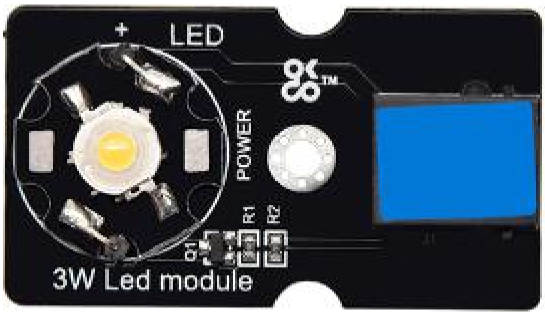
3W LED Module
(000x0000 Article Number)
(TS2186)

DESIGN THE WORLD

Product Details



This is the TelePort 3W LED module with high power. It can control the state of the plug-in LED. Its RJ11 port integrates G,V and S(Signal). When the S is at high levels, the LED will be on; when at low levels, the LED will be off.

Caution:
The module uses the RJ11 6P6C interface which is only compatible with the TelePort control boards and expansion boards of RJ11 6P6C series.

If you use a sensor or module with the RJ11 6P6C port from other companies, it won’t work with ours boards on grounds of different wire sequence.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- 3W LED module operates between 3.3-5V.
- Super bright yellow LED producing 200-220 lumens.
- Widely used in night lighting, corridor lights and advertising lights.

Technical Specifications

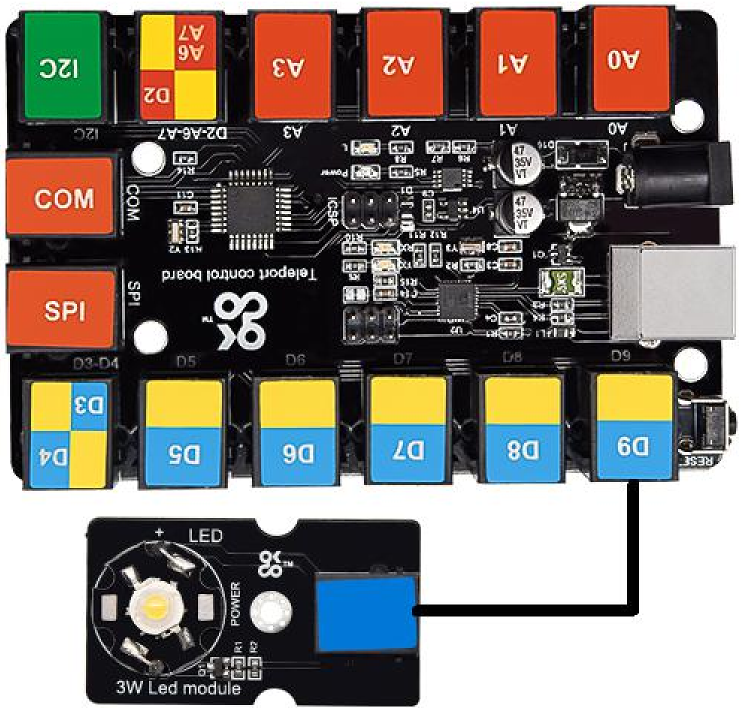
Sensor type	Digital output
Working voltage	3.3V-5V
Emitting color	Yellow
Recommended use current	600 - 700ma
Luminance	200 - 220LM

Applications

- Reading lights (car, bus, aircraft)
- Portable (flashlight, bicycle)
- Uplights/Downlights
- Decorative/Entertainment
- Bollards/Security/Garden
- Cove/Under shelf/Task
- Indoor/Outdoor Commercial and Residential Architectural
- Automotive Ext (Stop-Tail-Turn
- LCD backlights

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 led = 9;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

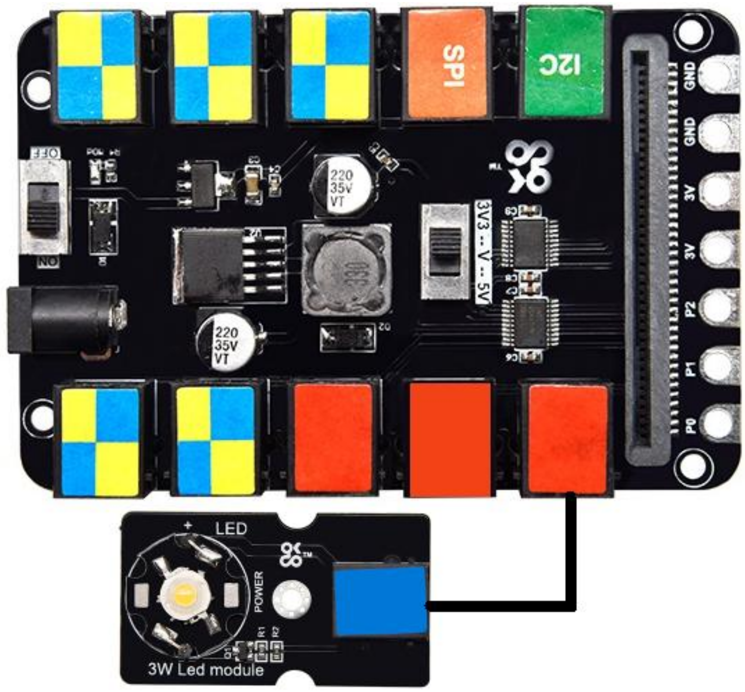
Test Result

Wire up, upload the test code and power it up.

Then you will see the indicator on the module flash with an interval of 1s.

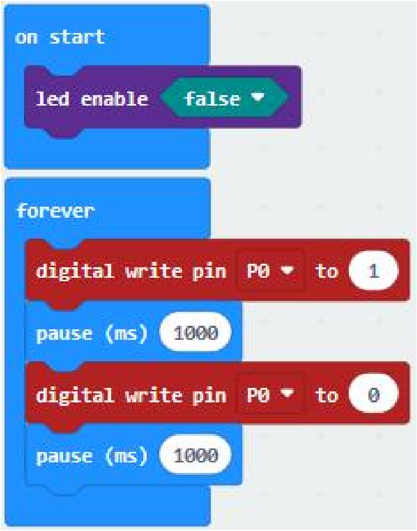
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

- **Micro:bit Application**



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

Test Code



```
on start
    led enable false
forever
    digital write pin P0 to 1
    pause (ms) 1000
    digital write pin P0 to 0
    pause (ms) 1000
```

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

..............②the LED matrix of the Micro:bit is turned off

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

..............④Set P0 to high level(1), turn on LED

..............⑤delay in 1000ms

..............⑥Set P0 to low level(0), turn off LED

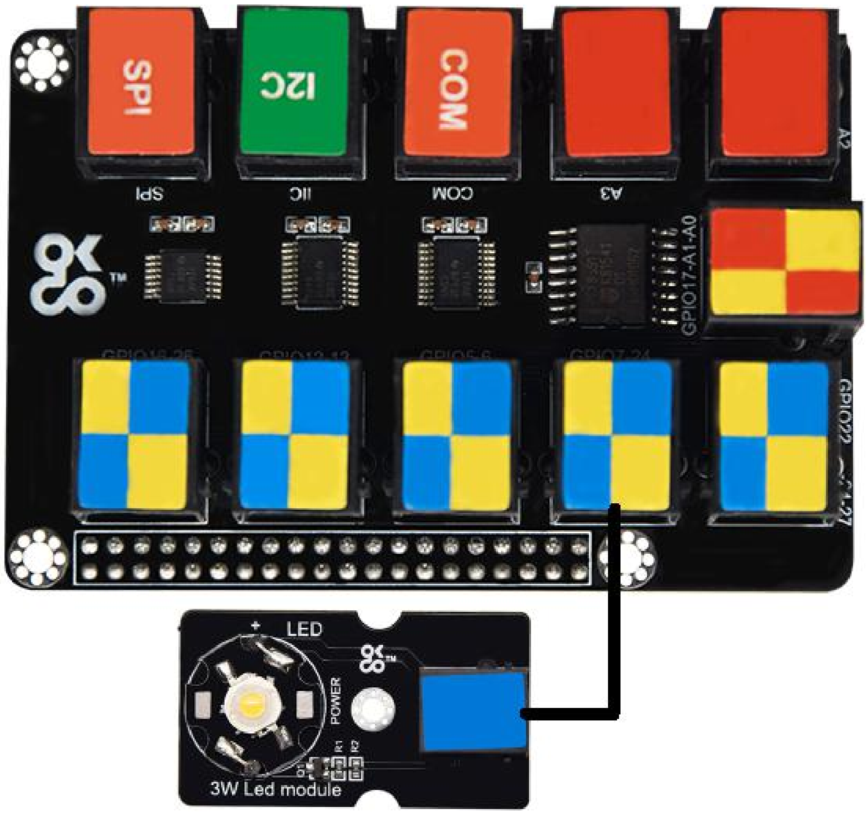
..............⑦delay in 1000ms

Test Result

Wire up, insert the Micro:bit into the shield, turn DIP switch to 3V3, upload test code and power it up. Then you will see the indicator on the module flash with an interval of 1s.

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

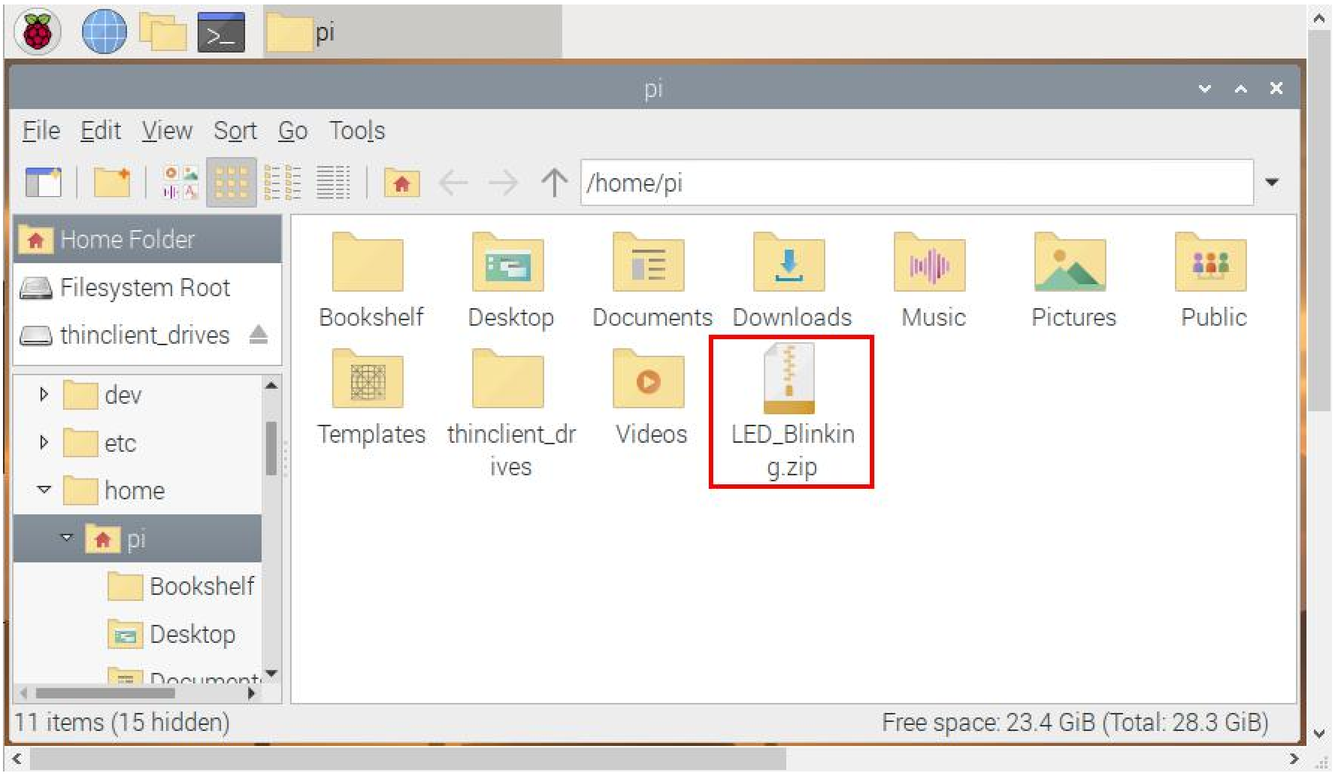
- **Raspberry Pi Application**



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

Copy the test code to Raspberry Pi system to run it

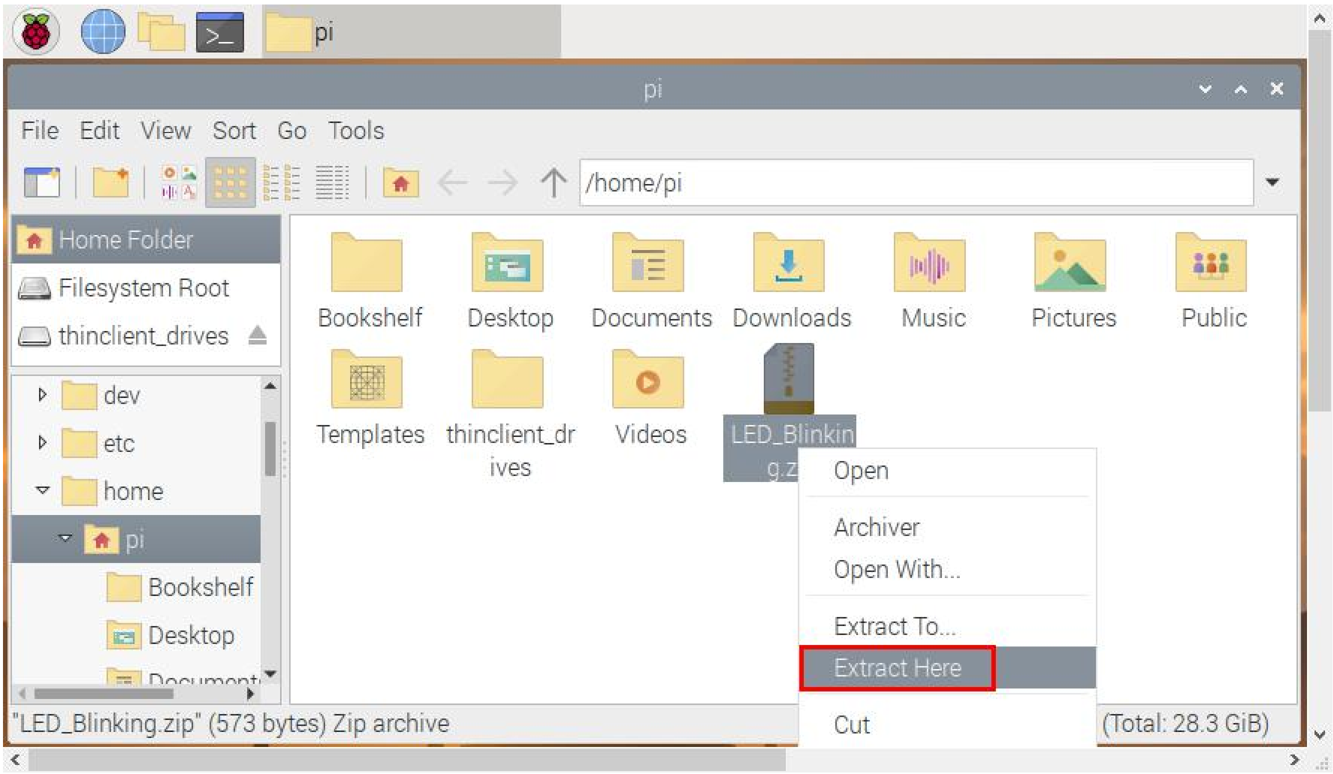
(1) Save the test code in the **pi** folder of Raspberry Pi system. Then place the LED_Blinking.zip file we provide in the **pi** folder, right-click and click **Extract Here.** As shown below:



/home/pi

11 items (15 hidden)

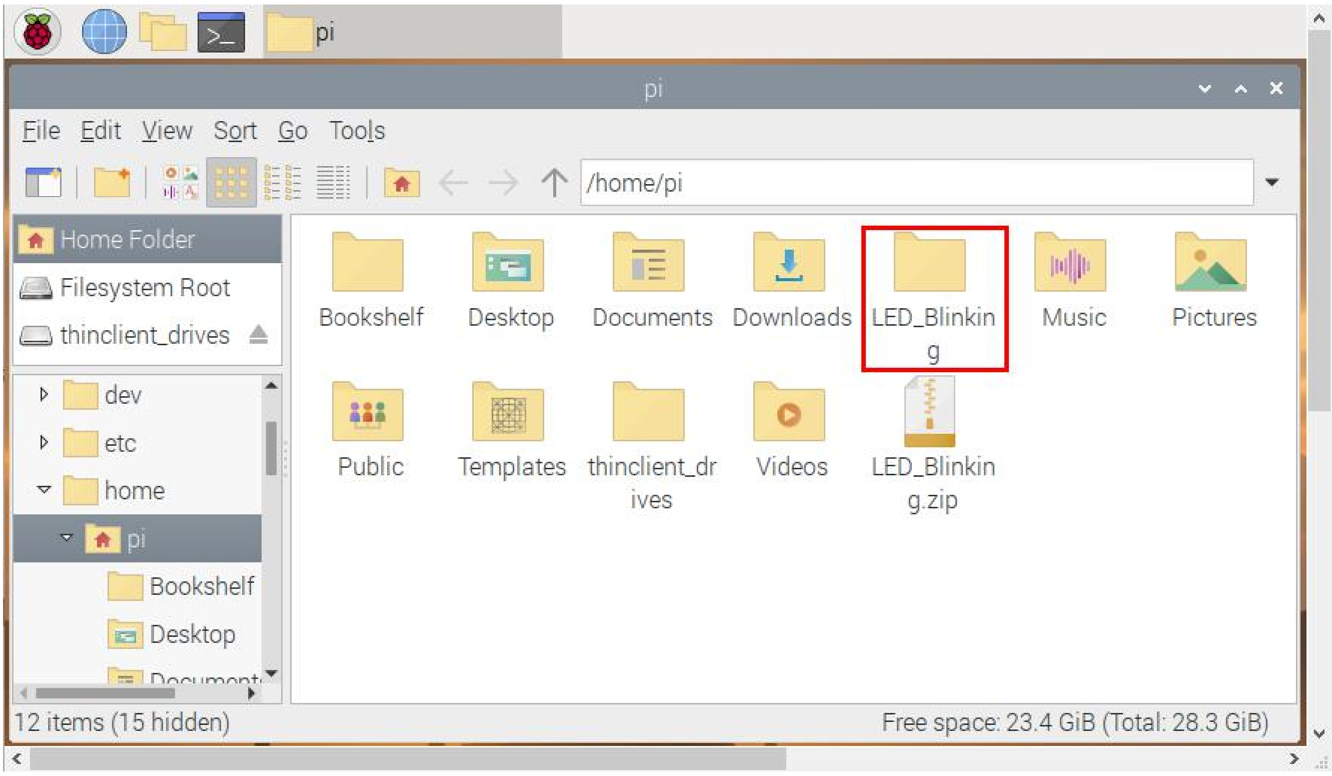
Free space: 23.4 GiB (Total: 28.3 GiB)



/home/pi

"LED_Blinking.zip" (573 bytes) Zip archive

(Total: 28.3 GiB)



(2) Compile and run test code：

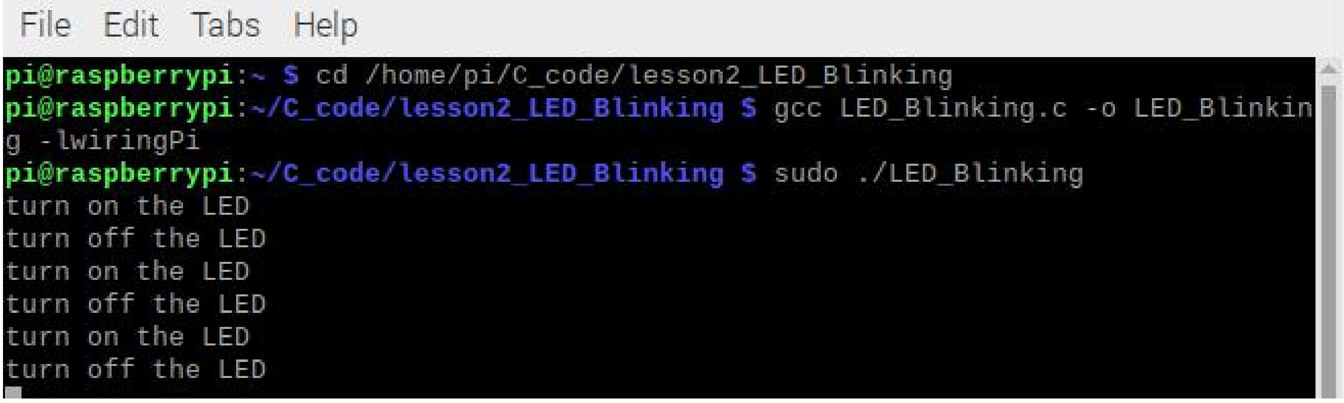
Input the following code and press“Enter”

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

(3) Test Result：

Insert the shield into the Raspberry Pi board. After programming finishes, then you will see the terminal print data and LED flash.

Note: press Ctrl + C to exit code running



```
pi@raspberrypi:~ $ cd /home/pi/C_code/lesson2_LED_Blinking
pi@raspberrypi:~/C_code/lesson2_LED_Blinking $ gcc LED_Blinking.c -o LED_Blinkin
g -lwiringPi
pi@raspberrypi:~/C_code/lesson2_LED_Blinking $ sudo ./LED_Blinking
turn on the LED
turn off the LED
turn on the LED
turn off the LED
turn on the LED
turn off the LED
```

Test Code

File Name: LED_Blinking.c

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

#define ledPin 11  //define led pin, BCM GPIO 7
```

```
int main()
{
  wiringPiSetup();  //Initialize wiringPi

  pinMode(ledPin,OUTPUT);  //set the ledPin OUTPUT mode

  while(1)
  {
        digitalWrite(ledPin,HIGH);  //turn on led
        printf("turn on the LED\n");
        delay(1000);       //delay 500ms
        digitalWrite(ledPin,LOW);   //turn off led
        printf("turn off the LED\n");
        delay(1000);
  }
}
```

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

END