# Starter Kit for RPI

# Introduction

**WiringPi**

WiringPi is a PIN based GPIO access library written in C for the BCM2835 used in the **Raspberry Pi**. It's released under the GNU LGPLv3 license and is usable from C, C++ and RTB (BASIC) as well as many other languages with suitable wrappers. It's designed to be familiar to people who have used the Arduino "wiring" system.

**********

* About Elecrow:
* We are a leading manufacturer of electronic components for Arduino and Raspberry Pi.
* We have a professional engineering team dedicated to providing tutorials and support to help you get started.
* If you have any technical questions or suggestions, please feel free to contact our support staff via email at techsupport@elecrow.com
* We truly hope you enjoy the product, for more great products please visit our company website: https://www.elecrow.com
or Amazon store: www.amazon.com/shops/elecrow

## Catalog

# Operation demo

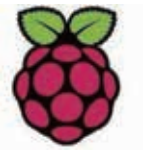
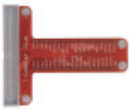
## Step 1: Download and install wiringpi

Tips: Most of the Rapbian versions have been preloaded with wiringPi, and the test methods are as follows:

Open terminal and run: gpio -v

If you get something, then you have it already installed.
If not, please refer it:
http://wiringpi.com/download-and-install/

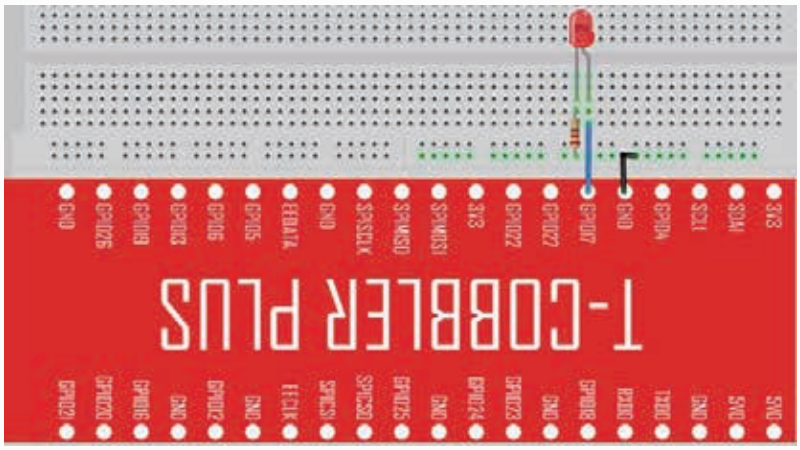
## Step 2: Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | 220/330Ω resistor | 1 |
|  | LED | 1 |
|  | Raspberry Pi Board | 1 |
|  | T-Cobbler Plus | 1 |
|  | 40P GPIO Cable | 1 |
|  | Jumper wires | Several |
|  | Breadboard | 1 |

## Step 3: Circuit connection

Because it is using the wiringPi library, the pins have been re-layout. The pins on the “T- Cobbler Plus” is basic for BCM which is different from wPi, please refer the “wPi” number.

```
pi@raspberrypi:~ $ gpio readall
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT5 | TxD     |  15 |  14 |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT5 | RxD     |  16 |  15 |
 |  17 |   0 | GPIO. 0 |  OUT | 0 | 11 || 12 | 0 | ALT5 | GPIO. 1 |   1 |  18 |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 |   4 |  23 |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 |   5 |  24 |
 |  10 |  12 |    MOSI | ALT0 | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 0 | 21 || 22 | 0 | IN   | GPIO. 6 |   6 |  25 |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | OUT  | CE0     |  10 |   8 |
 |     |     |      0v |      |   | 25 || 26 | 1 | OUT  | CE1     |  11 |   7 |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   |  31 |   1 |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 |  26 |  12 |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 |  27 |  16 |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 |  28 |  20 |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 |  29 |  21 |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
```




Connection:

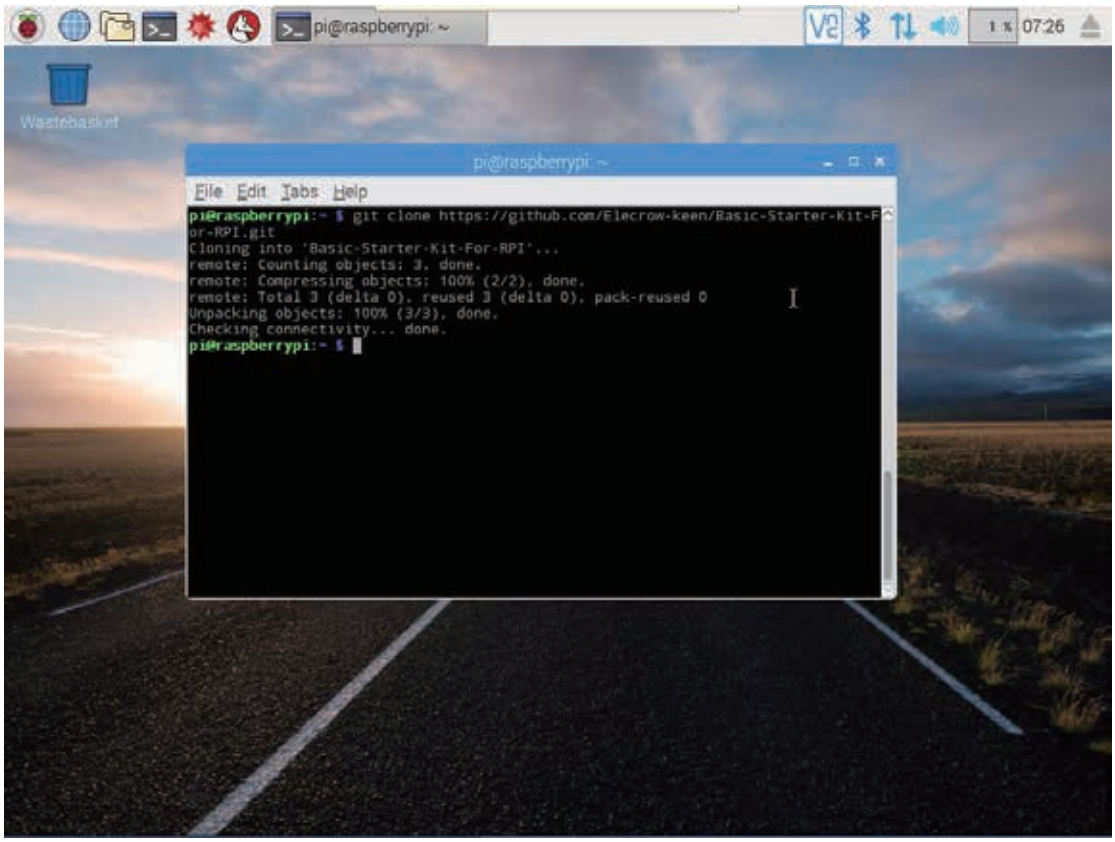
| RPI | LED |
|---|---|
| GPIO17 | Long pin |
| GND | Short pin |

## Step 4: Open Terminal and Download the Demo code

**Run:**

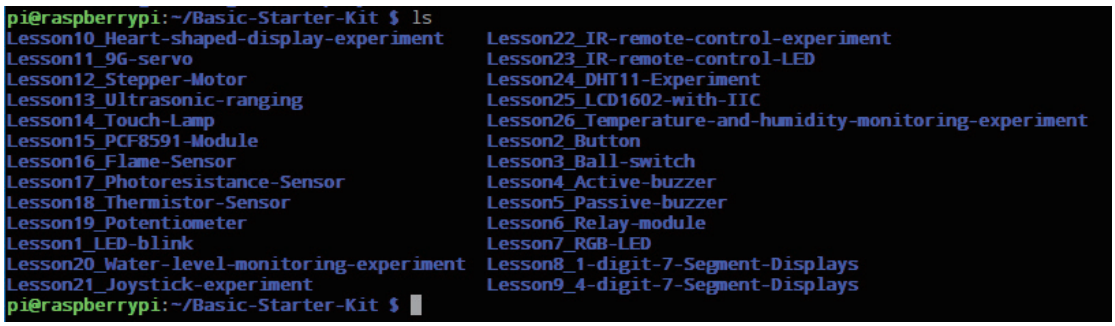
git clone https://github.com/Elecrow-keen/Basic-Starter-Kit.git

**Tips:** We will continue to update our resources on Github, please always pay attention the url: https://github.com/Elecrow-keen/Update-for-Basic-Starter-Kit

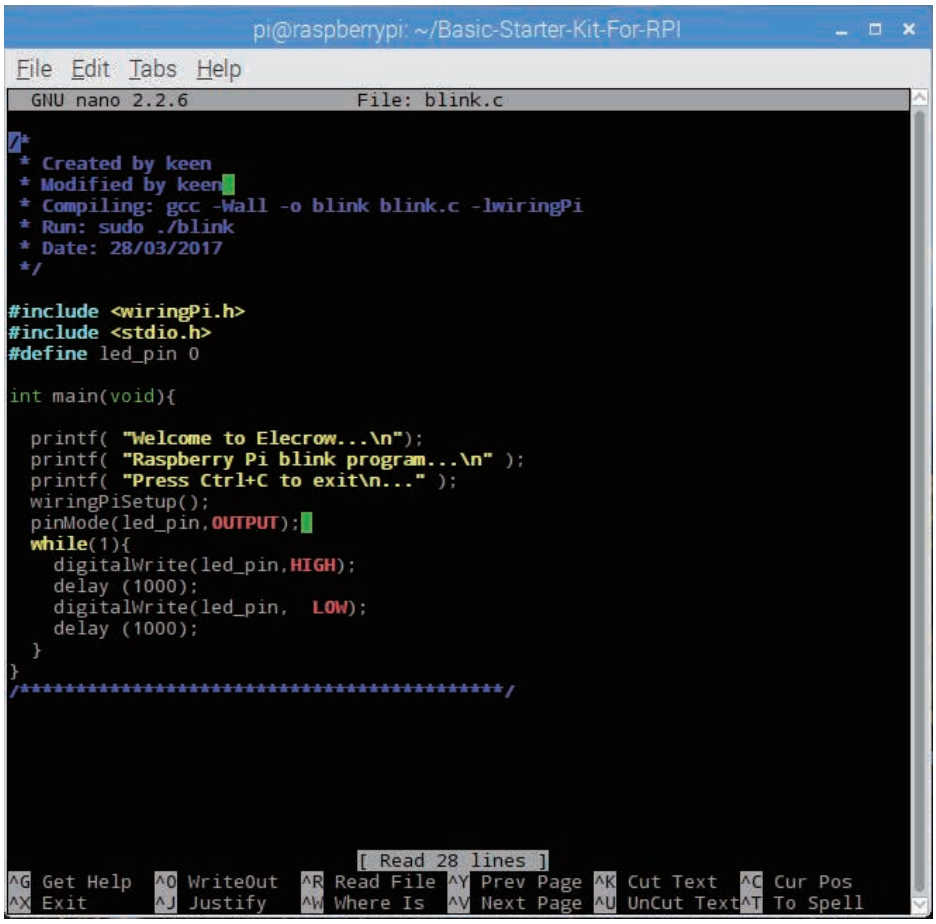



## Step 5: Enter Demo code

Run:
cd Basic-Starter-Kit
cd Lesson1_LED-blink




sudo nano blink.c




**Tips**: you can edit the Demo code.

The compiling is: gcc -Wall -o blink blink.c -lwiringPi

And run the program is: sudo ./blink

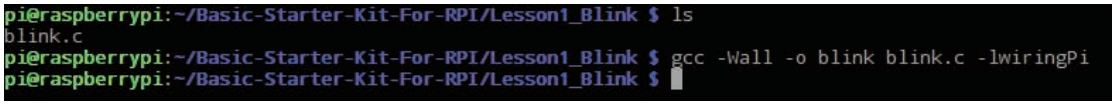
## Step 6: Exit and save

When you want to exit ,you need to “Ctrl+O” and “Ctrl+X”

## Step 7: Compiling

Run:

gcc -Wall -o blink blink.c -lwiringPi




Tips: if you want to compile “xxx.c” and you need run it by following this way.

Run: gcc -Wall -o xxx xxx.c -lwiringPi

Or: g++ -Wall -o xxx xxx.c -lwiringPi

## Step 8: Run the program

Run:
sudo ./blink

```
pi@raspberrypi:~/Basic-Starter-Kit-For-RPI $ ls
blink  blink.c
pi@raspberrypi:~/Basic-Starter-Kit-For-RPI $ sudo ./blink
Welcome to Elecrow...
Raspberry Pi blink program...
Press Ctrl+C to exit
```

Tips: Exit the Program -> “Ctrl+c”.

## Step 9: Application effect

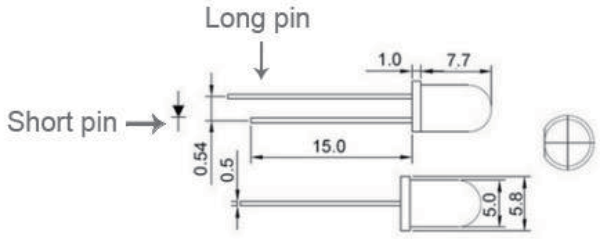
Turns on an LED on for one second, then off for one second, repeatedly.

# Lesson 1: LED blink

## Overview

The LED is designed for the beginners of RPI. It is the best way to step into the from RPI what it's I/O pins. The LED is the best choice to help you learn I/O pins.

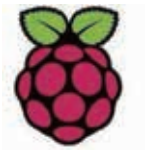
## Specification




## Pin definition

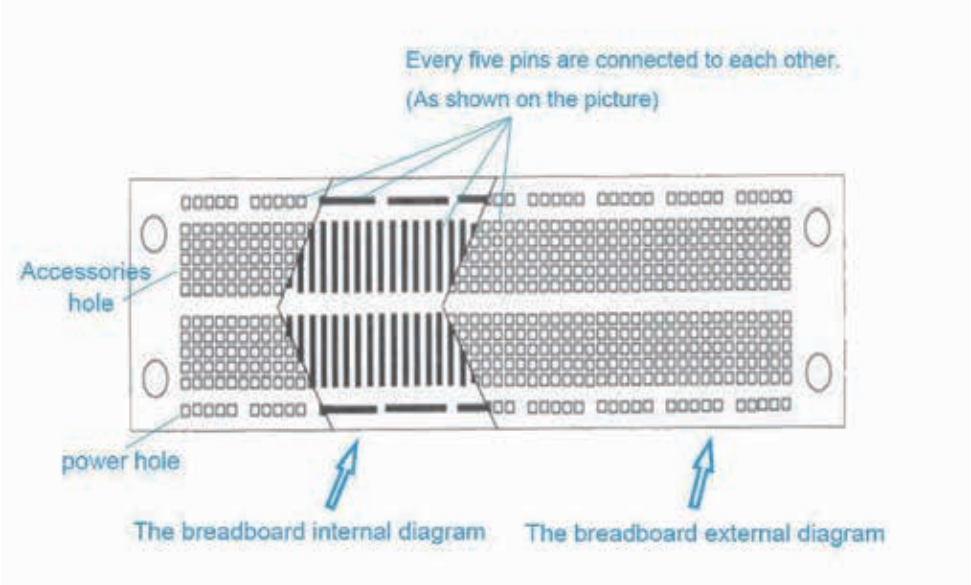
| | | |
|---|---|---|
| LED | | RPI |
| Long pin | -> | GPIO17 |
| Short pin | -> | GND |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | LED | 1 |
|  | 220/330Ω resistor | 1 |
|  | Raspberry Pi Board | 1 |

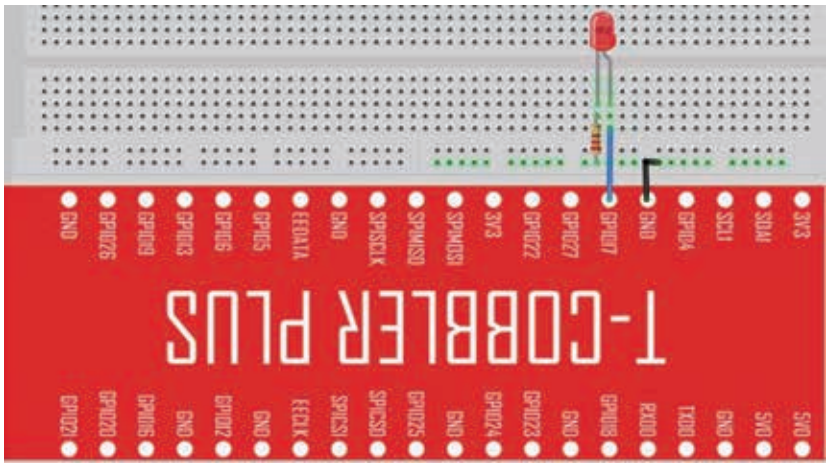
| | | |
|---|---|---|
|  | T-Cobbler Plus | 1 |
|  | 40P GPIO Cable | 1 |
|  | Breadboard | 1 |
|  | Jumper wires | Several |

## Breadboard schematic




All the tie points (indicated in the picture) of the different colors are connected together.

## Connection diagram




**Connection**:

| RPI | LED |
|---|---|
| GPIO17 | Long pin |
| GND | Short pin |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, turns on an LED on for one second, then off for one second, repeatedly.

# Lesson 2: Button

## Overview

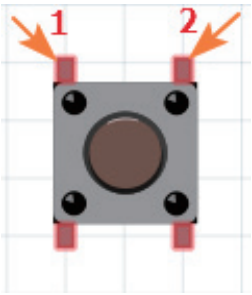
This lesson will teach you how to use button.

## Specification

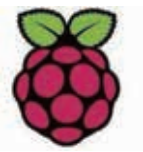
Size: 6 x 6 x 5mm
Temperature: -30 ~ +70 Centigrade

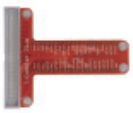
## Pin definition

It is the definition of Button pin：

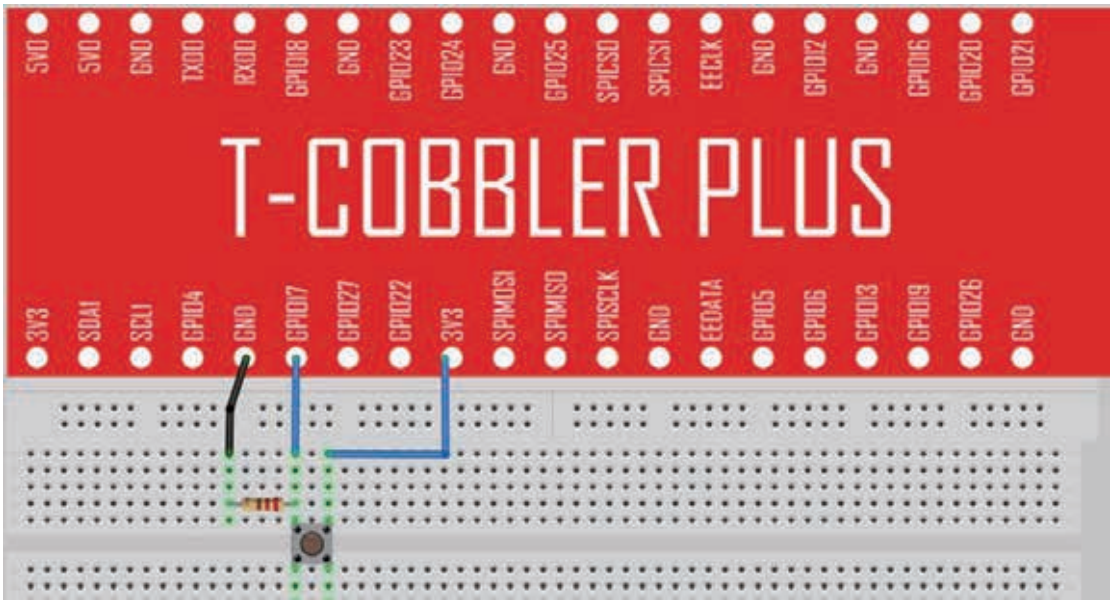



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Button | 1 |
|  | 10KΩ resistor | 1 |
|  | Raspberry Pi Board | 1 |

| | | |
|---|---|---|
|  | T-Cobbler Plus | 1 |
|  | 40P GPIO Cable | 1 |
|  | Breadboard | 1 |
|  | Jumper wires | Several |

## Connection diagram




Connection

| RPI | Button |
|---|---|
| GPIO17 | 1 |
| 3V3 | 2 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

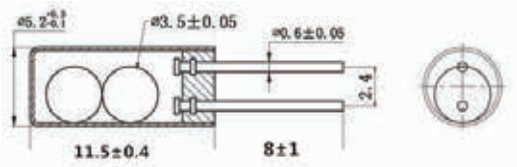
Running the program ,and pressing the button, the screen will show the state of the button.

# Lesson 3: Ball switch

## Overview

This lesson will teach you how to use ball module, which is simple and easy to use.

## Specification



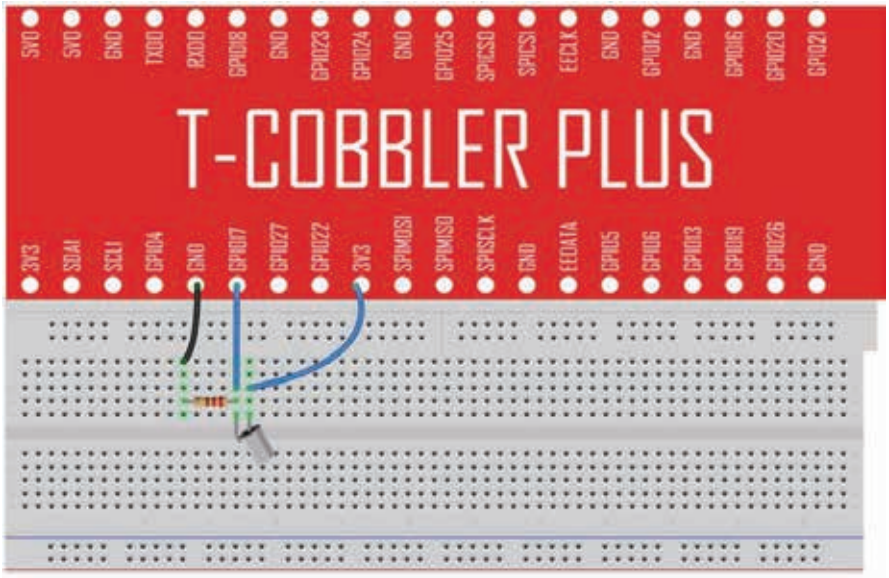

## Pin definition

Pin non polarity.

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Ball Switch | 1 |
|  | 10KΩ resistor | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| RPI | Ball Switch |
|---|---|
| GPIO17 | pin 1 |
| 3v3 | pin 2 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program ,and shocking the ball switch, the screen will show the state of the ball switch.

# Lesson 4: Active buzzer

## Overview


This is an active buzzer experiment. Active means that the direct power supply can make a sound.

## Specification

Voltage: DC 5V
Min Sound Output at 10cm: 85dB;
Total Size (Pin Not Included): 12 x 9mm/0.47" x 0.35"(D*H)

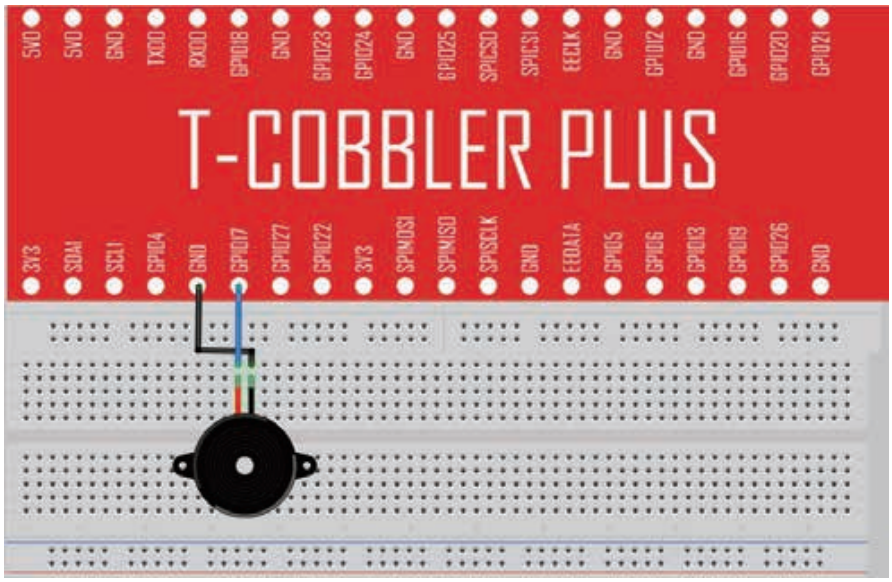
## Pin definition

| Active Buzzer | | RPI |
|---|---|---|
| Long pin/+ | -> | GPIO17 |
| Short pin | -> | GND |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Active buzzer | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| | | |
|---|---|---|
| Active Buzzer | | RPI |
| Long pin/+ | -> | GPIO17 |
| Short pin | -> | GND |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, the buzzer will be ringing.

# Lesson 5: Passive buzzer

## Overview

This lesson will teach you how to use Passive buzzer, which is simple and easy to use.

## Specification

Working Voltage: 3V/5V
Resistance: 16Ohm
Resonance Frequency: 2KHZ

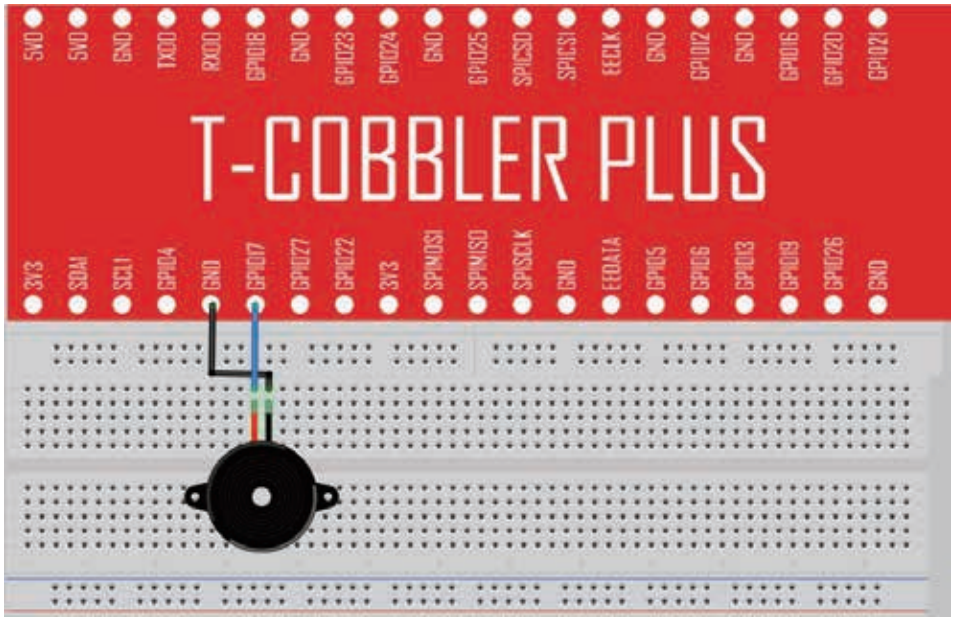
## Pin definition

| Passive Buzzer | | RPI |
|---|---|---|
| Long pin/+ | -> | GPIO17 |
| Short pin | -> | GND |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Passive buzzer | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| Passive Buzzer | | RPI |
|---|---|---|
| Long pin/+ | -> | GPIO17 |
| Short pin | -> | GND |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, the buzzer will be ringing.

# Lesson 6: Relay module

## Overview

This lesson will teach you how to use Relay module, which is simple and easy to use.

## Specification

Null

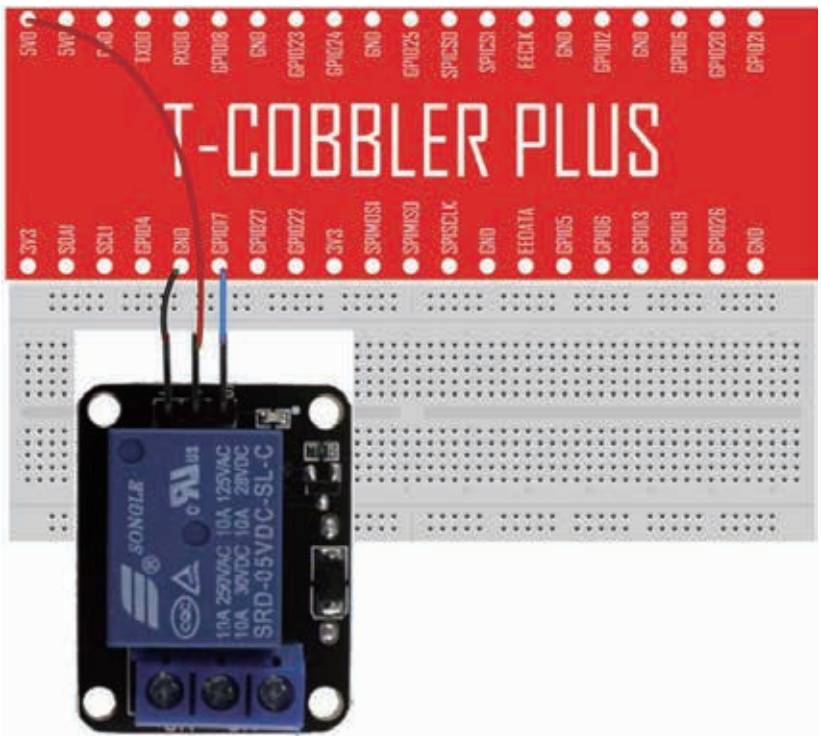
## Pin definition

| RPI | Relay Module |
|---|---|
| GPIO17 | S |
| 5v0 | + |
| GND | - |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Relay Module | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| RPI | Relay Module |
|---|---|
| GPIO17 | S |
| 5V0 | + |
| GND | - |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, according to the screen tips control relay.

# Lesson 7: RGB LED

## Overview

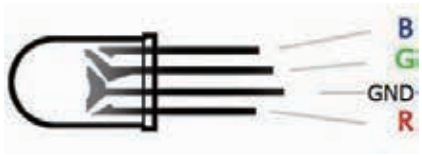
This lesson will teach you how to use a RGB (Red Green Blue) LED with an RPI, which is simple and easy to use.

## Specification

RGB led:

Emitting Light Color: Blue, Red, Green
Size(Approx): 5 x 35mm/ 0.2" x 1.37" (D * L)
Forward Voltage: 3.0-3.4V
Luminous Intensity: 12000-14000mcd

## Pin definition



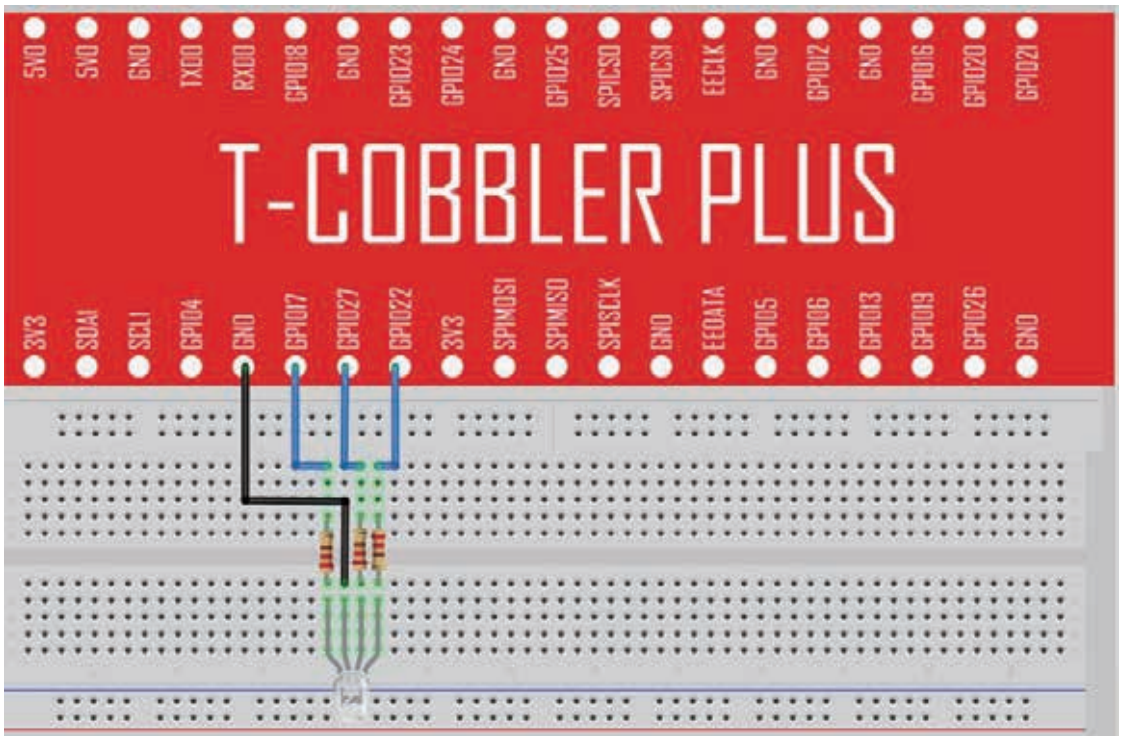

| RGB LED | | RPI |
|---|---|---|
| R | -> | GPIO17 |
| GND | -> | GND |
| G | -> | GPIO27 |
| B | -> | GPIO22 |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | RGB LED | 1 |
| | 220/330Ω resistor | 3 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |

| | | |
|---|---|---|
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

RGB LED RPI
R -> GPIO17
GND -> GND
G -> GPIO27
B -> GPIO22

## Compile and Run

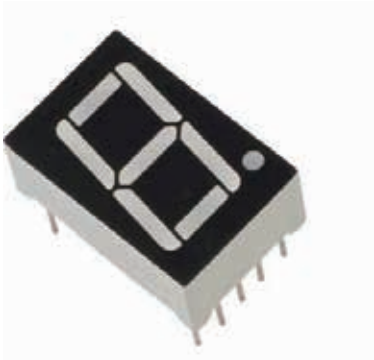
Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the LED loop emit 6 different colors of light.

# Lesson 8: 1 digit 7 Segment Displays

## Overview

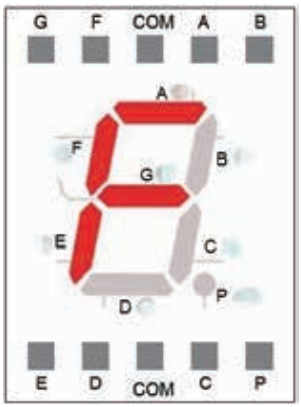


This experiment is similar to the LED experiment, the same is the control of LED, but the experiment can achieve time counting function.

## Specification

Null

## Pin definition

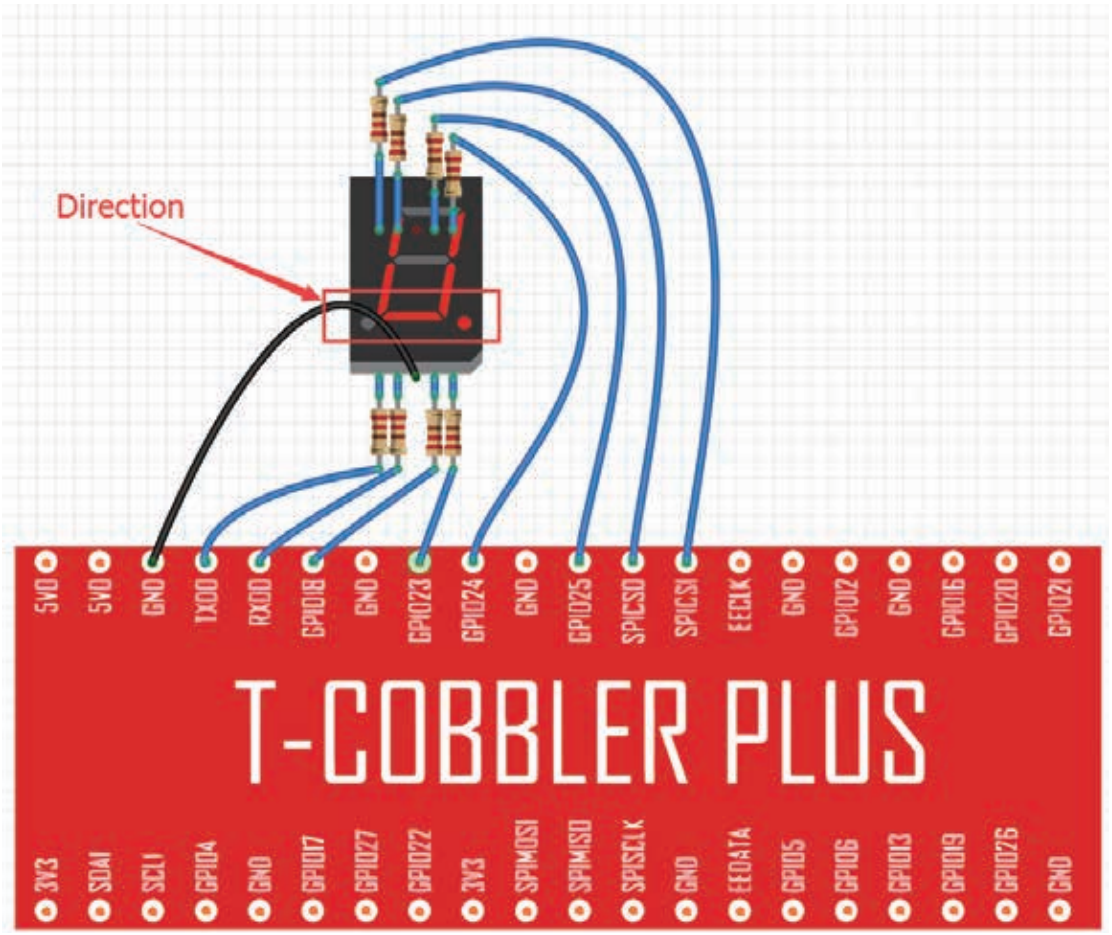



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | 1 Digit 7 Segment displays | 1 |
| | 220/330Ω | 8 |
| | Raspberry Pi Board | 1 |

| | | |
|---|---|---|
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Note： Pay attention to the direction of digital tube.

Connection:

| RPI | | SEG |
|---|---|---|
| TXD0 | -> | P |
| RXD0 | -> | C |
| GPIO18 | -> | D |
| GPIO23 | -> | E |
| GPIO24 | -> | G |
| GPIO25 | -> | F |
| SPICSO | -> | A |
| SPICSI | -> | B |
| GND | -> | COM |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

You will see the number on the digital tube increased from 0 to 9.

# Lesson 9: 4 digit 7 Segment Displays

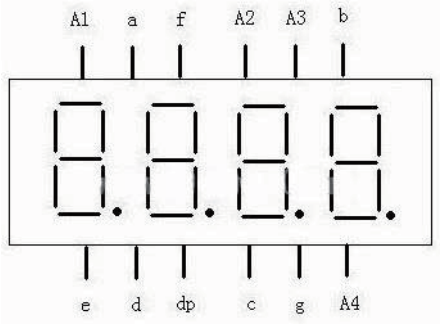
## Overview


This experiment is similar to the LED experiment, the same is the control of LED, but the experiment can achieve time counting function.

## Specification

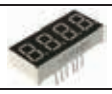
Null

## Pin definition

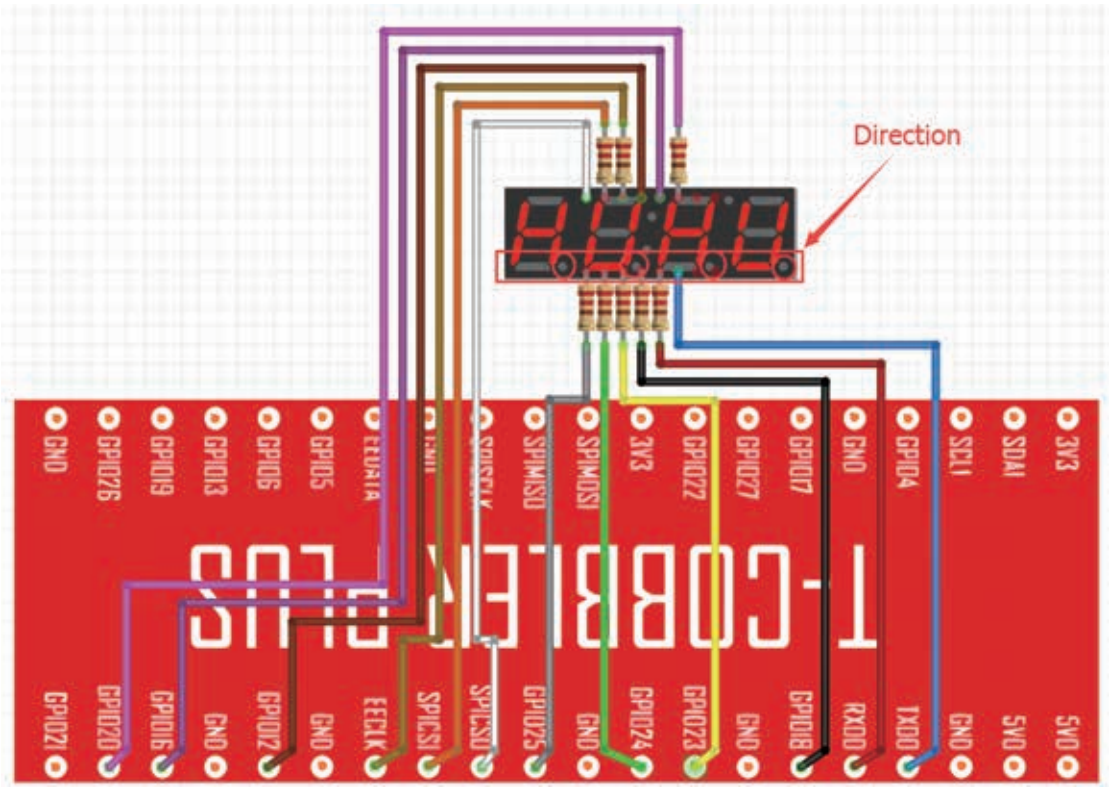



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|

| | | |
|---|---|---|
|  | 4 Digit 7 Segment displays | 1 |
| | 220/330Ω | 8 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Note： Pay attention to the direction of digital tube.

Connection:

| RPI | | SEG |
|---|---|---|
| TXD0 | -> | A4 |
| RXD0 | -> | g |
| GPIO18 | -> | c |
| GPIO23 | -> | DP |
| GPIO24 | -> | d |
| GPIO25 | -> | e |
| SPICSO | -> | A1 |
| SPICSI | -> | a |

| | | |
|---|---|---|
| EECLK | -> | f |
| GPIO12 | -> | A2 |
| GPIO16 | -> | A3 |
| GPIO20 | -> | b |

### Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

### Application effect

Running the program, you will see the number of digital tube display increasingly.

# Lesson 10: Heart-shaped display experiment

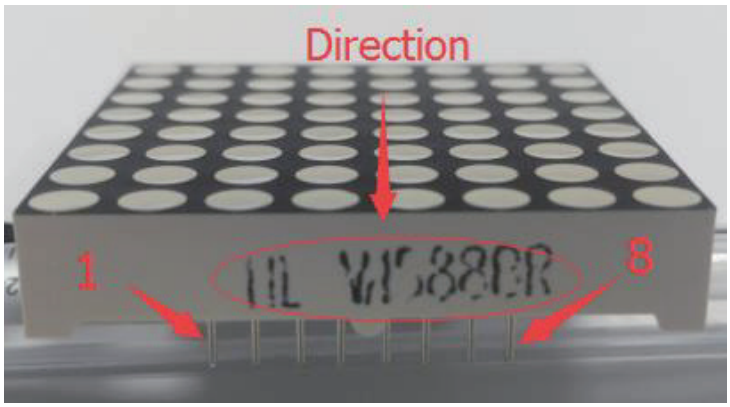
### Overview

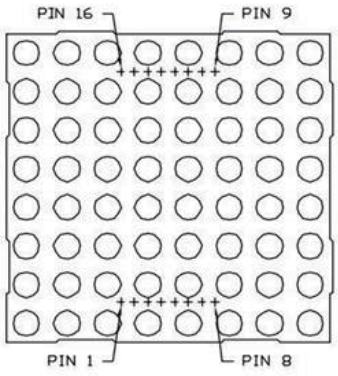
This experiment using 8*8 dot matrix display a beating heart animation.

### Specification

Please view 1588 ABxx.pdf.
Path: \Datasheet\1588 ABxx.pdf

## Pin definition

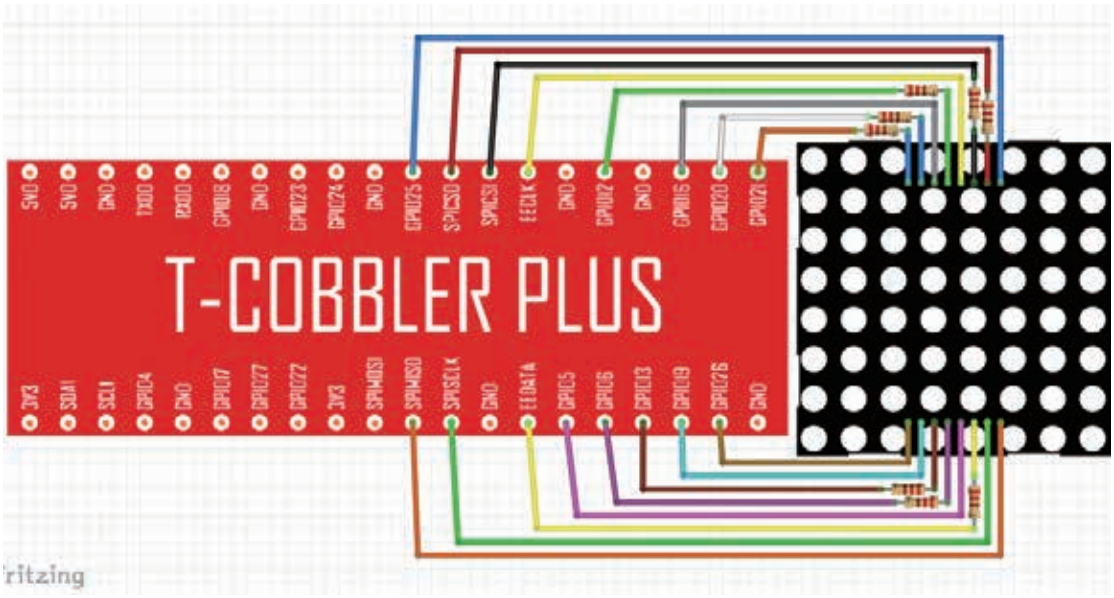



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | LED matrix | 1 |
| | 220/330Ω | 8 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection:**

LED Matrix

| | | |
|---|---|---|
| pin1 | -> | GPIO26 |
| pin2 | -> | GPIO19 |
| pin3 | -> | GPIO13 |
| pin4 | -> | GPIO6 |
| pin5 | -> | GPIO5 |
| pin6 | -> | EEDATA |
| pin7 | -> | SPISCLK |
| pin8 | -> | SPIMISO |
| pin9 | -> | GPIO25 |
| pin10 | -> | SPICSO |
| pin11 | -> | SPICSI |
| pin12 | -> | EECLK |
| pin13 | -> | GPIO12 |
| pin14 | -> | GPIO16 |
| pin15 | -> | GPIO20 |
| pin16 | -> | GPIO21 |

**Note：** Part of the pin needs to be connected to a resistor.

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Please ensure that the connection correct, then run the program, you will see the heart beating animation.

# Lesson 11: 9G servo

## Overview

This lesson will teach you how to use 9G servo. Sweeps the shaft of a RC servo motor back and forth across 180 degrees.

## Specification

Please view SG90Servo-datasheet.pdf.
Path: \Datasheet\ SG90Servo-datasheet.pdf

## Pin definition

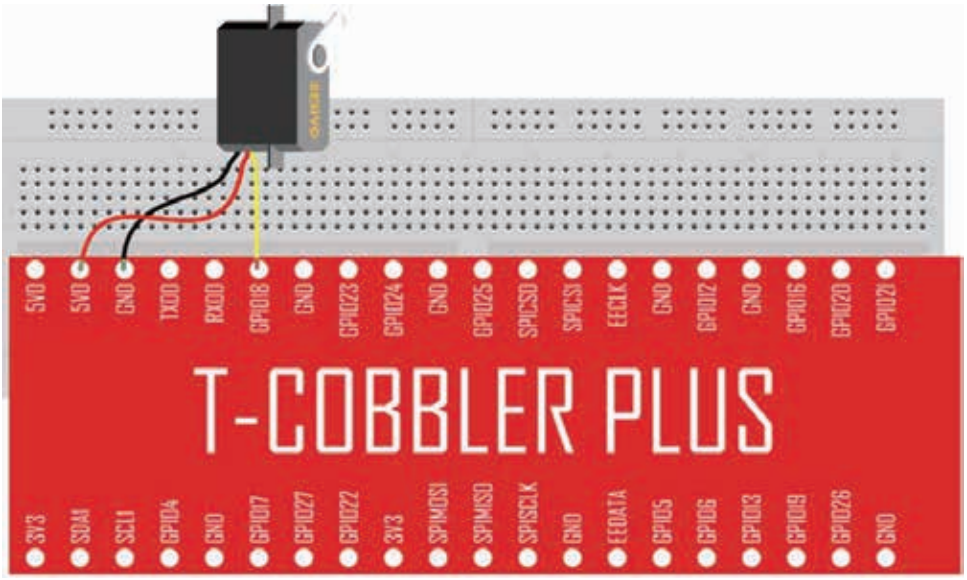

| RPI | Servo Motor |
|---|---|
| GPIO18 | Yellow Wire |
| 5V0 | Red Wire |
| GND | Black Wire |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Servo Motor | 1 |

| | | |
|---|---|---|
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| | |
|---|---|
| RPI | Servo Motor |
| GPIO18 | Yellow Wire |
| 5V0 | Red Wire |
| GND | Black Wire |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, then you can control the Servo motor by the screen tips.

# Lesson 12: Stepper Motor

## Overview

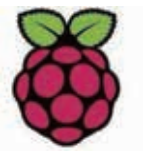
This lesson will teach you how to use Stepper Motor.

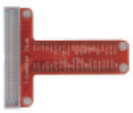
## Specification

Please view “Stepper-Motor.pdf”
Path: \Datasheet\ Stepper-Motor.pdf

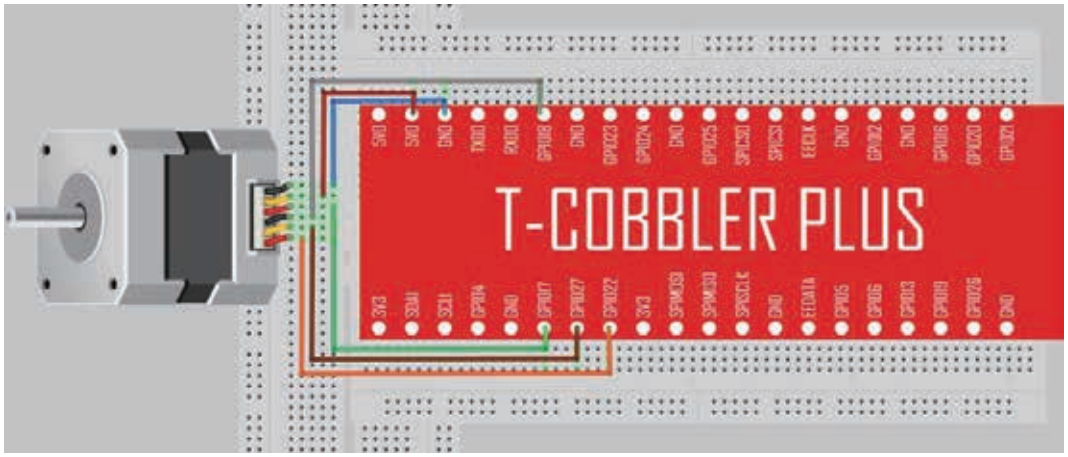
## Pin definition


## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Step motor | 1 |
|  | ULN2003 step motor driver board | 1 |
|  | Raspberry Pi Board | 1 |

| | | |
|---|---|---|
|  | T-Cobbler Plus | 1 |
|  | 40P GPIO Cable | 1 |
|  | Breadboard | 1 |
|  | Jumper wires | Several |

## Connection diagram




**Connection**

| RPI | Stepper Motor |
|---|---|
| GPIO17 | IN1 |
| GPIO18 | IN2 |
| GPIO27 | IN3 |
| GPIO22 | IN4 |
| “+” | 5V0 |
| “-” | GND |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, then you can control the relay by the screen tips.

# Lesson 13: Ultrasonic ranging

## Overview

This lesson will teach you how to use HC-SR04 module to test distance. It is generally used in the robot.

## Specification

Please view “HCSR04.pdf”
Path: \Datasheet\ HCSR04.pdf

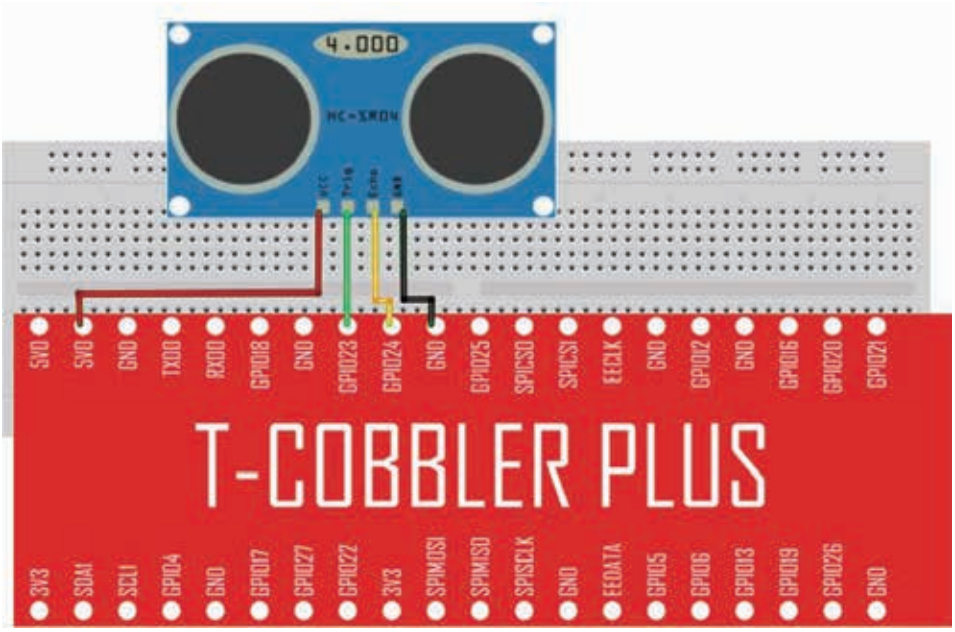
## Pin definition

| HC SR04 | | RPI |
|---|---|---|
| Vcc | -> | 5V0 |
| Trig | -> | GPIO23 |
| Echo | -> | GPIO24 |
| Gnd | -> | GND |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | HCSR04 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| HC SR04 | | RPI |
|---|---|---|
| Vcc | -> | 5V0 |
| Trig | -> | GPIO23 |
| Echo | -> | GPIO24 |
| Gnd | -> | GND |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the parameters returned by the ultrasonic module.

# Lesson 14: Touch Lamp

## Overview


This is a touch sensor to control the LED lamp experiment, it can control each LED light, but also can achieve the effect of breathing light.

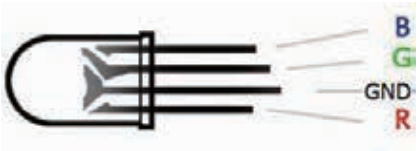
## Specification

Null

## Pin definition

Touch sensor

| | | |
|---|---|---|
| GND | -> | GND |
| VCC | -> | 5V |
| SIG | -> | data |

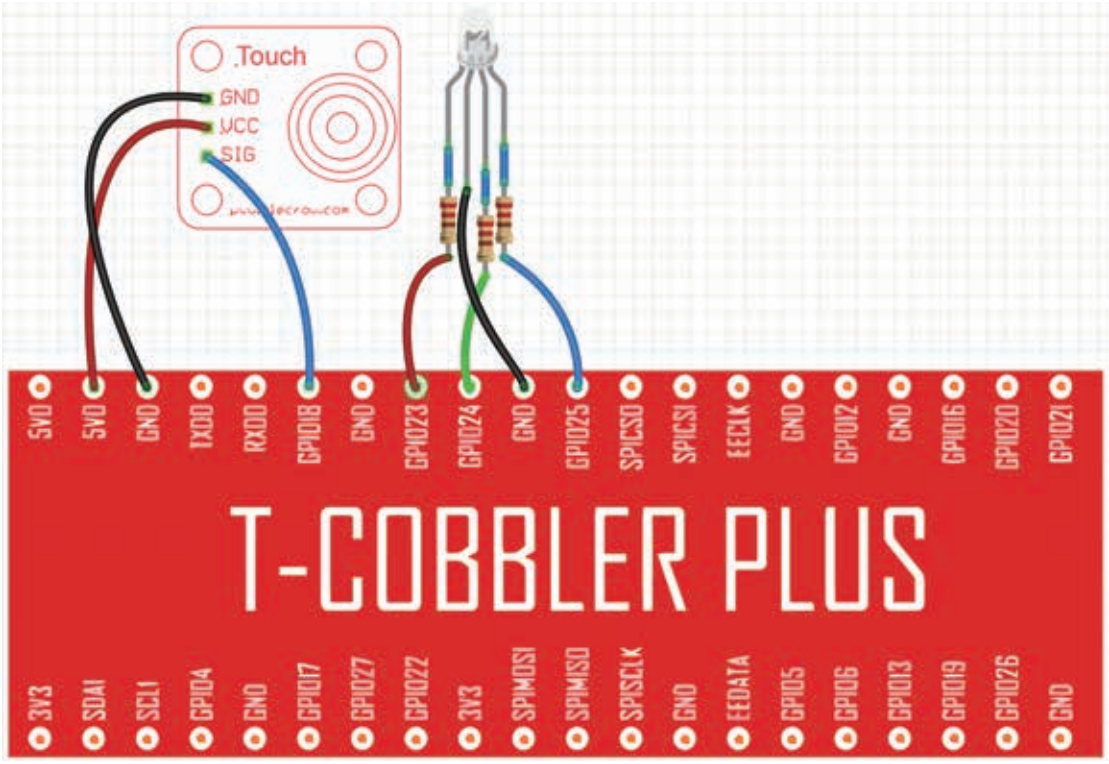



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Touch Sensor | 1 |
| | RGB LED | 1 |
| | 220/330Ω | 3 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |

| | | |
|---|---|---|
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Touch sensor RPI
GND -> GND
VCC -> 5V0
SIG -> GPIO18
RGB LED
R -> GPIO23
GND -> GND
G -> GPIO24
B -> GPIO25

## Compile and Run

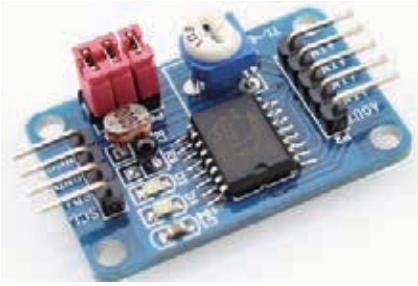
Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Through the touch panel, you can control the LED light.

# Lesson 15: PCF8591 Module

## Overview



The PCF8591 module an 8-bit A/D Converter & D/A Converter PCF8591 with four analog inputs, one analog output and a serial I2C-bus interface.

The PCF8591 module features I2C pinheader on one side, and I2C connector on the opposite side. Hence, it's more flexible to connect the board to your development system. The board also supports I2C cascading, allowing the use of multi module connected to the I2C bus at the same time by connecting the pinheader and connector.

## Specification

Please view “PCF8591-datasheet.pdf”
Path: \ Datasheet\ PCF8591-datasheet.pdf

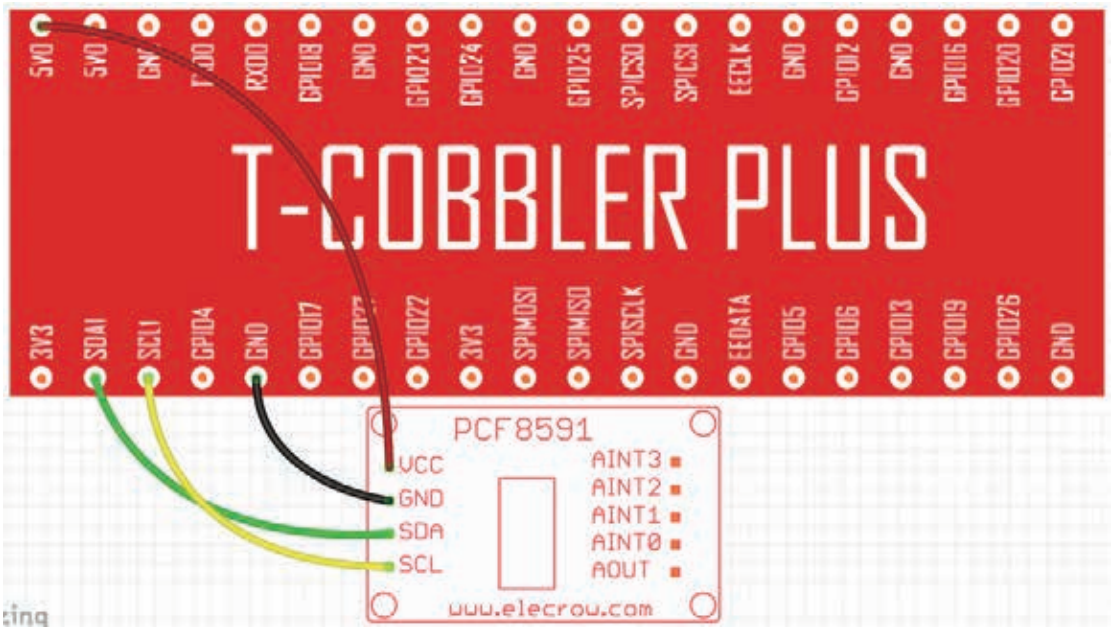
## Pin definition

P4 -> Select P4, the thermistor to be use.
P5 -> Select P5, the photosistence to be use.
P6 -> Select P6, the potentiometer to be use.

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Tips: Select photosistence, so we need to use P5, P4 sand P6 is null.

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).
If print: Unable to open I2C device

You need to open IIC. Enter Desktop: Preferences->Raspberry Pi Configuration->Interfaces->Enabled I2C->reboot.

## Application effect

Running the program, you will see the analog value of photosistence sensor.

# Lesson 16: Flame Sensor

## Overview

The flame sensor can be used to detect fire or other wavelength at 760 nm ~ 1100 nm light. In the fire-fighting robot game, the flame plays an important role in the probe, which can be used as the robot's eyes to find fire source or football. It can make use of fire-fighting robots, soccer robots.
The flame sensor's operating temperature is -25 degrees Celsius to 85 degrees Celsius.

## Specification

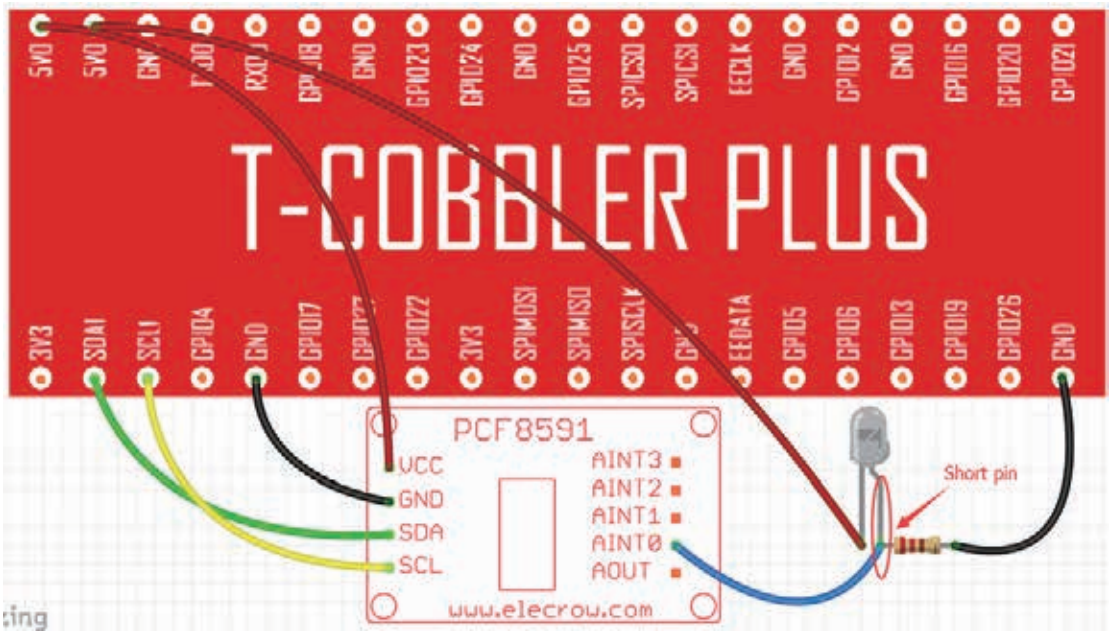
Null

## Pin definition


## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Flame sensor | 1 |
|  | 10KΩ resistor | 1 |
| | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Tips: P4, P5 and P6 is null.

Flame sensor

| | | |
|---|---|---|
| Short Pin | -> | 5V0 |
| Long Pin | -> | AINT0 |

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

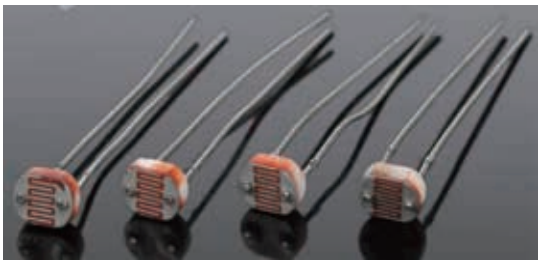
Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the analog value of Flame sensor.

# Lesson 17: Photoresistance Sensor

## Overview



As the resistance of the sensor varies depending on the amount of light it is exposed to, the output voltage changes with the ligth intensity. It can be used to trigger other modules.

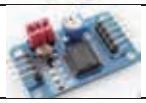
## Specification

Null

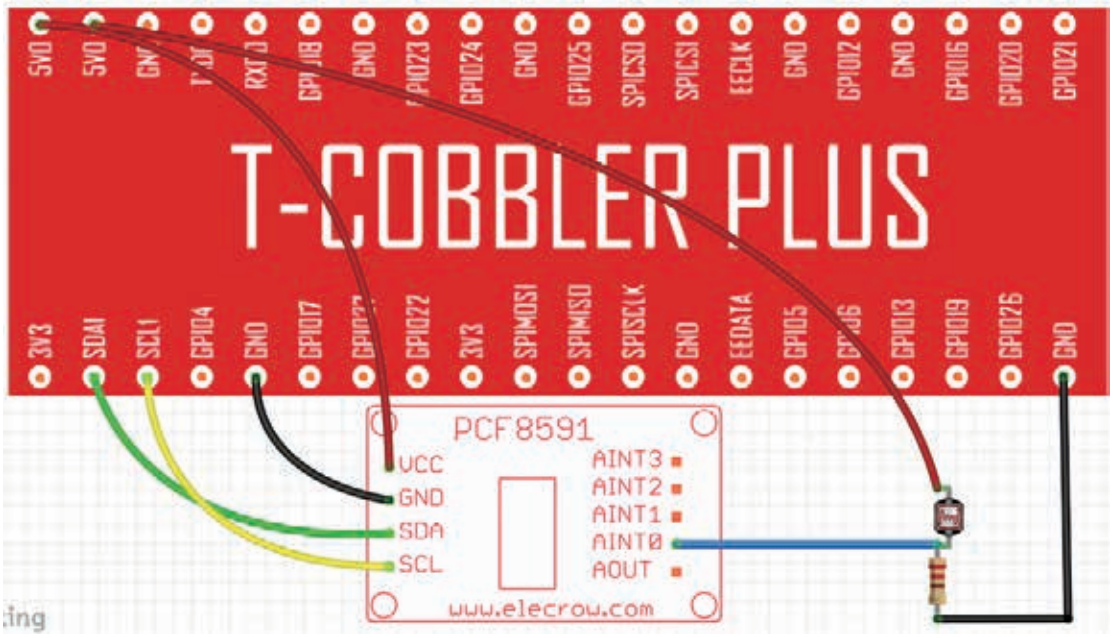
## Pin definition

Nonpolar.

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Photoresistance sensor | 1 |
|  | 10KΩ resistor | 1 |
|  | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Tips: P4, P5 and P6 is null.

Photoresistance sensor

| | | |
|---|---|---|
| Pin1 | -> | 5V0 |
| Pin2 | -> | AINT0 |

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the analog value of Photoresistance sensor.

# Lesson 18: Thermistor Sensor

## Overview

The resistance of a thermistor increases when the ambient temperature decreases, so the RPI can detects the voltage and thus to caculate the current temperature.The detection range of this sensor is between -40 to 125 degrees Celsius with an accuracy of ±1.5℃.

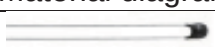
## Specification

Model: MF52-103
Insulation Material: Ceramic
Color: Black
Rated Power: 0.05W
Resistance Value: 10k
Resistance Tolerance: H (Â±3%)
B Value: 3950K
Pin Pitch: 1.5mm / 0.059"

## Pin definition

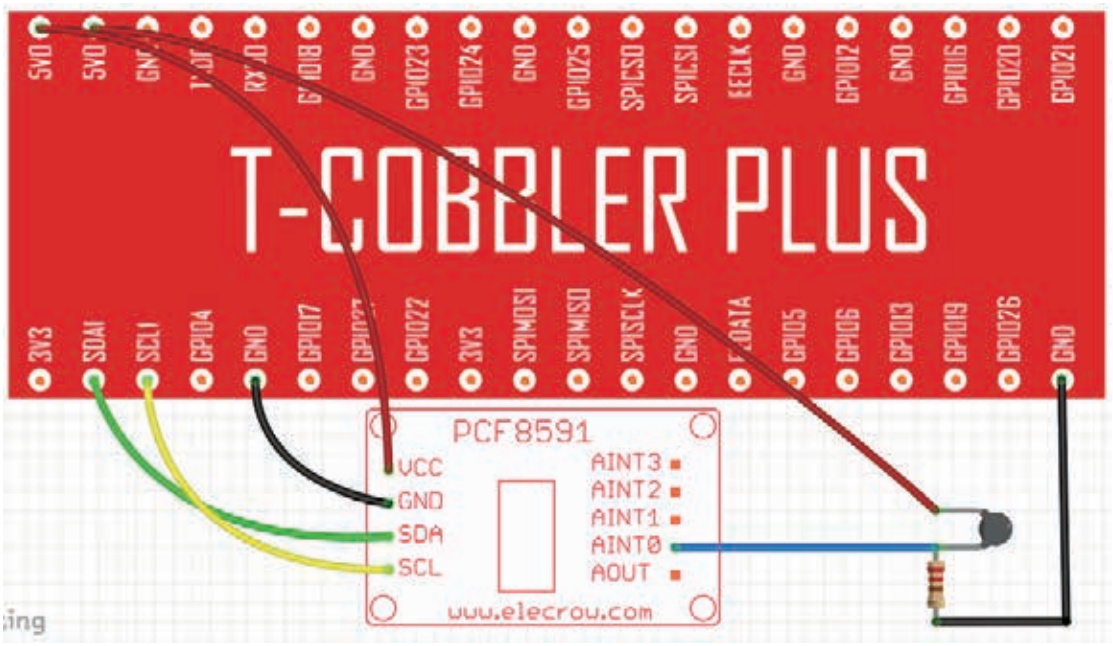
Nonpolar.

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Thermistor sensor | 1 |
|  | 10KΩ resistor | 1 |
| | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |

| | | |
|---|---|---|
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Tips:** P4, P5 and P6 is null.

Photoresistance sensor

| | | |
|---|---|---|
| Pin1 | -> | 5V0 |
| Pin2 | -> | AINT0 |

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the analog value of Thermistor sensor.

# Lesson 19: Potentiometer

## Overview

In this example, we use a potentiometer, we read its value using one analog input of an RPI board and we change the blink rate of the built-in LED accordingly. The resistor's analog value is read as a voltage because this is how the analog inputs work.

## Specification

Product Name: Potentiometer;
Resistance Value: 10K ohm;
Adjustment Type: Top Adjustment

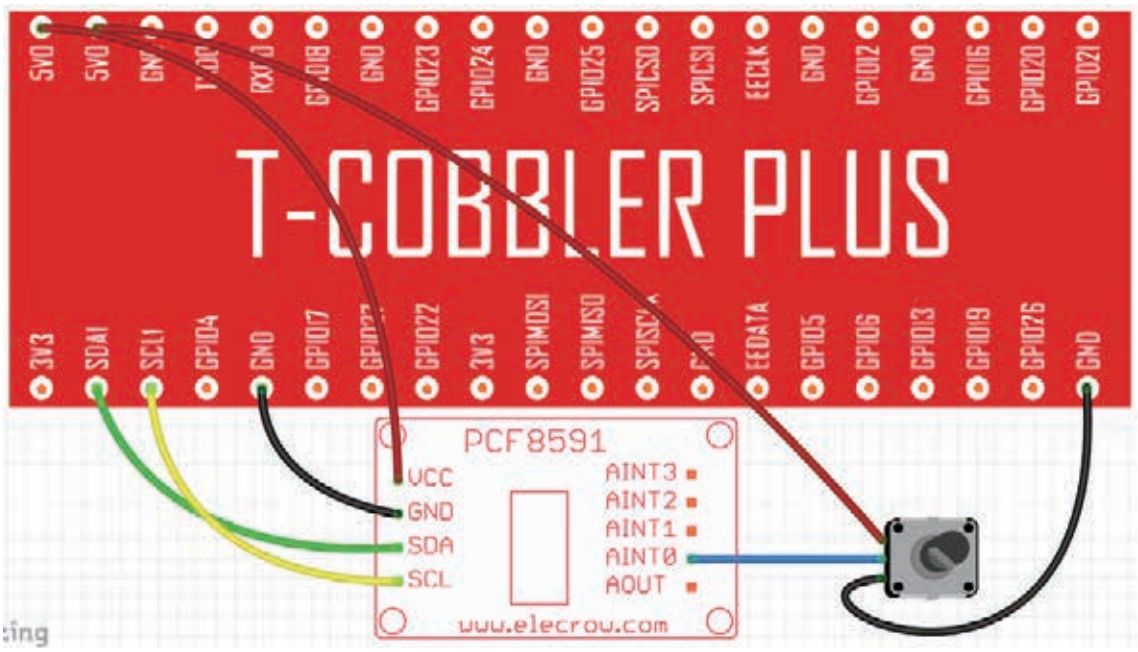
## Pin definition

Null

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | 10KΩ Potentiometer | 1 |
| | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Tips:** P4, P5 and P6 is null.

| | | |
|---|---|---|
| Potentiometer sensor | | |
| Up Pin | -> | 5V0 |
| Mid Pin | -> | AINT0 |
| Down Pin | -> | GND |
| PCF 8591 | | |
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, and rotating the Potentiometer that you will see the analog value of Potentiometer .

# Lesson 20: Water level monitoring experiment

## Overview

This is a water level measurement experiment, it is relatively simple to achieve, only need to read the value of the analog port(A0 or others), and then converted to a percentage.

## Specification

Operating voltage: DC3-5V
Operating current: less than 20mA
Sensor Type: Analog
Production process: FR4 double-sided HASL
Humidity: 10% -90% non-condensing
Detection Area: 40mmx16mm
Product Dimensions: 62mmx20mmx8mm

## Pin definition

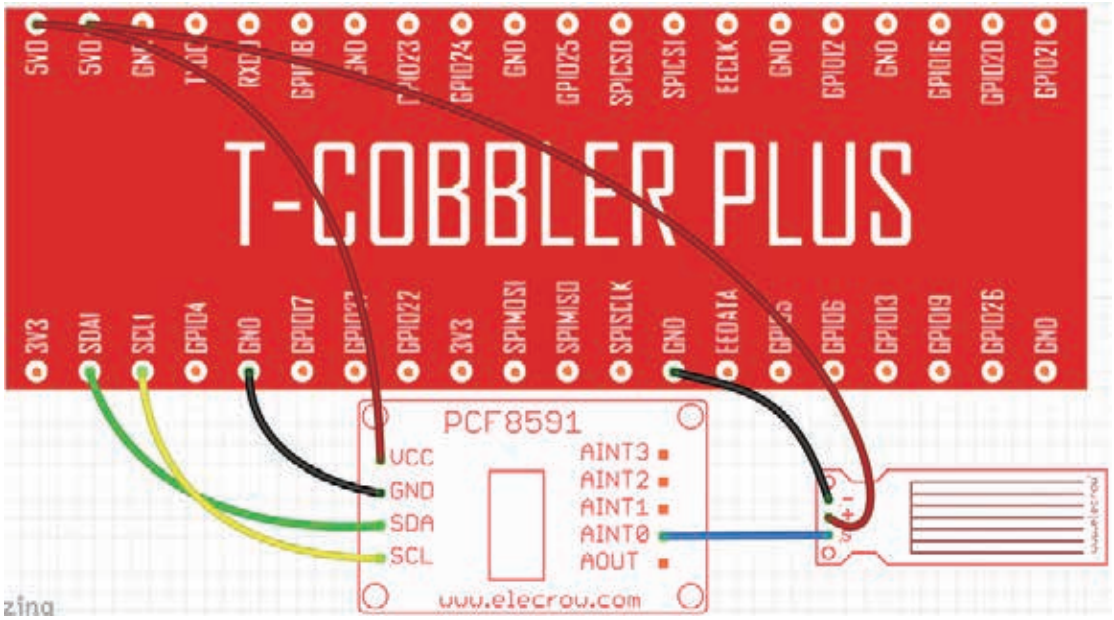
| | | |
|---|---|---|
| OUT | -> | analog data |
| ‘ + ’ | -> | VCC |
| ‘ - ’ | -> | GND |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Water Sensor | 1 |
| | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |

| | | |
|---|---|---|
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Tips:** P4, P5 and P6 is null.

Water sensor

| | | |
|---|---|---|
| - | -> | GND |
| + | -> | 5V0 |
| S | -> | AINT0 |

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, put the water level sensor enter water and you will see the analog value of water sensor.

# Lesson 21: Joystick experiment

## Overview


This experiment is to learn how to use the joystick of the analog output and digital output.

## Specification

Null.

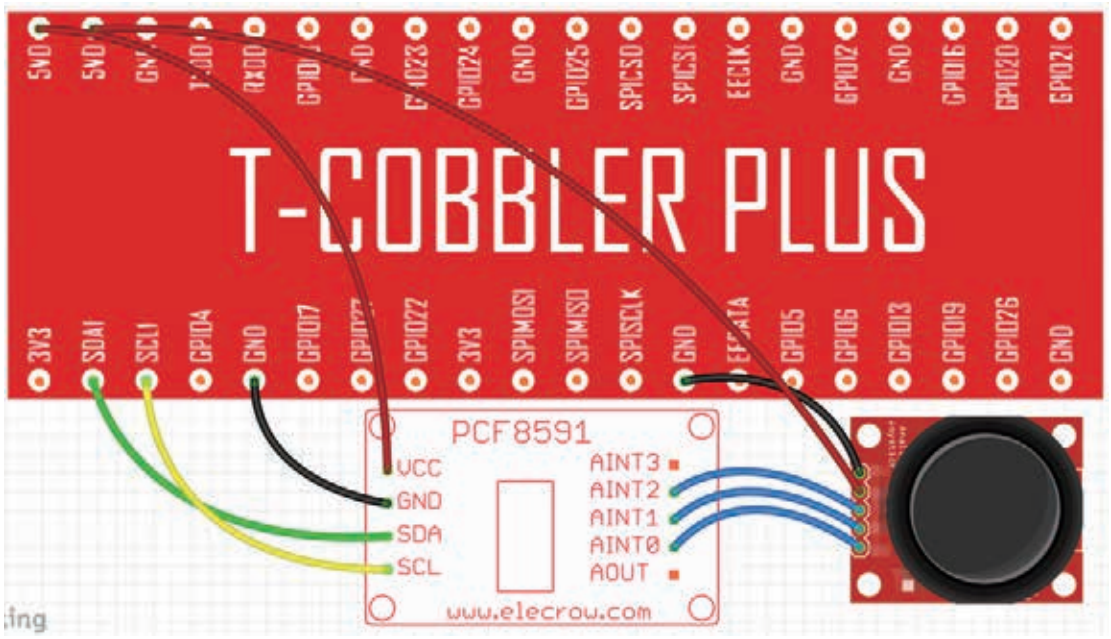
## Pin definition

| | | |
|---|---|---|
| GND | -> | GND |
| +5V | -> | VCC |
| VRx | -> | I/O |
| VRy | -> | I/O |
| SW | -> | I/O |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | Joystick Module | 1 |
| | PCF8591 | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Tips:** P4, P5 and P6 is null.

Joystick Module

| | | |
|---|---|---|
| GND | -> | GND |
| +5V | -> | 5V0 |
| VRX | -> | AINT2 |
| VRY | -> | AINT1 |
| SW | -> | AINT0 |

PCF 8591

| | | |
|---|---|---|
| VCC | -> | 5V0 |
| GND | -> | GND |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

By rotating or pressing the joystick, you will see the change in value.

# Lesson 22: IR remote control experiment

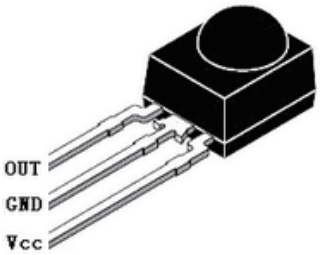
## Overview


In this lesson, we use the **lirc** library to read infrared signals returned by buttons of the remote control and translate them to button values. When a button is pressed, the IR transmitter in the remote control will send out the corresponding IR encoding signals. On the other side, when the IR receiver receives certain encoding signals, it will decode them to identify which button is pressed.

## Specification

IR Receiver:
Please view “IR Receiver-datasheet.pdf”
Path: \ Datasheet\ IR Receiver-datasheet.pdf

## Pin definition

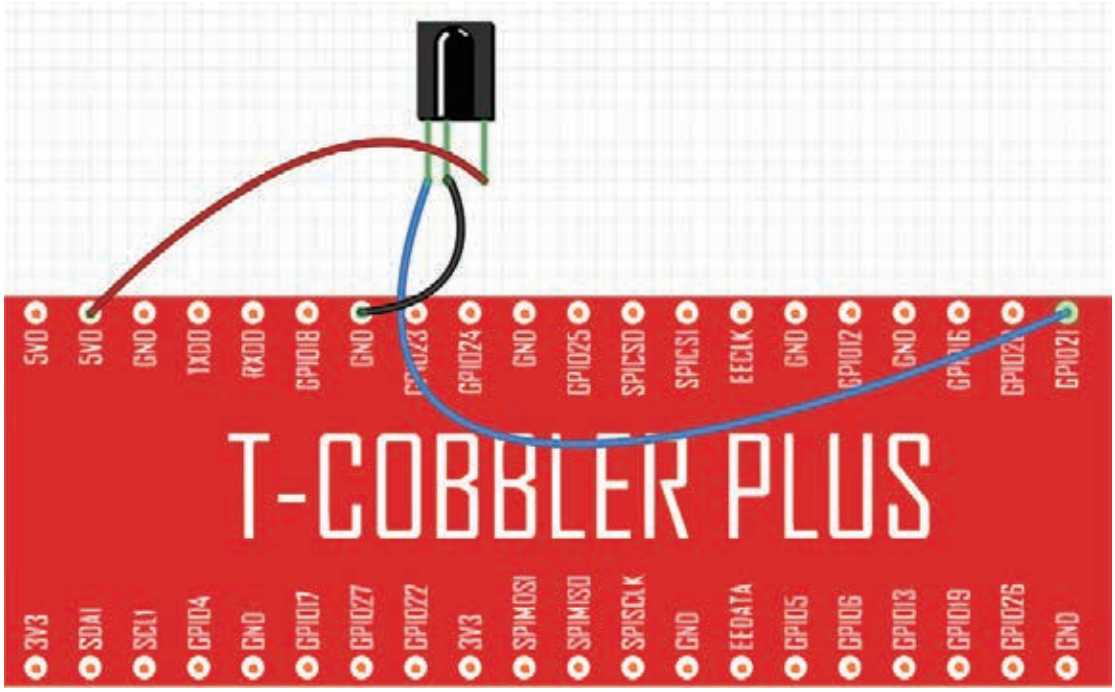



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | IR Remote | 1 |
|  | IR Receiver | 1 |
| | Raspberry Pi board | 1 |
| | T-Cobbler Plus | 1 |

| | | |
|---|---|---|
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Note： Please view **Pin definition.**

Connection

| RPI | | IR Receiver |
|---|---|---|
| GPIO21 | -> | OUT |
| GND | -> | GND |
| 5V0 | -> | VCC |

## Compile and Run

Open terminal and install the LIB of **lirc**, **run:**
git clone https://github.com/Elecrow-keen/Elecrow-lirc-setup.git

cd Elecrow-lirc-setup
sudo ./setup

And next, open the lesson of IR-Remote, compile and run.
**Tips:**

Compiling: gcc -Wall -o remote remote.c -lwiringPi -llirc_client

## Application effect

Running the program, press the button of the remote control, you will see that each button will have the corresponding coding.

# Lesson 23: IR remote control LED

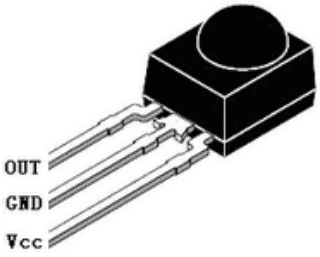
## Overview


In this lesson, we use the Remote to control a LED.

## Specification

IR Receiver:
Please view “IR Receiver-datasheet.pdf”
Path: \ Datasheet\ IR Receiver-datasheet.pdf

## Pin definition

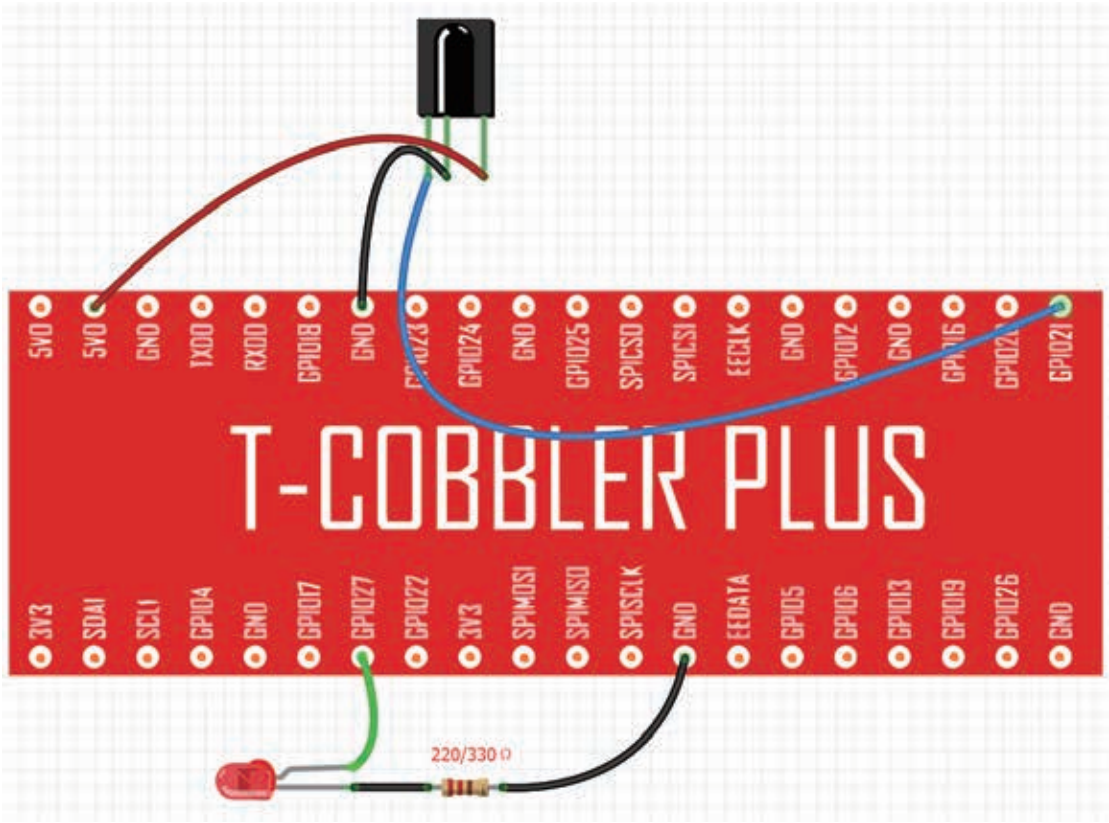



## Hardware required

| Material diagram | Material name | Number |
|---|---|---|

| | | |
|---|---|---|
|  | IR Remote | 1 |
|  | IR Receiver | 1 |
|  | LED | 1 |
|  | 220/330Ω resistor | 1 |
| | Raspberry Pi board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




Note： Please view **Pin definition.**

Connection

| RPI | | IR Receiver |
|---|---|---|
| GPIO21 | -> | OUT |
| GND | -> | GND |
| 5V0 | -> | VCC |

## Compile and Run

(Tips: If you have already added, skip this step) Open terminal and install the LIB of **lirc**, **run:**
git clone https://github.com/Elecrow-keen/Elecrow-lirc-setup.git

cd Elecrow-lirc-setup
sudo ./setup

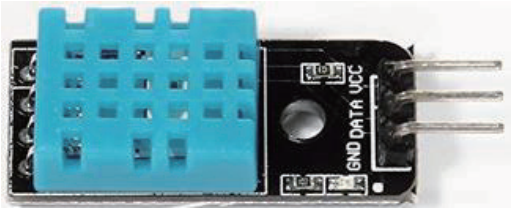
And next, open the lesson of IR-Remote, compile and run.

## Application effect

Running the program, press the button “CH-” or “CH+”, you will see the LED Open or Close.

# Lesson 24: DHT11 Experiment

## Overview




This lesson will teach you how to use DHT11 module, which is simple and easy to use.

## Specification

Please view DHT11-datasheet.pdf.
Path: \Datasheet\ DHT11-datasheet.pdf

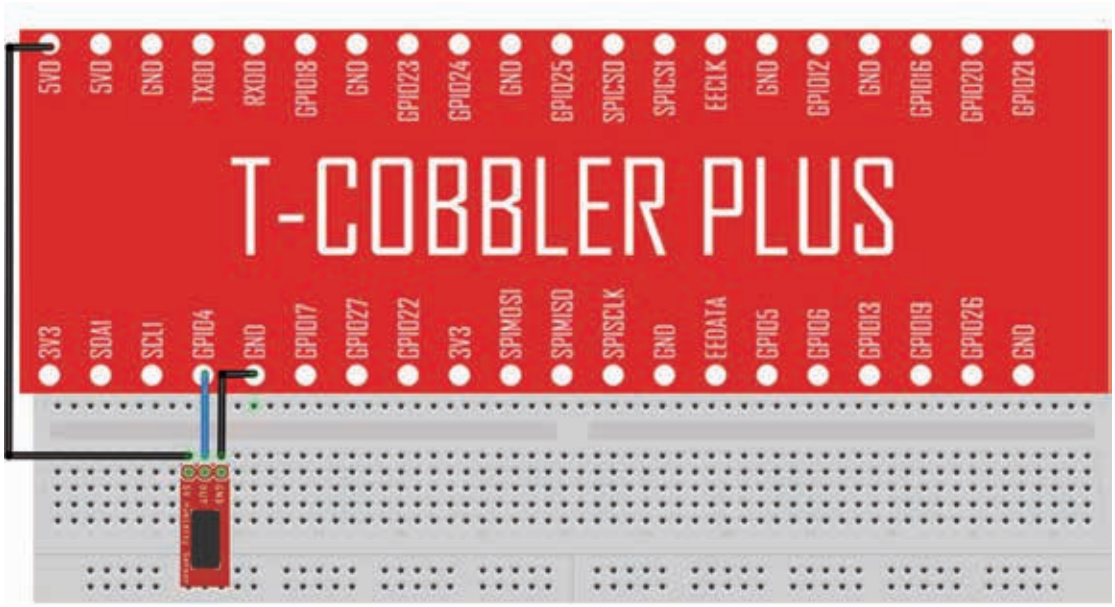
## Pin definition

| RPI | | DHT11 |
|---|---|---|
| GND | -> | GND/’-’ |
| GPIO4 | -> | DATA/’out’ |
| 5V0 | -> | VCC/’+’ |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | DHT11 Module | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| RPI | DHT11 |
|---|---|
| GND | GND/'-' |
| GPIO4 | DATA/'out' |
| 5V0 | VCC/'+' |

## Compile and Run

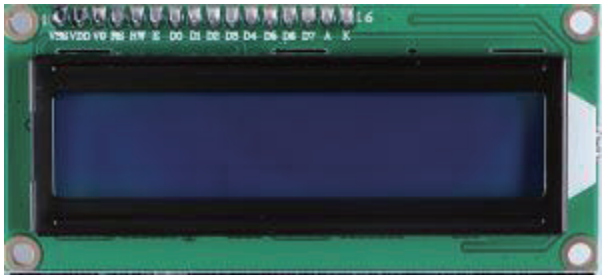
Tips: Refer to the operation demo (Step4 to Step7).

## Application effect

Running the program, you will see the parameters returned by the DHT11 module.

# Lesson 25: LCD1602 with IIC

## Overview



This is an experiment on how to use LCD1602 with IIC, the next lesson will do a temperature and humidity monitoring experiment.

## Specification

Please view LCD1602-datasheet.pdf.
Path: \Datasheet\LCD1602-datasheet.pdf

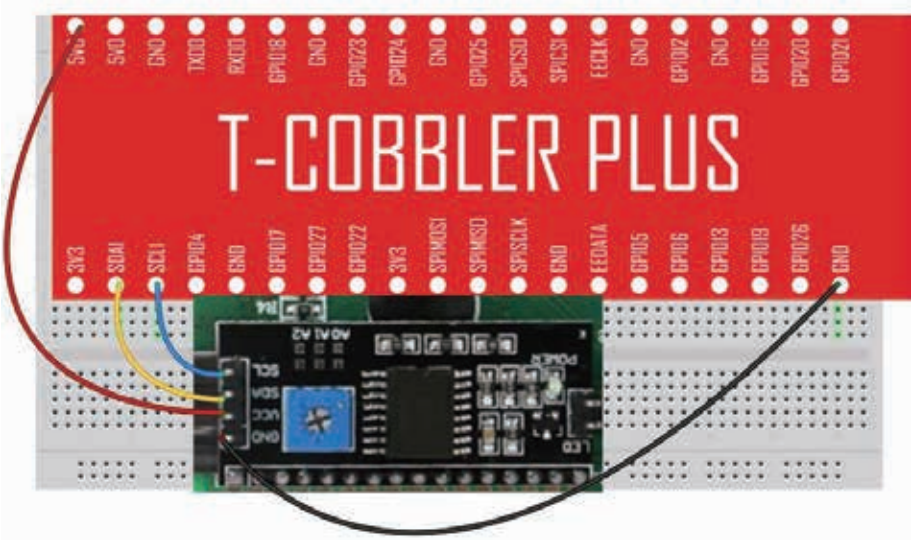
## Pin definition

| LCD1602 | | RPI |
|---|---|---|
| GND | -> | GND |
| VCC | -> | 5V0 |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | LCD1602 with IIC | 1 |
| | Raspberry Pi Board | 1 |
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| LCD1602 | | RPI |
|---|---|---|
| GND | -> | GND |
| VCC | -> | 5V0 |
| SDA | -> | SDA1 |
| SCL | -> | SCL1 |

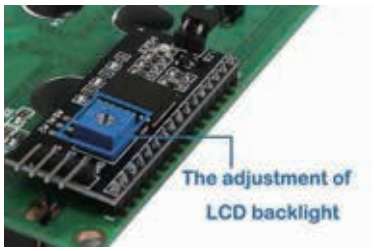
## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).
If print: Unable to open I2C device

You need to open IIC. Enter Desktop: Preferences->Raspberry Pi Configuration->Interfaces->Enabled I2C->reboot.

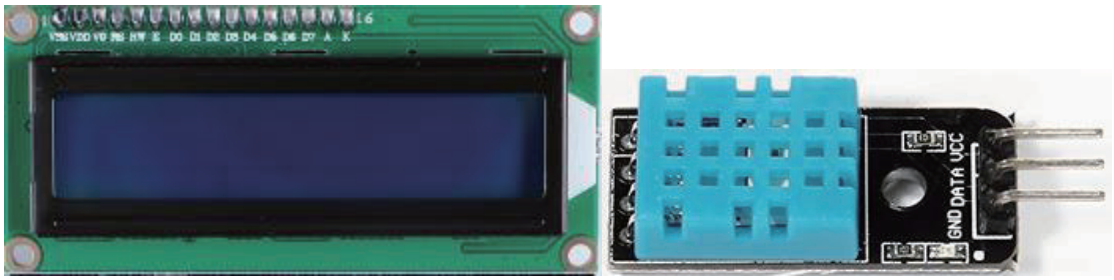
## Application effect

Running the program, LCD display string. If the LCD display is abnormal, please check the lines or adjust the potentiometer.



# Lesson 26: Temperature and humidity monitoring experiment

## Overview




This is a more complex experiment, it can realize the monitoring of indoor temperature and humidity, and in the LCD above display value.

## Specification

Please view LCD1602-datasheet.pdf、DHT11-datasheet.pdf and PCF8574.pdf.

Path: \Datasheet

## Pin definition

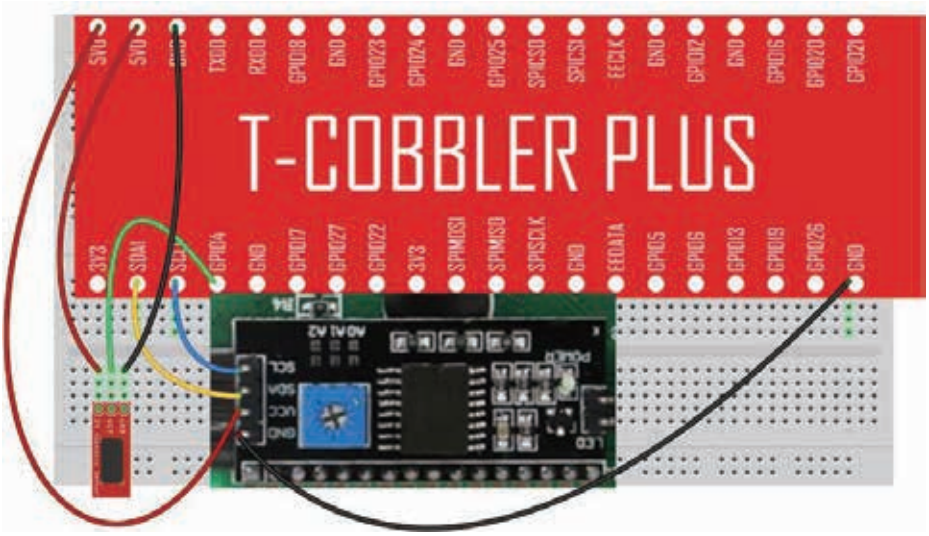
| RPI | | DHT11 |
|---|---|---|
| GND | -> | GND/'-' |
| GPIO4 | -> | DATA/'out' |
| 5V0 | -> | VCC/'+' |
| LCD1602 | | |
| GND | -> | GND |
| 5V0 | -> | VCC |
| SDA1 | -> | SDA |
| SCL1 | -> | SCL |

## Hardware required

| Material diagram | Material name | Number |
|---|---|---|
|  | DHT11 Module | 1 |
|  | LCD1602 with IIC | 1 |
| | Raspberry Pi Board | 1 |

| | | |
|---|---|---|
| | T-Cobbler Plus | 1 |
| | 40P GPIO Cable | 1 |
| | Breadboard | 1 |
| | Jumper wires | Several |

## Connection diagram




**Connection**

| RPI | | DHT11 |
|---|---|---|
| GND | -> | GND/'-' |
| GPIO4 | -> | DATA/'out' |
| 5V0 | -> | VCC/'+' |

| RPI | | LCD1602 |
|---|---|---|
| GND | -> | GND |
| 5V0 | -> | VCC |
| SDA1 | -> | SDA |
| SCL1 | -> | SCL |

## Compile and Run

Tips: Refer to the operation demo (Step4 to Step7).
If print: Unable to open I2C device

You need to open IIC. Enter Desktop: Preferences->Raspberry Pi Configuration->Interfaces->Enabled I2C->reboot.

## Application effect

Running the program, you will see the value of temperature and humidity on

the LCD.