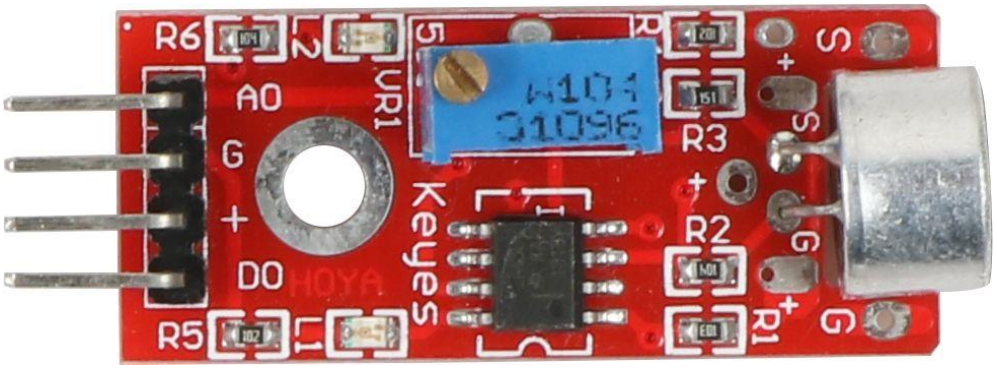
Introduction

The function of the sound sensor module is as a microphone (microphone). It is used to receive sound waves and display vibrating images of sound but it cannot measure the intensity of noise.

The sensor has a built-in condenser electret microphone that is sensitive to sound. The sound wave vibrates the electret film in the microphone which causes the change of the capacitance and generate a small voltage corresponding to the change. Then this voltage is converted into a voltage of 0-5V which is accepted by the data collector through A/D conversion and sent to the main control chip.



Sound Sensor Module

Experimental Purpose

- Understand how the sound sensor work;

Learn to use Raspberry Pi board and sound sensor module to realize the function of sound control LED.

Component List

- Raspberry Pi main board
- Raspberry Pi expansion board
- Cable
- Sound Sensor Module * 1
- LED Module * 1
- Several jumper wires

Wiring

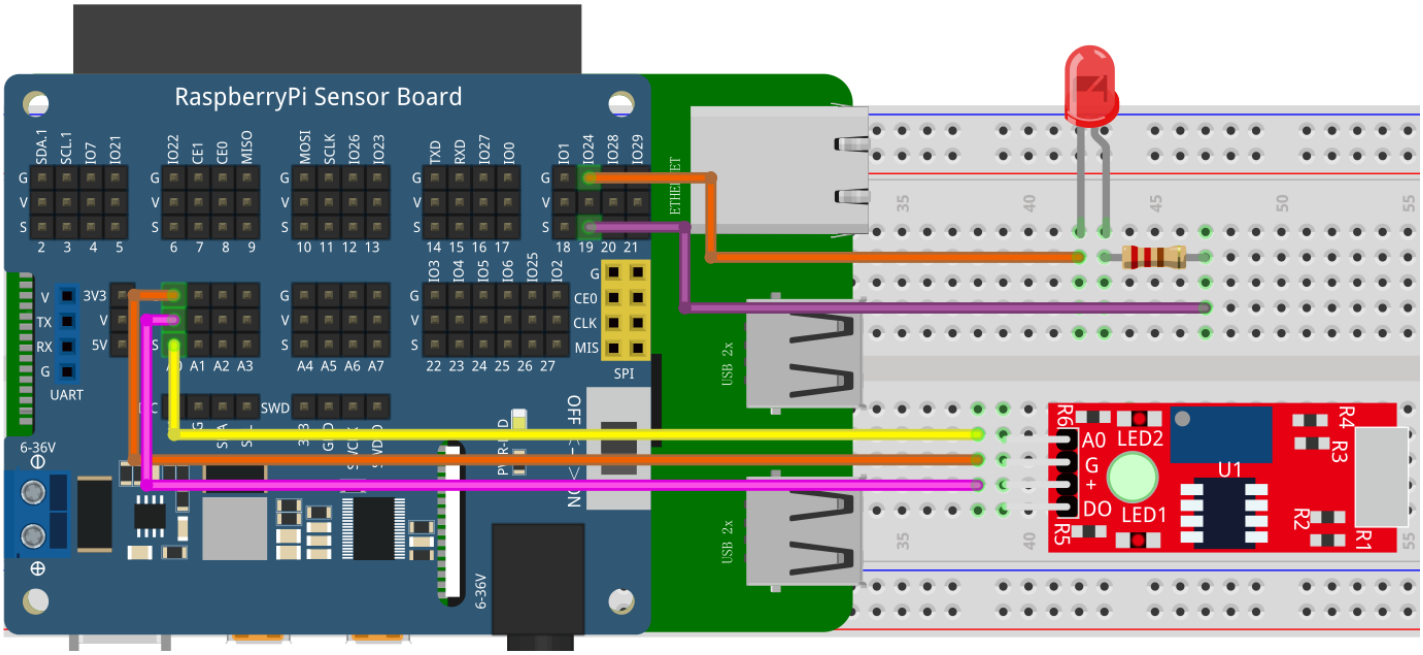
Connect the sound sensor signal AO pin to the Raspberry Pi A0 pin and connect the anode of the LED to the IO24 (wiringPi)/19 (BCM) pin of the Raspberry Pi board to complete the wiring of the entire experiment.

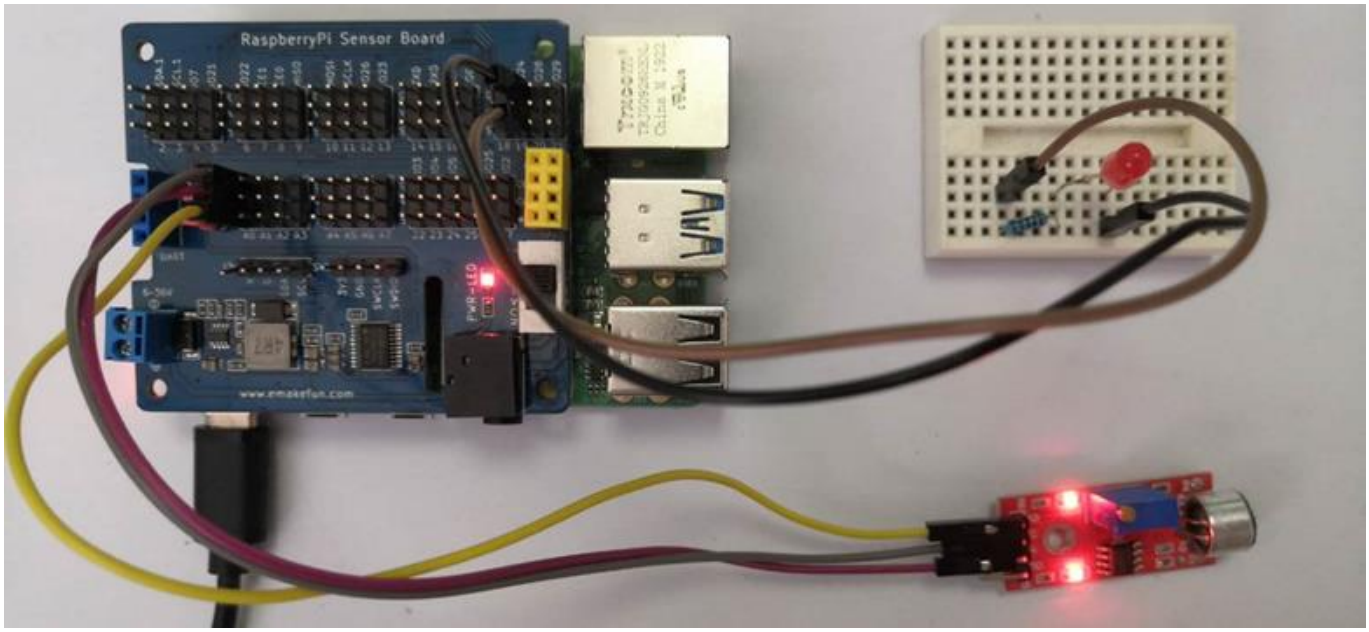
Sound Sensor Module Wiring

Sound Sensor Module	Raspberry Pi
AO(signal pin)	A0
G（GND）	GND
+ （Power Supply）	5V

LED Wiring

LED Module	Raspberry Pi
G（GND）	GND
R（pin）	IO24(wiringPi)/19(BCM)





Experimental Principle

The sound sensor convert the detected sound analog value into the voltage value.

C++ program

```
#include <stdio.h>//Import the basic library
#include <wiringPi.h>//Import the Raspberry Pi WiringPi encoding IO control library
#include <wiringPiI2C.h>//Import the Raspberry Pi WiringPi coding I2C control library

int  LEDPIN = 24; //LED light is connected to IO0 port
int  value = 0 ;
float voltage = 0.0 ;
int main()
{
    wiringPiSetup();
    wiringPiI2CSetup(0x04);
    pinMode(LEDPIN, OUTPUT);
    while(1)
    {
        value =  wiringPiI2CReadReg8(0x04, 0x10);
        if(value >= 100)
        {
            digitalWrite(LEDPIN, HIGH);
            delay(500);
        }else{
            digitalWrite(LEDPIN, LOW);
        }
        printf("%d\n",value); // Print Value
        delay(1000); // Delay for 1 second
    }
}
```

Python program

```
import time
import smbus as smbus
import RPi.GPIO as GPIO

buzzer = 27
ADC=smbus.SMBus(1)#Declare to use I2C 1
GPIO.setmode(GPIO.BCM)
GPIO.setup(buzzer, GPIO.OUT)
```

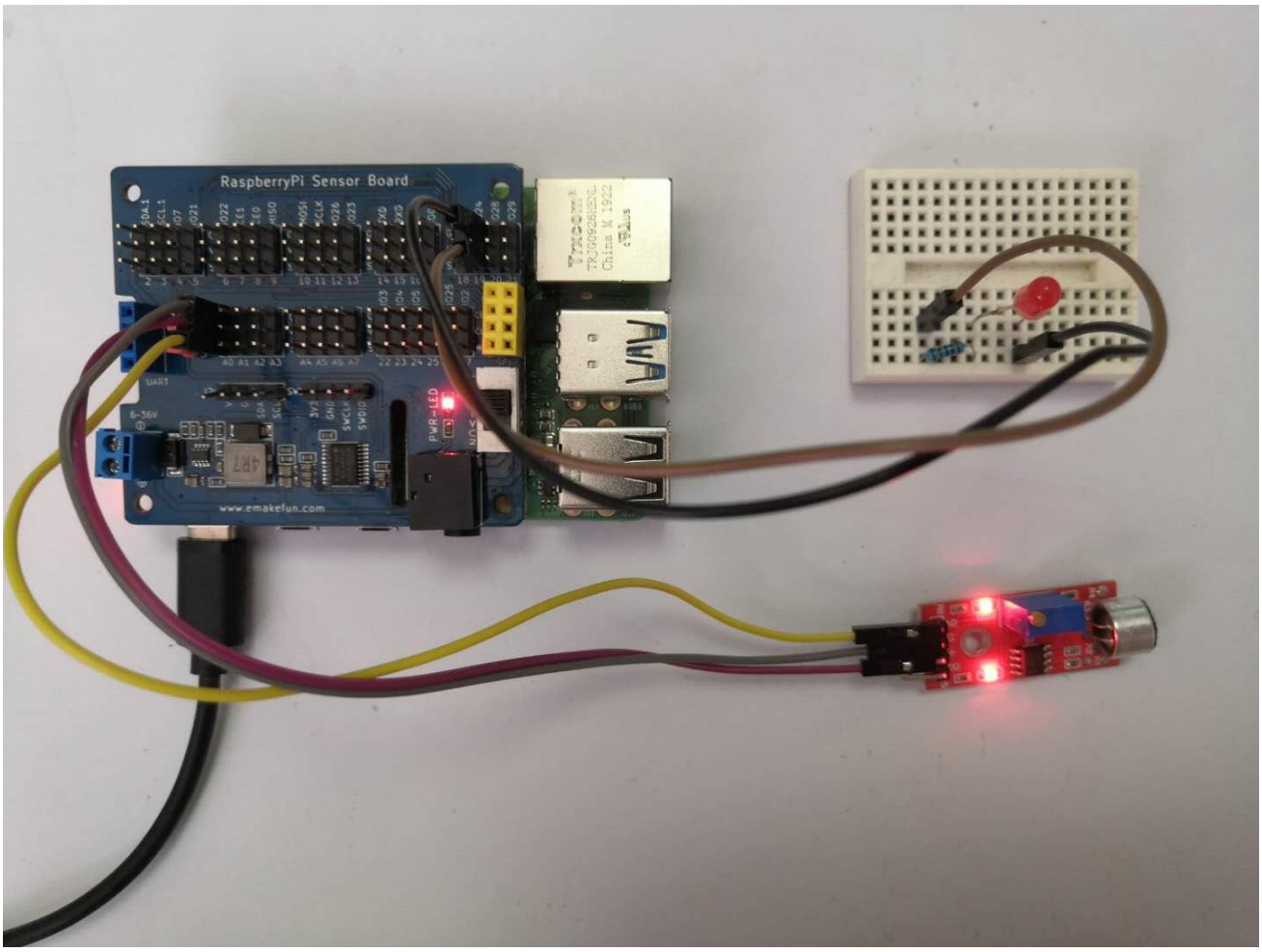
```
while True:
    ADC.write_byte(0x04, 0x10)#Write a byte to the slave
    val = ADC.read_word_data(0x04, 0x10);
    if val <= 550:
        count = count + 1
    else:
        count = 0
    if count >= 2:
        GPIO.output(buzzer, True)
        time.sleep(0.8)
    else:
        GPIO.output(buzzer, False)
    print(val)#Raspberry Pi reads the data returned by the expansion board and prints it
out
    time.sleep(1)#Delay 1 second
```

Java program

```
import com.pi4j.wiringpi.I2C;
import com.pi4j.wiringpi.Gpio;

public class Thermistor {
    static int led_pin = 24, value = 0, fd;

    static {
        Gpio.wiringPiSetup();
        fd = I2C.wiringPiI2CSetup(0x04);
        Gpio.pinMode(led_pin, Gpio.OUTPUT);
    }

    public static void main(String[] args){
        for ( ; ;){
            value =  I2C.wiringPiI2CReadReg16(fd, 0x10);
            if(value > 100) {
                Gpio.digitalWrite(led_pin, Gpio.HIGH);
                Gpio.digitalWrite(Buzzer, Gpio.HIGH);
                Gpio.delay(1000);
            } else {
                Gpio.digitalWrite(led_pin, Gpio.LOW);
                Gpio.digitalWrite(Buzzer, Gpio.LOW);
```

```
            }
        }
    }
}
```

Experimental Effect



While the sound sensor detects that the surrounding sound is greater than the threshold and it turns on the LED otherwise turns off the LED light.