

Water Level Sensor (000x0000 Article Number) (TS2135)



Product Details

This is the TelePort water level sensor. It is not only an analog sensor but also a water level recognition sensor. The water level can be detected by the exposed parallel wires. The larger the immersed coverage of the sensor, the larger the detected value. It outputs the analog value by converting the water content into the analog signals.



Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Detect the presence of water from single rain drops to level measurements in a container.
- Analog output direct to any microcontroller.

Technical Specifications

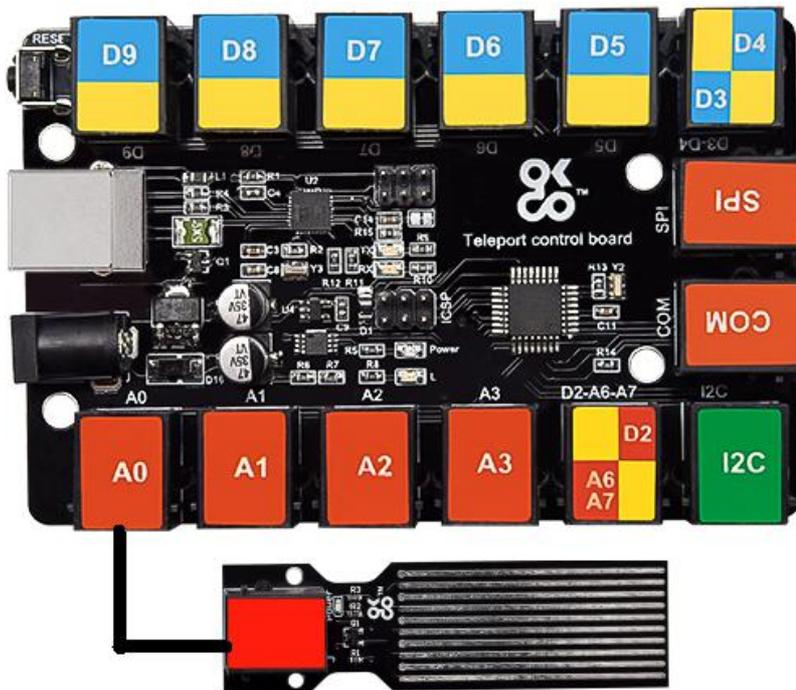
Sensor type	Analog input
Operating voltage	3.3V-5V
Operating current	<20mA
Detection area	40mm x16mm
Operating temperature	10%~90% without condensation

Applications

- Water level alarm
- Water level detection for fish tanks
- Soilless culture

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

➤ Arduino Application



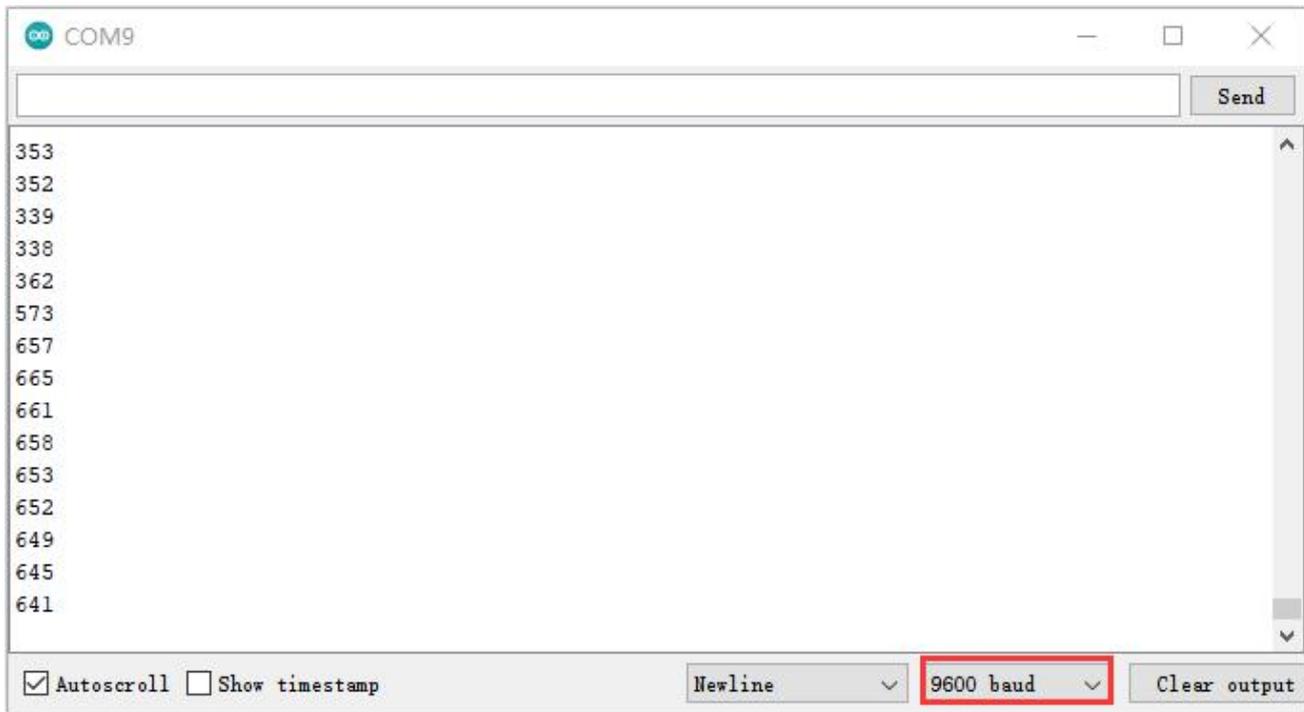
This module is compatible with the TS2178 TelePort control board.

Test Code

```
int analogPin = A0; //connect water sensor to analog interface A0
int led = 13; //LED to digital interface 13
int val = 0; //define the initial value of variable 'val' as 0
int data = 0; //define the initial value of variable 'data' as 0
void setup()
{
  pinMode(led, OUTPUT); //define led as output pin
  Serial.begin(9600); //set baud rate at 9600
}
void loop()
{
  val = analogRead(analogPin); //read and assign analog value to variable 'val'
  if(val>700){ //decide whether variable 'val' is over 700
    digitalWrite(led,HIGH); //turn on LED when variable 'val' is over 700
  }
  else{
    digitalWrite(led,LOW); //turn off LED when variable 'val' is under 700
  }
  data = val; //variable 'val' assigns value to variable 'data'
  Serial.println(data); //print variable 'data' by Serial.print
  delay(100);
}
```

Test Result

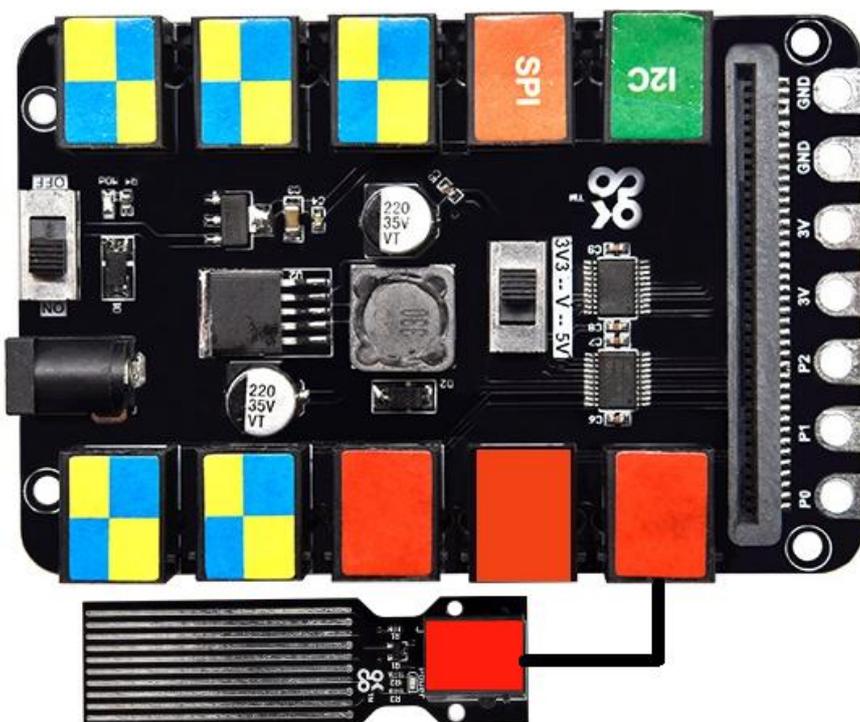
Wire up, upload test code, power it up, open serial monitor and set baud rate to 9600. Then the LED indicator on the water level sensor will be on. You will spot that the larger the immersed coverage of the sensor, the larger the detected value.



Meanwhile, when the value is more than 700, LED will be on; if not, it will be off.

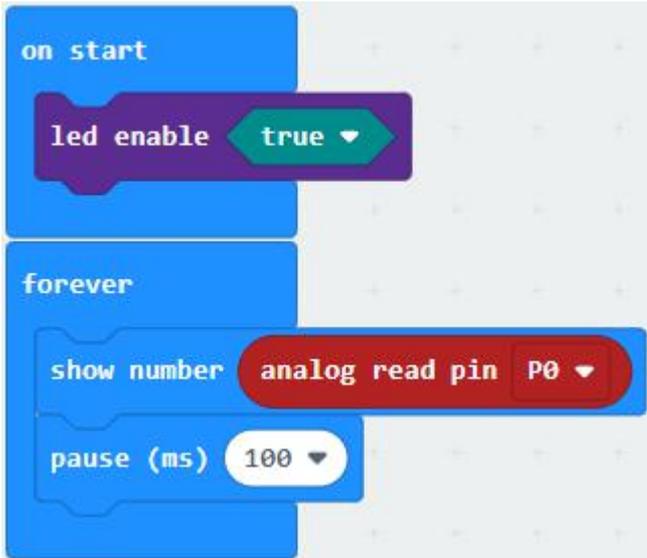
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➤ Micro:bit Application



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

Test Code



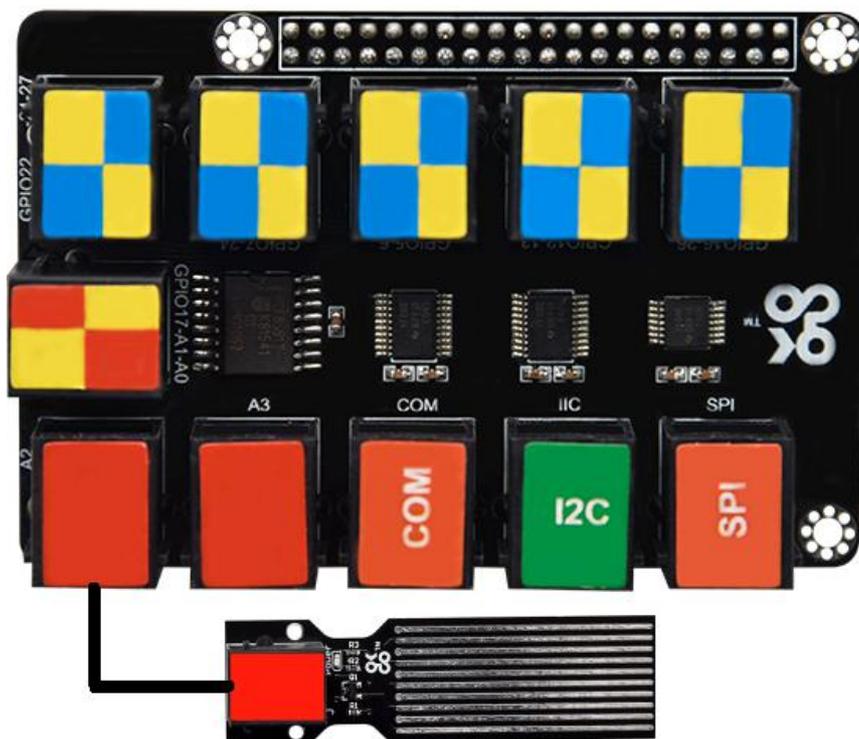
-①Run the “on start” block to boot the program
-②Open the LED matrix of the Micro:bit
-③The program is run circularly under the command of “forever” block
-④Micro:bit shows the detected analog values by the photoresistor
-⑤delay in 100ms

Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn DIP switch to 3V3, upload test code and power it up. Then Micro:bit will show the analog values detected by the water level sensor.

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:

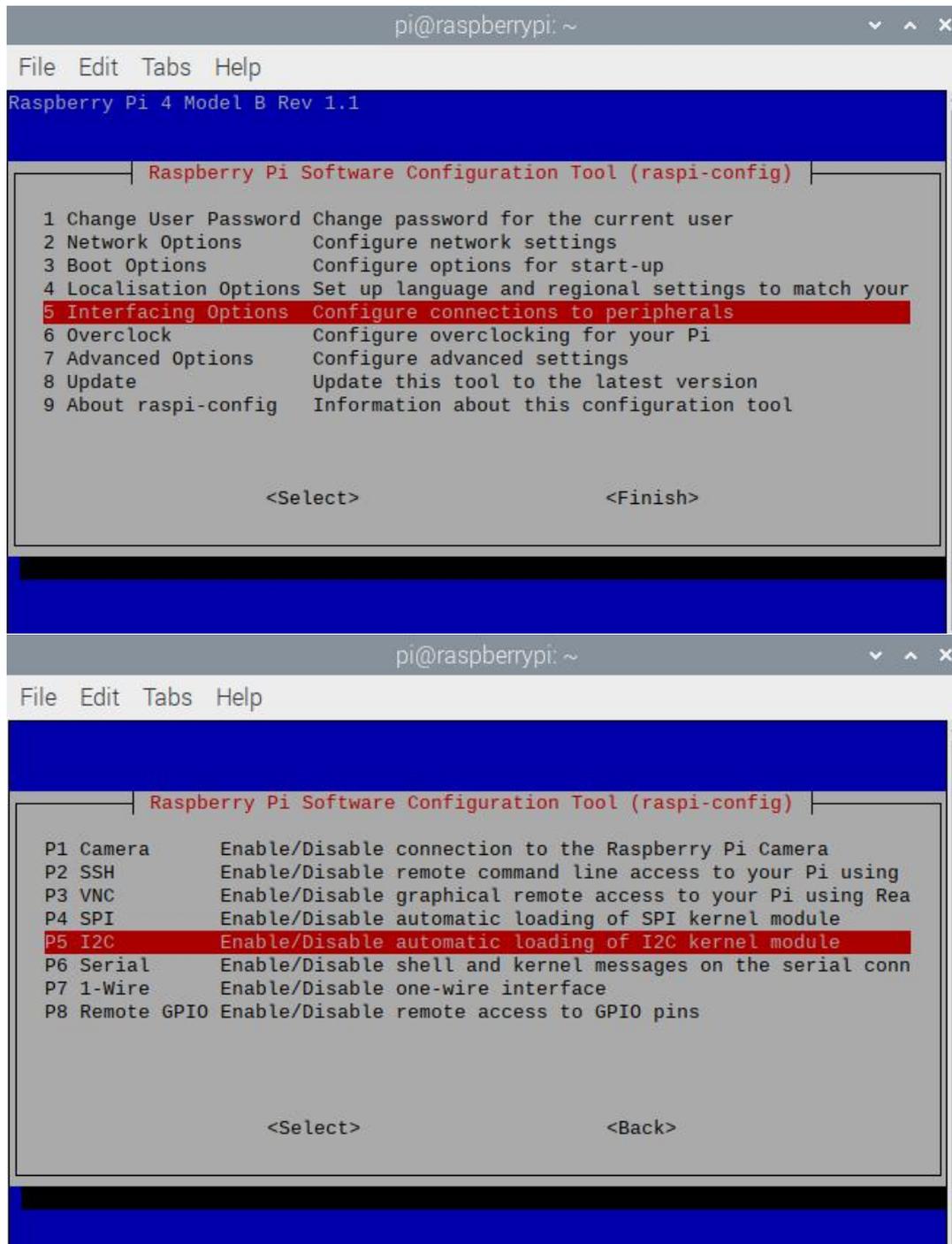
The Raspberry Pi itself does not have AD/DA function; therefore an expansion board with this function is required when connected to external analog sensors. And here we use a PCF8591 A/D converter with I2C communication.

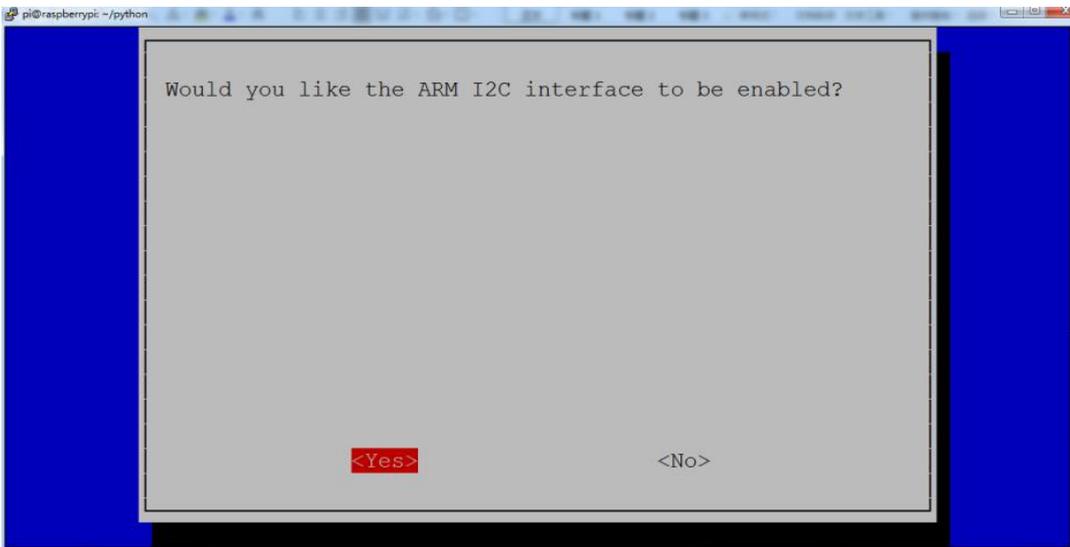
Enable the I2C communication function of the Raspberry Pi as follows:

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
```

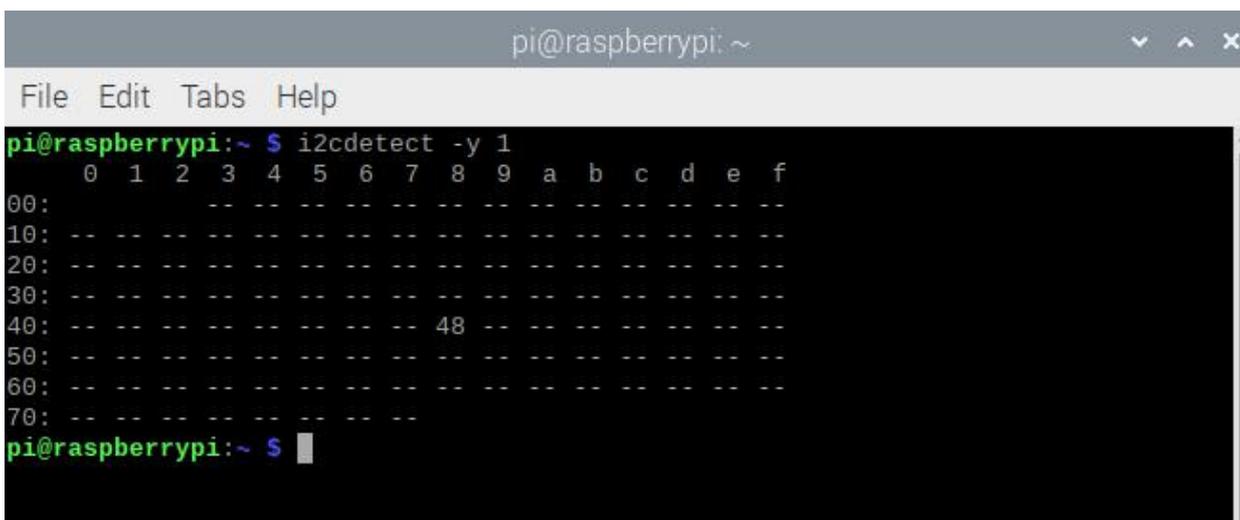
b. Follow the below instructions to enable the I2C function of Raspberry Pi:(press ←,↑,↓,→ then“Enter”)





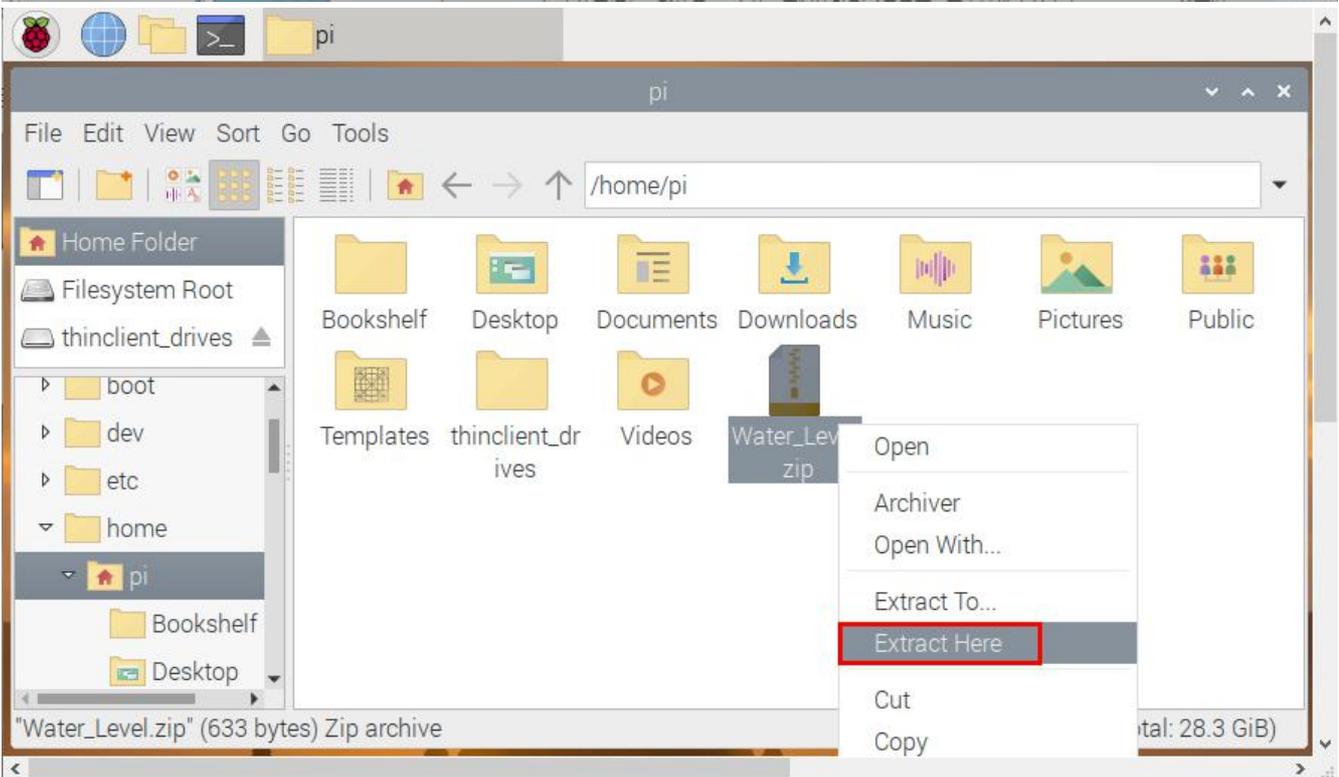
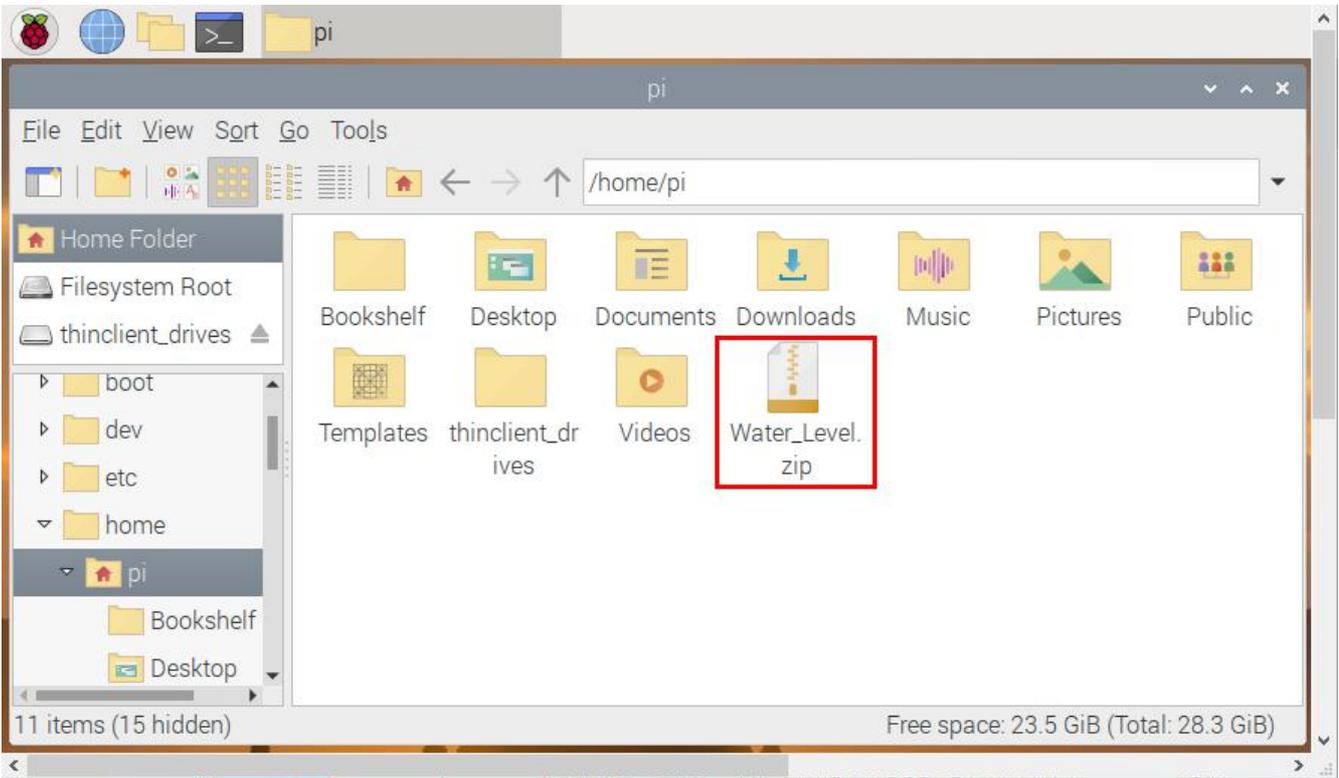
Check the address of the I2C module (PCF8591) connected to the Raspberry Pi, enter the command `i2cdetect -y 1`, and then press **Enter**.

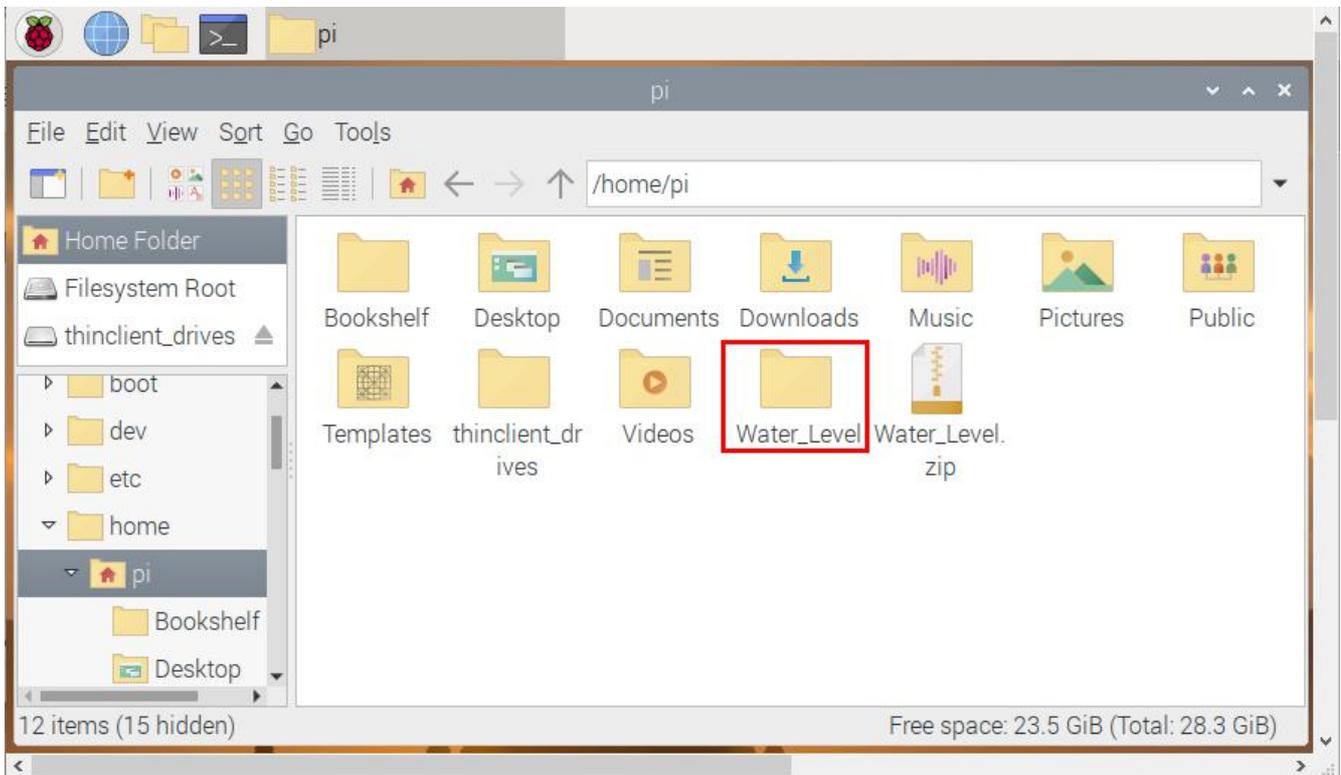
From below picture, it is known that the I2C address of PCF8591 is 0x48 .



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 **Water_Level.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/Water_Level  
gcc Water_Level.c -o Water_Level -lwiringPi  
sudo ./Water_Level
```

(3) Test Result:

Insert the shield into the Raspberry Pi board. After programming finishes, the terminal will show the detected signals by the water level sensor.

Note: press Ctrl + C to exit the code running

```
File Edit Tabs Help
A2:60
A2:53
A2:55
A2:72
A2:83
A2:86
A2:88
A2:91
A2:94
A2:97
A2:101
A2:104
A2:105
A2:107
A2:108
A2:108
A2:108
A2:108
A2:108
A2:109
A2:108
A2:108
A2:109
```

Test Code

File name: [Water_Level.c](#)

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

#define Address 0x48 //iic 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 value;
    wiringPiSetup();
    pcf8591Setup(BASE,Address); //Initialize the pcf8591

    while(1)
    {
        value=analogRead(A2); //Read the analog value of pin A2
        printf("A2:%d\n",value); //The terminal prints the simulated value
        delay(100);
    }
}
```

```
}  
}
```

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