4-Digit 8-Segment Display (000x0000 Article Number) (TS2169)

DESIGN THE WORLD

Product Details

This is a 0.36 inch common anode TelePort 4-digit 8-segment LED display module with 12 pins and score points. Only two digital pins on the control board can control what the digital display shows and brightness. RJ11 port
has four pins(GND, VCC, DIO and CLK). GND is grounded, VCC is power supply, DIO is data input/output pin and CLK is the clock signal pin.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Control the 4-digit display using I2C bus including brightness.
- Ideal to display time, scoring device, counter, and other information.

Technical Specifications

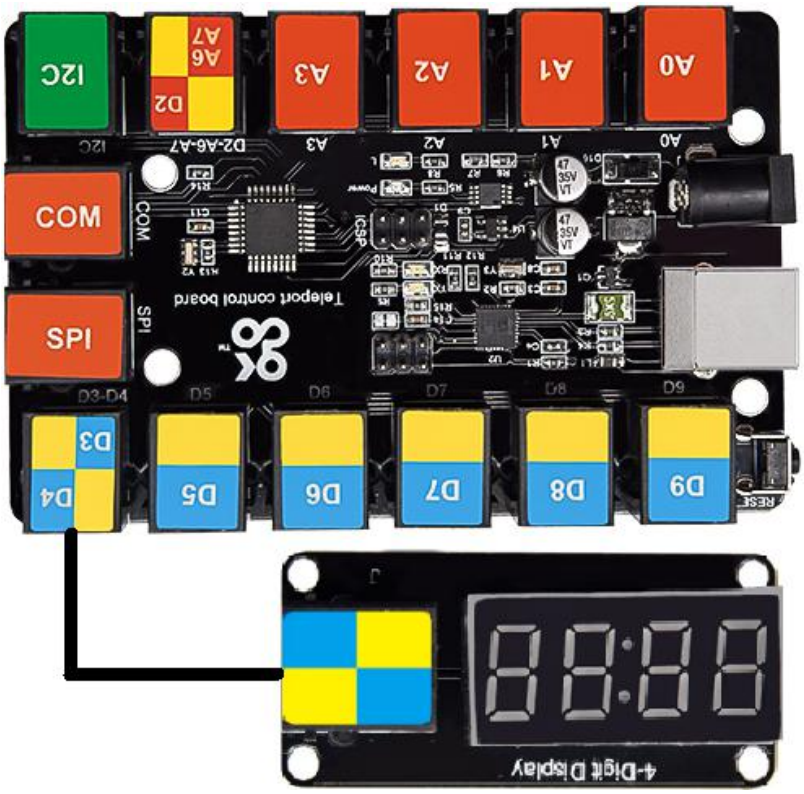
Sensor type	I2C
Working voltage	3.3V-5V
Control chip	TM1637
Module current	30--80MA
Tube colour	0.36 inches LED, red highlights
Dimensions	50mm*23mm*18mm
Weight	9g

Applications

- Time display
- Stopwatch
- Sensors input display

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

Before compiling test code, remember to place the **DigitalTube** library in the libraries of Arduino IDE.
Unzip the library files, that is, copy the unzipped the **DigitalTube** folder into the libraries of Arduino IDE.
After pasting it, reboot the compiler.
For instance: C:\Program Files\Arduino\libraries

```
#include "TM1637.h"
#define CLK 4//pins definitions for TM1637 and can be changed to other ports
#define DIO 3
TM1637 tm1637(CLK,DIO);
void setup()
{
  tm1637.init();
  tm1637.set(BRIGHT_TYPICAL);//BRIGHT_TYPICAL = 2,BRIGHT_DARKEST = 0,BRIGHTEST = 7;
}
void loop()
{
  //int8_t NumTab[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};//0~9,A,b,C,d,E,F
  int8_t NumTab[] = {0,1,2,3,4,5,6,7,8,9};//0~9,A,b,C,d,E,F
  int8_t ListDisp[4];
  unsigned char i = 0;
  unsigned char count = 0;
  delay(150);
  while(1)
  {
```

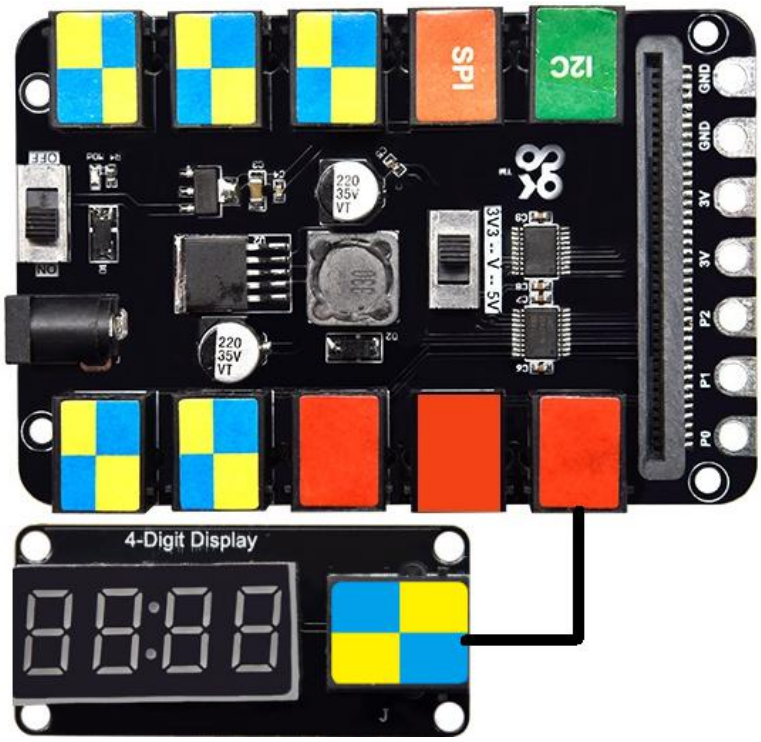
```
    i = count;
    count ++;
    if(count == sizeof(NumTab)) count = 0;
    for(unsigned char BitSelect = 0;BitSelect < 4;BitSelect ++)
    {
      ListDisp[BitSelect] = NumTab[i];
      i ++;
      if(i == sizeof(NumTab)) i = 0;

    tm1637.display(0,ListDisp[0]);
    tm1637.display(1,ListDisp[0]);
    tm1637.display(2,ListDisp[0]);
    tm1637.display(3,ListDisp[0]);
    tm1637.point(POINT_ON);
    delay(100);
    }
  }
}
```

Test Result

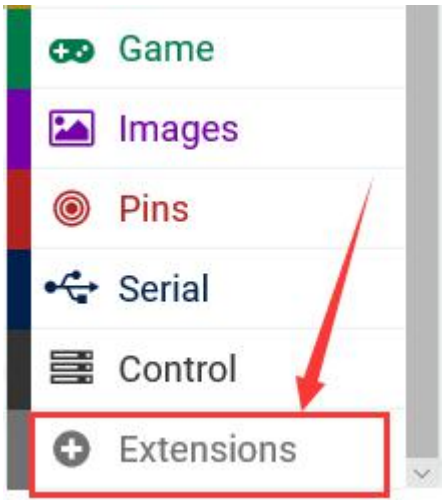
Wire up, upload code and power it up. Then 4-digit 8-segment display will show numbers from 0 to 9, and display same four numbers.

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

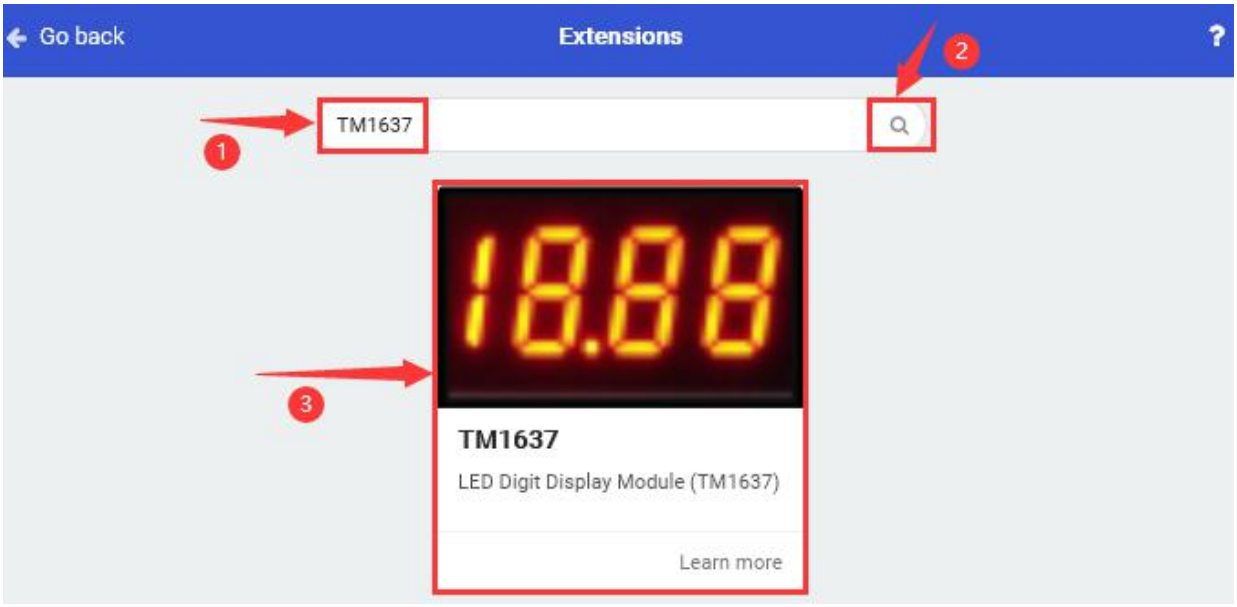
- **Micro:bit Application**



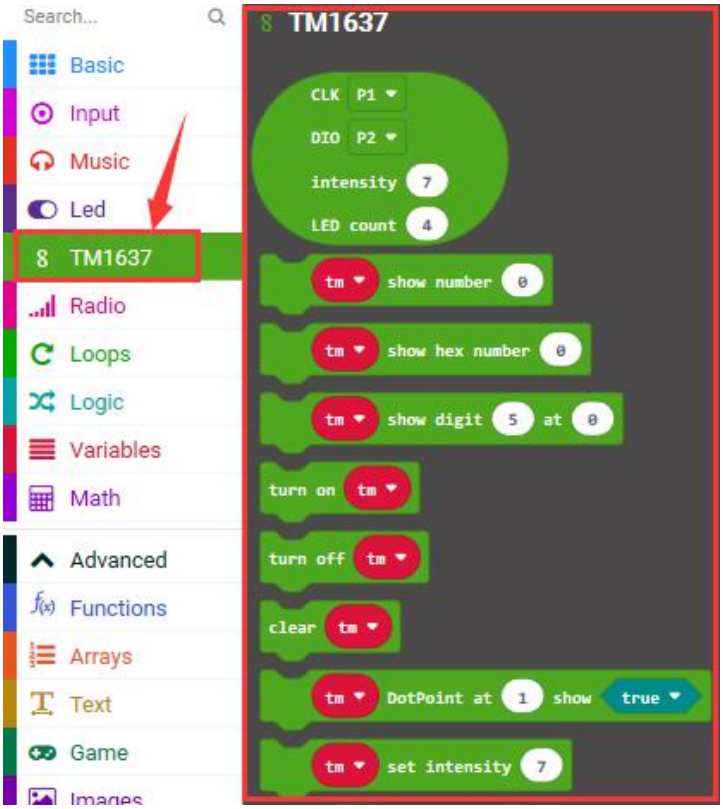
It is compatible with the Micro:bit board and the TS2179 Micro:bit expansion board.
Add the library of the 4-digit 8-segment display , as shown below;
Use the library file to set code, click“Extensions”



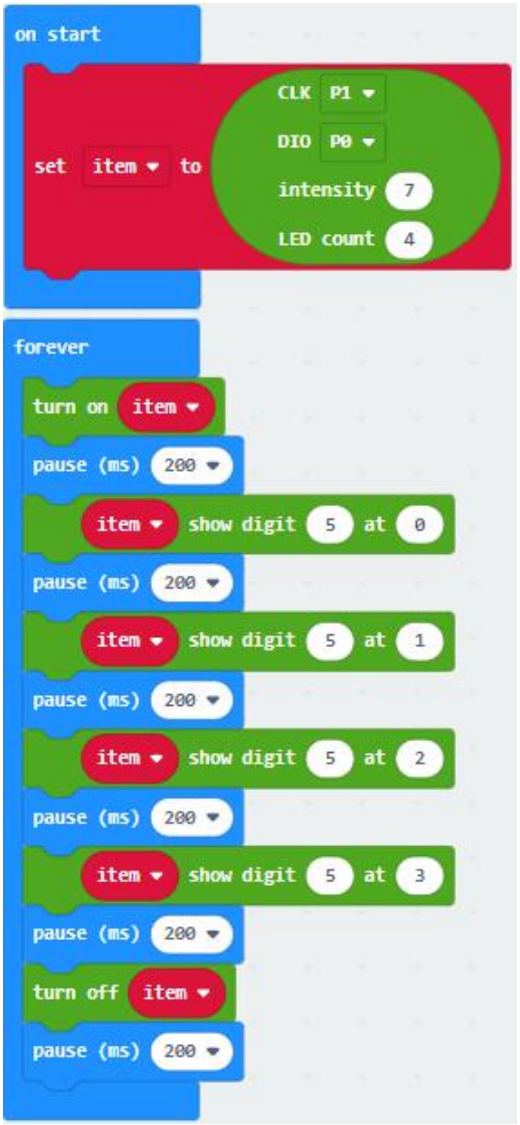
Enter“TM1637”to search, as shown below, click the library file and download it automatically.



After the library of the 4-digit 8-segment display is installed, then you can view the corresponding block in the blocks list.



Test Code



.....①Run the "on start" block to boot the program
.....②Set item to CLK P1 DIO P0 intensity 7 LED count 4
..... ③ The program is run circularly under the command of "forever" block
.....④open variable item
.....⑤delay in 200ms
.....⑥The 4-digit 8-segment display will show 5 at 0 bit

.....⑦delay in 200ms
.....⑧The 4-digit 8-segment display will show 5 at 1 bit

.....⑨delay in 200ms
.....⑩The 4-digit 8-segment display will show 5 at 2 bit

.....⑪delay in 200ms
.....⑫The 4-digit 8-segment display will show 5 at 3 bit

.....⑬delay in 200ms

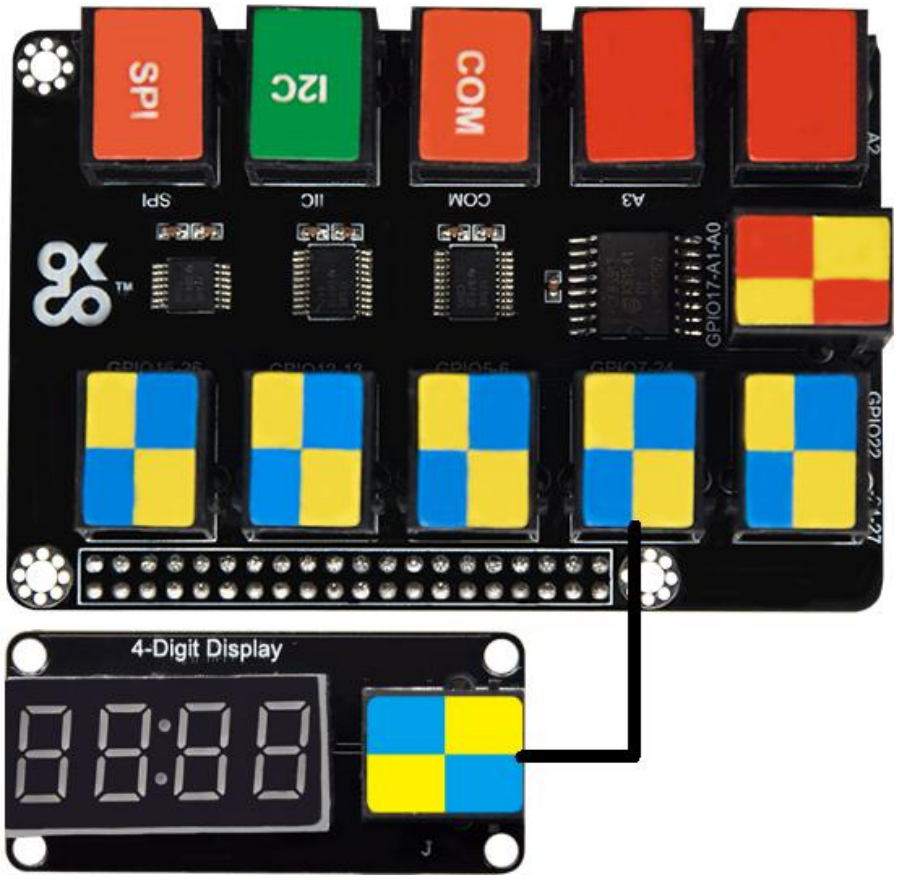
.....⑭turn off variable item
.....⑮delay in 200ms

Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn DIP switches to 5V and ON end, upload test code and power it up. Then you will view the flashing“5555”on the 4-digit 8-segment display.

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

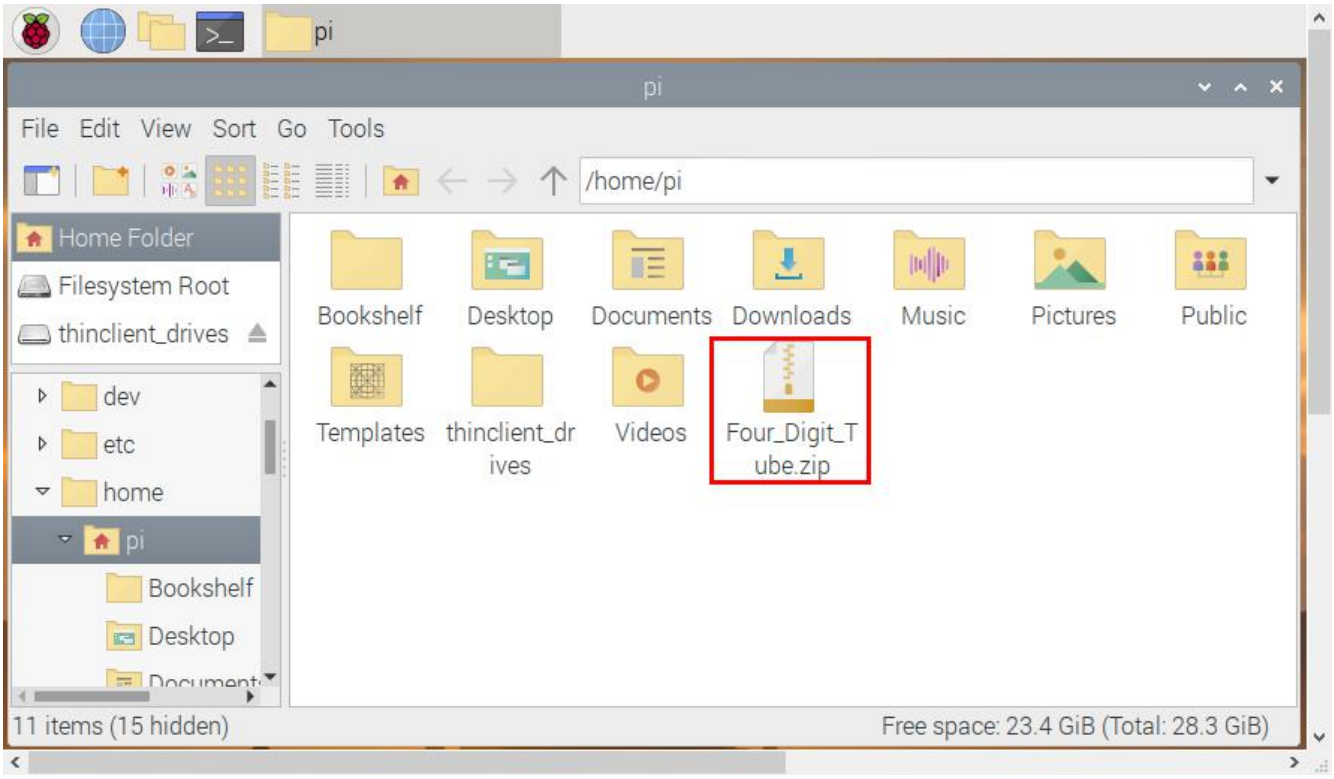
➢ **Raspberry Pi Application**

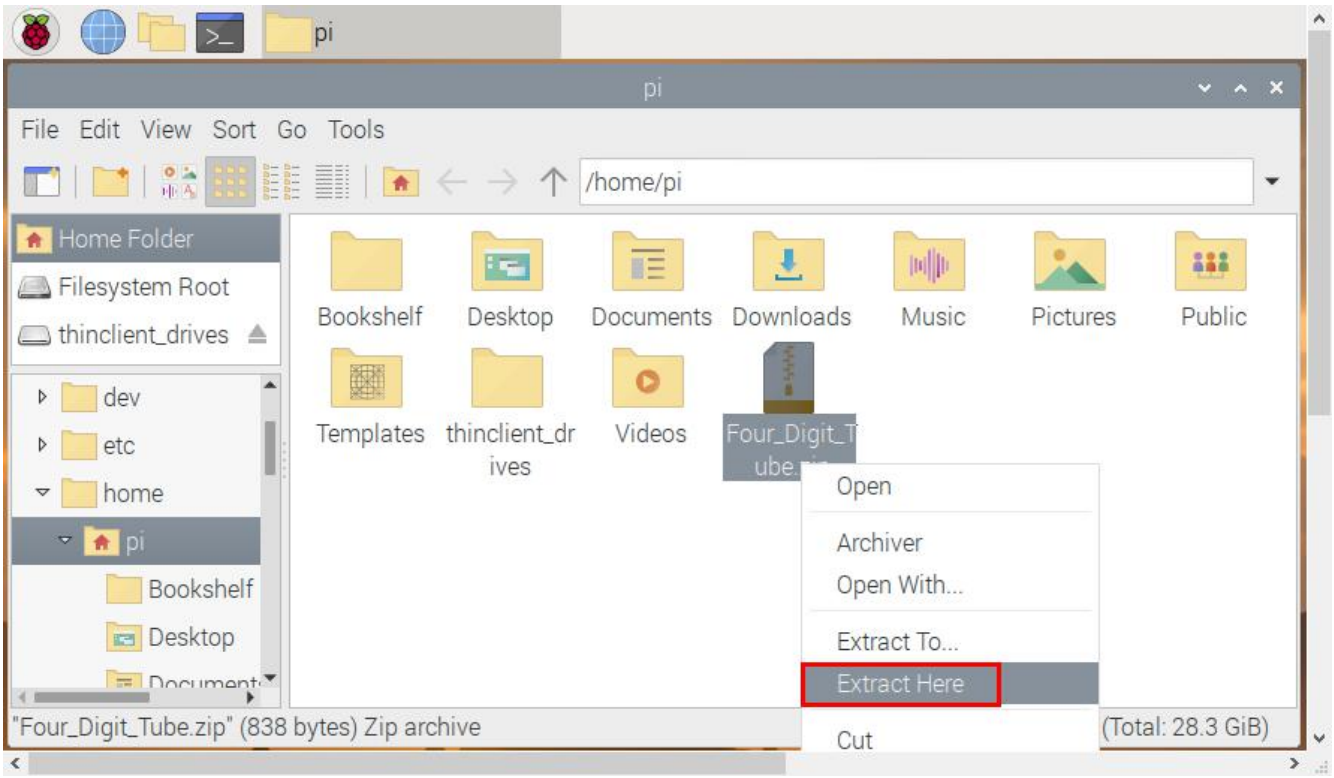


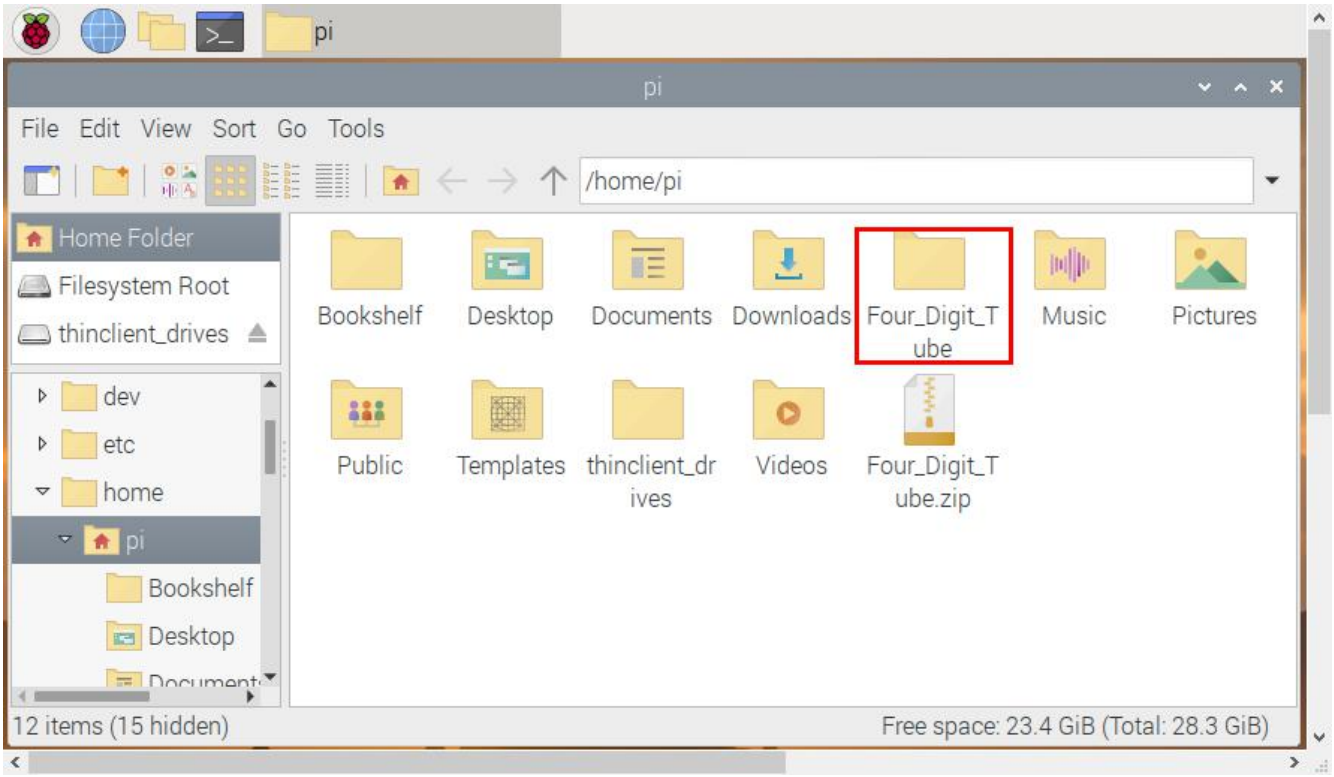
It is compatible with the Raspberry Pi board and the TS2180 Raspberry Pi shield.

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 Four_Digit_Tube.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
Four_Digit_Tube.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
Four_Digit_T
Open
Archiver
Open With...
Extract To...
Extract Here
Cut
"Four_Digit_Tube.zip" (838 bytes) Zip archive
(Total: 28.3 GiB)



(2) Compile and run test code:

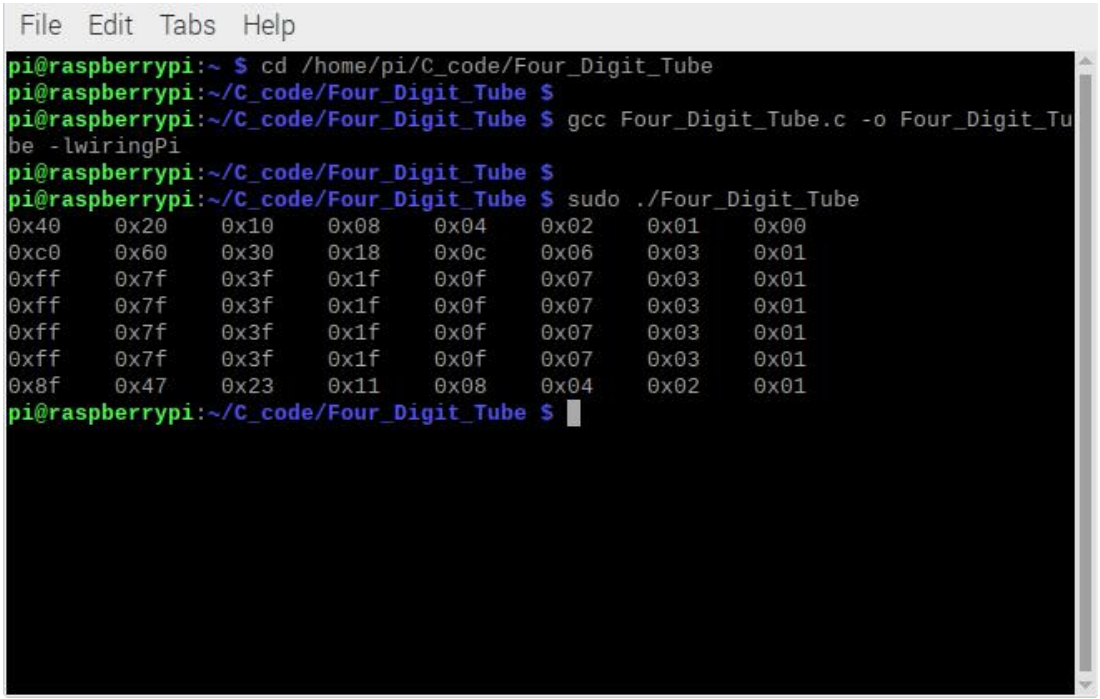
Input the following code and press"Enter"

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

(3) Test Result:

Insert the shield into the Raspberry Pi board. After programming finishes, the 4-digit 8-segment display will show 88:88.

Note: press Ctrl + C to exit code running



Test Code

File name: Four_Digit_Tube.c

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

#define CLK 5    // BCM GPIO 24
#define DIO 11    // BCM GPIO 7

#define CMD_AUTOINCREASE 0x40
#define CMD_STARTADDRESS 0xC0
#define CMD_BRIGHTNESS 0x8F

void delay(unsigned int time){
        int intime = time;
        while(intime--){
                for(int i=255;i>0;i--){ }
        }
}

void writeData(unsigned char data){
        unsigned char indata = data;
        for(int i=0;i<8;i++){
                digitalWrite(CLK,LOW);
                delay(1);
                if((indata & 0x01) == 0x01){
                        digitalWrite(DIO,HIGH);
                        delay(1);
                }else{
                        digitalWrite(DIO,LOW);
                        delay(1);
                }
                printf("0x%02x\t",indata);
                indata = (indata >> 1);
                digitalWrite(CLK,HIGH);
                delay(1);
        }
        printf("\n");
        digitalWrite(CLK,LOW);
        delay(1);
        digitalWrite(CLK,HIGH);
        delay(1);
        digitalWrite(CLK,LOW);
        delay(1);
```

```
}

void startDisp(){
	digitalWrite(CLK,HIGH);
	delay(1);
	digitalWrite(DIO,HIGH);
	delay(1);
	digitalWrite(DIO,LOW);
	delay(1);
	digitalWrite(CLK,LOW);
	delay(1);
}

void stopDisp(){
	digitalWrite(CLK,LOW);
	delay(1);
	digitalWrite(DIO,LOW);
	delay(1);
	digitalWrite(CLK,HIGH);
	delay(1);
	digitalWrite(DIO,HIGH);
	delay(1);
}

void disp(){
	startDisp();writeData(CMD_AUTOINCREASE);stopDisp();
	startDisp();writeData(CMD_STARTADDRESS);
	writeData(0b11111111);
	writeData(0b11111111);
	writeData(0b11111111);
	writeData(0b11111111);
	stopDisp();
	startDisp();writeData(CMD_BRIGHTNESS);stopDisp();
}

int main(){
	wiringPiSetup();

	pinMode(CLK,OUTPUT);
	pinMode(DIO,OUTPUT);
	digitalWrite(CLK,LOW);
	digitalWrite(DIO,LOW);
```

```
        disp();

        return 0;
}
```

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

END