

Lesson 2 Sound Detection and Placement

1. Project Principle

Sound sensor has a in-built capacitive microphone. Sound waves cause the diaphragm in the microphone to vibrate, resulting in capacitance change, which in turn produces a correspondingly small change in voltage. This voltage is then converted into a voltage in the range of 0-5 V and compared with an adjustable voltage with adjusted sensitivity by means of a comparator.

Then it is received by the data collector through the A/D conversion of module, and the range is 0-1023. The value is higher as the sound intensity increases, so the detected sound is proportional to the output analog quantity.



The path of program file: "6. Secondary Development/Sensor-extension Game/ Python Development/ Program Files/ Light Detection and Placement/main.py"

```

19 # 初始化光线传感器
20 light_sensor = ADC(Pin(32))
21 light_sensor.atten(ADC.ATTN_11DB)
22 light_sensor.width(ADC.WIDTH_10BIT)
23
24 if __name__ == '__main__':
25     arm.go_home() #机械臂复位，回到初始位置
26     nozzle.set_angle(0,1000) #喷嘴角度复位
27     time.sleep_ms(2000)
28     num = 0 #声音计数变量
29     angle = [12, 35, 55] #角度单位
30     while True:
31         light = light_sensor.read() #光线传感器检测当前
32         print(light)
33         if light > 900: # 光线传感器检测
34             print('num:',num+1)
35             buzzer.setBuzzer(100) #蜂鸣器响一下
36             time.sleep_ms(500)
37             arm.set_position((0,-165,100),1200) #机械臂移动到吸取位置上方，等待2秒后吸取
38             time.sleep_ms(2000)
39             arm.set_position((0,-165,86),600) #吸取木球

```

Firstly, import sensor library files and initialize sensor.

Then create sound detection function to detect sound. After the sound is detected, buzzer will sound once for responding.

Then use kinematics to control robotic arm and place block to the corresponding position.

2. Preparation

2.1 Hardware

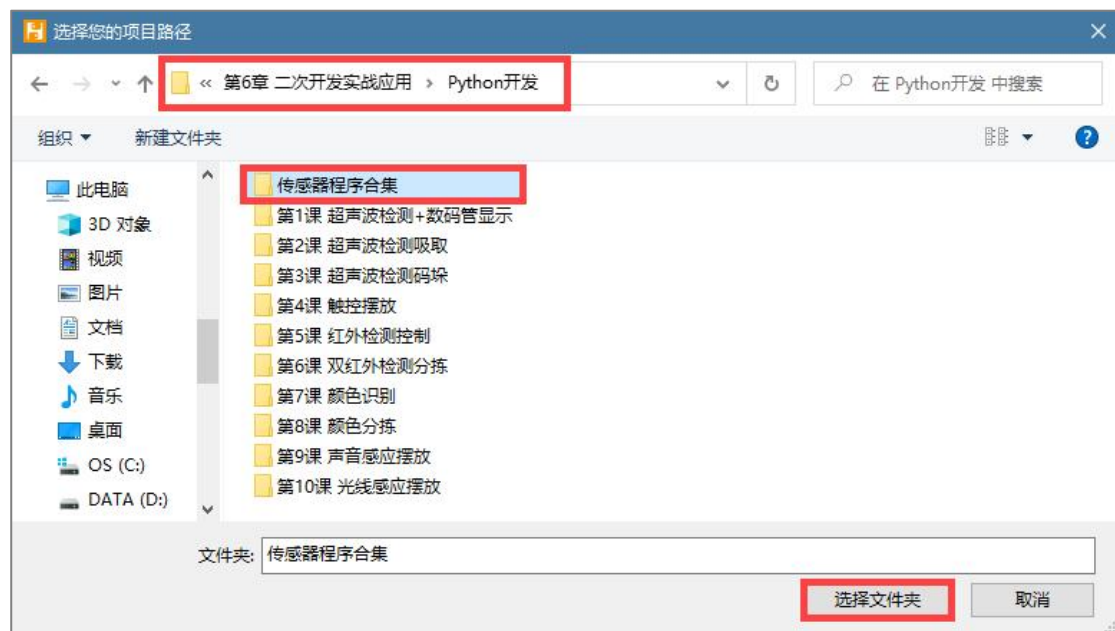
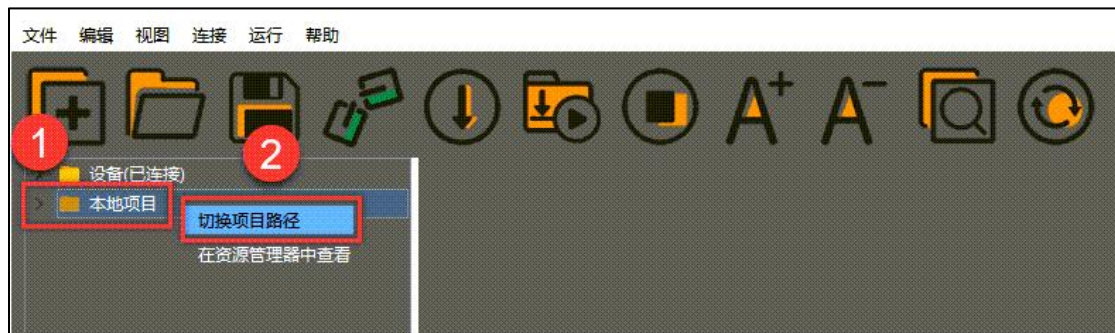
Please assemble sound sensor to the corresponding position on MaxArm according to the tutorial in folder “Lesson 1 Sensor Assembly” under the same directory.

2.2 Software

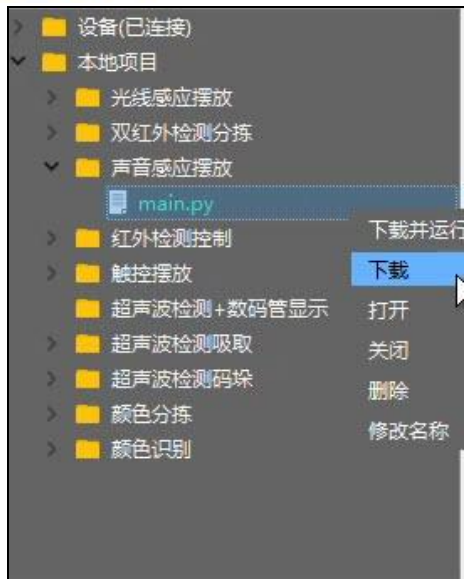
Please connect MaxArm to Python editor according to the tutorial in folder “4. MaxArm Underlying Program Learning/Python Development/Lesson 1 Set Development Environment”.

3. Program Download

- 1) After connecting, change the path of Workspace to “6. Secondary Development / Python Development/Sensor-extension Game”, and select “Program Files”.



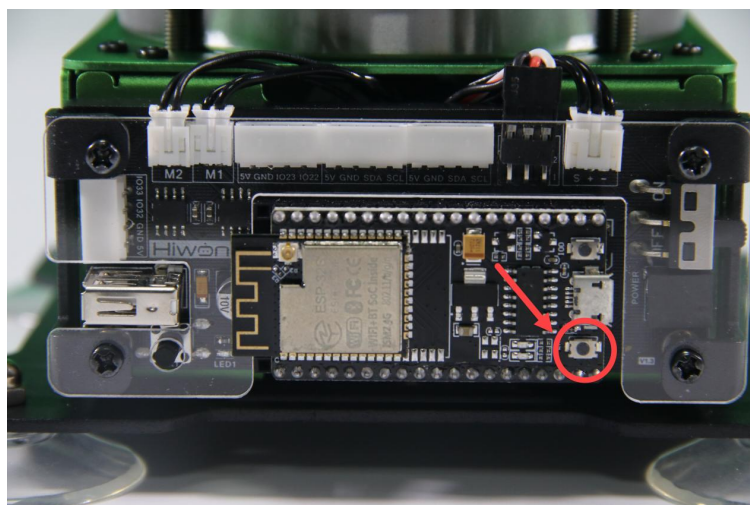
- 2) Click the folder “Sound Detection and Placement”, and then select all the program files in the folder. Then right click to download all the program files to the controller.



When the terminal prints the prompt as shown in the image below, it means download completed.

```
正在将文件下载到控制板.....
main.py文件下载完成！
>>>
```

- 3) After downloading, click on the reset icon or press the reset button on ESP32 controller to run program.



4. Project Outcome

When a knock sound is detected by sensor, the buzzer will sound once. According to the number of knocks, robot arm will place object to the corresponding position in the right side (take robot arm as the first person view) . You can view the demo video for the specific effect.

5. Program Instruction

5.1 Import Function Library

The path of program file: “6. Secondary Development/Sensor-extension Game/ Python Development/ Program Files/ Light Detection and Placement/main.py”

Before the program is executed, the related Python function libraries including buzzer, PWM servo, bus servo, air pump and other related Python function libraries are required to import.

```
1 import time
2 from machine import Pin,ADC
3 from Buzzer import Buzzer
4 from espmax import ESPMax
5 from PWMServo import PWMServo
6 from BusServo import BusServo
7 from RobotControl import RobotControl
8 from SuctionNozzle import SuctionNozzle
```

5.2 Initialization

Initialize sound sensor and robotic arm, which makes robotic arm back to the initial posture.


```

19 # 初始化声音传感器
20 sound_sensor = ADC(Pin(32))
21 sound_sensor.atten(ADC.ATTN_11DB)
22 sound_sensor.width(ADC.WIDTH_10BIT)
23
24 if __name__ == '__main__':
25     arm.go_home() #机械臂复位，回到初始位置
26     nozzle.set_angle(0,1000) #吸嘴角度置0
27     time.sleep_ms(2000)
28     angle = [12, 35, 55] #角度补偿
29     time_ms = time.ticks_ms()
30     num_st = False
31     num = 0

```

5.3 Sound Detection

When the sound is detected, the number of knocks plus one. If the number of detected knocks is greater than 3, then game will start from a new round.

```

32
33 while True:
34     sound = sound_sensor.read() #声音传感器检测函数
35
36     if sound > 50:
37         if num == 0 or (time.ticks_ms()-time_ms) < 1000:
38             time_ms = time.ticks_ms()
39             time.sleep_ms(80)
40             num += 1
41
42         if num and (time.ticks_ms()-time_ms) > 1500:
43             num = 3 if num > 3 else num
44             num_st = True
45             print(num)

```

5.4 Control Robotic Arm

When sound is detected, robotic arm will suck up block and place it to the corresponding position according to the number of knocks.

```

47 if num_st:
48     buzzer.setBuzzer(100) # 蜂鸣器响一下
49     arm.set_position((0,-160,100),1200) # 机械臂移动到初始位置上方，等待2秒后吸取
50     time.sleep_ms(2000)
51     arm.set_position((0,-160,86),600) # 吸取木块
52     nozzle.on() # 打开气泵
53     time.sleep_ms(1000)
54     arm.set_position((0,-160,180),1000) # 机械臂抬起
55     time.sleep_ms(1000)
56     arm.set_position((120,-20-60*(num-1),180),1500) # 移动到放置位置上方
57     nozzle.set_angle(angle[num-1],1500) # 设置角度并给，使木块放正
58     time.sleep_ms(1500)
59     arm.set_position((120,-20-60*(num-1),88),1000) # 放置木块
60     time.sleep_ms(1200)
61     nozzle.off() # 关闭气泵
62     arm.set_position((120,-20-60*(num-1),200),1000) # 机械臂抬起
63     time.sleep_ms(1000)
64     arm.go_home() # 机械臂复位，回到初始位置
65     nozzle.set_angle(0,1800) # 喷嘴角度置0
66     time.sleep_ms(2000)
67     num_st = False
68     num = 0

```

The specific program for block placement control is as follow:

The program use the buzzer.setBuzzer() function to control buzzer, as the figure shown below:

```

48 buzzer.setBuzzer(100) # 蜂鸣器响一下

```

Take the code “buzzer.setBuzzer(100)” for example:

The parameter “100” represent the sounding duration of buzzer and the unit is ms.

The use the arm.set_position() function to control robot arm to locate above the block, as the figure shown below:

```

49 arm.set_position((0,-160,100),1200) # 机械臂移动到初始位置上方，等待2秒后吸取
50 time.sleep_ms(2000)

```

Take the code “arm.set_position((0,-160,100),1200)” for example:

The first parameter “(0, -160, 100)” represents the coordinate of end effector. (You can read “Basic Kinematics Learning (must read!!!)” in folder “6. Secondary Development/ 6. Sensor-extension Game/ Python Development” to learn about space coordinate system.)

The second parameter “1200” is the running time and the unit is ms.

Next, use the nozzle.on() function to turn on air pump. If it is nozzle.off() function, air pump will be off.

```
52 nozzle.on() # 打开气泵
```

Then, use the nozzle.set_angle() function to control the rotation angle of suction nozzle, as the figure shown below:

```
56 arm.set_position((120,-20-60*(num-1),180),1500) # 移动到放置位置上升
```

Take the code “arm.set_position((120,-20-60*(num-1),180),1500)” for example.

The first parameter “120” represents the x-axis value of suction nozzle.

The second parameter “20-60*(num-1)” represents the y-axis of suction nozzle. The value of y-axis is calculated according to the number of detected knocks.

The third parameter “1500” is the running time and the unit is ms.

After block moves to the corresponding position, use the nozzle.set_angle () function to adjust the angle of suction nozzle to place block in a neat direction.

```
57 nozzle.set_angle(angle[(num-1)],1500) # 设置角度补偿,使不倾斜正
58 time.sleep_ms(1500)
```

Take the code “nozzle.set_angle(angle[(num-1)],1500)” as example:

The first parameter “angle[(num-1)]” is the angle setting. The different angles corresponding to three detection results are 12° , 35° and 55° individually, as the figure shown below:

```
28 angle = [12, 35, 55] # 角度补偿
```

Finally, robotic arm will be place block to the corresponding position and turn off air pump. Robot arm will return to initial position and suction nozzle restore to the initial position.


```
59     arm.set_position((120,-20-60*(num-1),88),1000) # 放置冰块
60     time.sleep_ms(1200)
61     nozzle.off() # 关闭喷嘴
62     arm.set_position((120,-20-60*(num-1),200),1000) # 机械臂抬起
63     time.sleep_ms(1000)
64     arm.go_home() # 机械臂复位，回到初始位置
65     nozzle.set_angle(0,1800) # 喷嘴角度置0
```