Joystick Module
(000x0000 Article Number)
(TS2184)

DESIGN THE WORLD

Product Details

The joystick module mainly uses the PS2 joystick component. In fact, it has three signal terminal pins, which simulate X axis, Y axis and Z axis.

When controlling, the X and Y signal terminals of the module are connected to the analog port of the control board. The signal terminal B is generally connected to the digital port and used as a button. You can determine how the joystick on the module works by reading two analog values and high/low levels at a digital port.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- High-quality rocker provides long life with reliable performance.
- Two analog outputs,you can read through the AD conversion twist angle.
- For two degrees of freedom servo PTZ control or other robots remote control.

Technical Specifications

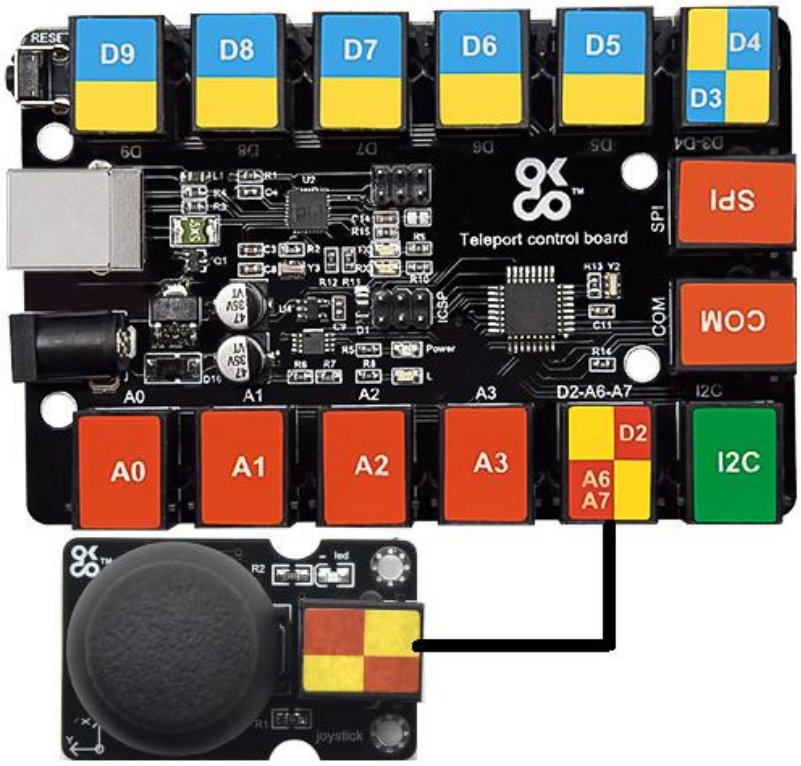
Sensor type	Digital and Analog
Working voltage	5V
Internal Potentiometer value	10k
Operating temperature	0 to 70 °C
Dimensions	45mm*28mm*33mm
Weight	12.5g

Applications

- Hand game console
- Joystick excavator

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

➢ Arduino Application



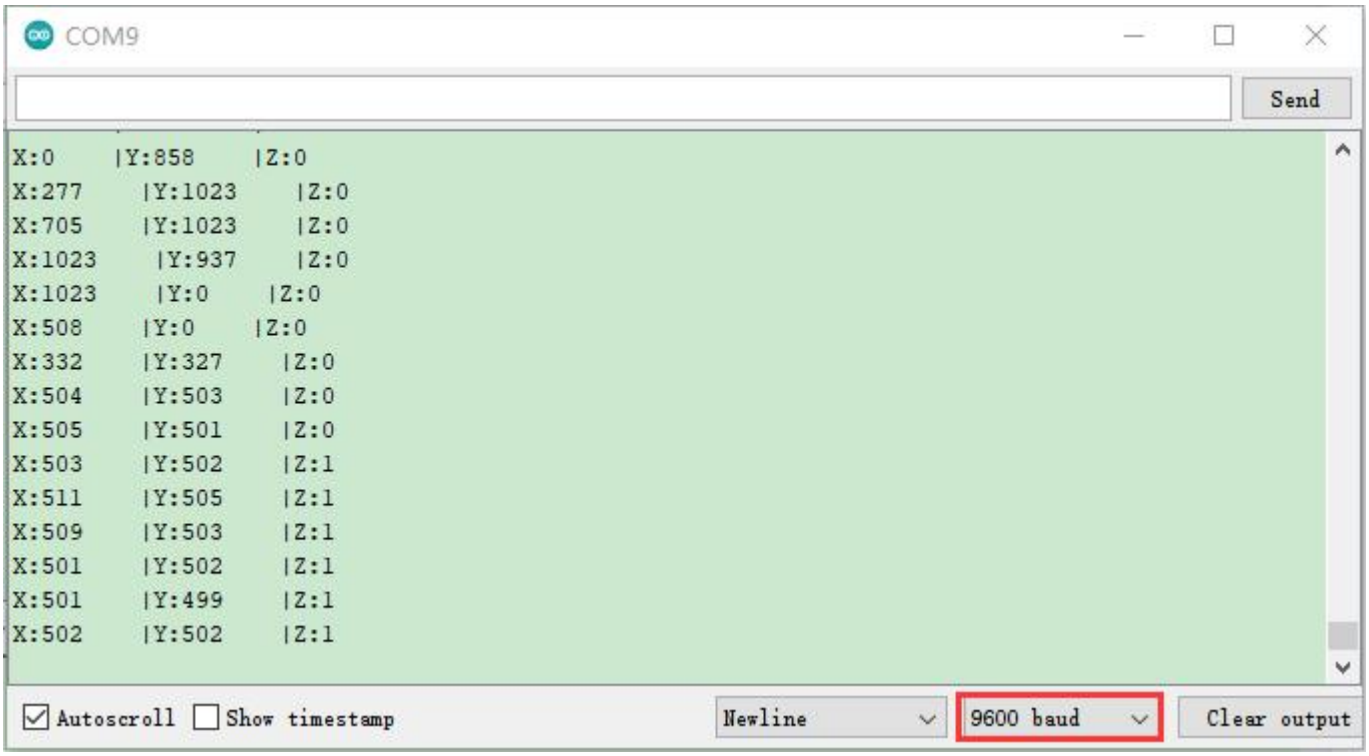
This module is compatible with the TS2178 TelePort control board.

Test Code

```
int JoyStick_X = 0; //Define analog port A0
int JoyStick_Y = 1; //Define analog port A1
int JoyStick_Z = 2; //Defining digital port 2
void setup()
{
pinMode(JoyStick_Z, INPUT);//Set JoyStick_Z as INPUT
Serial.begin(9600); // Set baud rate 9600
}
void loop()
{
int x,y,z; //Defining numerical variables x y z
x=analogRead(JoyStick_X); //Set x to the value of A0 read
y=analogRead(JoyStick_Y);//Set y to the value of A1 read
z=digitalRead(JoyStick_Z);//Set z to the value read at numeric port 2
Serial.print("X:"); //display character X：
Serial.print(x ,DEC); //display x values
Serial.print("  |Y:"); //display character  |Y：
Serial.print(y ,DEC);//display y values
Serial.print("  |Z:");//display character  |Z：
Serial.println(z ,DEC);//display z values，and wrap it
delay(100);//delay 0.1S
}
```

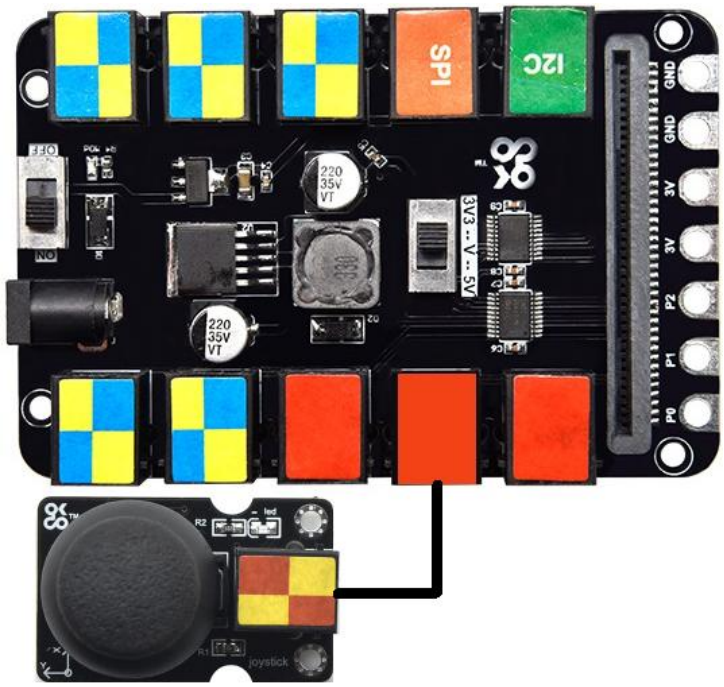
Test Result

Wire up, upload the test code and power it up, open serial monitor and set baud rate to 9600.
Push the thumb button leftward (along the X axis), the analog value of X axis is in the range of 0-500; if you push it rightward, the analog value is in the range of 500-1023; if you push the thumb button forward(along the Y axis), the analog value of Y axis is in the range of 0-497; if you push it backward, the data of Y axis is in the range of 497-1023; for Z axis, press the thumb button, 1 will be displayed.



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

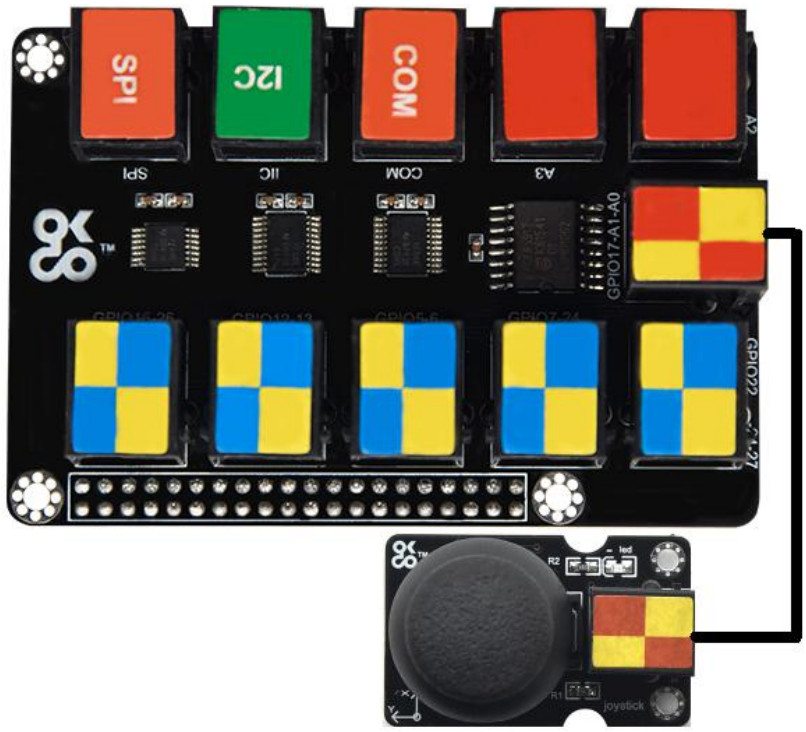
.....①Run the "on start" block to boot the program
.....②Open the LED matrix of the Micro:bit
.....③The program is run circularly under the command of "forever" block
.....④The Micro:bit shows the analog value at the port P3(x axis)
.....⑤The Micro:bit displays the analog value at the port P4(Y axis)
.....⑥The Micro:bit shows the digital signals shown at the port 2(Z axis)

Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn the DIP switch to 3V3, upload test code and power it up. The Micro:bit will display the analog values at P3(x axis), P4(y axis) and P2 port(z axis).

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：

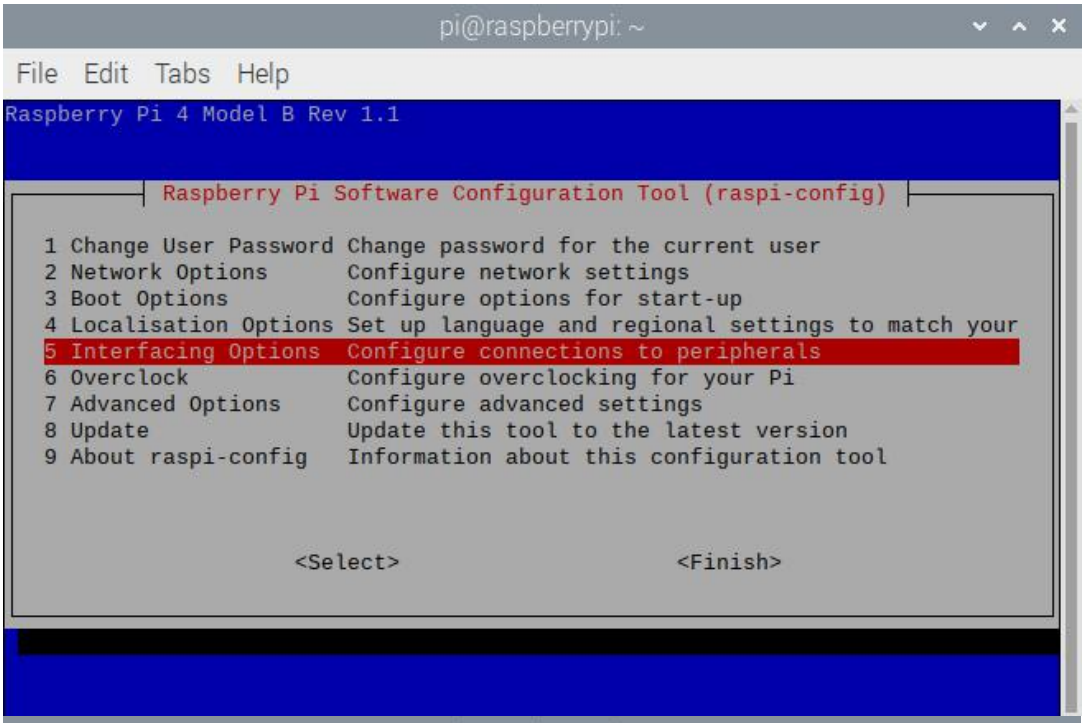
The Raspberry Pi itself does not have AD/DA function; therefore, an expansion board with this function is required when connected to external analog sensors. And here we use a PCF8591 A/D converter with I2C communication.

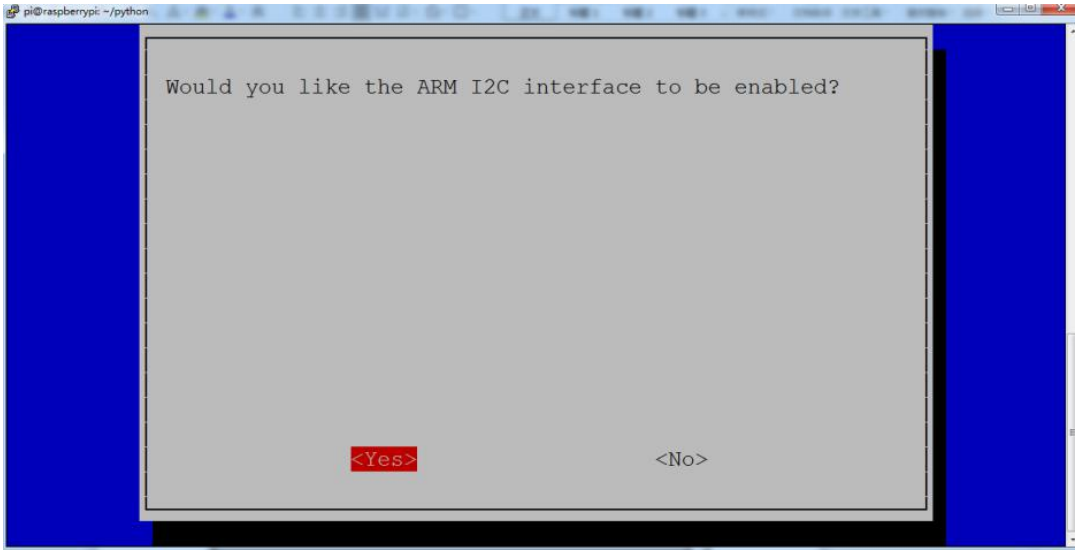
Enable the I2C communication function of the Raspberry Pi as follows:

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.

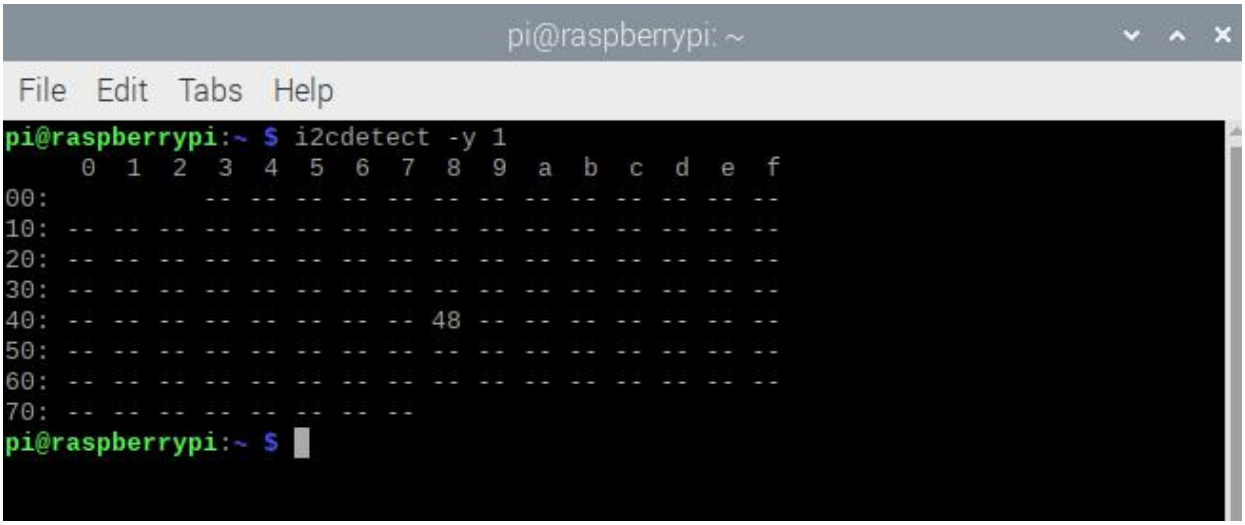
```
pi@raspberrypi:~/python $ sudo raspi-config
```

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



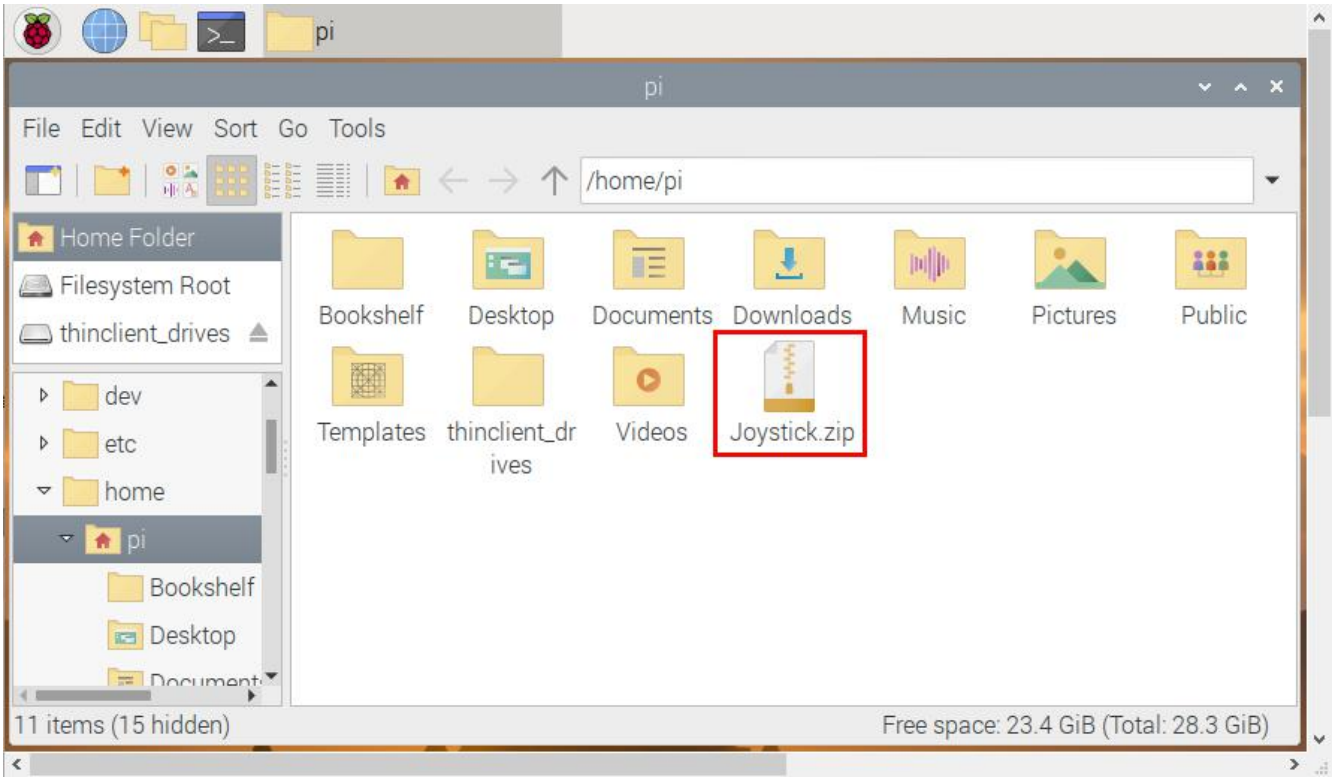


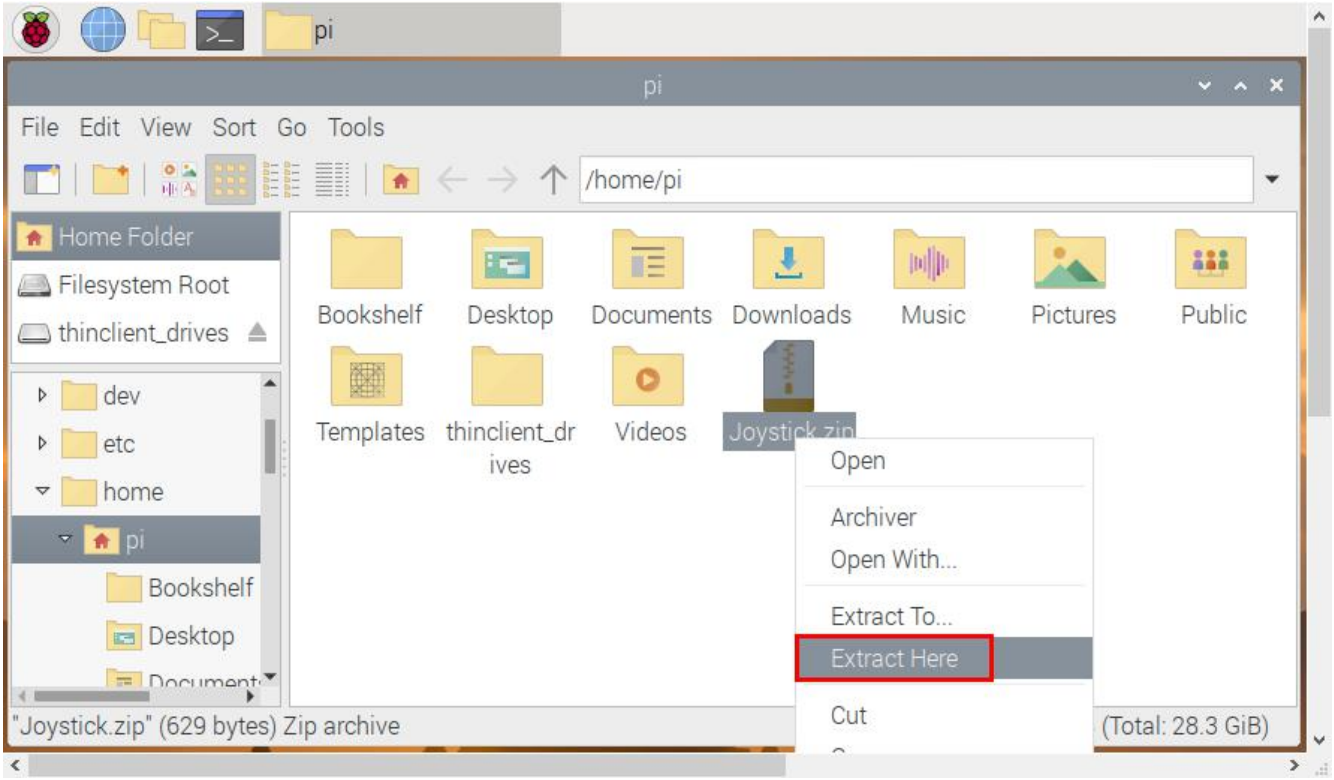
Check the address of the I2C module (PCF8591) connected to the Raspberry Pi, enter the command i2cdetect -y 1, and then press **Enter**.
From the below picture, it is known that the I2C address of PCF8591 is 0x48 .

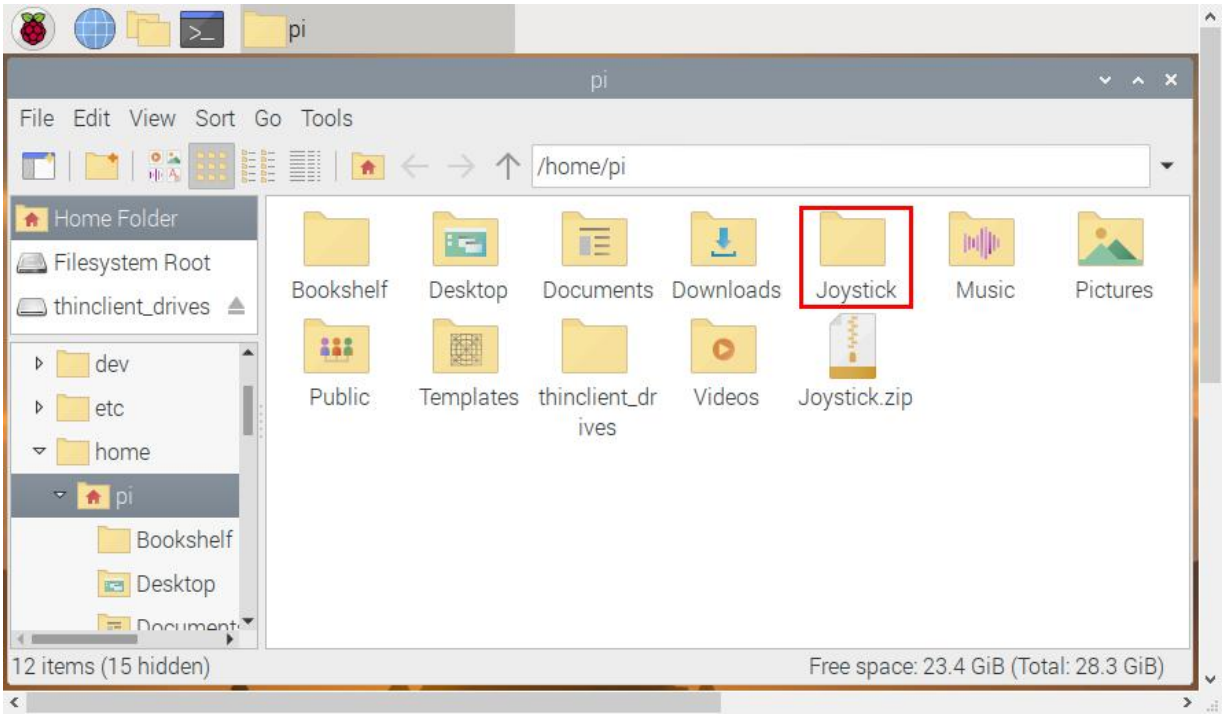


Copy the test code to Raspberry Pi system to run it

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


pi
pi
File Edit View Sort Go Tools
/home/pi
Home Folder
Filesystem Root
thinclient_drives
dev
etc
home
pi
Bookshelf
Desktop
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
thinclient_drives
Videos
Joystick.zip
11 items (15 hidden)
Free space: 23.4 GiB (Total: 28.3 GiB)


pi
pi
File Edit View Sort Go Tools
/home/pi
Home Folder
Filesystem Root
thinclient_drives
dev
etc
home
pi
Bookshelf
Desktop
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
thinclient_drives
Videos
Joystick.zip
Open
Archiver
Open With...
Extract To...
Extract Here
Cut
"Joystick.zip" (629 bytes) Zip archive
(Total: 28.3 GiB)



(2) Compile and run test code:

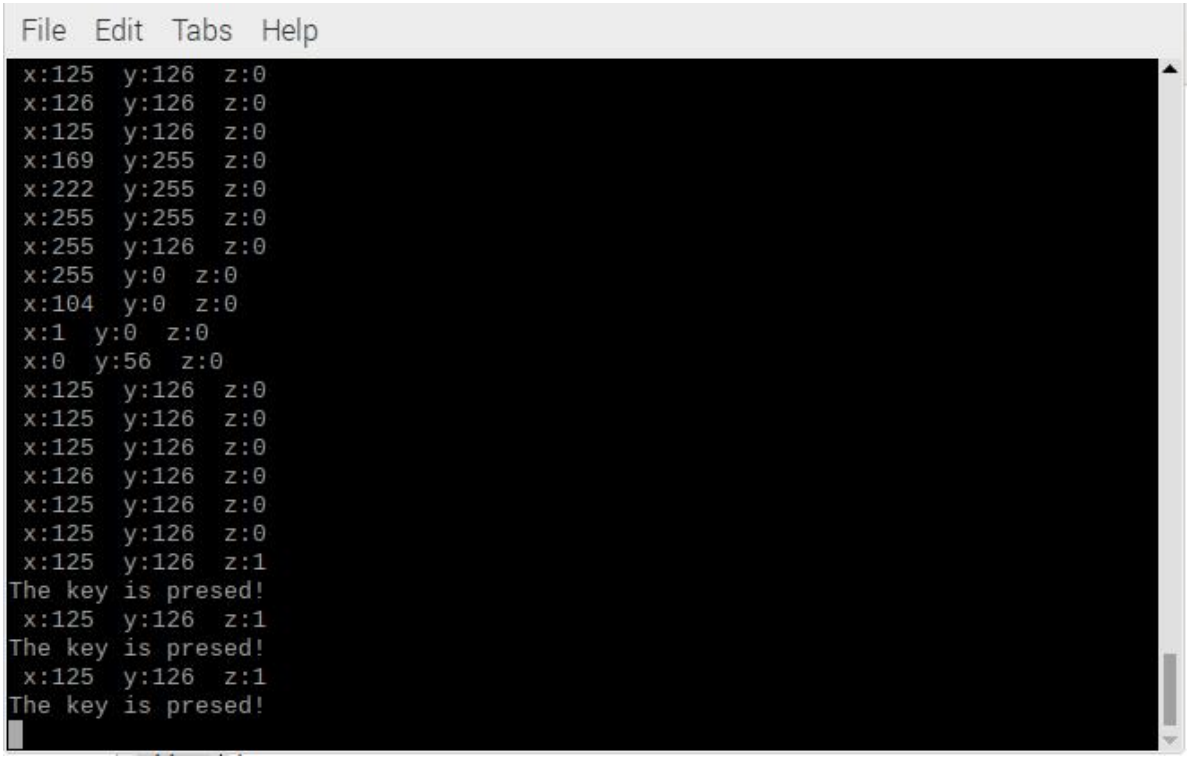
Input the following code and press"Enter"

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

(3) Test Result:

Insert the shield into the Raspberry Pi board. After programming finishes, push the joystick, the terminal will show the value changes; press it, the terminal will display"The key is pressed".

Note: press Ctrl + C to exit code running



Test Code

File name: Joystick.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

#define btnPin 0 //GPIO 17

int main(void)
{
  unsigned char x_val;
  unsigned char y_val;
  unsigned char z_val;
  wiringPiSetup();
  pcf8591Setup(BASE,Address);
  pinMode(0,INPUT);

  while(1)
  {
    x_val=analogRead(A1);  //read x
    y_val=analogRead(A0);  //read y
    z_val=digitalRead(0); //read z, button
    printf(" x:%d  y:%d  z:%d\n", x_val,y_val,z_val);
    if(z_val==1)
    printf("The key is presed!\n");
    delay(100);
  }
}
```

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

END