4.2 Sound Sensor

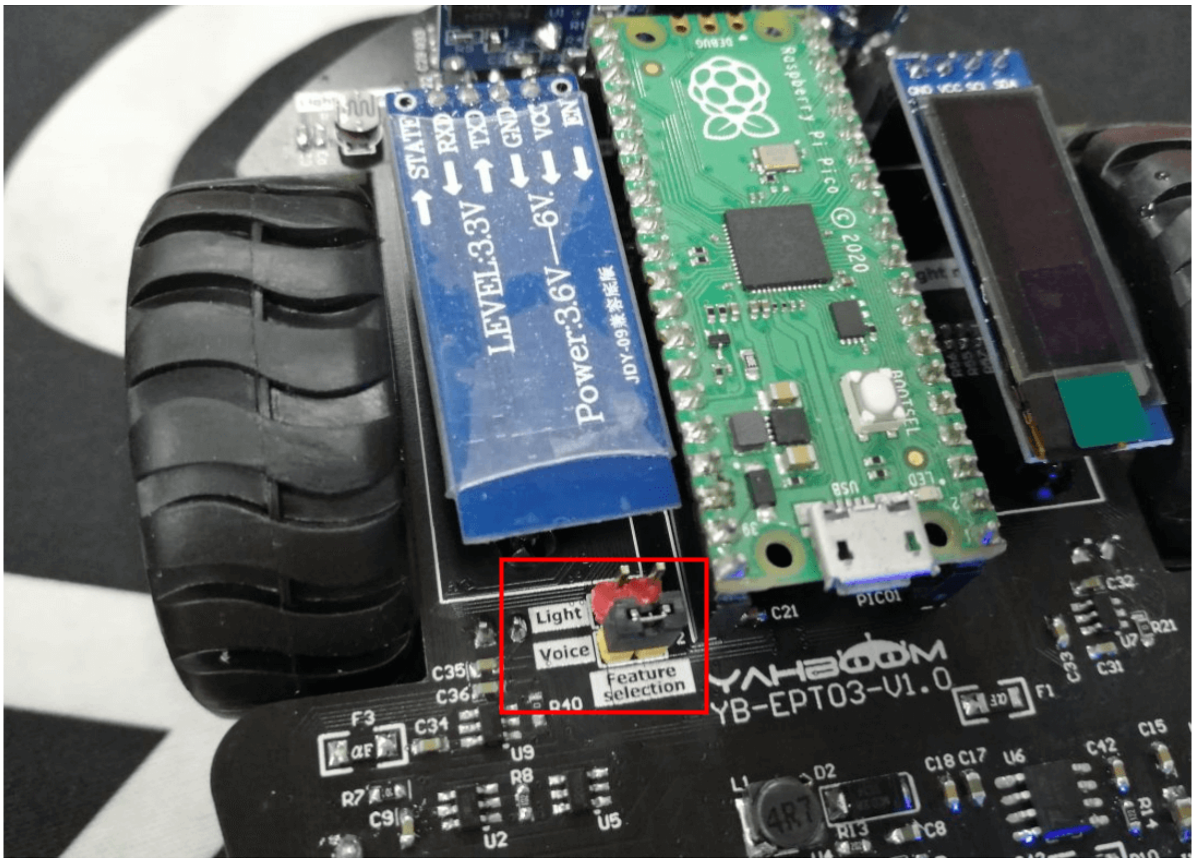
Note: To prevent false triggering, the detection reference value of the sound sensor is set to a large value. If the recognition effect is not good, you can modify the detection reference value appropriately, or blow air to trigger the sound sensor.

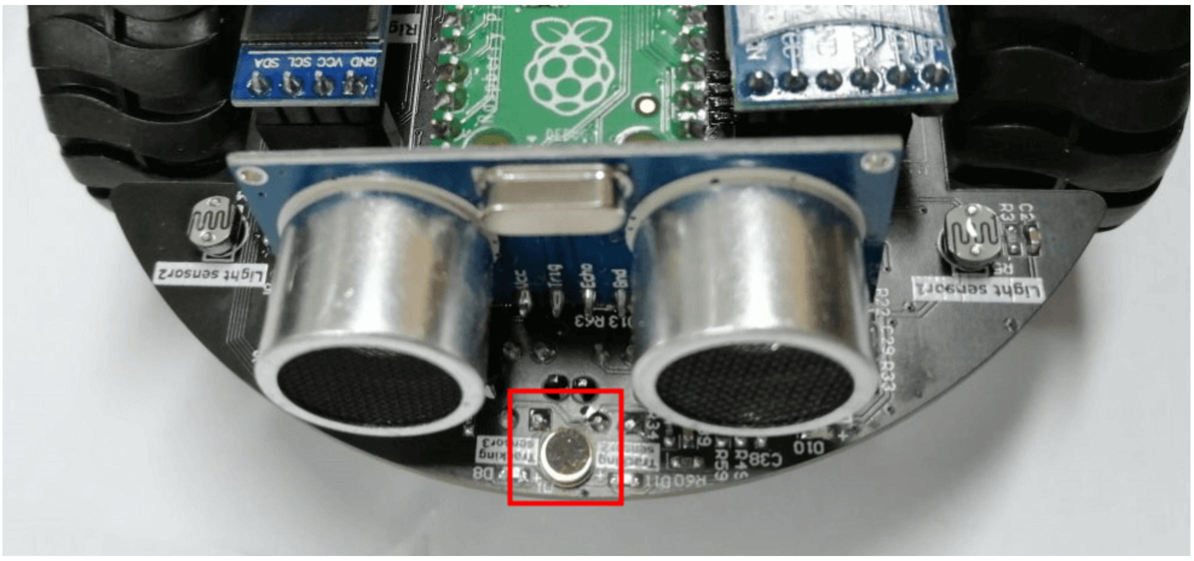
I. Learning Objectives

1. Learn to combine the sound sensor and OLED of the Raspberry Pi Pico 2/Pico mainboard and the car expansion board to conduct experiments.
2. Understand the use of sound sensors.

II. Hardware Usage

This course uses the Pico 2/Pico mainboard and the sound sensor and OLED of the car expansion board. **Please connect the jumper cap to the Voice pin header before running**.





The sensor has a built-in condenser electret microphone that is sensitive to sound. The sound waves make the electret film in the microphone vibrate, causing the capacitance to change, and a small voltage corresponding to the change is generated. This voltage is converted into a suitable voltage through operational amplification and transmitted to the PICO development board. Because it has no ability to detect noise, try to use it in a quiet environment.

3. Program Analysis

Code path: Code -> 2.Sensor advanced course -> 2. Sound sensor.py

```
from pico_car import SSD1306_I2C
from machine import Pin, I2C, ADC
import time
#initialization oled
i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)
oled = SSD1306_I2C(128, 32, i2c)
#initialization ADC
Sound = machine.ADC(27)

while True:
    #get value
    sounds = Sound.read_u16()
    print(sounds)
    oled.text('Sound:', 0, 0)
    oled.text(str(sounds), 50, 0)
    #Display sound on OLED
    for i in range(10):
        oled.pixel(i, 30, 1)
        oled.pixel(i, 29, 1)
    if sounds > 5000:
        for i in range(10):
            for j in range(4):
                oled.pixel(i+10, 27+j, 1)
    if sounds > 10000:
        for i in range(10):
            for j in range(10):
                oled.pixel(i+20, 21+j, 1)
    if sounds > 20000:
        for i in range(10):
            for j in range(20):
                oled.pixel(i+30, 11+j, 1)
    oled.show()
    oled.fill(0)
    time.sleep(0.1)
```

from pico_car import SSD1306_I2C

Use SSD1306_I2C from pico_car.

import time

The “time” library. This library handles everything to do with time, from measuring it to inserting delays into your program. The unit is seconds.

from machine import Pin, I2C, ADC

The machine library contains all the instructions MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing. Here, the Pin, ADC and I2C libraries are used.

i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)

Set IIC 1 pin to SCL 15, SDA 14, and frequency to 100000.

oled = SSD1306_I2C(128, 32, i2c)

Initialize the size of OLED to 128*32, and pass the previously set IIC parameters into it.

Sound = machine.ADC(27)

Initialize ADC port 27.

oled.text(str(sounds), 50, 0)

Convert the sound value into a string and display it at the 50,0 position of the OLED.

oled.pixel(i, 30, 1)

Display a dot at the position of i, 30, and display it on the OLED through a for loop drawing.

oled.show()

Display the set OLED content.

oled.fill(0)

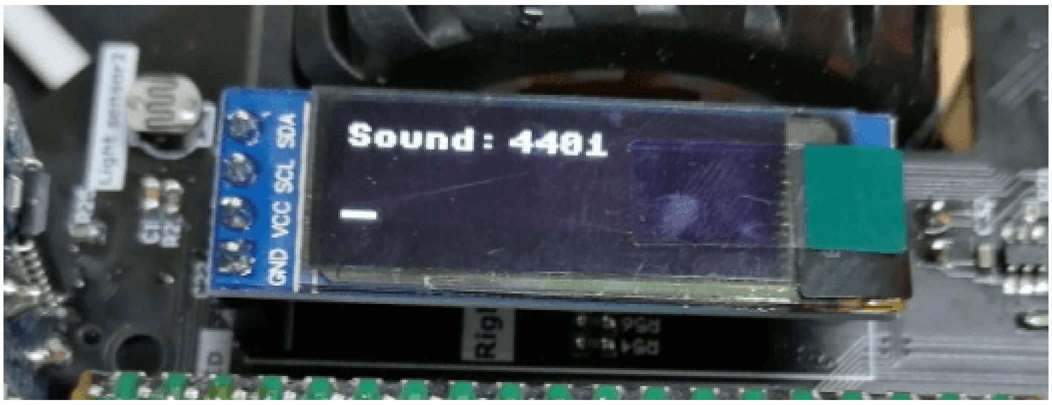
Clear the set content and prepare for the next display.

sounds = Sound.read_u16()

The Sound.read_u16() function is used to detect the value of the sound sensor and assign it to the variable sounds.

Fourth, experimental phenomenon

After the program is downloaded, we can see that the first line of OLED displays the value of Sound, and the second line displays the volume of the sound. Blowing air to the sound sensor, the value will change and the volume will fluctuate. At the same time, the value can also be seen in the Shell.



Note that the sound sensor detection value will be affected by the battery power. It is recommended to test first and then adjust the comparison value. When testing, it is recommended to turn off loads with high power consumption such as motors and RGB lights.