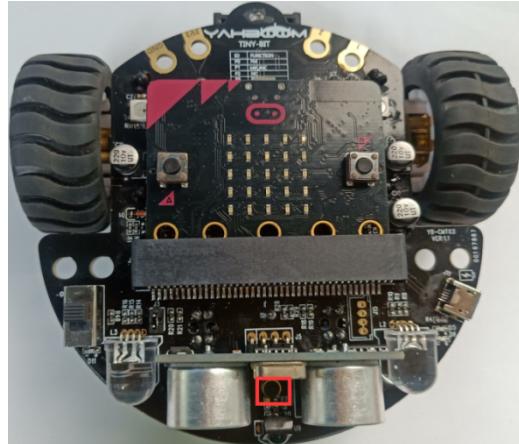


Courses10 --- Sound sensor

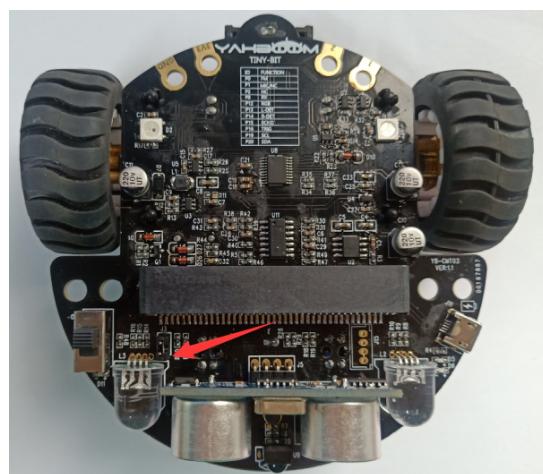
Learning goal:

This lesson learns how to use sound sensor by Python programming.

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:

```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4 from random import randint
5
6 display.show(Image.HAPPY)
7 item = 0
8
9
10 while True:
11     voice = tinybit.getVoicedata()
12     if voice > 100:
13         item = randint(1, 6)
14
15     if item == 1:
16         display.show(Image.HEART)
17     elif item == 2:
18         display.show(Image.COW)
19     elif item == 3:
20         display.show(Image.DUCK)
21     elif item == 4:
22         display.show(Image.TARGET)
23     elif item == 5:
24         display.show(Image.SNAKE)
25     elif item == 6:
26         display.show(Image.GIRAFFE)
27

```

- 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.

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

item = 0: Declare item as a global variable, and initialize it to 0.

voice = tinybit.getVoicedata(): Read the data of the sound sensor and save it to the variable voice.

If the variable voice> 100, it means that the light needs to be switched. First, a random number of 1 to 6 is generated and stored in the variable item. Then the color of the front RGB searchlight and the body colorful light are randomly set.

If the variable voice> 100, it means that the sound sensor detects the signal, and a random number of 1 to 6 is generated and stored in the variable item. If the test response is not sensitive in the experiment, you can change the number 100 to a smaller value; Then judge the value of item. Different values correspond to different patterns and are displayed on the micro:bit dot matrix.

According to the hardware interface reference manual, the sound sensor connect to P1 of micro:bit.

Category	Function	Number	Drive	The number of Drive pin	The number of connected to the controller	micro:bit
Buzzer	Buzzer	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	Micro:bit drive directly		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-INB		PC6/TIM1_CH1		
	Left motor Reverse	L-INB		PC7/TIM1_CH2		
	Right motor Forward	R-INB		PC3/TIM1_CH3		
	Right motor Reverse	R-INB	STM8S	PC4/TIM1_CH4		
	Red	LED-R		PD5/TIM2_CH1		
RGB Searching light	Green	LED-G		PD3/TIM2_CH2		
	Blue	LED-B		PD2/TIM2_CH3		

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.

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Sound sensor.py ×
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4 from random import randint
5
6 display.show(Image.HAPPY)
7 item = 0

```

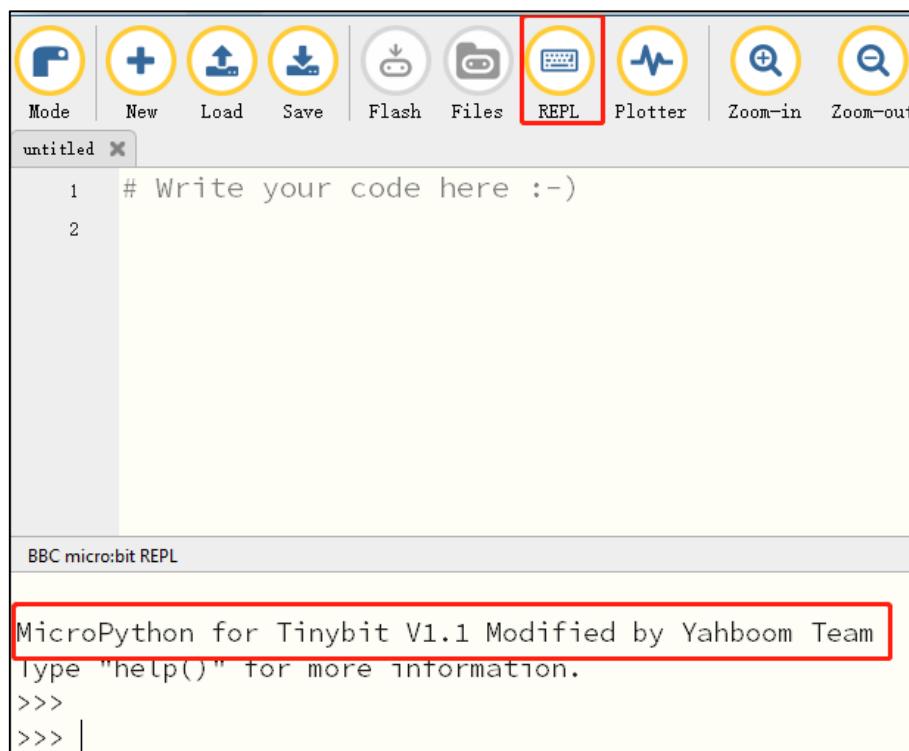
2. You can click the “Check” button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.

```

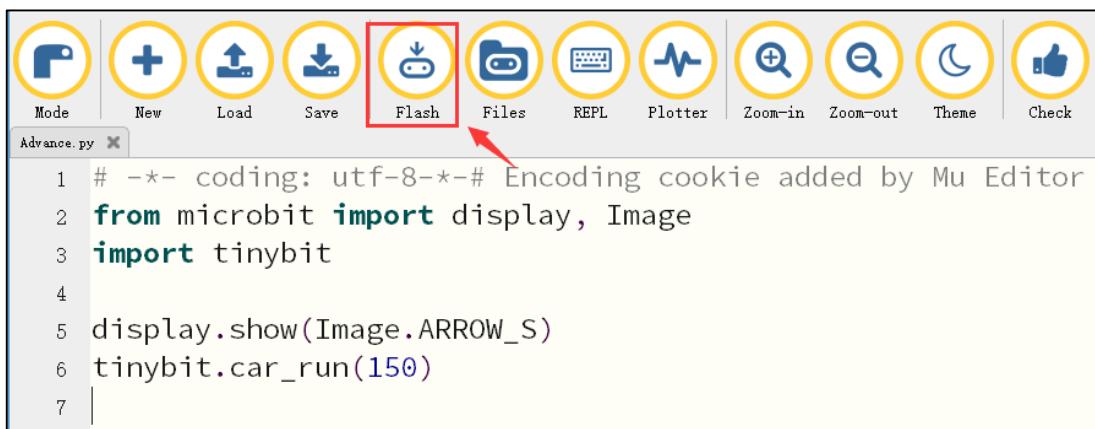
Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Voice control light.py ×
6
7 np = neopixel.NeoPixel(pin12, 2)
8 np.clear()
9 tinybit.car_HeadRGB(0, 0, 0)
10 display.show(Image.HAPPY)
11
12 item = 0
13
14
15 while True:
16     voice = tinybit.getVoicedata()
17     if voice > 100:

```

3. Click “REPL” button,check whether the tinybit library has been downloaded. If not, please refer to the [preparation before class]---> [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. Micro:bit will display a smile on dot matrix, we need to put robot car on the desk. When we Shooting table, the dot matrix will display different pattern.