Free your innovation

Freenove is an open-source electronics platform.
www.freenove.com

Warning

When you purchase or use Freenove products, please note the following:

- This product contains small parts. Swallowing or improper operation can cause serious infections and death. Seek immediate medical attention when the accident happened.
- Do not allow children under 3 years old to play with or near this product. Please place this product in where children under 3 years of age cannot reach.
- Do not allow children lack of ability of safe to use this product alone without parental care.
- Never use this product and its parts near any AC electrical outlet or other circuits to avoid the potential risk of electric shock.
- Never use this product near any liquid and fire.
- Keep conductive materials away from this product.
- Never store or use this product in any extreme environments such as extreme hot or cold, high humidity and etc.
- Remember to turn off circuits when not in use this product or when left.
- Do not touch any moving and rotating parts of this product while they are operating.
- Some parts of this product may become warm to touch when used in certain circuit designs. This is normal. Improper operation may cause excessively overheating.
- Using this product not in accordance with the specification may cause damage to the product.

Copyright

Contents

Preface

Freenove

Freenove is committed to provide high-quality products and services for customers, making it easy to get started with programing and electronics and launching innovative open source products. We value the user experience very much. Our services include:

- Electronic components and modules
- Learning kits for Arduino
- Learning kits for Raspberry Pi
- Learning kits for Technology
- Product customization service
- Robot kits

Please refer to:
http://www.freenove.com
http://www.freenove.com/store.html

Our code and circuit are open source. You can obtain the details and the latest information through visiting the following web sites:
http://www.freenove.com/tutorial.html
https://github.com/freenove

If you have any business matters, please feel free to contact us:
sale@freenove.com

Tutorial

This tutorial has two parts.
The part 1 is for Freenove Raspberry Pi starter or case kit. It will introduce how to assemble case and setup raspberry Pi.
The part 2 is for Raspberry Pi projects. If you have electronic components, you can learn our tutorial.

You can download the complete resource below:
http://www.freenove.com
https://github.com/freenove

Customer service

We value the user experience very much. And we are fully responsible for any concerns.

support@freenove.com

If you have any difficulties or feedbacks about our tutorial and kit, please feel free to contact us.

support@freenove.com

We usually reply to you within 24 hours on working day.
And we offer free and quick ethical support. If you have any questions, welcome to contact us.

Part 1 Assemble Case and Set Up Raspberry Pi

This part is for **Freenove Starter Kit for Raspberry Pi** and **Freenove Case Kit for Raspberry Pi**. It will introduce how to assemble the case and setup your Raspberry Pi to get it work.

You can get the products via below:

http://www.freenove.com/store.html

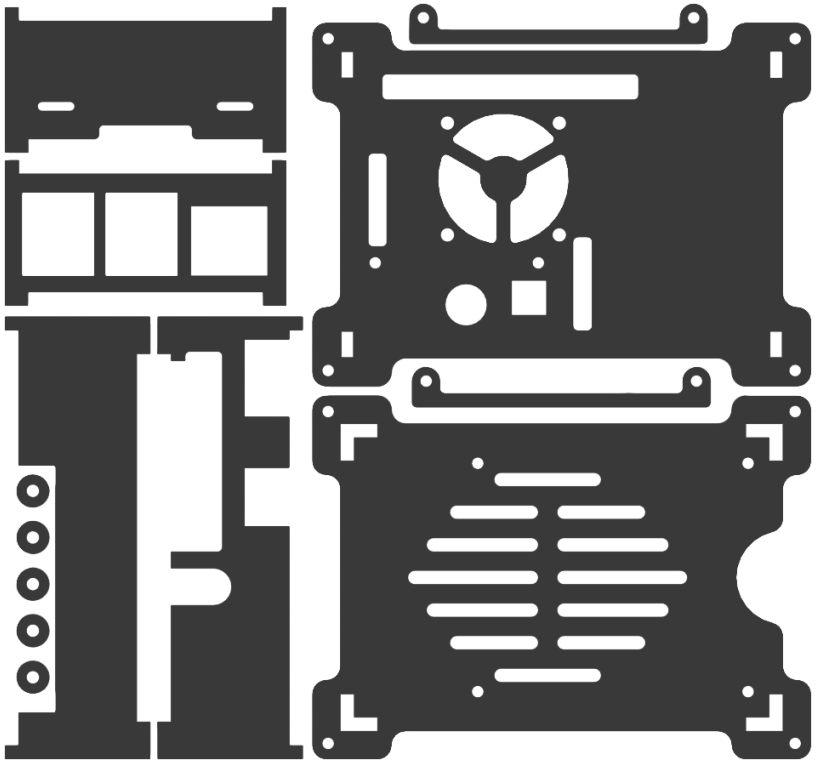
If you have any concerns, please feel free to contact us:

support@freenove.com

List

Acrylic and electronic parts

Acrylic



32 GB Samsung EVO **(not contained in case kit)**

Card Reader **(not contained in case kit)**

5V/3A Power Adapter (**you get only one of below, which is applied in your country**) **(not contained in case kit)**

MINI-HDMI Cable **(not contained in case kit)** 	USB-Type-C cable 10cm 
Power Fan board 	USB-Type-C cable 100cm **(not contained in case kit V1, contained in V2)** 
Fan & Screws and nuts for Fan 	Heat sink 

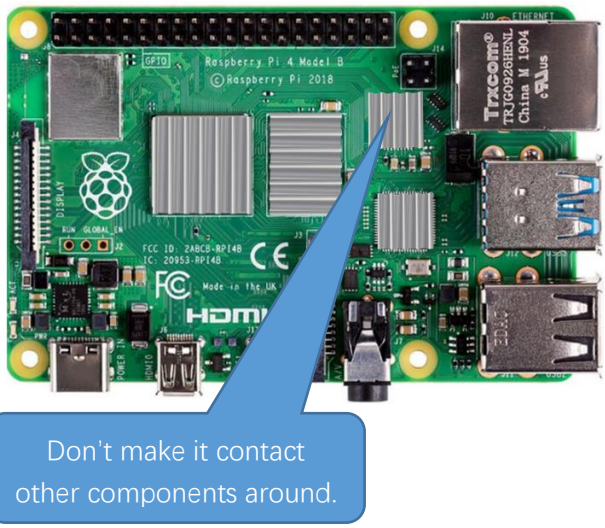
Mechanical Parts

M2.5*6 Screw  x14 Freenove	M2.5*10 Screw  x5 Freenove	M2.5 Nut  x5 Freenove
M2.5*25 Copper Standoff  x5 Freenove	M2.5*8 Copper Standoff  x3 Freenove	Screwdriver 

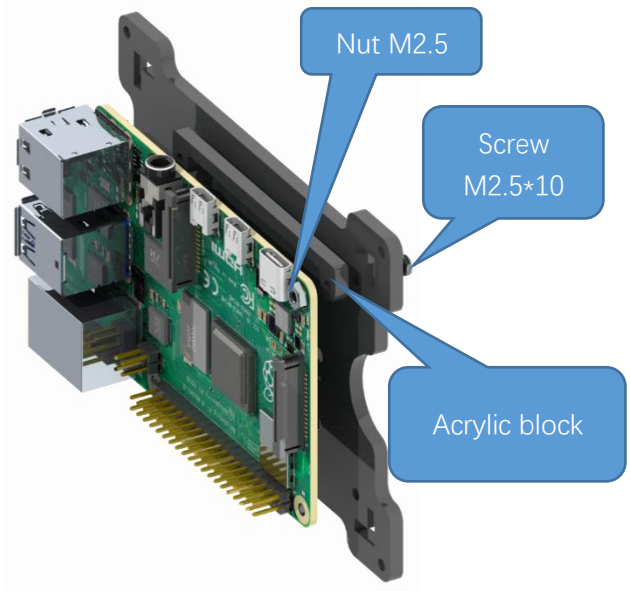
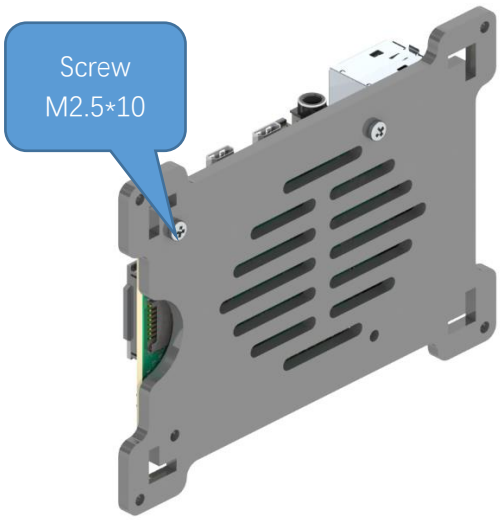
Assemble Case

Note this case supports Raspberry Pi 4 B. It can be used for 3B and 3B + when without the side acryic board for power port.

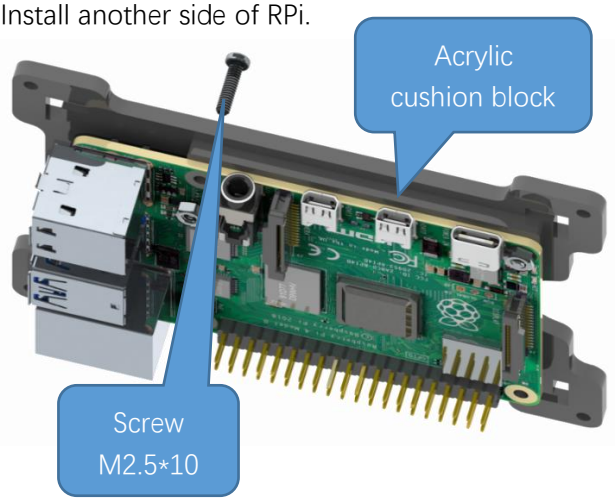
Step 1, paste heat sinks

Before	After
	
Tear the cover of sink and paste on Raspberry Pi 	The two small heat sinks are not necessary.

Step 2, assemble the case

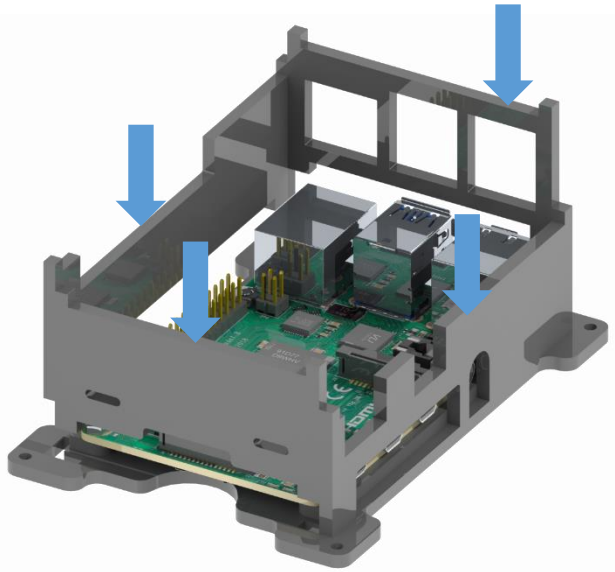
Install one side of RPi.	Bottom view.
	
These parts are spare: 	

Install another side of RPi.

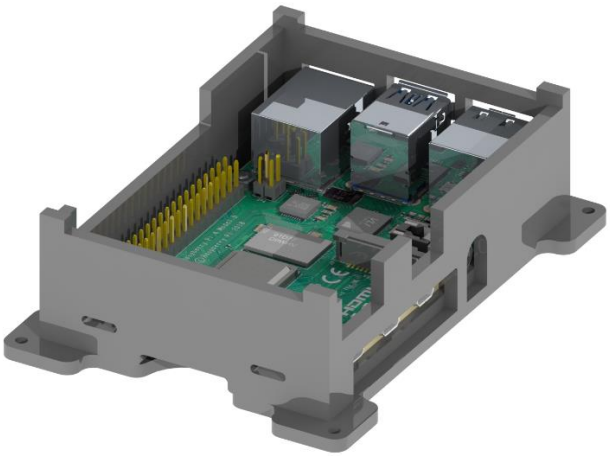


Raspberry pi is install to bottom board successfully.

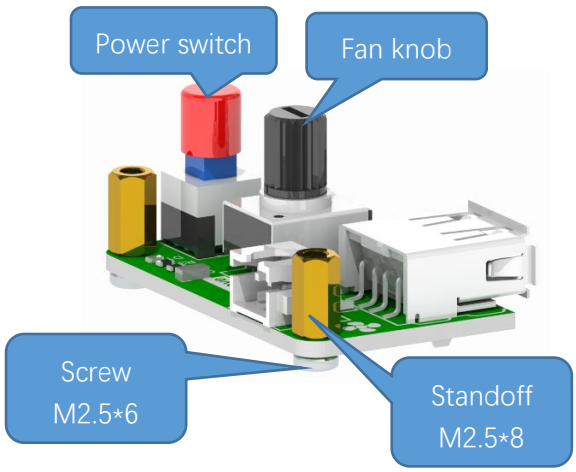
Install sideboard.



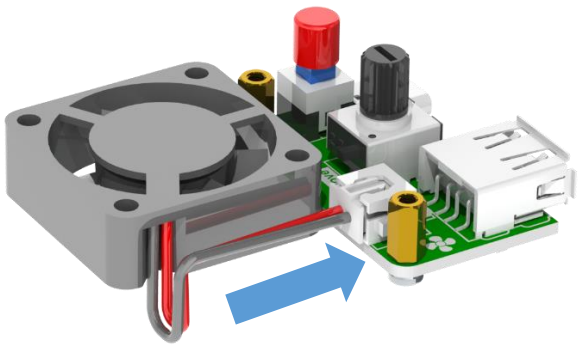
After installation.



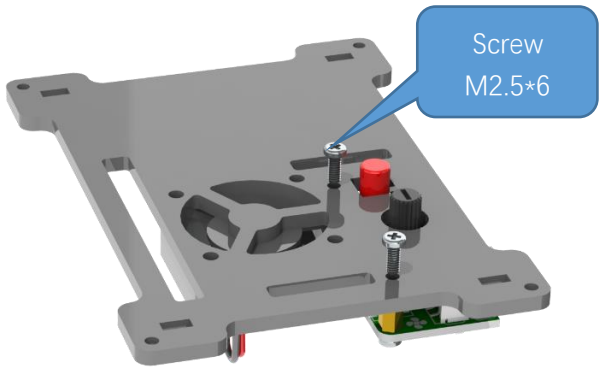
Install standoff to power board.



Connect fan cable to power board.

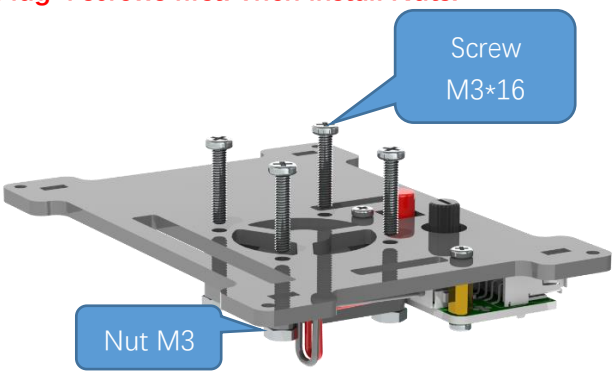


Install power board to top acrylic board.

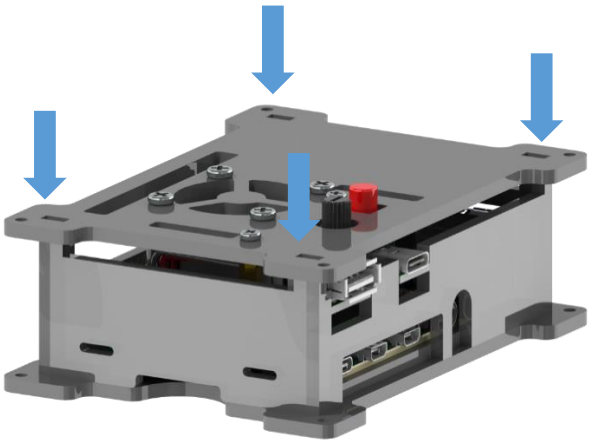


Install fan to top Acrylic board.

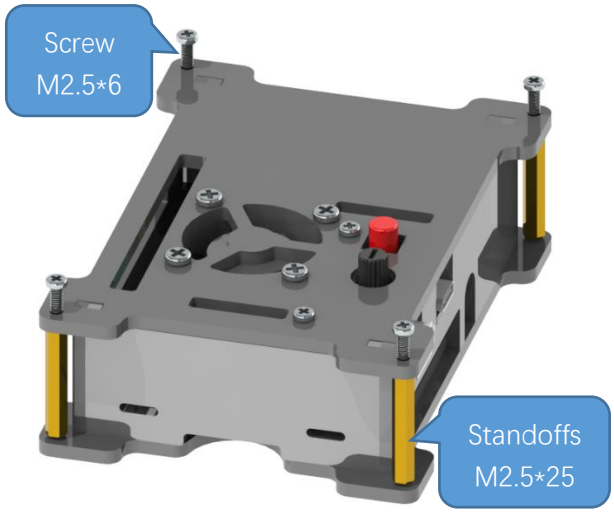
Plug 4 screws first. Then install Nuts.



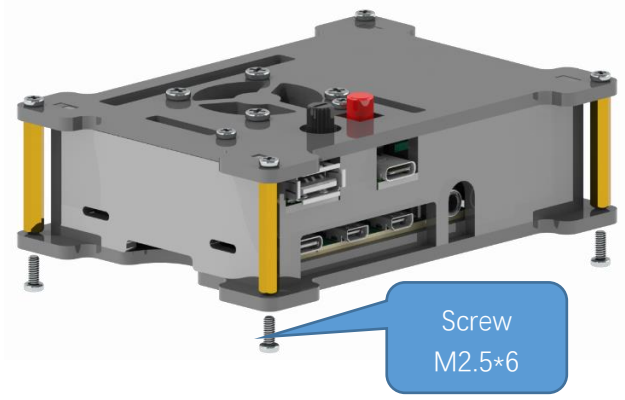
Install top acrylic borad.



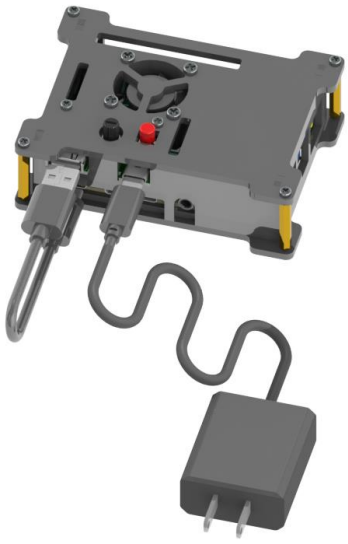
Install standoff to top acrylic board.



Install standoff to bottom acrylic board.



Next, you need to connect cables as below.



If you have any concerns (like something broken), please contact us:

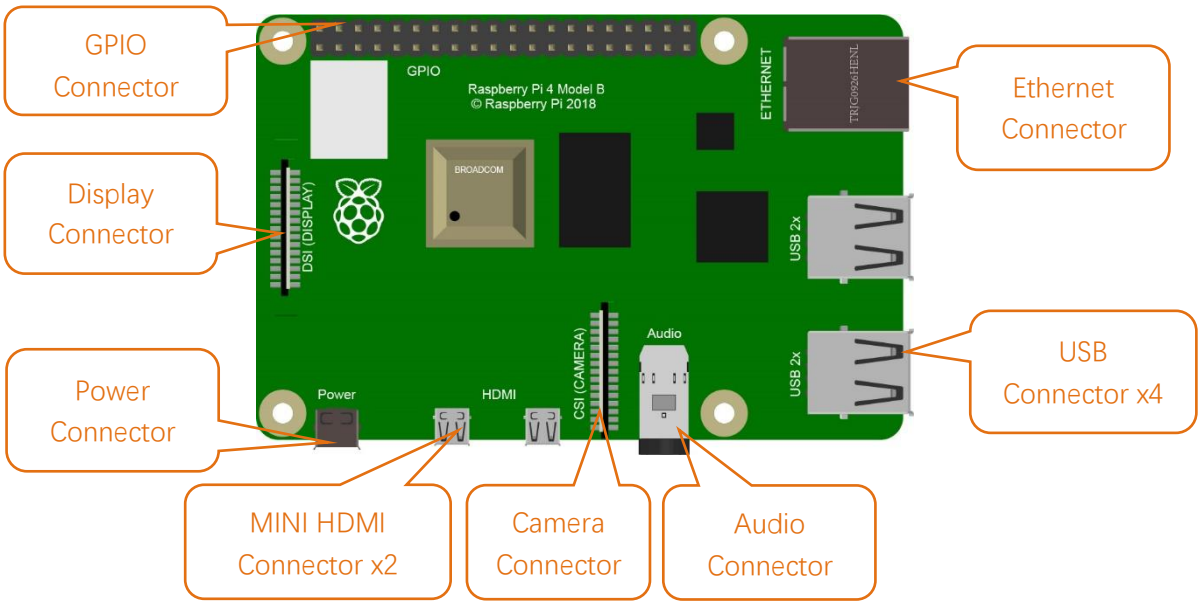
support@freenove.com

Raspberry Pi

Raspberry Pi (RPi, RPI, RasPi, these words will be used later), is a micro-computer with size of a card. When you learn it. You can take it as a computer. It is widely used in desktop workstation, media center, smart home, robots, and even the servers, etc.

So far, Raspberry Pi has developed to the fourth generation. Changes in versions are accompanied by increase and upgrades in hardware. A type and B type, the first generation of products, have been stopped due to various reasons. Other versions are popular and active and the most important is that they are consistent in the order and number of pins, which makes the compatibility of peripheral devices greatly enhanced between different versions.

Hardware interface diagram of RPi 4B is shown below:

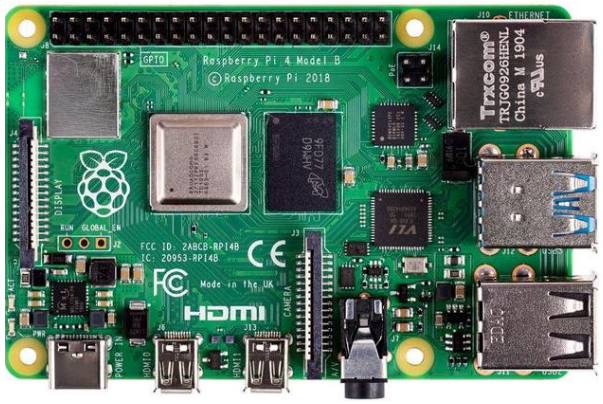
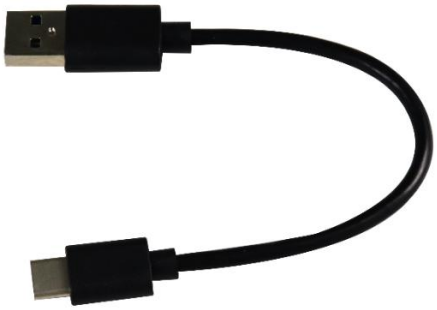
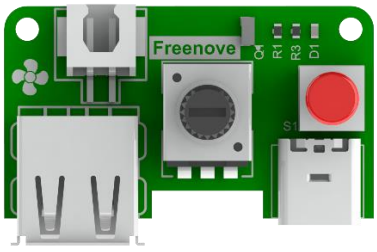


Install the System

Firstly, install a system for your RPi.

Component List

Necessary Components for this case

Any Raspberry Pi 	5V/3A Power Adapter. **Choose the one applied in your country.** 
USB-Type-C 10cm 	Micro SD Card(TF Card), Card Reader 
USB-Type-C cable 100cm  **If you get Freenove USB-Type-C cable 100cm, you need use it. Or something may be wrong.**	Power Fan board 

Power requirement of different versions of Raspberry Pi is shown in following table:

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	400mA
Raspberry Pi 3 Model A+	2.5A	Limited by PSU, board, and connector ratings only.	350mA
Raspberry Pi 3 Model B+	2.5A	1.2A	500mA
Raspberry Pi 4 Model B	3.0A	1.2A	600mA
Raspberry Pi Zero W	1.2A	Limited by PSU, board, and connector ratings only.	150mA
Raspberry Pi Zero	1.2A	Limited by PSU, board, and connector ratings only	100mA

For more details, please refer to https://www.raspberrypi.org/help/faqs/#powerReqs

In addition, RPi also needs a network cable used to connect it to wide area network.

All of these components are necessary. Among them, the power supply is required at least 5V/2.5A, because lack of power supply will lead to many abnormal problems, even damage to your RPi. So power supply with 5V/2.5A is highly recommend. SD Card Micro (recommended capacity 16GB or more) is a hard drive for RPi, which is used to store the system and personal files. In later projects, the components list with a RPi will contains these required components, using only RPi as a representative rather than presenting details.

Optional Components

Under normal circumstances, there are two ways to login to Raspberry Pi: using independent monitor, or remote desktop to share a monitor with your PC.

Required Accessories for Monitor

If you want to use independent monitor, mouse and keyboard, you also need the following accessories.

1. Display with HDMI interface
2. Mouse and Keyboard with USB interface

As to Pi Zero and Pi Zero W, you also need the following accessories.

1. HDMI to HDMI converter wire.
2. Micro-USB to USB-A Receptacles converter wire (Micro USB OTG wire).
3. USB HUB.
4. USB transferring to Ethernet interface or USB Wi-Fi receiver.

For different Raspberry Pi, the optional items are slightly different. But all of their aims are to convert the special interface to standard interface of standard Raspberry Pi.

<table>
<tr><th></th><th>Pi Zero</th><th>Pi Zero W</th><th>Pi A+</th><th>Pi 3A+</th><th>Pi B+/2B</th><th>Pi 3B/3B+/4B</th></tr>
<tr><td>Monitor</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Mouse</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Keyboard</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Mini-HDMI to HDMI converter wire</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td>Micro-USB to USB-A Receptacles converter wire (Micro USB OTG wire)</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td>USB HUB</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td></tr>
<tr><td>USB transferring to Ethernet interface</td><td rowspan="2">select one from two or select two from two</td><td>optional</td><td rowspan="2">select one from two or select two from two</td><td>optional</td><td>Internal Integration</td><td rowspan="2">Internal Integration</td></tr>
<tr><td>USB Wi-Fi receiver</td><td>Internal Integration</td><td>Internal Integration</td><td>optional</td></tr>
</table>

Required Accessories for Remote Desktop

If you do not have an independent monitor, or you want to use a remote desktop, first you need to login to Raspberry Pi through SSH, and then open the VNC or RDP service. So you need the following accessories.

	Pi Zero	Pi Zero W	Pi A+	Pi 3A+	Pi B+/2B	Pi 3B/3B+/4B
Micro-USB to USB-A Receptacles converter wire (Micro USB OTG wire)	Yes	Yes	No	NO		
USB transferring to Ethernet interface	Yes	Yes	Yes			

Raspbian System

If you are using MacBook, please refer to following page. Then jumper to next Remote desktop & VNC.
https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up

If you are using Windows system, please follow content below:

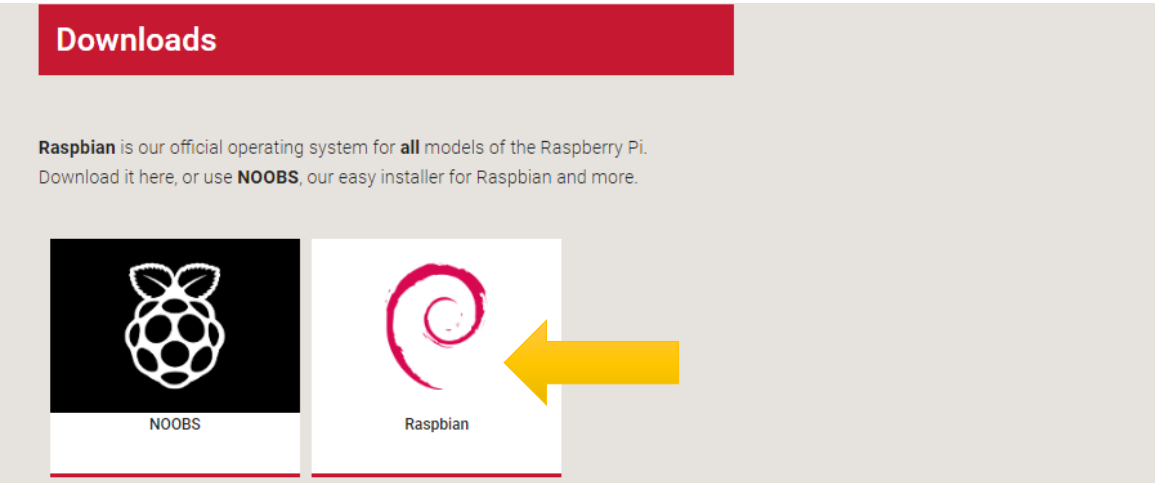
Tool and System image

Software Tool

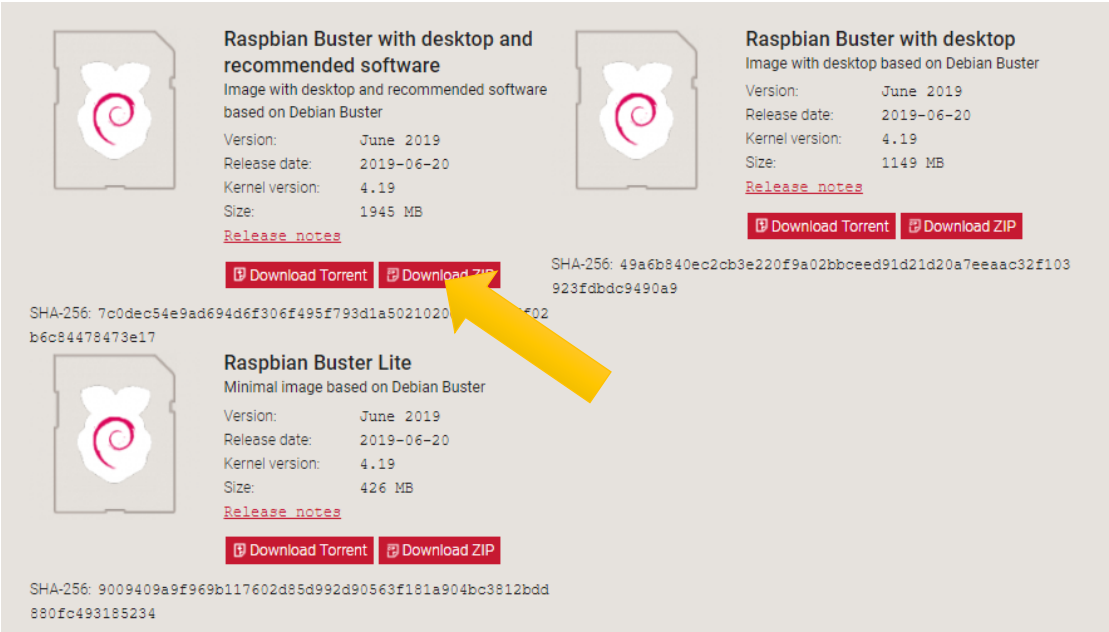
A tool Disk Imager Win32 is required to write system. You can download and install it through visiting the web site: https://sourceforge.net/projects/win32diskimager/

Selecting System

Visit RPi official website (https://www.RaspberryPi.org/), click "Downloads" and choose to download "RASPBIAN". RASPBIAN supported by RPI is an operating system based on Linux, which contains a number of contents required for RPi. We recommended RASPBIAN system to beginners. All projects in this tutorial are operated under the RASPBIAN system.



https://www.raspberrypi.org/downloads/raspbian/

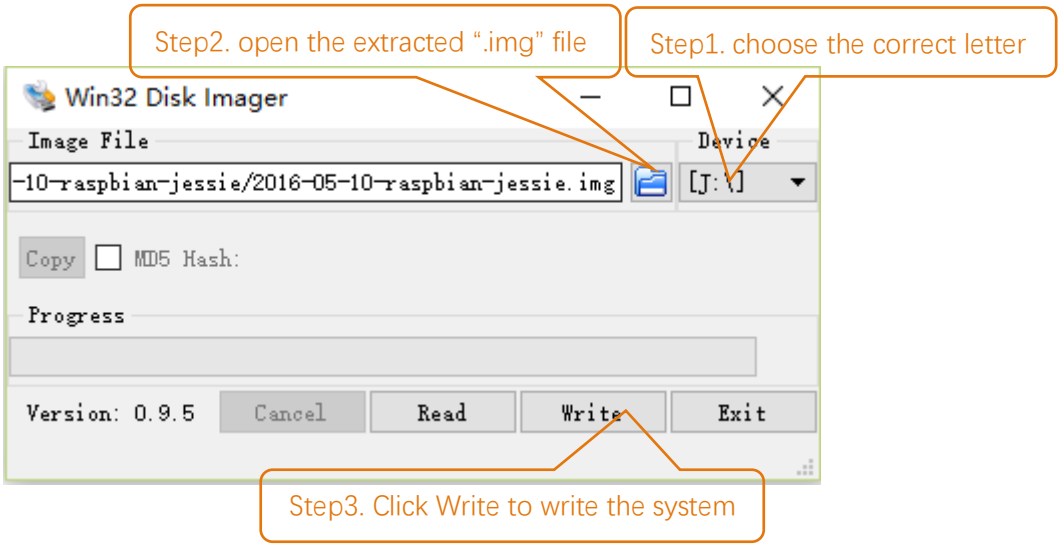


After download, extract file with suffix (.img). Preparation is ready to start making the system.

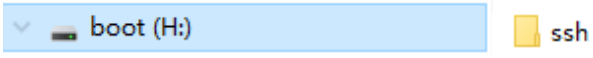
Write System to Micro SD Card

First, put your Micro SD card into card reader and connect card reader to USB port of your computer.

Then open Win32 disk imager, choose the correct letter of your Micro SD Card (here is "J"), open the extracted ".img" file and then click the "Write".



After the system is written. Create a new folder named "ssh" under generated boot disk.



If you have any concerns, please feel free to contact us: **support@freenove.com**

Start Raspberry Pi

If you have any concerns, please feel free to contact us: **support@freenove.com**

If you don't have a spare display, you can jumper to next section "Remote desktop & VNC".

If you have a spare monitor, please follow steps below.
After the system is written successfully, take out Micro SD Card and put it into the card slot of RPi. Then connect RPi to screen through the HDMI-micro HDMI, to mouse and keyboard through the USB port, to network cable through the network card interface and to the power supply. Then your RPi starts initially. Later, you need to enter the user name and password to login. The default user name: pi; password: raspberry. Enter and login. After login, you can enter the following interface.

Now, you have successfully installed the RASPBIAN operating system for your RPi.

Remote desktop & VNC

If you don't have a spare display, mouse and keyboard for your RPi, you can use a remote desktop to share a display, keyboard, and mouse with your PC. Below is how to use remote desktop to control RPi under the Windows operating system.

Under windows, Raspberry Pi can be generally accessed remotely through two applications. The first one is the windows built-in application remote desktop, which corresponds to the Raspberry Pi xrdp service. The second one is the free application VNC Viewer, which corresponds to the VNC interface of Raspberry Pi. Each way has its own advantages. You can choose either one or two.

Windows	Raspberry Pi
Remote Desktop Connection	Xrdp
VNC Viewer	VNC

VNC Viewer can not only run under Windows, but also under system MAC, Linux, IOS, Android and so on.

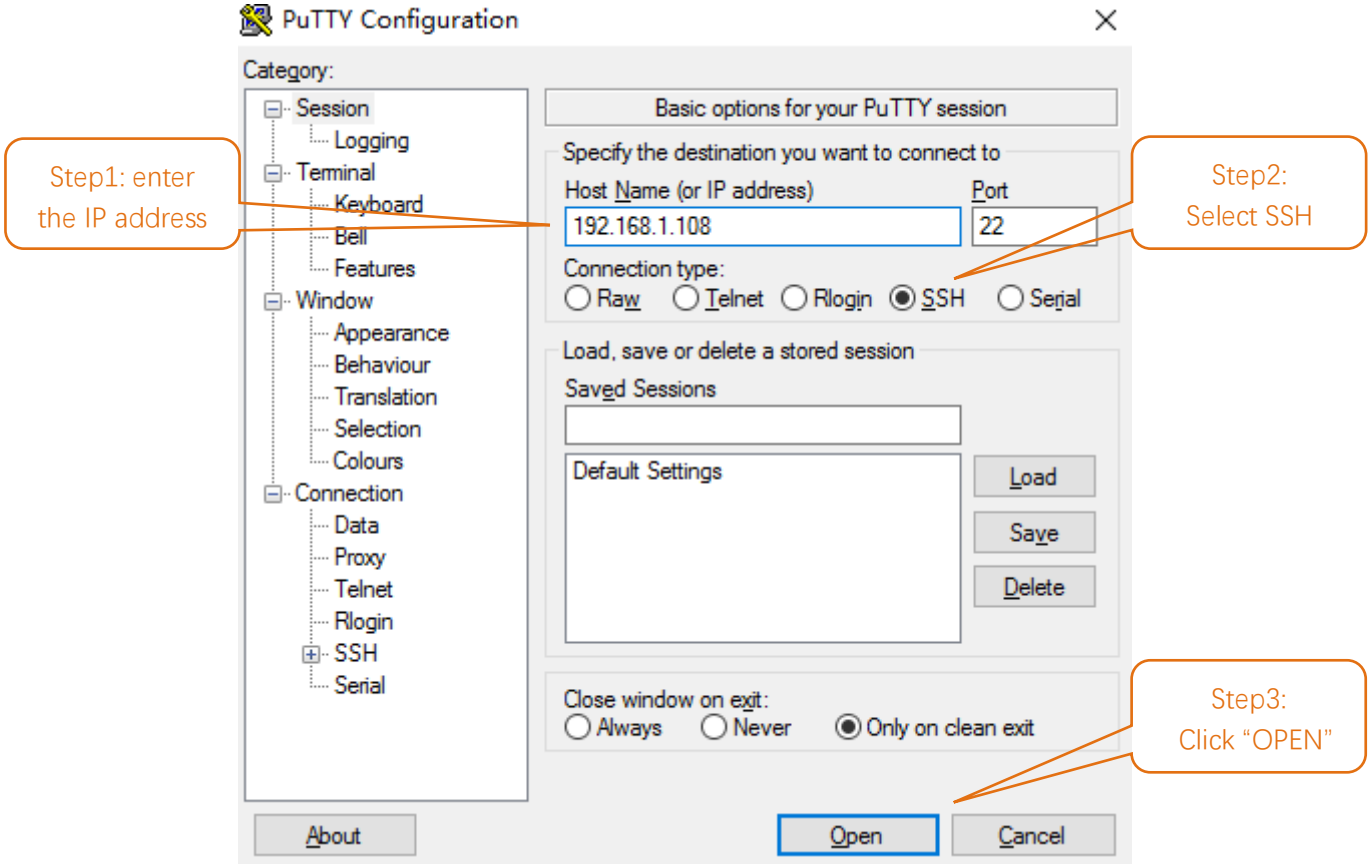
SSH

Under previous Raspbian system, SSH is opened by default. Under the latest version of Raspbian system, it is closed by default. So you need to open it first.

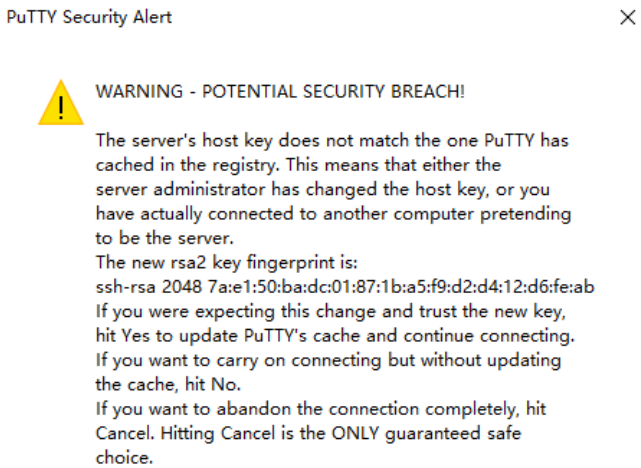
Method: after the system is written. Create a folder named “ssh” under generated boot disk, then the SSH connection will be opened.

Then, download the tool software Putty. Its official address: http://www.putty.org/
Or download it here: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

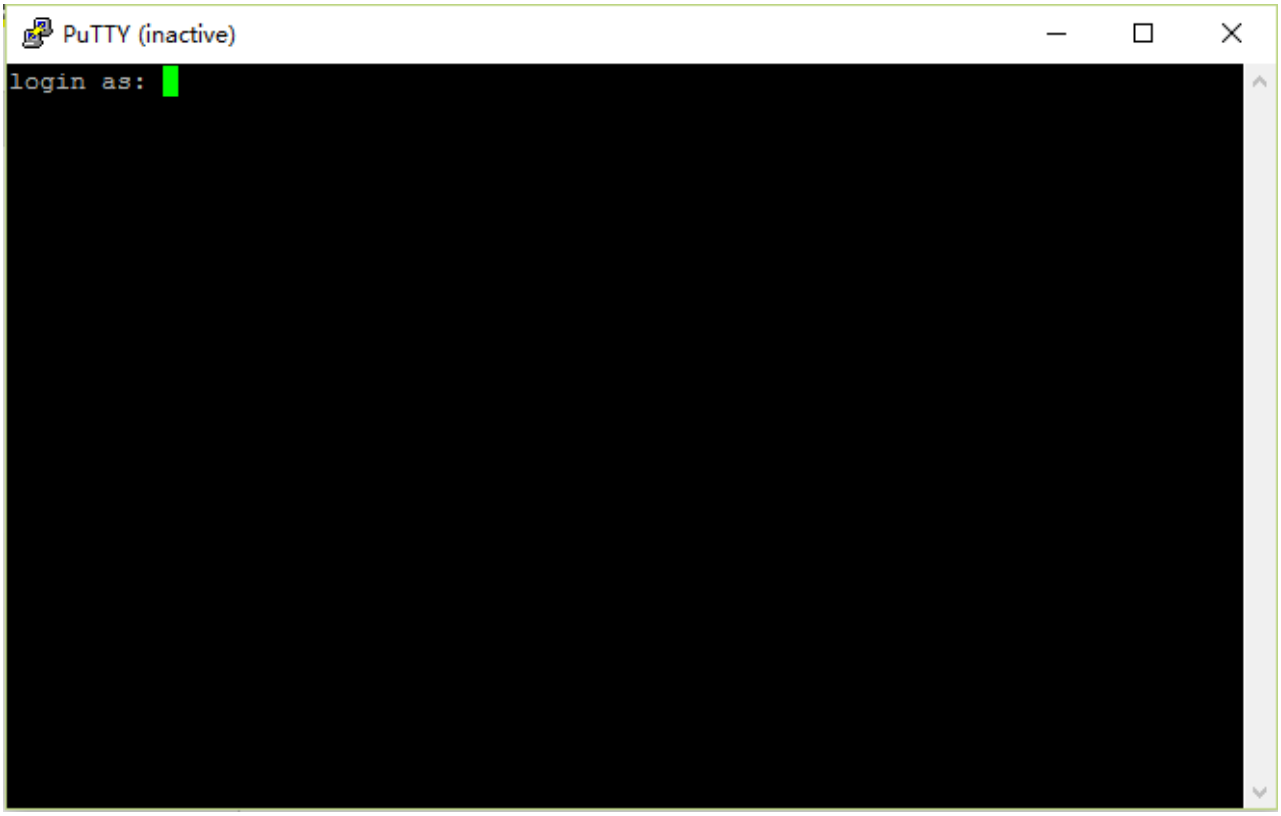
Then use cable to connect your RPi to the routers of your PC LAN, to ensure your PC and your RPi in the same LAN. Then put the system Micro SD Card prepared before into the slot of the RPi and turn on the power supply waiting for starting RPi. Later, enter control terminal of the router to inquiry IP address named “raspberry pi”. For example, I have inquired to my RPi IP address, and it is “192.168.1.108". Then open Putty, enter the address, select SSH, and then click "OPEN", as shown below:



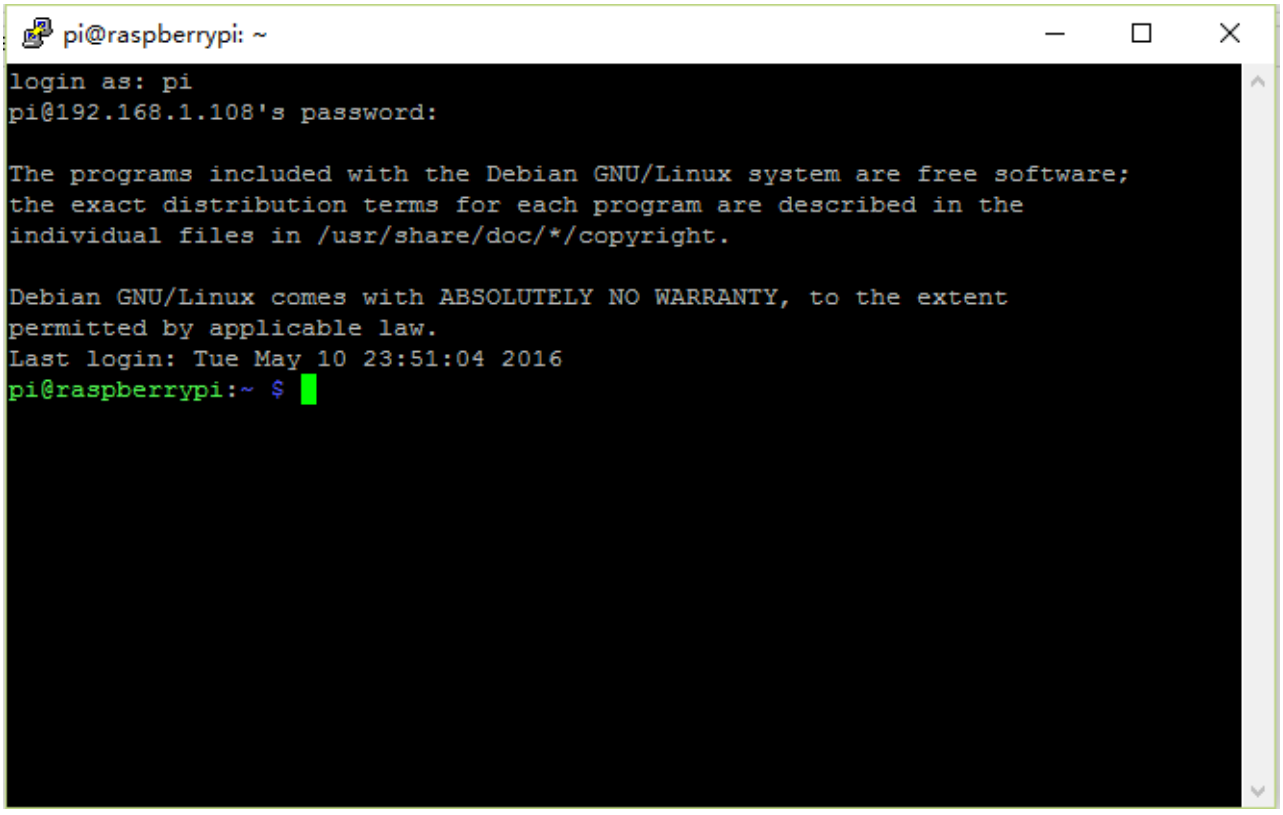
There will appear a security warning at first login. Just click “YES”.



Then there will be a login interface (RPi default user name: pi; the password: raspberry). When you enter the password, there will be no display on the screen. This is normal. After the correct output, press “Enter” to confirm.



Then enter the command line of RPi, which means that you have successfully login to RPi command line mode.



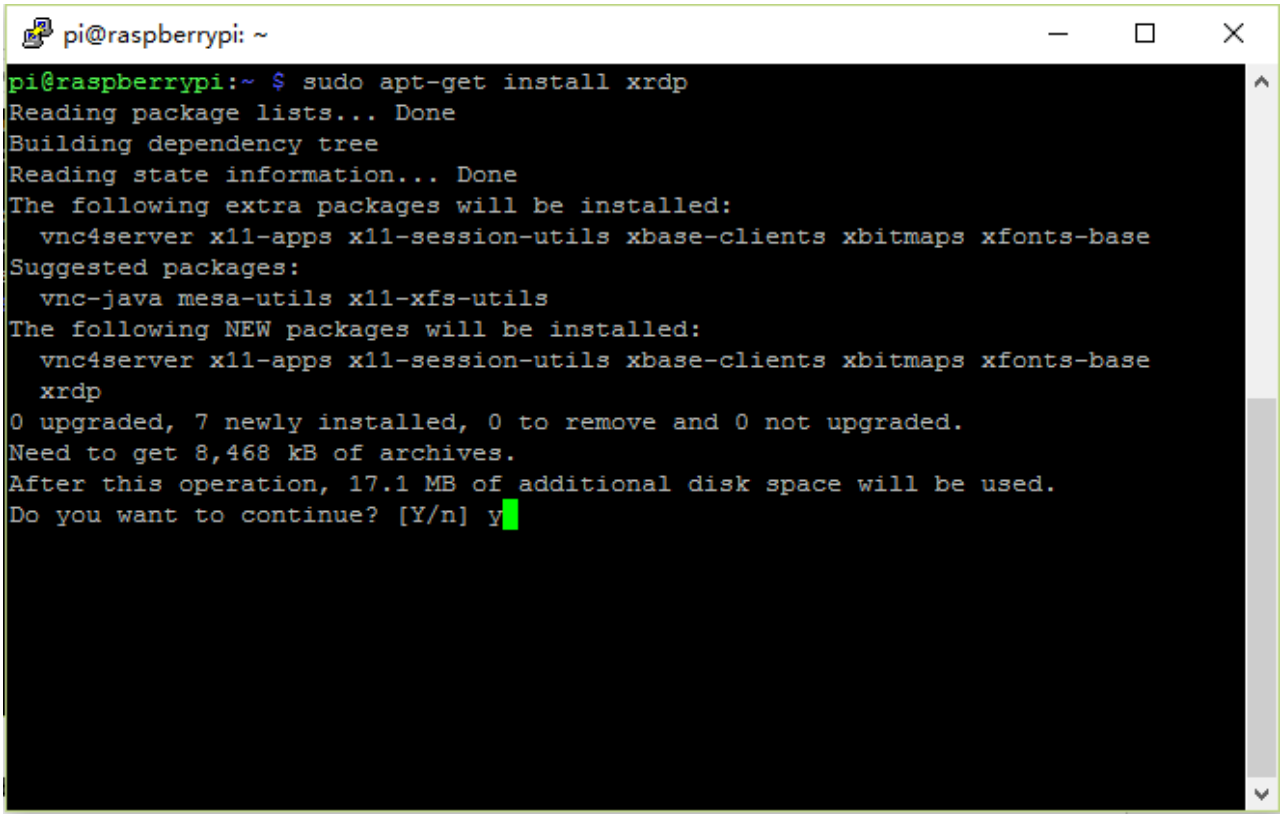
Remote Desktop Connection & xrdp

If you want to use built-in Remote Desktop Connection under Windows, you need install xrdp service on Raspberry Pi.

Next, install a xrdp service, an open source remote desktop protocol(rdp) server, for RPi. Type the following command, then press enter to confirm:

```
sudo apt-get install xrdp
```

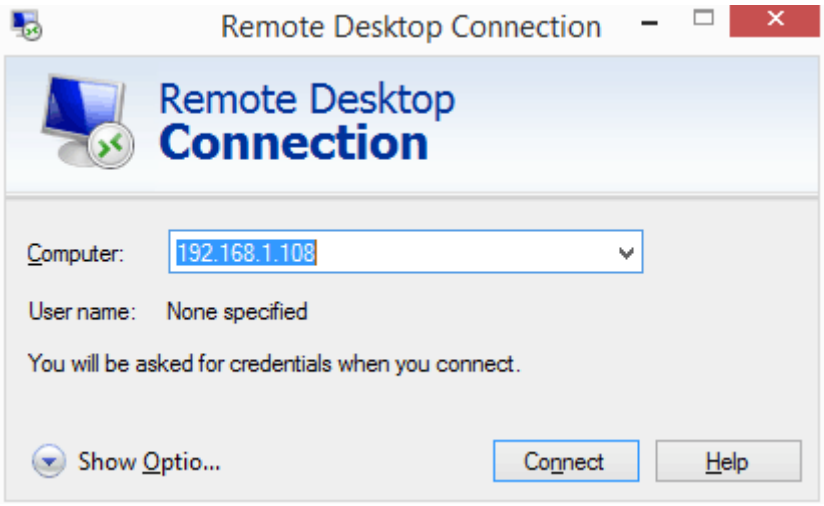
Later, the installation starts.



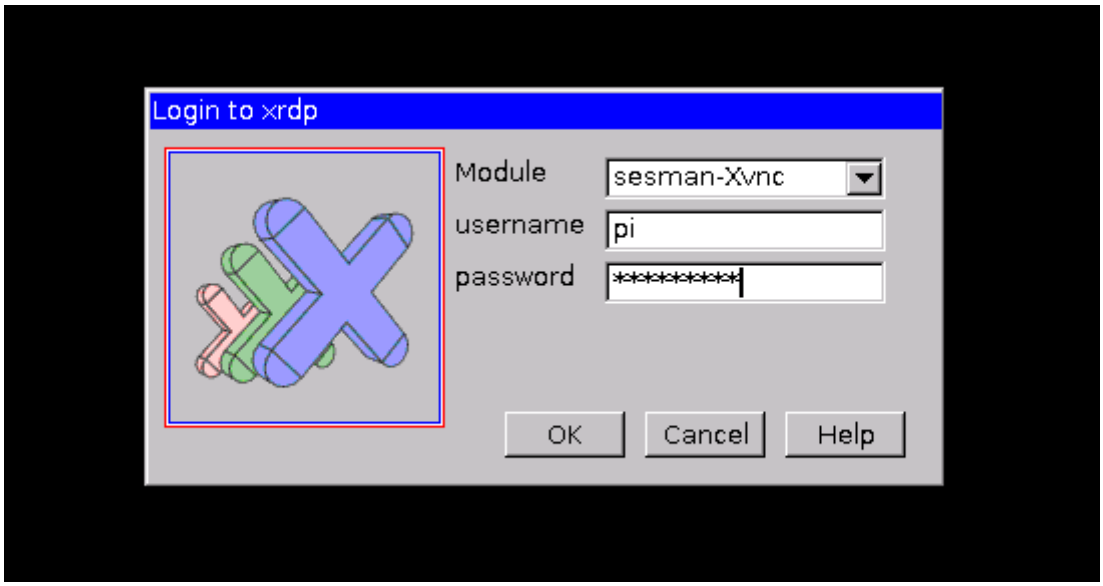
Enter "Y", press key “Enter” to confirm.
After the installation is completed, you can use Windows remote desktop applications to login to your RPi.

Login to Windows remote desktop

Use "WIN+R" or search function, open the remote desktop application "mstsc.exe" under Windows, enter the IP address of RPi and then click “Connect”.



Later, there will be xrdp login screen. Enter the user name and password of RPi (RPi default user name: **pi**; password: **raspberry**) and click “OK”.



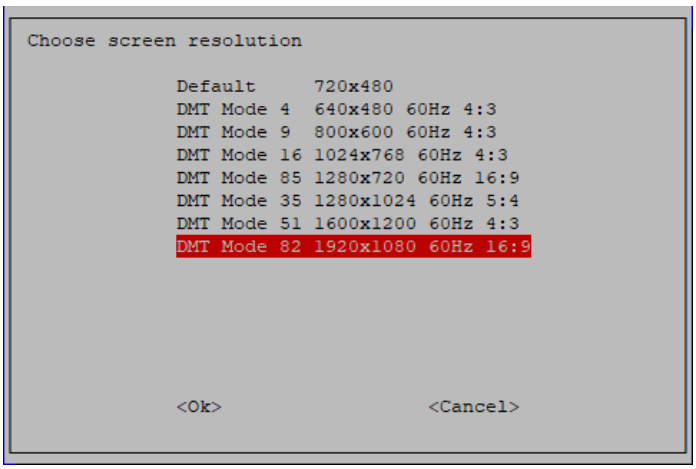
Later, you can enter the RPi desktop system.

Here, you have successfully used the remote desktop login to RPi.
If you have any concerns, please feel free to contact us: **support@freenove.com**

You can set a proper resolution ratio according to your monitor.
And if VNC doesn't work normal, you need adjust resolution to try.

```
sudo raspi-config
```

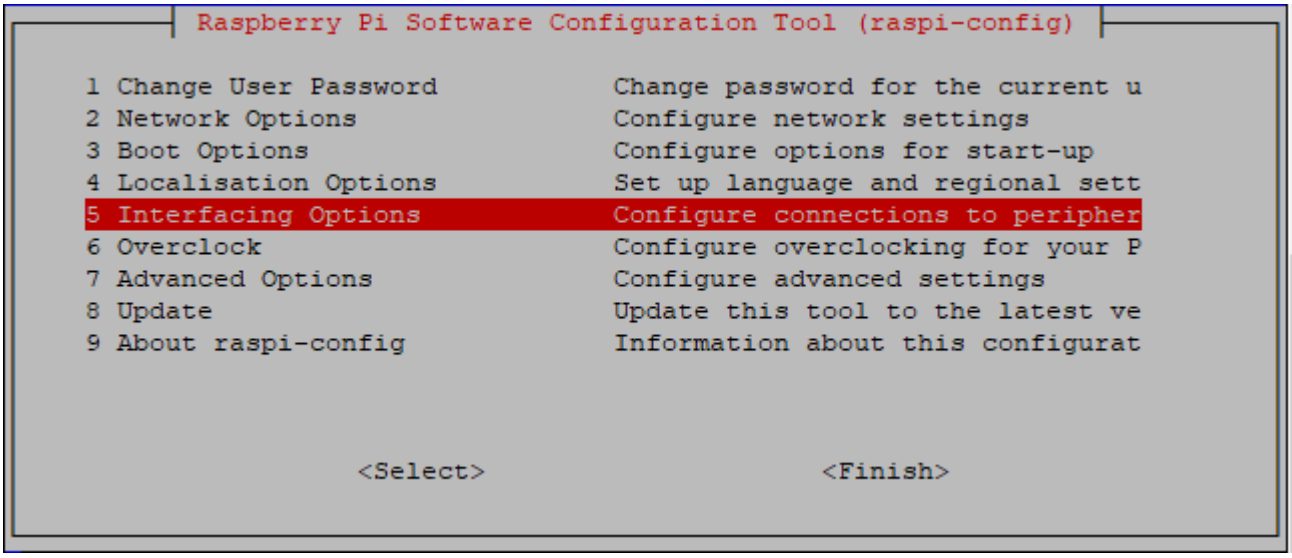
Select 7 Advanced Options→A5 Resolution→proper resolution ratio (set by yourself)→OK. If it needs restart, just restart.

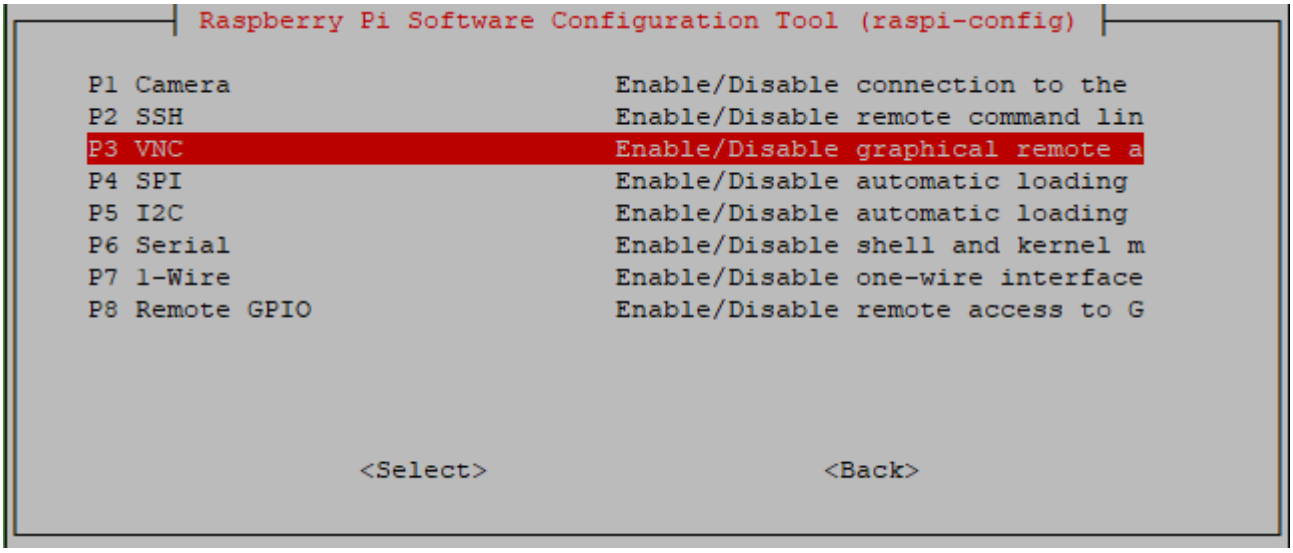


VNC Viewer & VNC

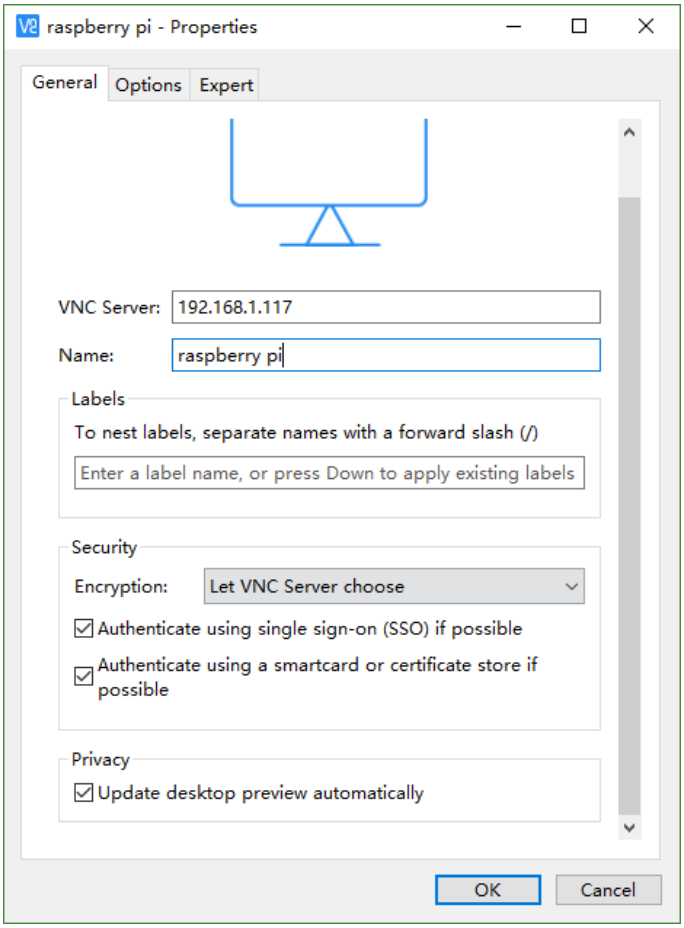
Type the following command. And select 5 Interfacing Options→P3 VNC →Yes→OK→Finish. Here Raspberry Pi may need be restarted, and choose ok. Then open VNC interface.

```
sudo raspi-config
```

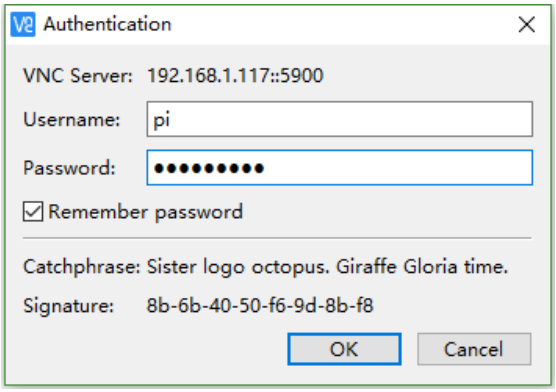




Then download and install VNC Viewer by click following link:
https://www.realvnc.com/en/connect/download/viewer/windows/
After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.

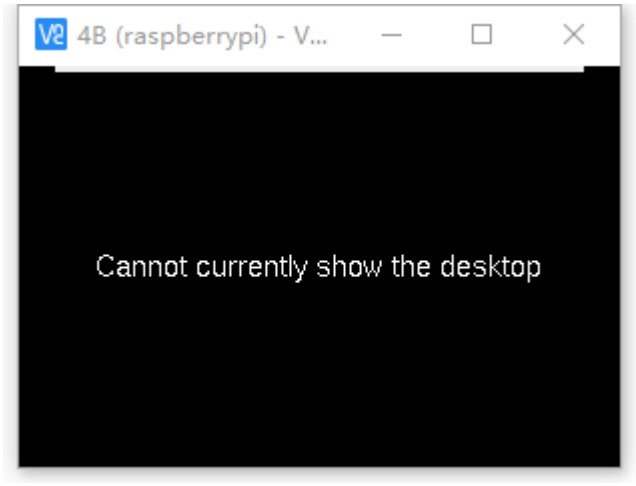


Enter IP address of your Raspberry Pi and fill in a Name. And click OK.
Then on the VNC Viewer panel, double-click new connection you just created, and the following dialog box pops up.



Enter username: pi and Password: raspberry. And click OK.

Here, you have logged in to Raspberry Pi successfully by using VNC Viewer
If the window does not work normal, as below.

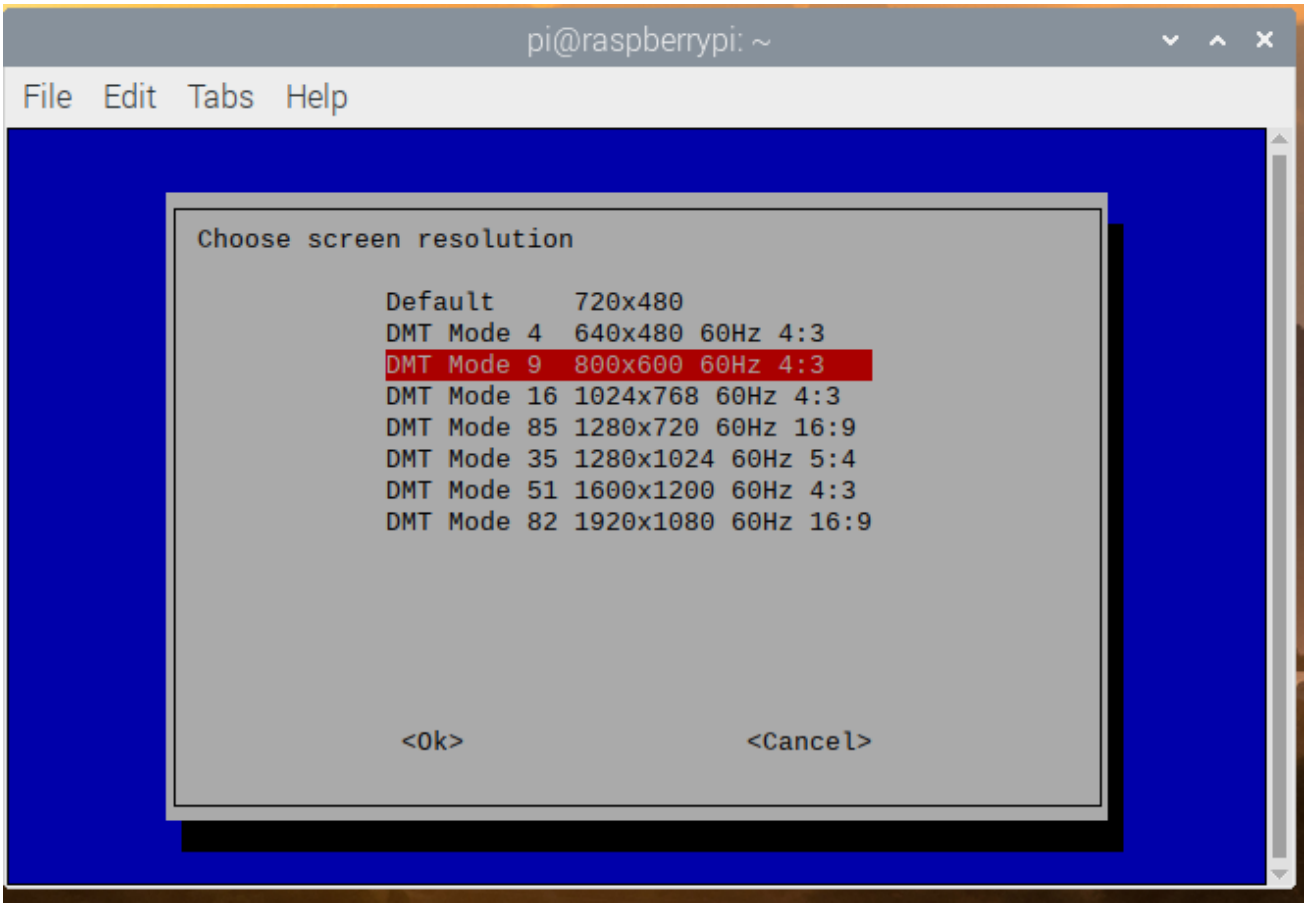


You need login to windows remote desktop and change resolution ratio.

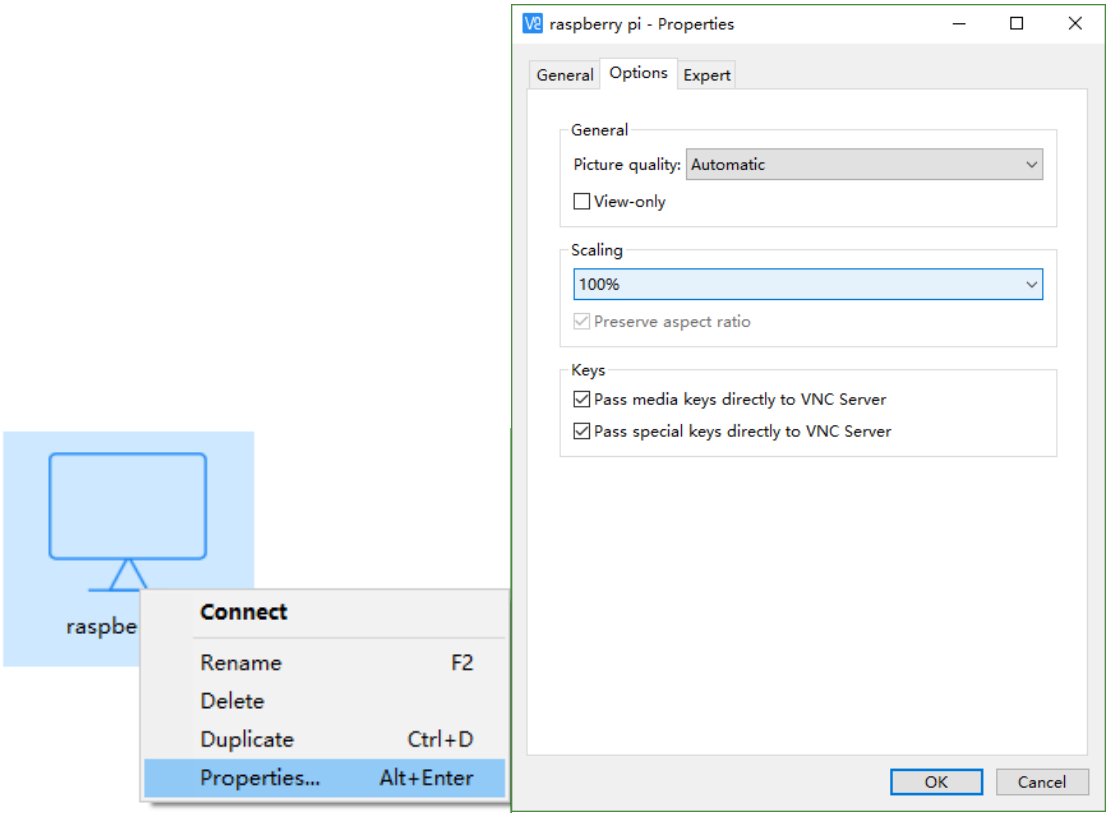
If you think resolution ratio is not OK, you can set a proper resolution ratio on set interface of Raspberry Pi.

```
sudo raspi-config
```

Select 7 Advanced Options→A5 Resolution→proper resolution ratio (set by yourself)→OK. If it needs restart, just restart.



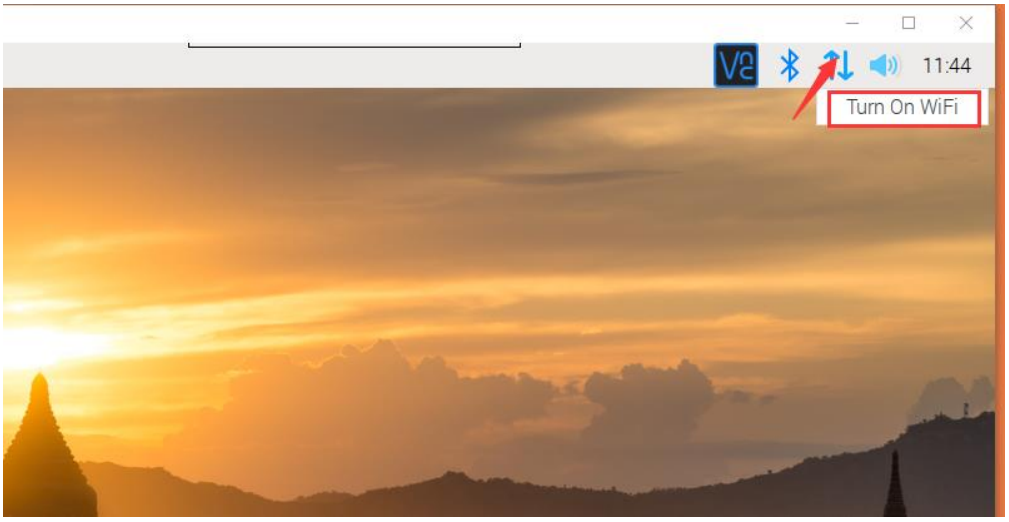
In addition, your VNC Viewer window may zoom your Raspberry Pi desktop. You can change it. On your VNC View control panel, click right key. And select Properties->Options label->Scaling. Then set proper scaling.



Here, you have logged in to Raspberry Pi successfully by using VNC Viewer and operated proper setting. Then continue to do some preparation work: install a GPIO library wiringPi for your RPi.

Wi-Fi

Raspberry Pi 4B/ 3B+/3B integrates a Wi-Fi adaptor. You can use it to connect to your Wi-Fi. Then you can use the wireless remote desktop to control your RPi. This will be helpful for the following work. Raspberry Pi of other models can use wireless remote desktop through accessing an external USB wireless card.



What's Next?

If you do not have Freenove project kit, you can visit official website to learn how to use Raspberry Pi.
https://projects.raspberrypi.org/en/projects/raspberry-pi-using

If you have Freenove Ultimate Starter Kit for Raspberry Pi or some electronic components, you can continue this tutorial to complete dozens of projects.

Part 2 Raspberry Pi Projects

This part is for Raspberry Pi projects.

1. If you already get **Freenove Ultimate Starter Kit for Raspberry Pi** or some electronic components, you can also download and learn the whole resource.

https://github.com/Freenove/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/archive/master.zip

It will teach you how to use raspberry Pi to control electronic components (such as LED, motor, servo, sensors) to create interesting projects. We provide code of both C and Python for each project, and code in Processing.pdf is based on Java.

2. We also have a **Raspberry Pi Car.** You can download the whole resource here.

https://github.com/Freenove/Freenove_Three-wheeled_Smart_Car_Kit_for_Raspberry_Pi/archive/master.zip

You can get Freenove Ultimate Starter Kit for Raspberry Pi, Freenove Three-wheeled Smart Car Kit for Raspberry Pi and other Freenove products here:

http://www.freenove.com/store.html

If you have any concerns, please feel free to contact us.

support@freenove.com

Following are tutorial examples of Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi.

Contents of Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi

You can download the whole resource here:

https://github.com/Freenove/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/archive/master.zip

Chapter 1 LED (Sample)

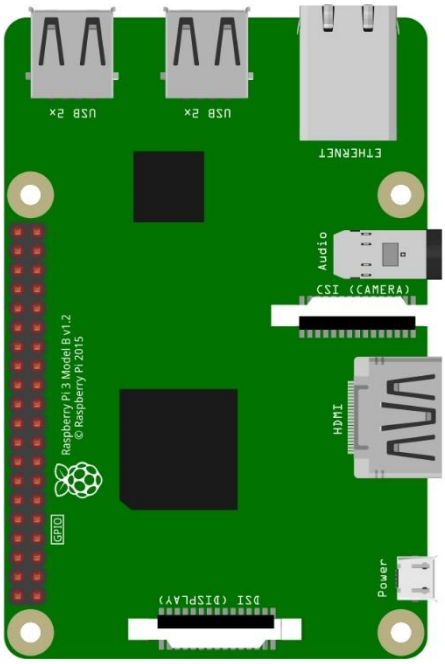
This chapter is the starting point of the journey to explore RPi electronic projects. Let's start with simple "Blink".

Project 1.1 Blink

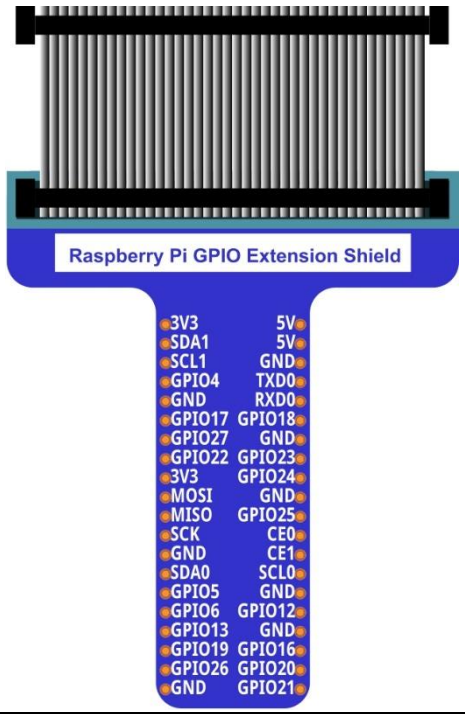
In this project, let's try to use RPi to control LED blinking.

Component List

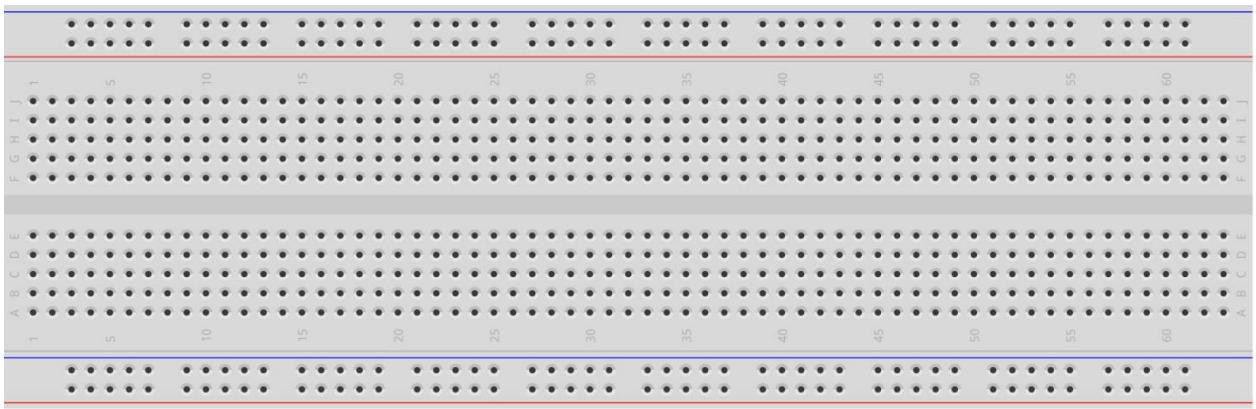
Raspberry Pi 3B x1



GPIO Extension Board & Wire x1



BreadBoard x1



LED x1

Resistor 220Ω x1

Jumper

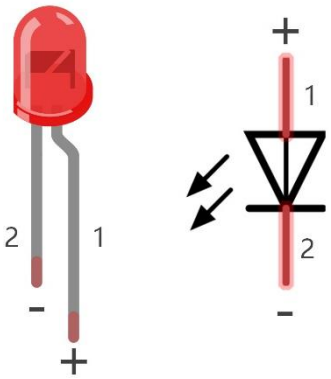
In the components list, 3B GPIO, Extension Shield Raspberry and Breadboard are necessary for each project. They will be listed only in text form later.

Component knowledge

LED

LED is a kind of diode. LED will shine only if the long pin of LED is connected to the positive electrode and the short pin is connected to negative electrode.

This is also the features of the common diode. Diode works only if the voltage of its positive electrode is higher than its negative electrode.



LED	Voltage	Maximum current	Recommended current
Red	1.9-2.2V	20mA	10mA
Green	2.9-3.4V	10mA	5mA
Blue	2.9-3.4V	10mA	5mA
Volt ampere characteristics conform to diode			

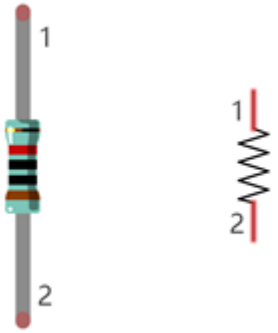
The LED can not be directly connected to power supply, which can damage component. A resistor with certain resistance must be connected in series in the circuit of LED.

Resistor

The unit of resistance(R) is ohm(Ω). 1mΩ=1000kΩ, 1kΩ=1000Ω.

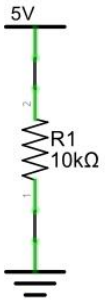
Resistor is an electrical component that limits or regulates the flow of current in an electronic circuit.

The left is the appearance of resistor, and the right is the symbol of resistor represented in circuit.



Color rings attached to the resistor is used to indicate its resistance. For more details of resistor color code, please refer to the appendix of this tutorial.

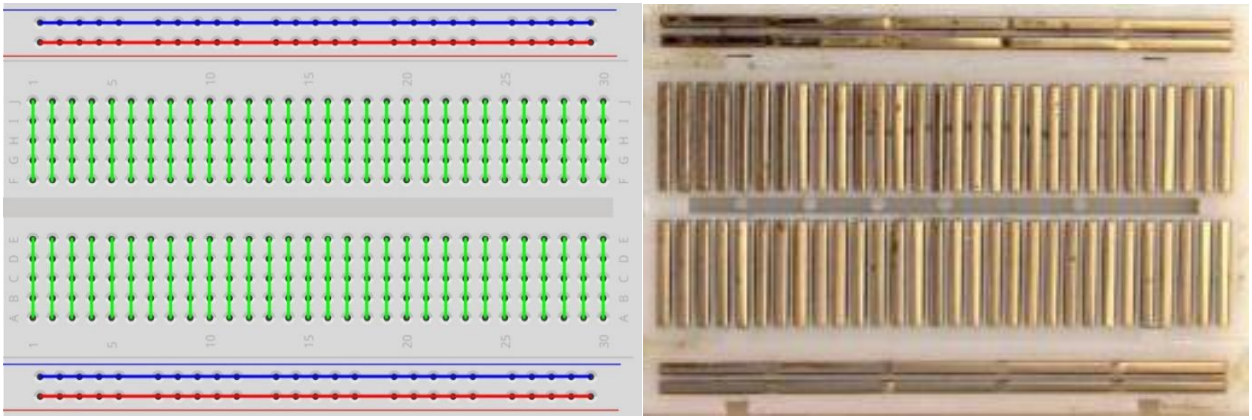
With the same voltage there will be less current with more resistance. And the links among current, voltage and resistance can be expressed by the formula below: I=U/R.
In the following diagram, the current through R1 is: I=U/R=5V/10kΩ=0.0005A=0.5mA.



Do not connect the two poles of power supply with low resistance, which will make the current too high to damage electronic components.
And resistor has no poles.

Breadboard

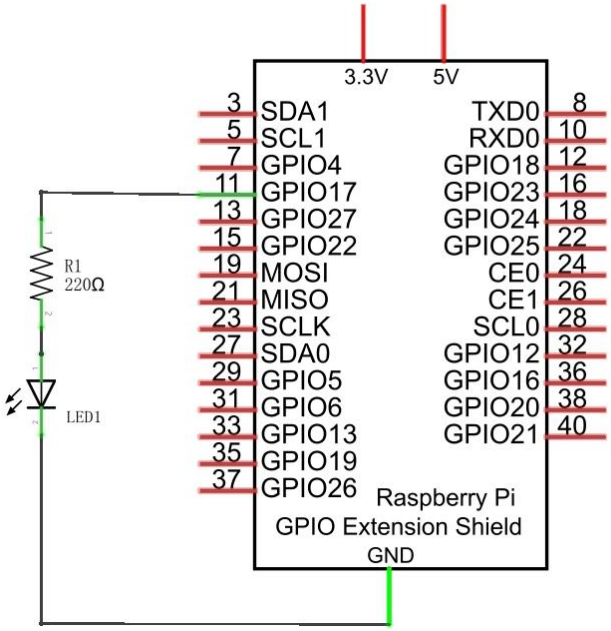
Take a short breadboard as an example to introduce its feature, as below.
The left picture shows the way of connection of pins. The right picture shows the pratical internal structure.



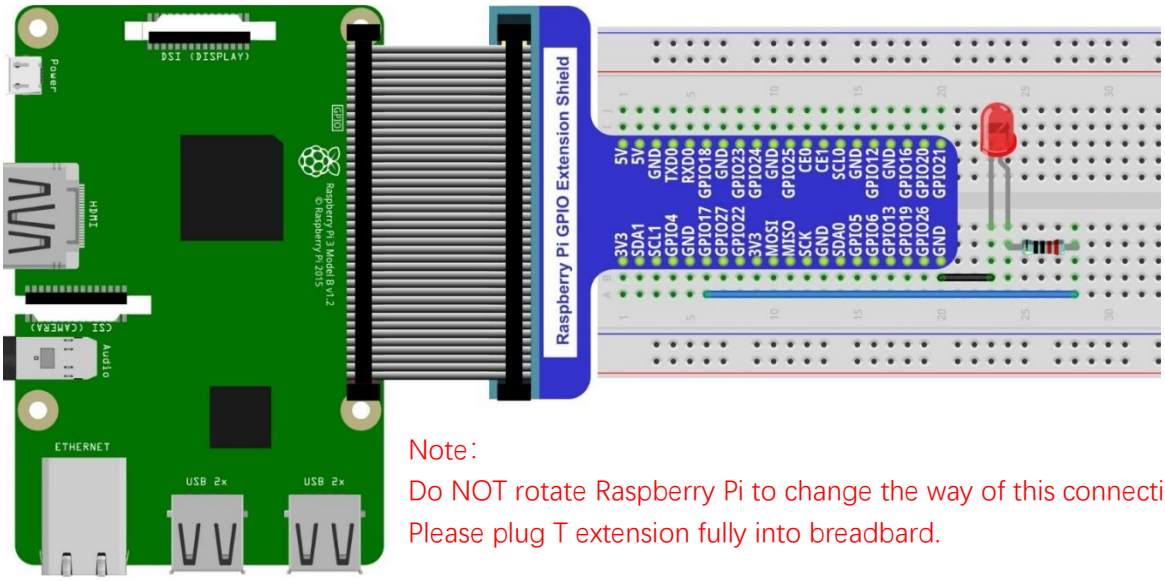
Circuit

Disconnect RPi from GPIO Extension Shield first. Then build the circuit according to the circuit diagram and the hardware connection diagram. After the circuit is built and confirmed, connect RPi to GPIO Extension Shield. In addition, short circuit (especially 5V and GND, 3.3V and GND) should be avoid, because short circuit may cause abnormal circuit work, or even damage to RPi.

Schematic diagram



Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



Note:
Do NOT rotate Raspberry Pi to change the way of this connection.
Please plug T extension fully into breadbard.

Because Numbering of GPIO Extension Shield is the same as RPi GPIO, later Hardware connection diagram will only show the part of breadboard and GPIO Extension Shield.

Code

According to the circuit, when the GPIO17 of RPi output high level, LED is turned on. Conversely, when the GPIO17 RPi output low level, LED is turned off. Therefore, we can let GPIO17 output high and low level in

cycle to make LED blink. We will use both C code and Python code to achieve the target.

C Code 1.1.1 Blink

First, execute command into the terminal one by one. Then observe the project result, and analyze the code.

1. Use cd command to enter 01.1.1_Blink directory of C code.

```
cd ~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/C_Code/01.1.1_Blink
```

2. Use the following command to compile the code "Blink.c" and generate executable file "Blink".

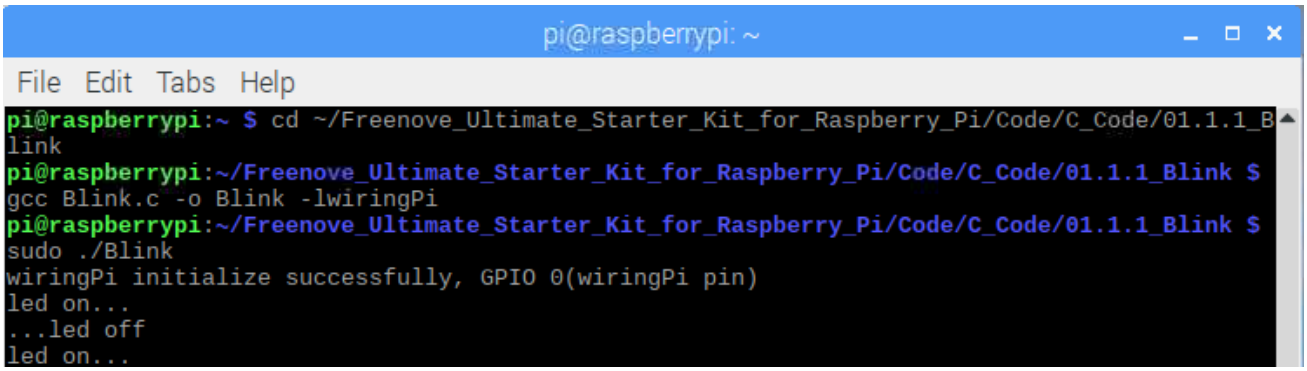
"l" of "lwiringPi" is low case of "L".

```
gcc Blink.c -o Blink -lwiringPi
```

3. Then run the generated file "blink".

```
sudo ./Blink
```

Now, LED start blink.



You can press "Ctrl+C" to end the program. The following is the program code:

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

#define  ledPin    0

int main(void)
{
    if(wiringPiSetup() == -1){ //when initialize wiring failed, print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }
    //when initialize wiring successfully, print message to screen
    printf("wiringPi initialize successfully, GPIO %d(wiringPi pin)\n",ledPin);

    pinMode(ledPin, OUTPUT);

    while(1){
            digitalWrite(ledPin, HIGH);  //led on
            printf("led on...\n");
            delay(1000);
            digitalWrite(ledPin, LOW);  //led off
```

```
22              printf("...led off\n");
23              delay(1000);
24      }
25
26      return 0;
27  }
```

GPIO connected to ledPin in the circuit is GPIO17. And GPIO17 is defined as 0 in the wiringPi numbering. So ledPin should be defined as 0 pin. You can refer to the corresponding table in Chapter 0.

```
#define  ledPin    0
```

In the main function main(), initialize wiringPi first, and then print out the initial results. Once the initialization fails, exit the program.

```
    if(wiringPiSetup() == -1){ //when initialize wiring failed, print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }
    //when initialize wiring successfully, print message to screen
    printf("wiringPi initialize successfully, GPIO %d(wiringPi pin)\n",ledPin);
```

After the wiringPi is initialized successfully, set the ledPin to output mode. And then enter the while cycle, which is an endless loop. That is, the program will always be executed in this cycle, unless it is ended outside. In this cycle, use digitalWrite (ledPin, HIGH) to make ledPin output high level, then LED is turned on. After a period of time delay, use digitalWrite(ledPin, LOW) to make ledPin output low level, then LED is turned off, which is followed by a delay. Repeat the cycle, then LED will start blinking.

```
    pinMode(ledPin, OUTPUT);
    while(1){
            digitalWrite(ledPin, HIGH);  //led is turned on
            printf("led on...\n");
            delay(1000);
            digitalWrite(ledPin, LOW);  //led is turned off
            printf("...led off\n");
            delay(1000);
    }
```

Among them, the configuration function for GPIO is shown below as:

void pinMode(int pin, int mode);

This sets the mode of a pin to either INPUT, OUTPUT, PWM_OUTPUT or GPIO_CLOCK. Note that only wiringPi pin 1 (BCM_GPIO 18) supports PWM output and only wiringPi pin 7 (BCM_GPIO 4) supports CLOCK output modes.

This function has no effect when in Sys mode. If you need to change the pin mode, then you can do it with the gpio program in a script before you start your program

void digitalWrite (int pin, int value);

Writes the value HIGH or LOW (1 or 0) to the given pin which must have been previously set as an output.

For more related functions, please refer to http://wiringpi.com/reference/

Python Code 1.1.1 Blink

Net, we will use Python language to make LED blink.

First, observe the project result, and then analyze the code.

1. Use cd command to enter 01.1.1_Blink directory of Python code.

```
cd ~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_Code/01.1.1_Blink
```

2. Use python command to execute python code blink.py.

```
python Blink.py
```

Now, LED start blinking.

```
pi@raspberrypi:~ $ cd ~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_Code/01.1.1_Blink
pi@raspberrypi:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_Code/01.1.1_Blink $ python Blink.py
```

You can press "Ctrl+C" to end the program. The following is the program code:

```
1  import RPi.GPIO as GPIO
2  import time
3
4  ledPin = 11    # RPI Board pin11
5
6  def setup():
7      GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
8      GPIO.setup(ledPin, GPIO.OUT)   # Set ledPin's mode is output
9      GPIO.output(ledPin, GPIO.LOW) # Set ledPin low to off led
10     print ('using pin%d'%ledPin)
11
12 def loop():
13     while True:
14         GPIO.output(ledPin, GPIO.HIGH)  # led on
15         print ('...led on')
16         time.sleep(1)                   # delay 1 second
17         GPIO.output(ledPin, GPIO.LOW) # led off
18         print ('led off...')
19         time.sleep(1)
20 def destroy():
21     GPIO.output(ledPin, GPIO.LOW)      # led off
22     GPIO.cleanup()                      # Release resource
23
24 if __name__ == '__main__':     # Program start from here
25     setup()
26     try:
27         loop()
28     except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the subprogram destroy() will
29 be  executed.
30         destroy()
31
```

In subfunction setup(), GPIO.setmode (GPIO.BOARD) is used to set the serial number for GPIO based on

physical location of the pin. GPIO17 use pin 11 of the board, so define ledPin as 11 and set ledPin to output mode (output low level).

```
ledPin = 11     # RPi Board pin11
def setup():
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    GPIO.setup(ledPin, GPIO.OUT)    # Set ledPin to output mode
    GPIO.output(ledPin, GPIO.LOW) # Set ledPin to low level to turn off led
    print ('using pin%d'%ledPin)
```

In loop(), there is a while cycle, which is an endless loop. That is, the program will always be executed in this cycle, unless it is ended outside. In this cycle, set ledPin output high level, then LED is turned on. After a period of time delay, set ledPin output low level, then LED is turned off, which is followed by a delay. Repeat the cycle, then LED will start blinking.

```
def loop():
    while True:
        GPIO.output(ledPin, GPIO.HIGH)  # led on
        print ('...led on')
        time.sleep(1)
        GPIO.output(ledPin, GPIO.LOW) # led off
        print ('led off...')
        time.sleep(1)
```

Finally, when the program is terminated, subfunction will be executed, the LED will be turned off and then the IO port will be released. If close the program terminal directly, the program will be terminated too, but destroy() function will not be executed. So, GPIO resources won't be released, in the warning message may appear next time you use GPIO. So, it is not a good habit to close the program terminal directly.

```
def destroy():
    GPIO.output(ledPin, GPIO.LOW)      # led is turned off
    GPIO.cleanup()                      # Release resource
```

About RPi.GPIO:

RPi.GPIO
This is a Python module to control the GPIO on a Raspberry Pi. It includes basic output function and input function of GPIO, and function used to generate PWM.
GPIO.setmode(mode)
Set the mode for pin serial number of GPIO. mode=GPIO.BOARD, which represents the GPIO pin serial number is based on physical location of RPi. mode=GPIO.BCM, which represents the pin serial number is based on CPU of BCM chip.
GPIO.setup(pin,mode)
Set pin to input mode or output mode. "pin" for the GPIO pin, "mode" for INPUT or OUTPUT.
GPIO.output(pin,mode)
Set pin to output mode. "pin" for the GPIO pin, "mode" for HIGH (high level) or LOW (low level).

For more functions related to RPi.GPIO, please refer to:
https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/
"import time" time is a module of python.
https://docs.python.org/2/library/time.html?highlight=time%20time#module-time

You can download the whole resource here:

https://github.com/Freenove/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/archive/master.zip

You can get Freenove Ultimate Starter Kit for Raspberry Pi, Freenove Three-wheeled Smart Car Kit for Raspberry Pi and other Freenove products here:

http://www.freenove.com/store.html

If you have any concerns, please feel free to contact us.

support@freenove.com