

MQ-135 Air Quality Sensor (000x0000 Article Number) (TS2157)



Product Details

MQ135 sensor uses SnO₂ sensitive material on the ground that SnO₂ has low electrical conductivity in the clean air.

When surrounded by the polluted air, the electrical conductivity of MQ135 sensor will increase with the increase of pollutants, and changes of electrical conductivity can be converted to corresponding output signal.

You can rotate the potentiometer on the sensor to adjust the sensitivity.

MQ135 has a high sensitivity to ammonia, sulfide, benzene vapor, smoke and other harmful gases.

Note:

The sensor has a potentiometer which can adjust the threshold value. When you rotate the potentiometer clockwise, the threshold value will increase; in counter-clockwise way, the threshold value will get reduced.



The sensor can't output stable data simultaneously ; therefore, we need to wait for about one minute.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Suitable for aromatic compound detection including: ammonia (NH₃), NO_x, benzene, smoke, CO₂ and other gases.
- Long stable service life with reliable fast response and recovery features.

Technical Specifications

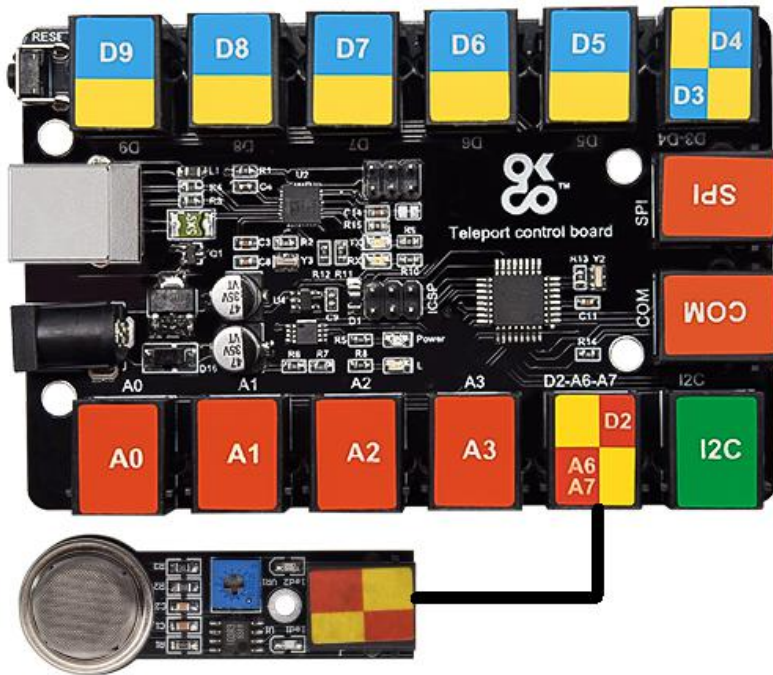
Sensor type	Digital and Analog
Working voltage	5V
Detect/Measure NH ₃ , NO _x , alcohol, Benzene, smoke, CO ₂ , etc.	
Analog output voltage	0V to 5V
Digital output voltage	0V or 5V (TTL Logic)
Preheat duration 20 seconds	
Dimensions	56mm*20mm*18mm
Weight	9.2g

Applications

- Air quality detector
- Environment monitoring

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.

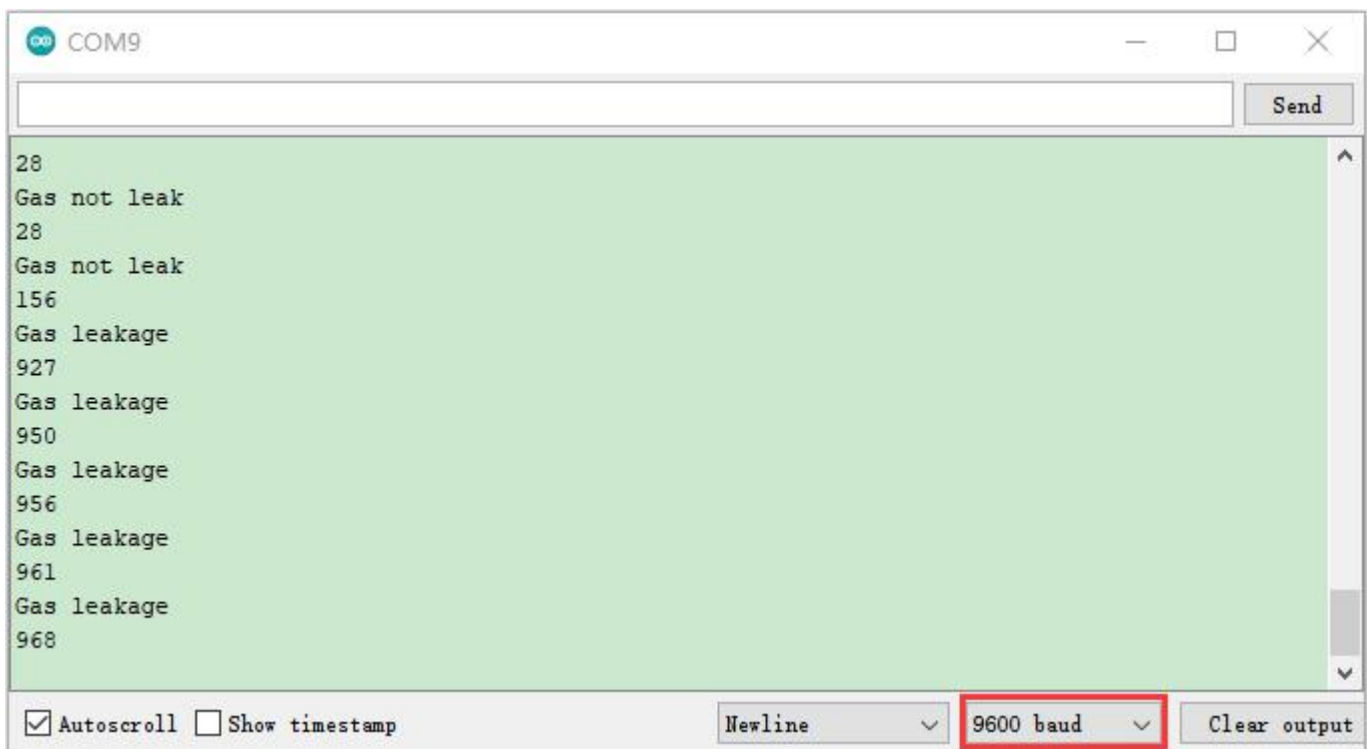
Test Code

```
int gas_din=2;
int gas_ain=A7;
int led=13;
int ad_value;
void setup()
{
  pinMode(led,OUTPUT);
  pinMode(gas_din,INPUT);
  pinMode(gas_ain,INPUT);
  Serial.begin(9600);
}
void loop()
{
  ad_value=analogRead(gas_ain);
  if(digitalRead(gas_din)==LOW)
  {
    digitalWrite(led,HIGH);
  }
}
```

```
Serial.println("Gas leakage");
Serial.println(ad_value);
}
else
{
  digitalWrite(led,LOW);
  Serial.println("Gas not leak");
  Serial.println(ad_value);
}
delay(500);
}
```

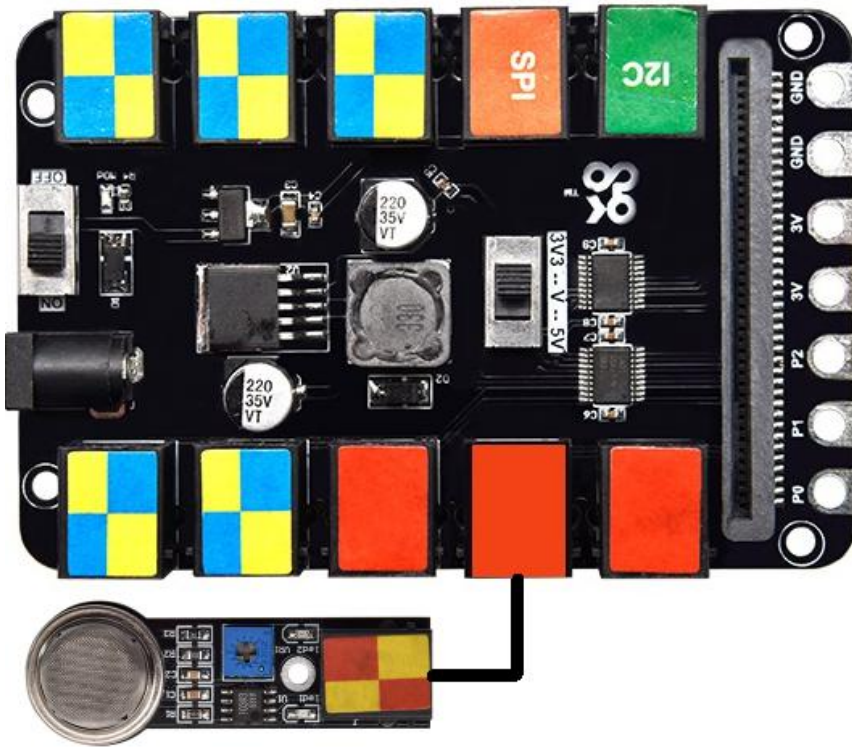
Test Result

Wire up, upload code, power it up, open serial monitor and set baud rate to 9600. When the MQ-135 air quality sensor detects the poisonous gas, the serial monitor will show "Gas leakage", the analog value will get larger and D13 on the module will light up; on the contrary, the serial monitor will show "Gas not leak" and D13 will be off.



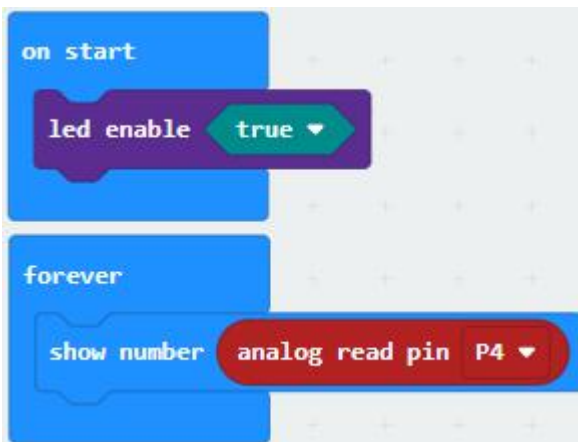
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➤ Micro:bit Application



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

Test Code



-①Run the “on start” block to boot the program
-②Open the LED matrix of the Micro:bit
-③The program is run circularly under the command of “forever” block
-④The Micro:bit shows the analog value of combustible gases detected by the MQ-135 air quality sensor

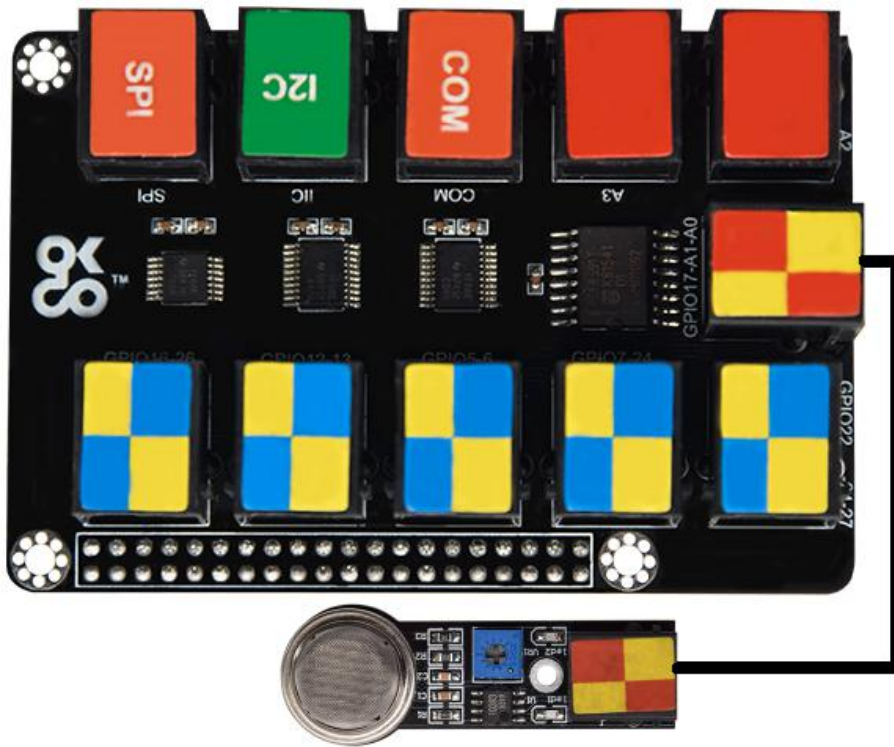
Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn the DIP switch to 5V and slide another switch to ON mode , upload test code, and power it up.

When the sensor detects the poisonous gas, the Micro:bit will show the detected analog value.

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

➤ Raspberry Pi Application



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.

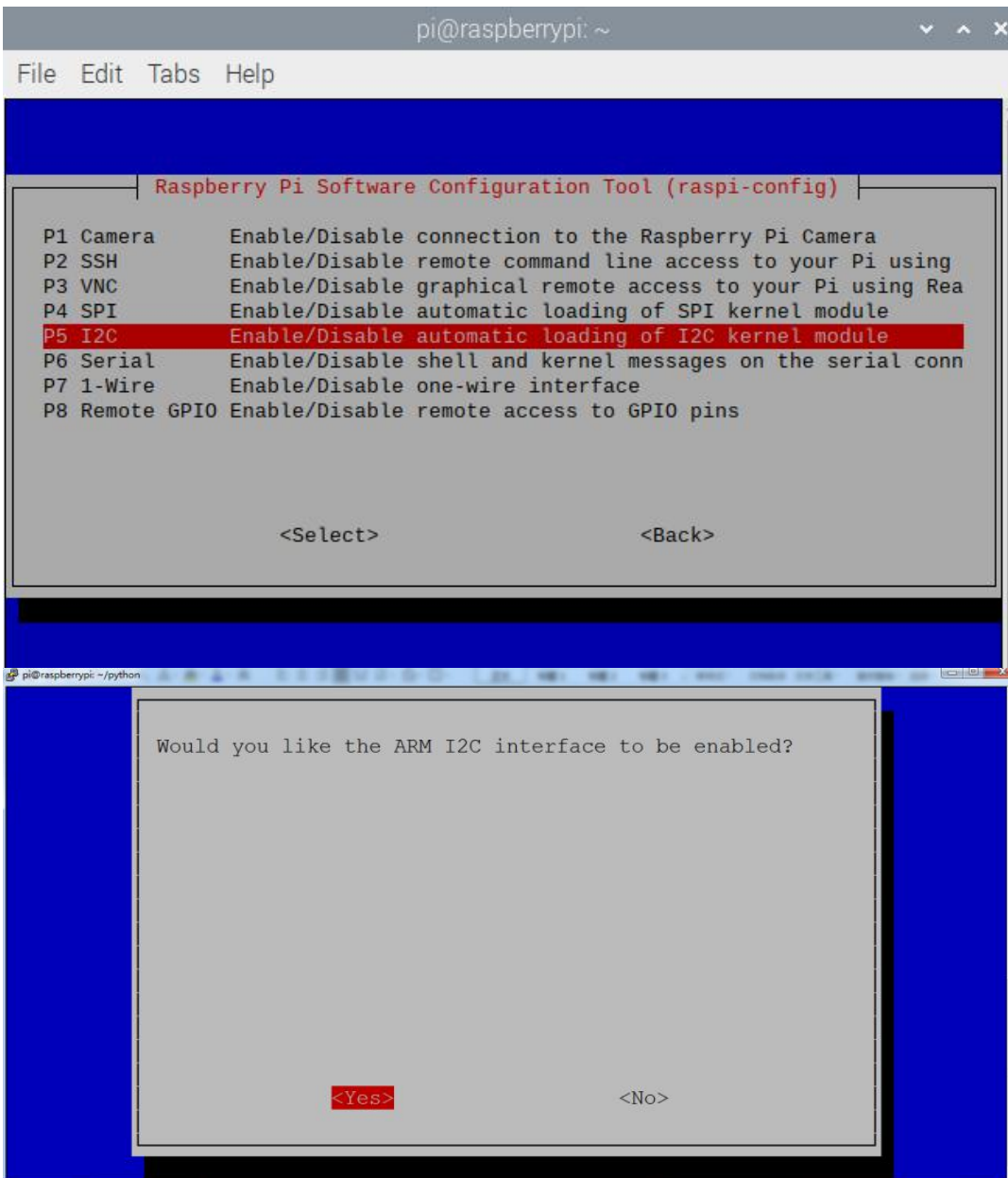
Enable the I2C communication function of the Raspberry Pi as follows:

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
```

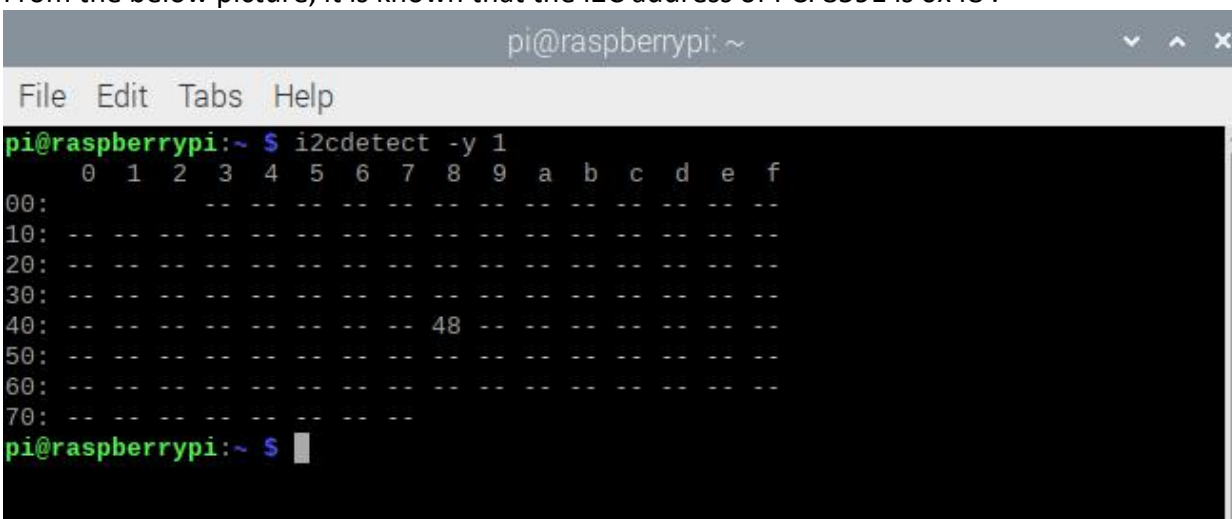
Follow the below instructions to enable the I2C function of Raspberry Pi:(press ←,↑,↓,→ then“Enter”)

```
pi@raspberrypi: ~  
File Edit Tabs Help  
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>
```



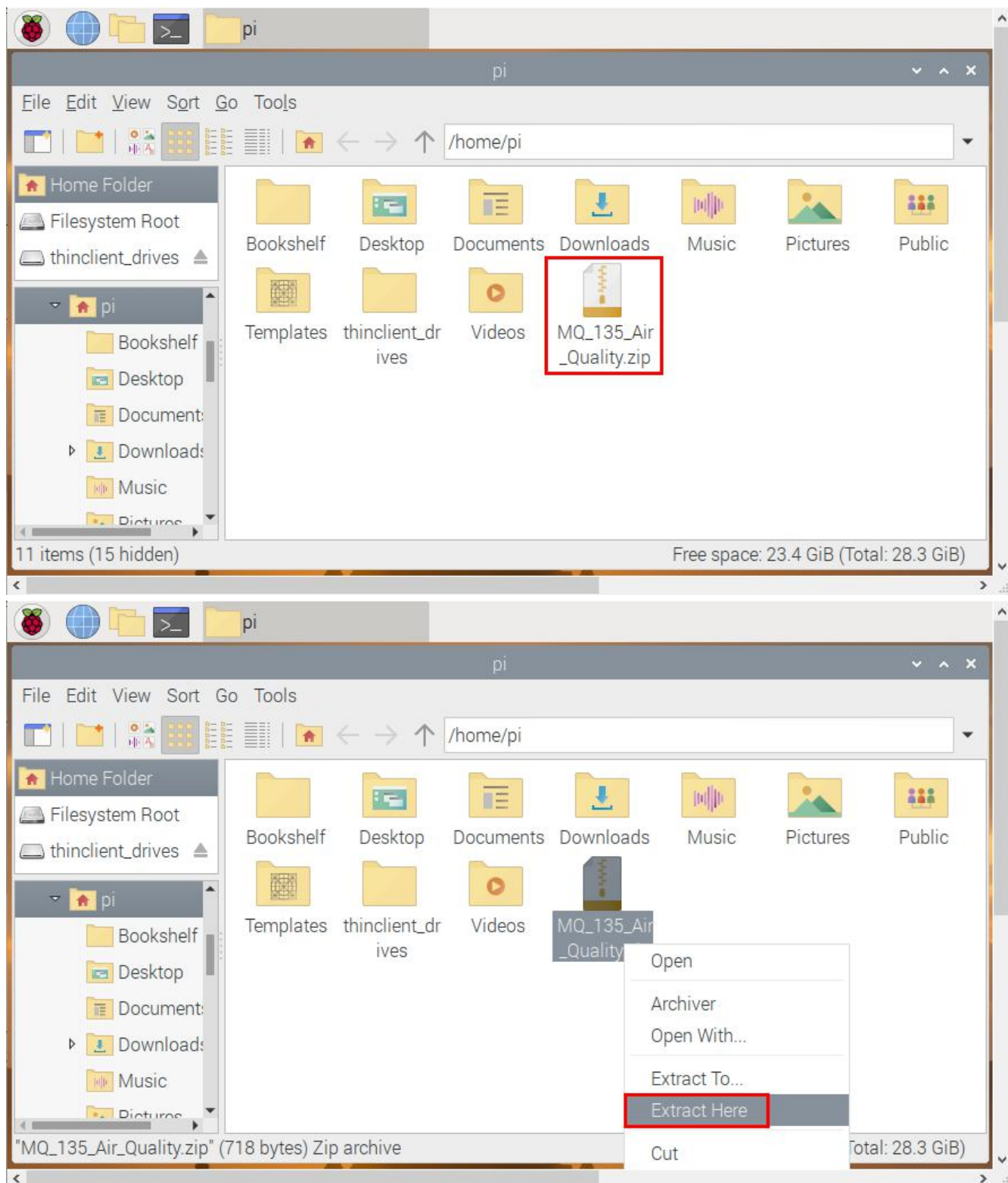
Check the address of the I2C module (PCF8591) connected to the Raspberry Pi, enter the command `i2cdetect -y 1`, and then press **Enter**.

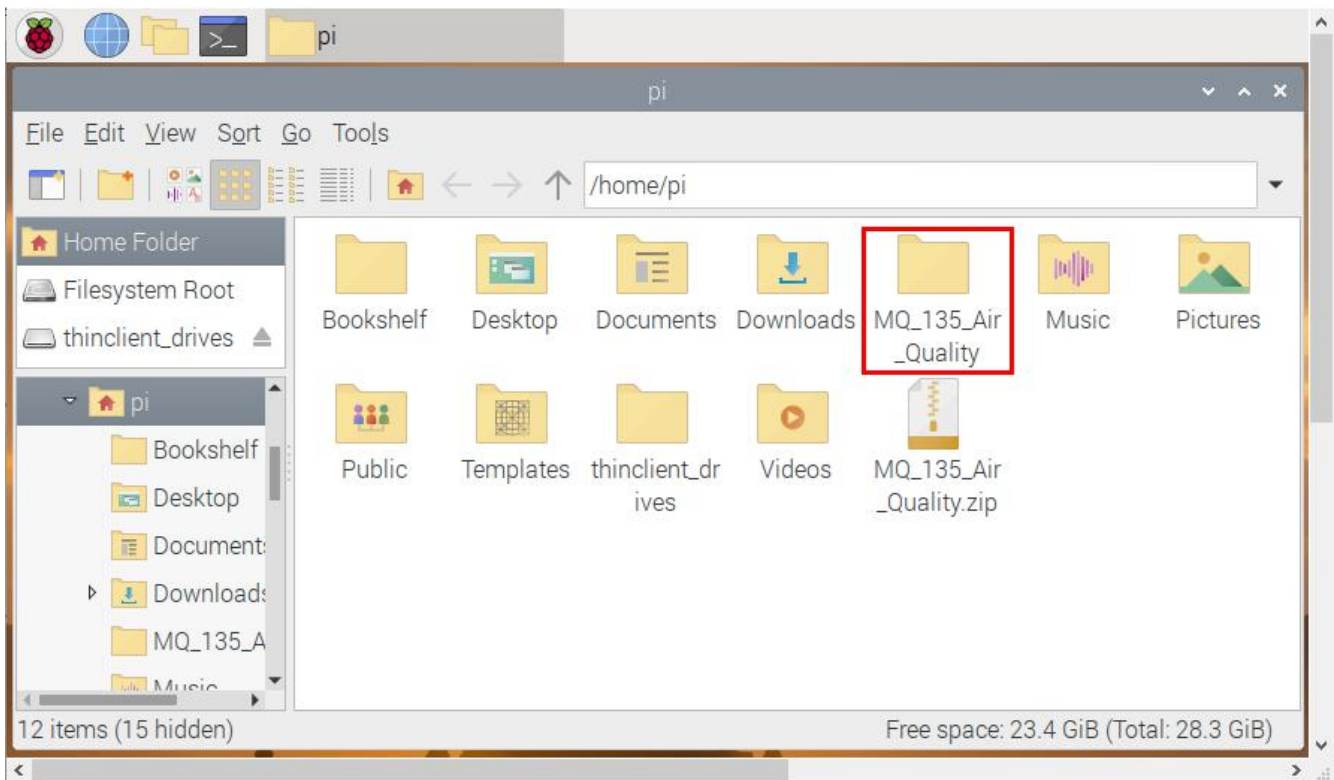
From the below picture, it is known that the I2C address of PCF8591 is 0x48 .



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 **MQ_135_Air_Quality.zip** file we provide in the **pi** folder, right-click and click **Extract Here**. As shown below:





(2) Compile and run test code :

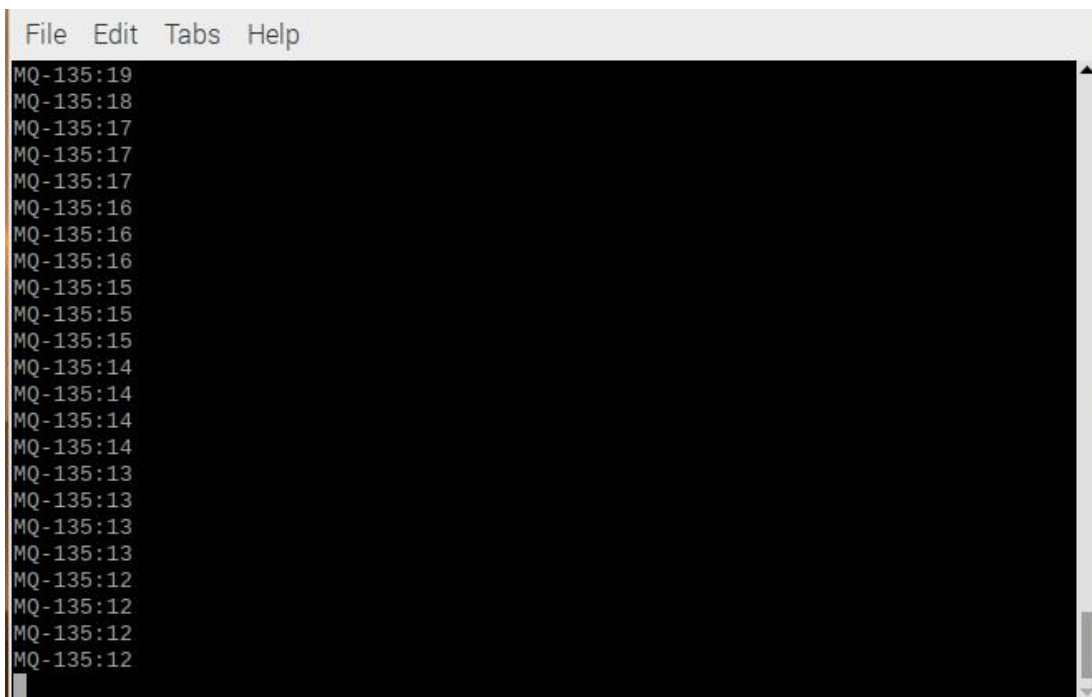
Input the following code and press“Enter”

```
cd /home/pi/MQ_135_Air_Quality
gcc MQ_135_Air_Quality.c -o MQ_135_Air_Quality -lwiringPi
sudo ./MQ_135_Air_Quality
```

(3) Test Result :

Insert the shield into the Raspberry Pi board. After programming finishes, then the terminal will display the analog value detected by the MQ-135 air quality sensor.

Note: press Ctrl + C to exit code running



Test Code

File name: [MQ_135_Air_Quality.c](#)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <wiringPi.h>
#include <pcf8591.h> //pcf8591 library

#define Address 0x48 //i2c address
#define BASE 64 //DAC write address
#define A0 BASE+0 //A0 analogRead address
#define A1 BASE+1 //A1 analogRead address
#define A2 BASE+2
#define A3 BASE+3

int main(void)
{
    unsigned char dat;
    wiringPiSetup();
    pcf8591Setup(BASE,Address); //Initialize the pcf8591
    if (wiringPiSetup() == -1){
        exit(1);
    }

    while(1){

        dat=analogRead(A0); //read the ADC value of channel 0
        printf("MQ-135:%d\n",dat); //The terminal prints the simulated value
        delay(100);
    }
    return 0;
}
```

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

END