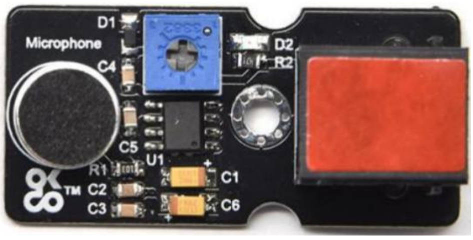
Analog Sound Sensor
(000x0000 Article Number)
(TS2133)

DESIGN THE WORLD

Product Details



This is the TelePort sound sensor, aiming at the detection of the sound intensity. Based on the LM386 amplifier, it can amplify the detected sound. Then the Arduino will sense it and turn sound signals into the the analog signals, that is, the voltage value by feedback can demonstrate the sound intensity. In addition, the blue potentiometer is used to adjust the sound intensity.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Used for detecting the ambient loudness in an environment.
- Make interactive circuits such as a voice-operated switch.
- The sound sensitivity can be adjusted by the potentiometer.

Technical Specifications

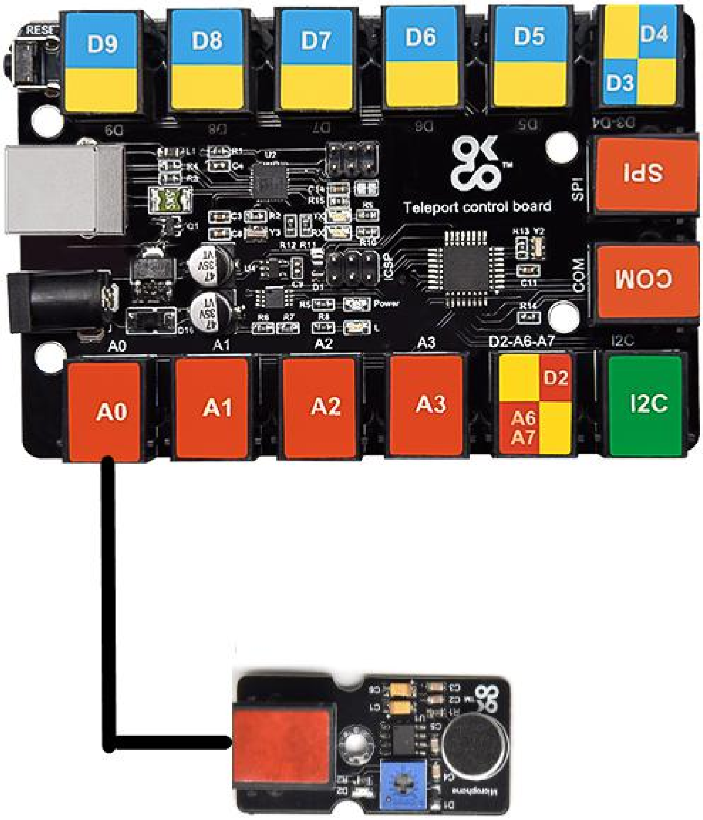
Sensor type	Analog input
Working voltage	3.3V-5V
Operating current	<10mA
Dimensions	42mm*20mm*18mm
Weight	6.0g

Applications

- Sound-activated robots
- Sound-activated switches
- Sound alert alarm

This module is compatible with the TS2180-Raspberry Pi shield, the TS2179-Micro:bit shield and the TS2178-TelePort main board.

➢ Arduino Application



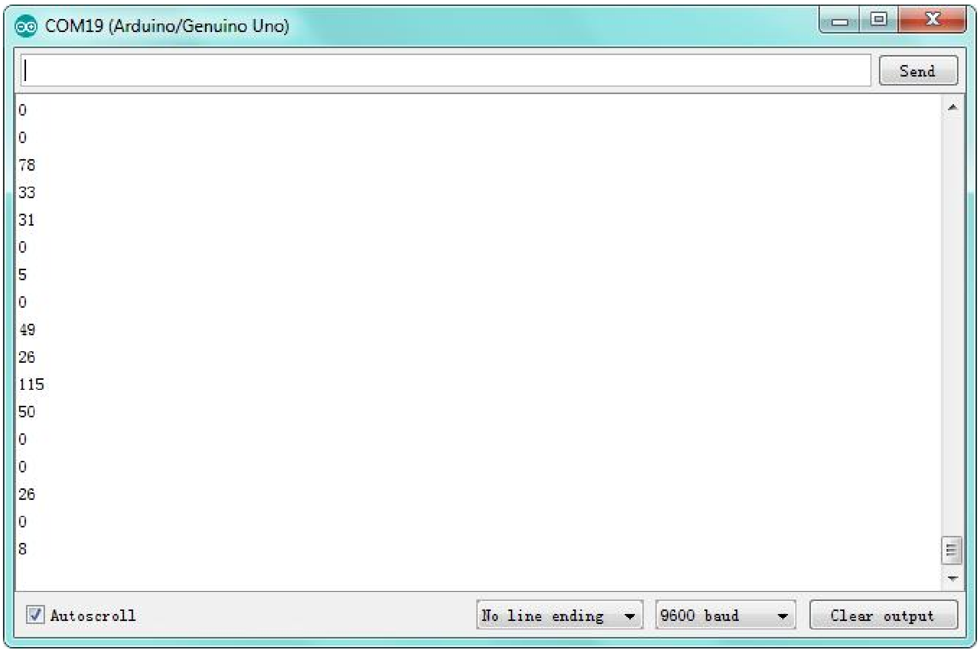
This module is compatible with the TS2178 TelePort control board.

Test Code

```
void setup()
{
Serial.begin(9600);// open serial port, set the baud rate to 9600 bps
}
void loop()
{
int val;
val=analogRead(A0); //connect mic sensor to Analog 0
Serial.println(val,DEC);
delay(100);
}
```

Test Result

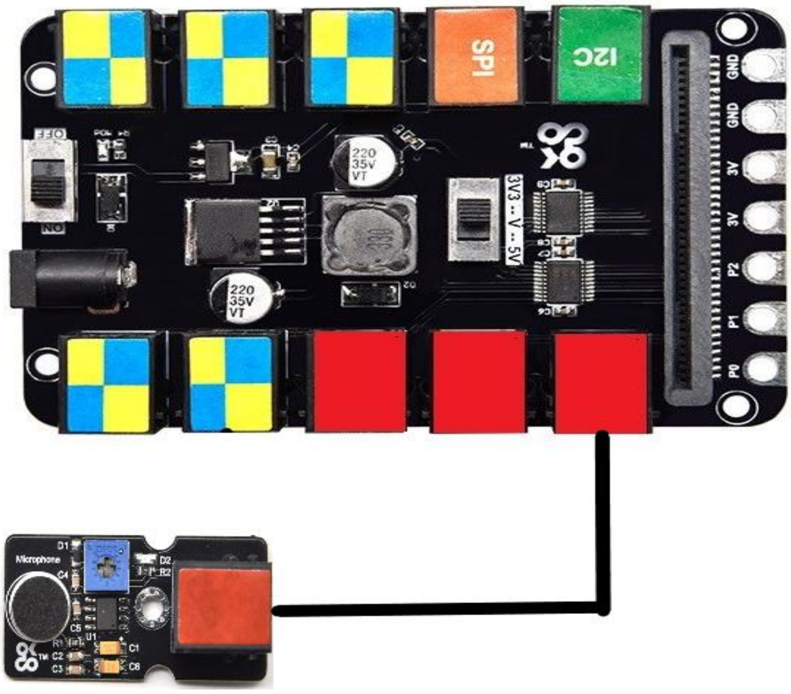
Wire up, upload test code and power it up. Change the sound intensity in the environment, then check the displayed results on the monitor, as shown below;



```
0
0
78
33
31
0
5
0
49
26
115
50
0
0
26
0
8
```

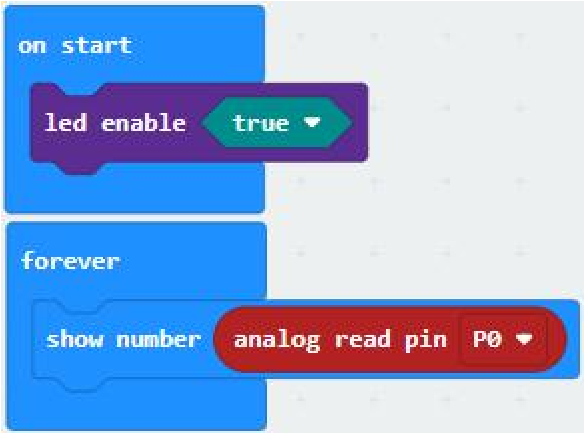
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➢ **Micro:bit Application**



It is compatible with the Micro:bit board and the TS2179 Micro:bit expansion board.

Test Code



```
on start
    led enable  true ▾
forever
    show number  analog read pin  P0 ▾
```

………………..①Run the “on start” block to boot the program

………………..②Turn on the LED matrix of the Micro:bit

………………..③The program is run circularly under the command of “forever” block

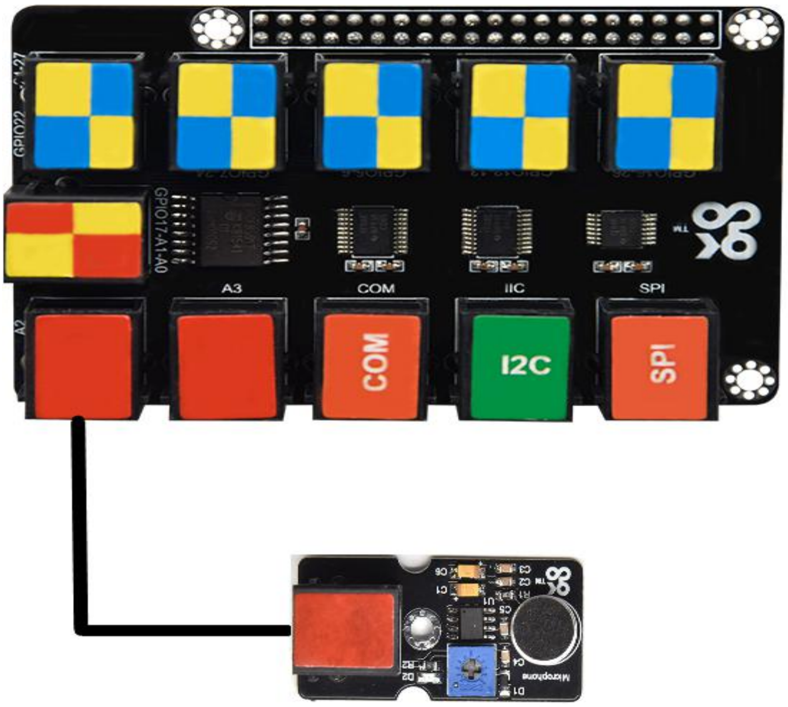
………………..④read the analog value of P0

Test Result

Wire up, upload the test code and power it up. Change the sound intensity in the environment, then check the displayed values on the Micro:bit board,

If you want to know more details about the Micro:bit board and Micro:bit shield, you can refer to TS2179.

- ➢ <u>**Raspberry Pi Application**</u>



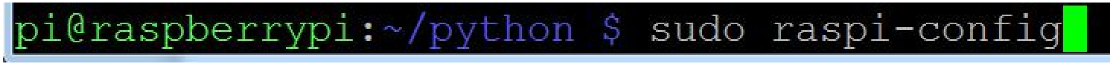
This module is compatible with the Raspberry Pi board and the TS2180 Raspberry Pi shield.

PCF8591 A/D Conversion：

The Raspberry Pi itself does not have AD/DA function; therefore an expansion board with this function is required when connected to external analog sensors. And here we use a PCF8591 A/D converter with I2C communication.

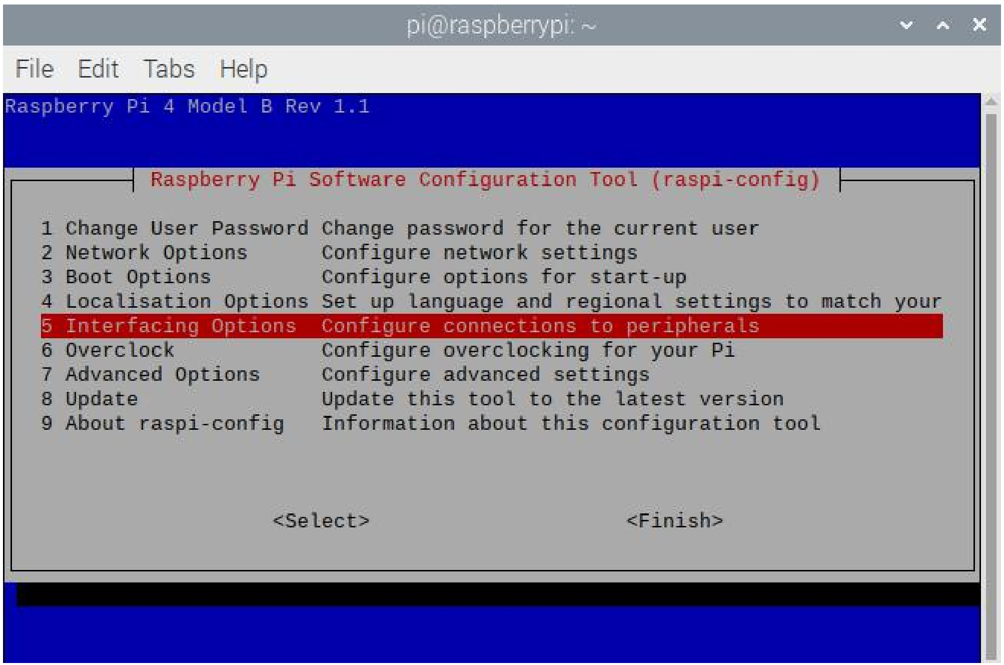
Enable the I2C communication function of the Raspberry Pi as follows:

a. Raspberry Pi does not enable the I2C function by default. Enter sudo raspi-config in the terminal to enter the Raspberry Pi configuration interface.

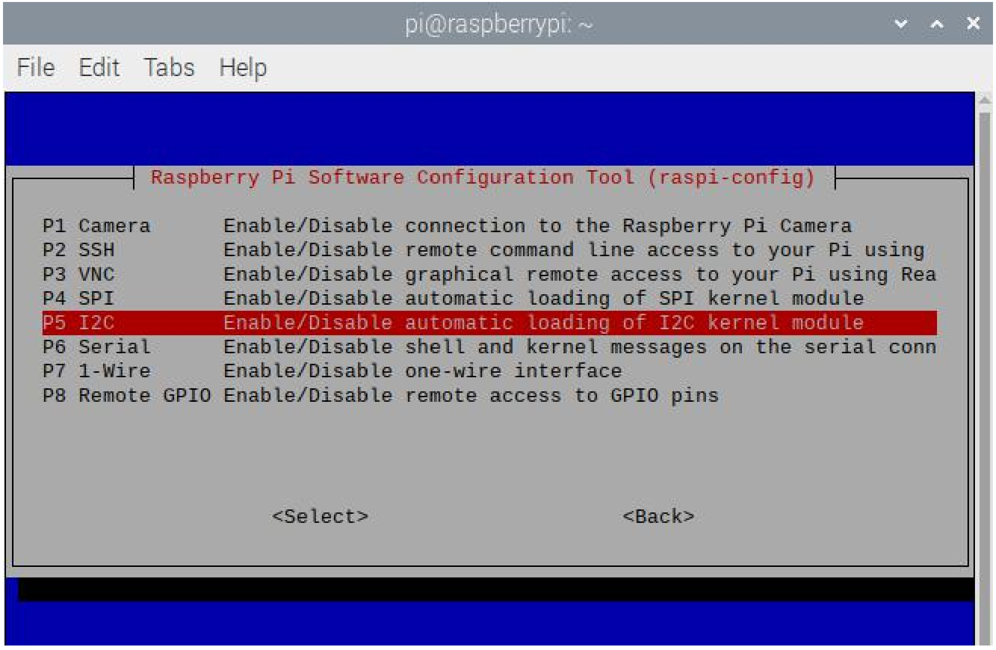


```
pi@raspberrypi:~/python $ sudo raspi-config
```

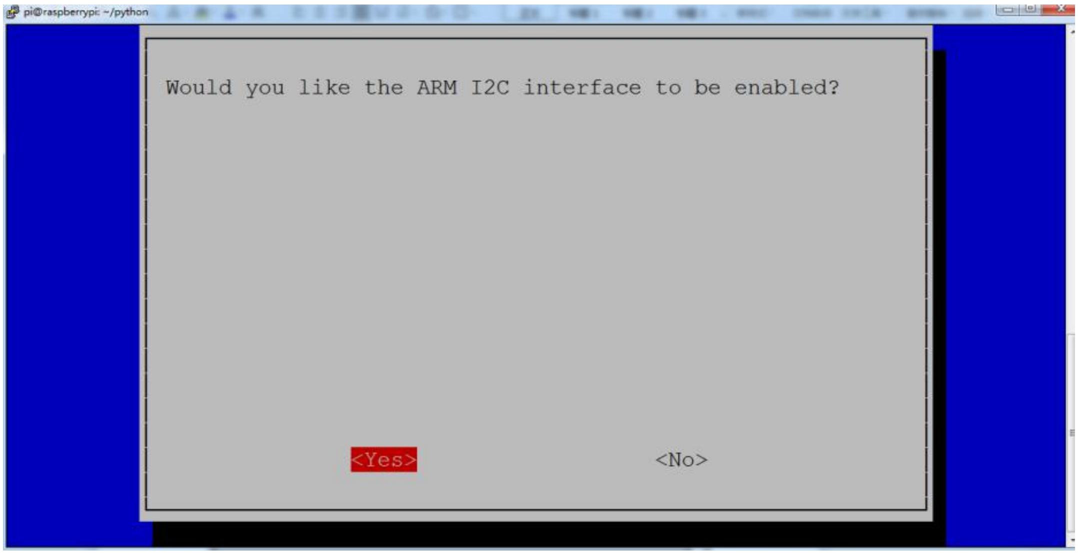
Follow the below instructions to enable the I2C function of Raspberry Pi:(press ←,↑,↓,→ then“Enter”)



```
Raspberry Pi 4 Model B Rev 1.1

          Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                   <Select>                       <Finish>
```



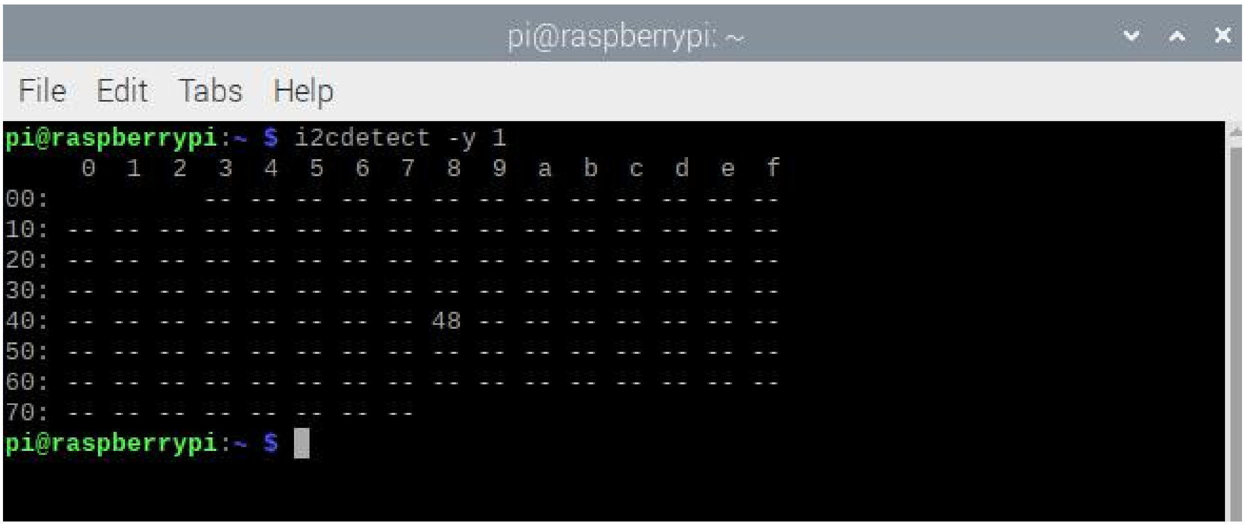
```
          Raspberry Pi Software Configuration Tool (raspi-config)

 P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
 P2 SSH         Enable/Disable remote command line access to your Pi using
 P3 VNC         Enable/Disable graphical remote access to your Pi using Rea
 P4 SPI         Enable/Disable automatic loading of SPI kernel module
 P5 I2C         Enable/Disable automatic loading of I2C kernel module
 P6 Serial      Enable/Disable shell and kernel messages on the serial conn
 P7 1-Wire      Enable/Disable one-wire interface
 P8 Remote GPIO Enable/Disable remote access to GPIO pins

                   <Select>                       <Back>
```



```
Would you like the ARM I2C interface to be enabled?

            <Yes>                    <No>
```

Check the address of the I2C module (PCF8591) connected to the Raspberry Pi, input the command i2cdetect -y 1, and then press **Enter**.

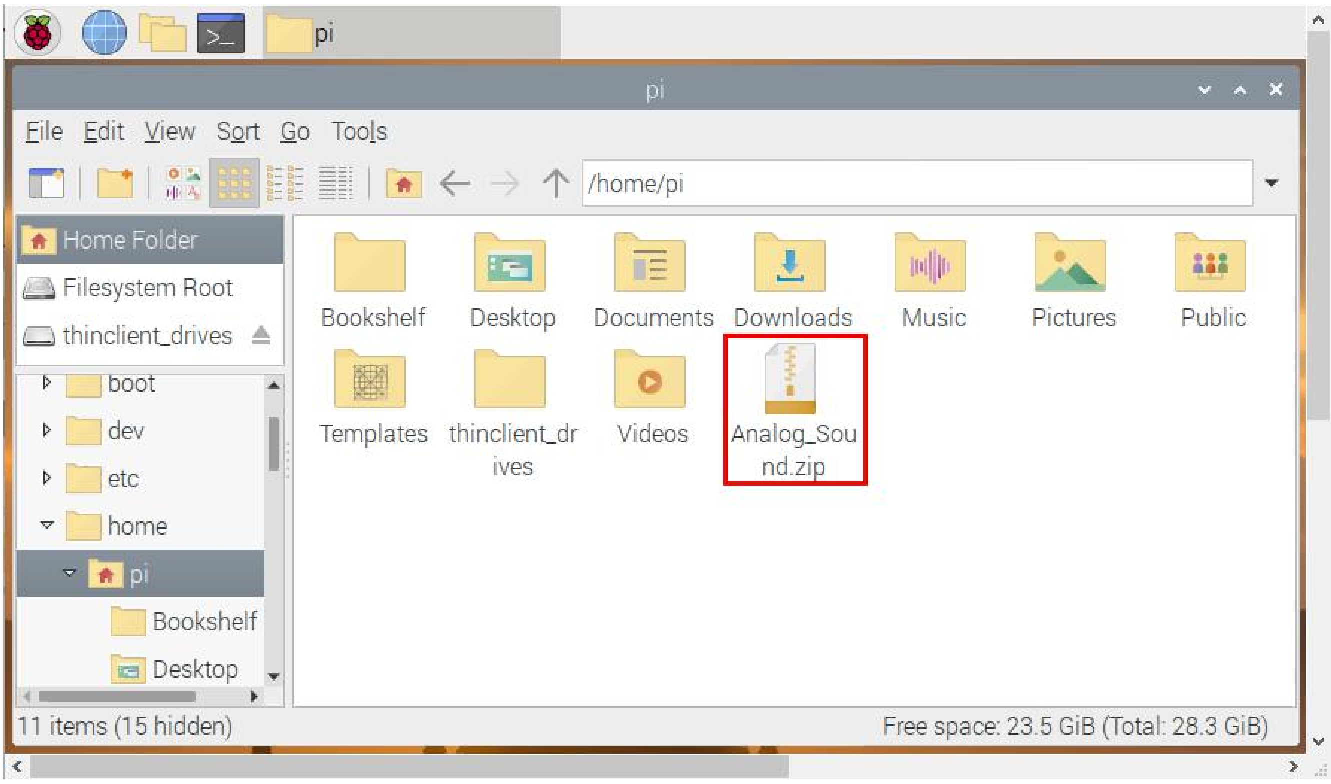
From below picture, the I2C address of PCF8591 is 0x48 .



```
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $ 
```

Copy the test code to Raspberry Pi system to run it

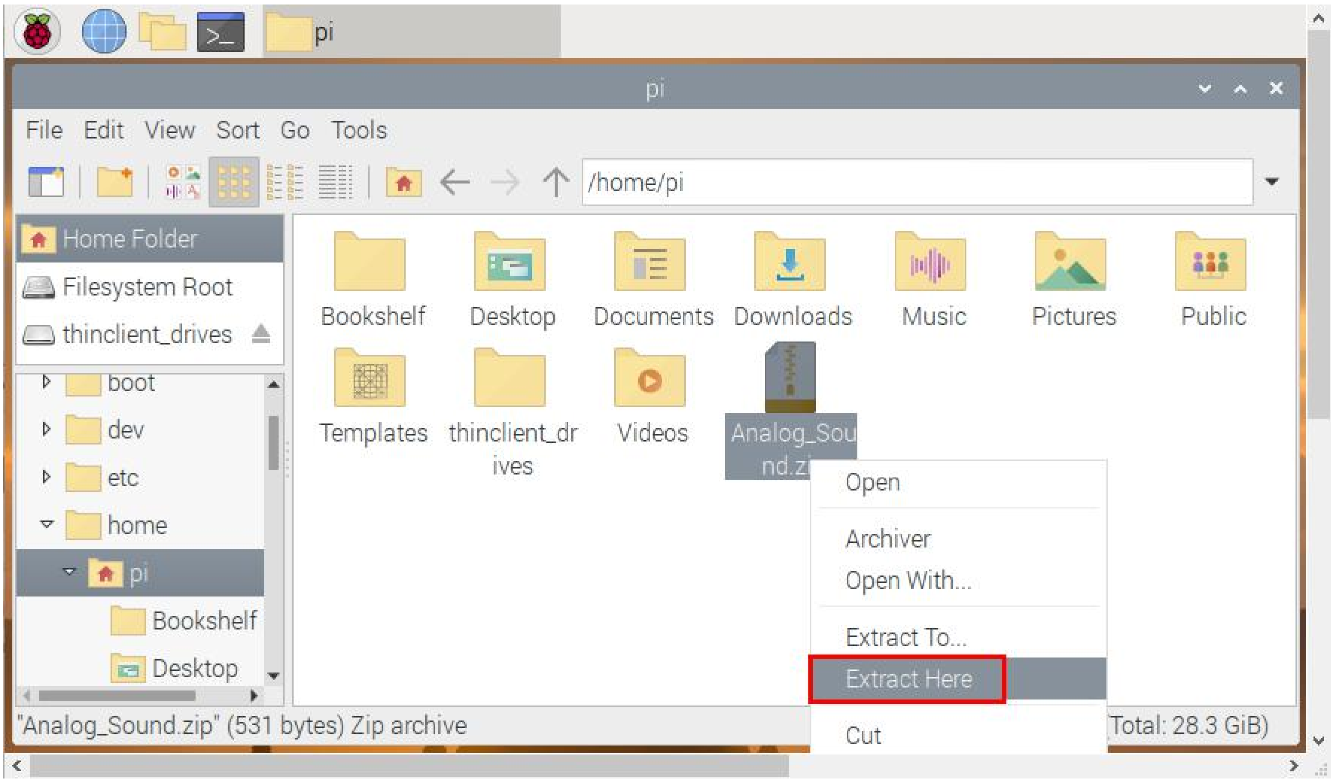
(1) Save the test code in the **pi** folder of Raspberry Pi system. Then place the Analog_Sound.zip file we provide in the **pi** folder, right-click and click **Extract Here.** As shown below:



/home/pi

11 items (15 hidden)

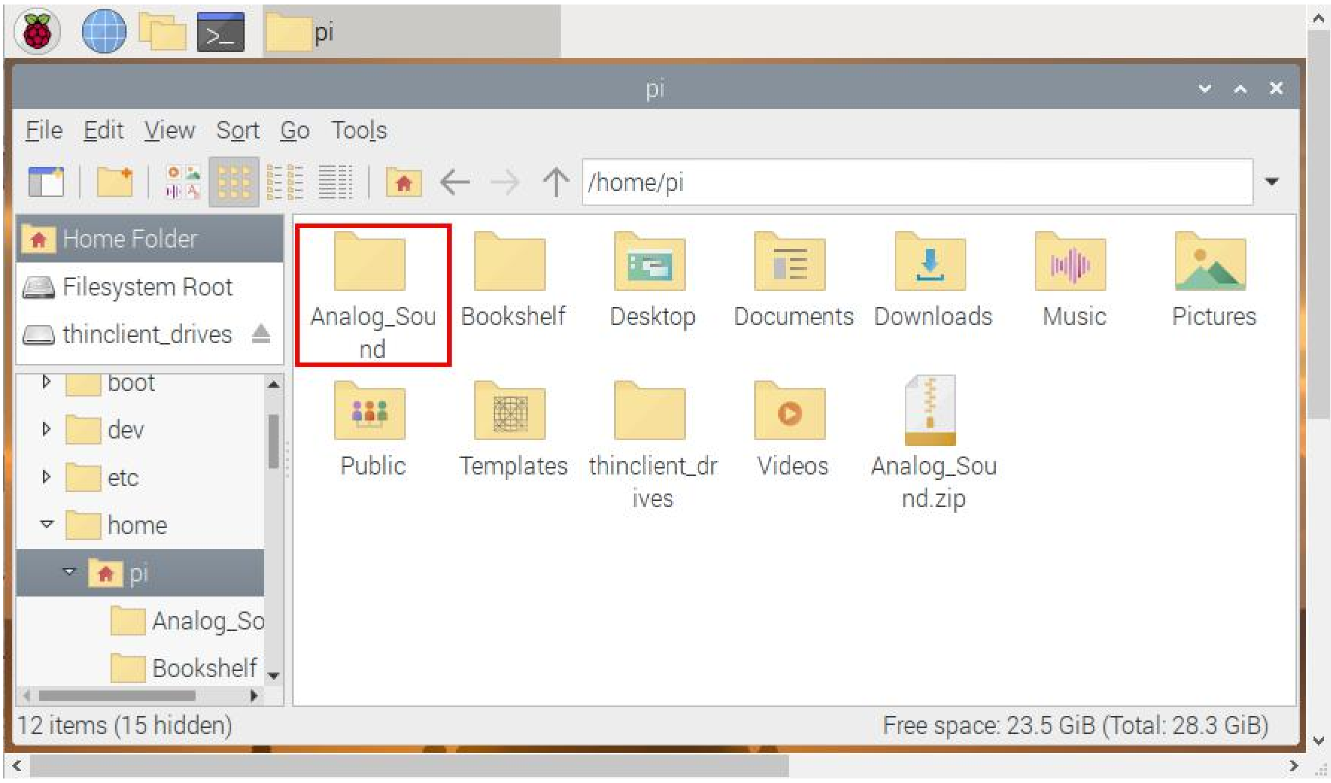
Free space: 23.5 GiB (Total: 28.3 GiB)



/home/pi

"Analog_Sound.zip" (531 bytes) Zip archive

Total: 28.3 GiB)



(2) Compile and run test code：

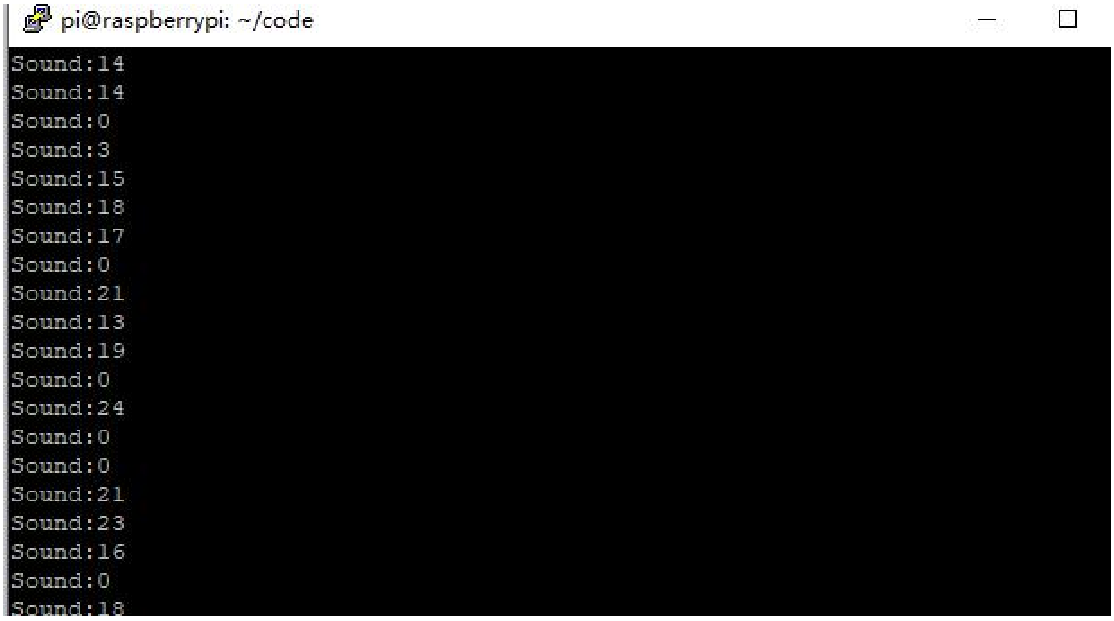
Input the following code and press“Enter”

```
cd /home/pi/Analog_Sound
gcc Analog_Sound.c -o Analog_Sound -lwiringPi
sudo ./Analog_Sound
```

(3) Test Result：

Insert the shield into the Raspberry Pi board. After programming finishes, then the terminal will print the detected sound analog values.

Note: press Ctrl + C to exit code running



```
Sound:14
Sound:14
Sound:0
Sound:3
Sound:15
Sound:18
Sound:17
Sound:0
Sound:21
Sound:13
Sound:19
Sound:0
Sound:24
Sound:0
Sound:0
Sound:21
Sound:23
Sound:16
Sound:0
Sound:18
```

Test Code

File name: Analog_Sound.c

```
#include <wiringPi.h>
#include <pcf8591.h>
#include <stdio.h>

#define Address 0x48
#define BASE 64
#define A0 BASE+0
#define A1 BASE+1
#define A2 BASE+2
#define A3 BASE+3

int main(void){
        wiringPiSetup();
        pcf8591Setup(BASE,Address);
        int value;

        while(1)
        {
                value=analogRead(A2);
                printf("Sound:%d\n",value);
                delay(100);
        }
}
```

If you want to know how to utilize Raspberry Pi and the Raspberry Pi shield, you can refer to TS2180.