tinybit.car_stop()
        np.clear()
        tinybit.car_HeadRGB(0, 0, 0)
        display.show(level_0)
    elif voice >= 10 and voice < 40:
        display.show(level_1)
        tinybit.car_HeadRGB(0, 0, 50)
        np[0] = (255, 0, 0)
        np[1] = (255, 0, 0)
        np.show()
        tinybit.car_spinleft(50)
        sleep(delay)
        tinybit.car_spinright(50)
        sleep(delay)
```

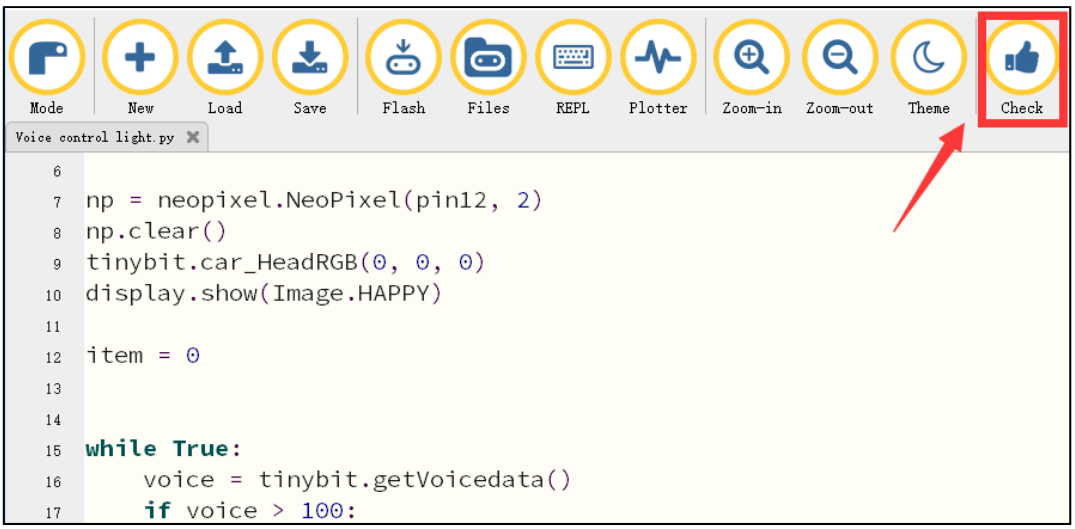
Programming and downloading：

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

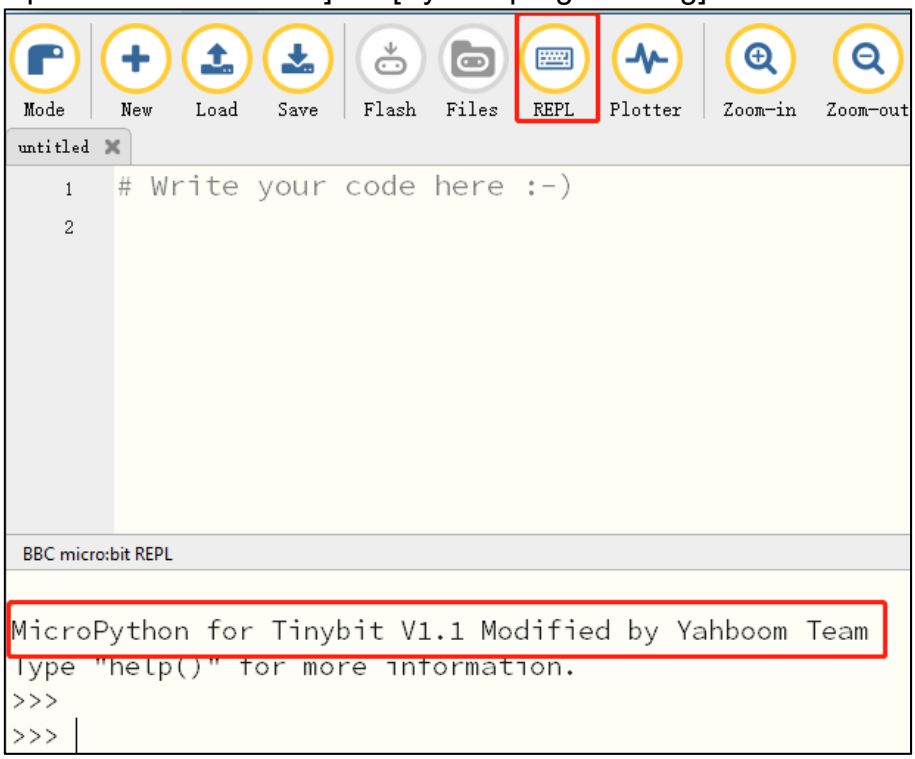
Note! All English and symbols should be entered in English, and the last line must be a space.



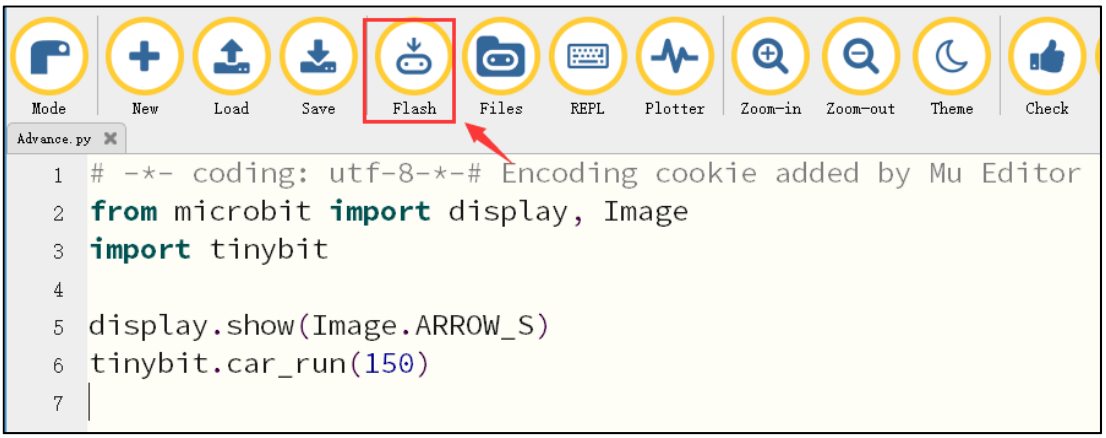
2.You can click the **“Check”** button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.



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.When we make sounds with different intensities next to Tiny-bit, the brightness of the lights on Tiny-bit will change, and the micro: bit dot matrix will display different patterns at the same time. At the same time, the sound intensity will control Tiny- Bit body swing, the louder the sound, the greater the body swing.