Getting Started

Thank you for choosing Freenove products!
After downloading the ZIP file we provide, unzip it and you will get a folder contains several files and folders. There are two PDF files:

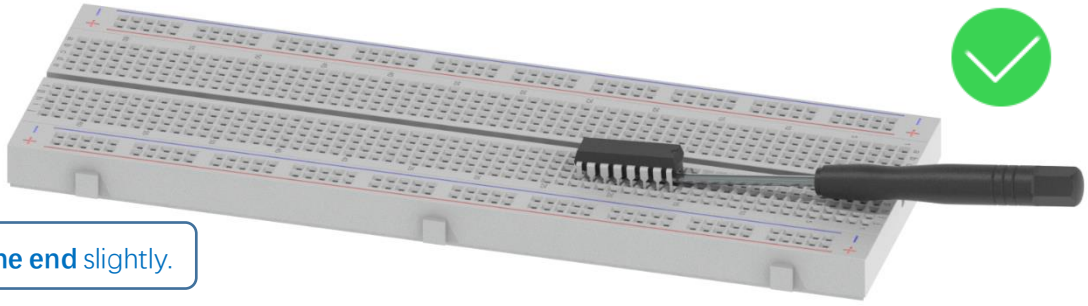
- **Tutorial.pdf**
  It contains basic operations such as installing system for Raspberry Pi.
  The code in this PDF is in C and Python.
- **Processing.pdf** in Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi\Processing
  The code in this PDF is in Java.

- **Scratch.pdf in** Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi\Scratch3
  The code in this PDF is in Scratch.

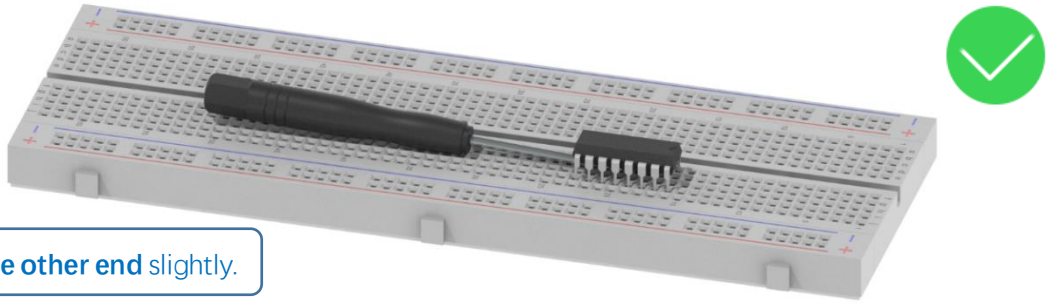
We recommend you to start with Tutorial.pdf first.
If you want to start with Processing.pdf or skip some chapters of Tutorial.pdf, you need to finish necessary steps in **Chapter 7 AD/DA** of **Tutorial.pdf** first.

Remove the Chips

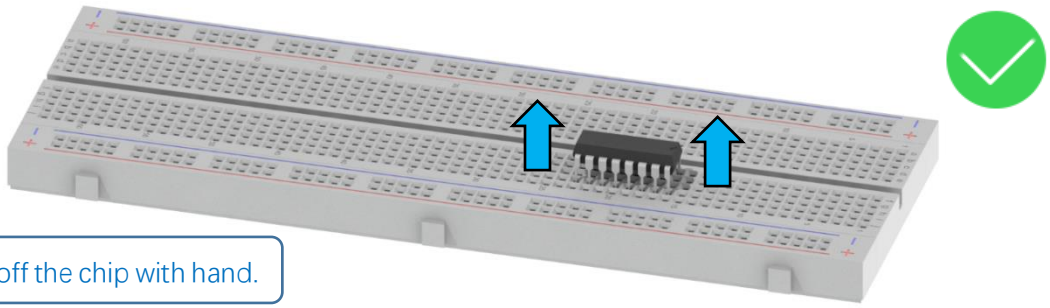
Some chips and modules are inserted into the breadboard to protect their pins.
You need to remove them from breadboard before use. (There is no need to remove GPIO Extension Board.)
Please find a tool (like a little screw driver) to remove them as below:



Step 1, lift **one end** slightly.

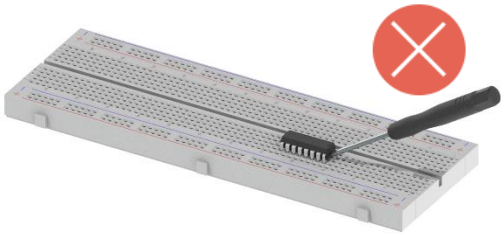


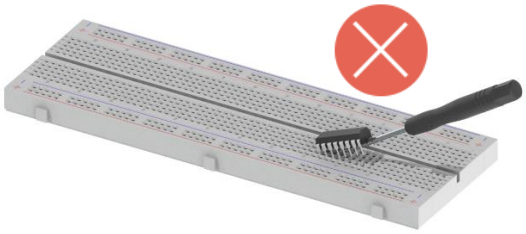
Step 2, lift **the other end** slightly.



Step 3, take off the chip with hand.

Avoid lifting one end with big angle directly.





Get Support and Offer Input

Freenove provides free and responsive product and technical support, including but not limited to:

- Product quality issues
- Product use and build issues
- Questions regarding the technology employed in our products for learning and education
- Your input and opinions are always welcome
- We also encourage your ideas and suggestions for new products and product improvements

For any of the above, you may send us an email to:

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:

- Keep this product out of the reach of children under 6 years old.
- This product should be used only when there is adult supervision present as young children lack necessary judgment regarding safety and the consequences of product misuse.
- This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.
- When the product is turned ON, activated or tested, some parts will move or rotate. To avoid injuries to hands and fingers, keep them away from any moving parts!
- It is possible that an improperly connected or shorted circuit may cause overheating. Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it

cools down! When everything is safe and cool, review the product tutorial to identify the cause.
- Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
- Store the product in a cool dry place and avoid exposing the product to direct sunlight.
- After use, always turn the power OFF and remove or unplug the batteries before storing.

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

http://www.freenove.com

Copyright

Contents

Preface

Raspberry Pi is a low cost, **credit card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is an incredibly capable little device that enables people of all ages to explore computing, and to learn how to program in a variety of computer languages like Scratch and Python. It is capable of doing everything you would expect from a desktop computer, such as browsing the internet, playing high-definition video content, creating spreadsheets, performing word-processing, and playing video games. For more information, you can refer to Raspberry Pi official website. For clarification, this tutorial will also reference Raspberry Pi as RPi, RPI and RasPi.

In this tutorial, most chapters consist of **Components List**, **Component Knowledge**, **Circuit**, and **Code** (**C** code and **Python** code). We provide both C and Python code for each project in this tutorial. After completing this tutorial, you can learn Java by reading Processing.pdf.

This kit does not contain **Raspberry and its accessories**. You can also use the components and modules in this kit to create projects of your own design.

Additionally, if you encounter any issues or have questions about this tutorial or the contents of kit, you can always contact us for free technical support at:

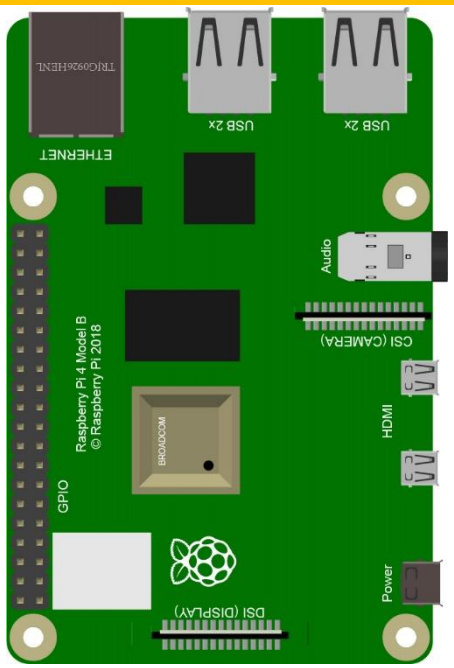
support@freenove.com

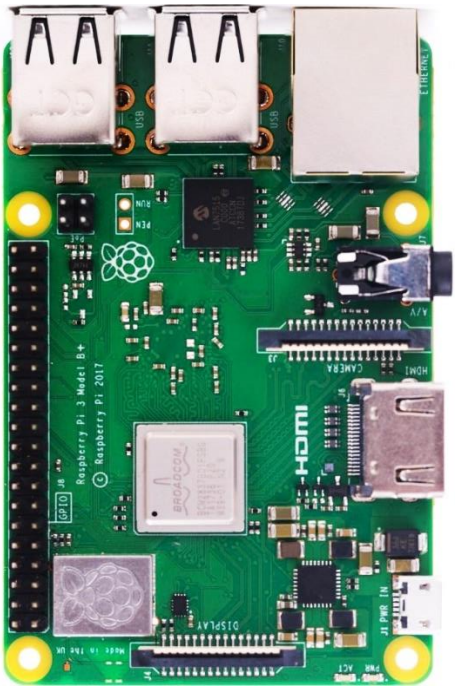
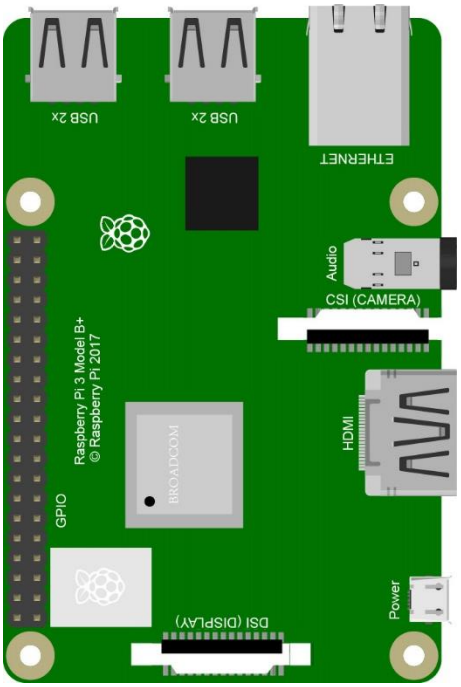
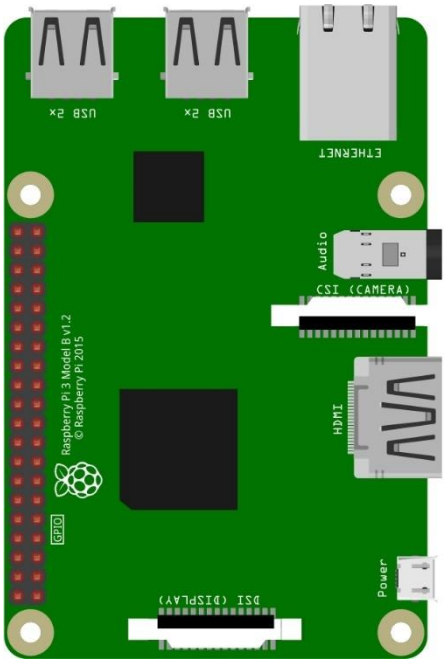
Raspberry Pi

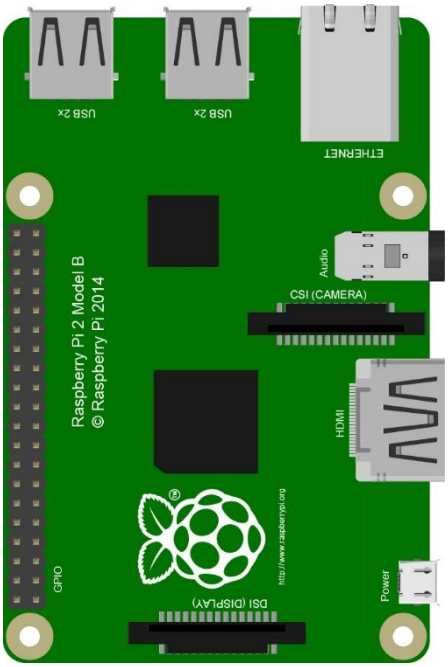
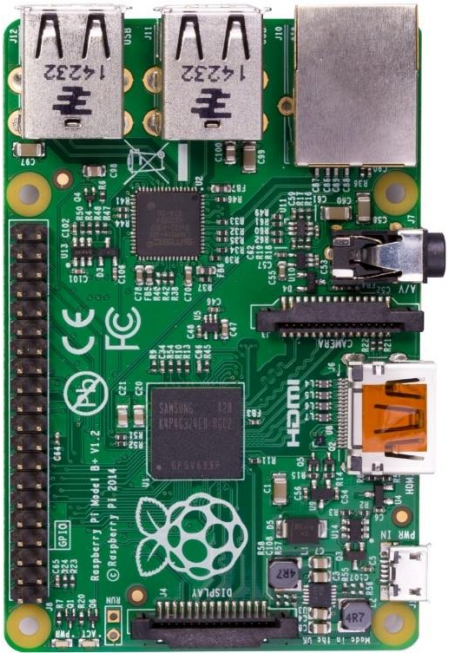
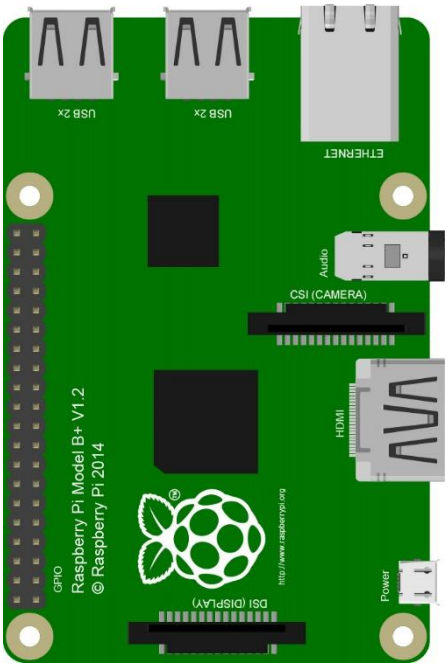
So far, at this writing, Raspberry Pi has advanced to its fourth generation product offering. Version changes are accompanied by increases in upgrades in hardware and capabilities.

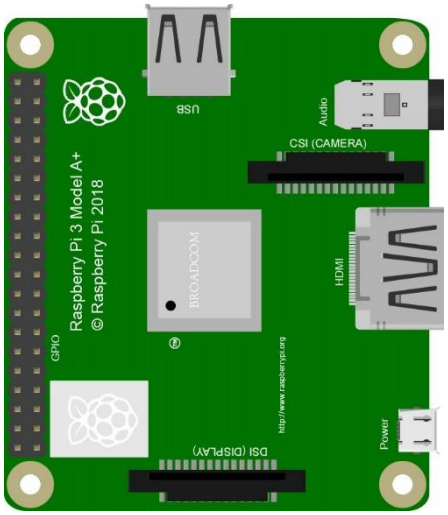
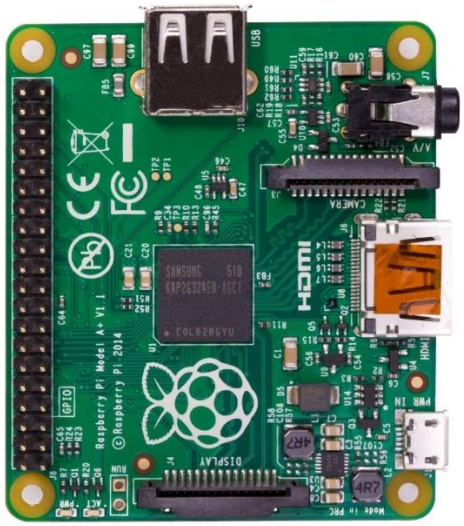
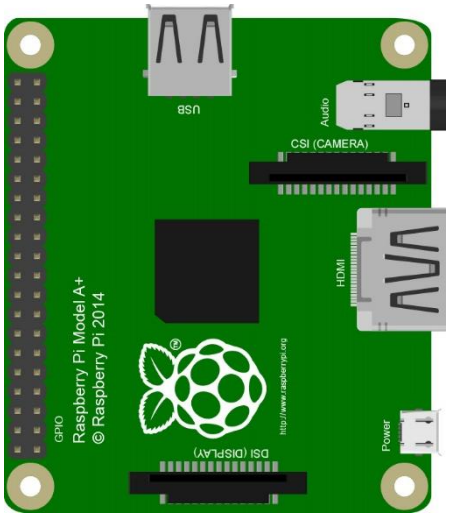
The A type and B type versions of the first generation products have been discontinued due to various reasons. What is most important is that other popular and currently available versions are consistent in the order and number of pins and their assigned designation of function, making compatibility of peripheral devices greatly enhanced between versions.

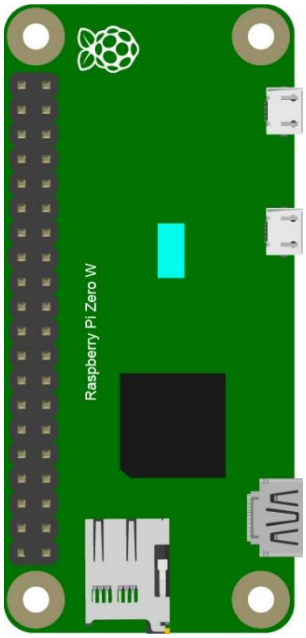
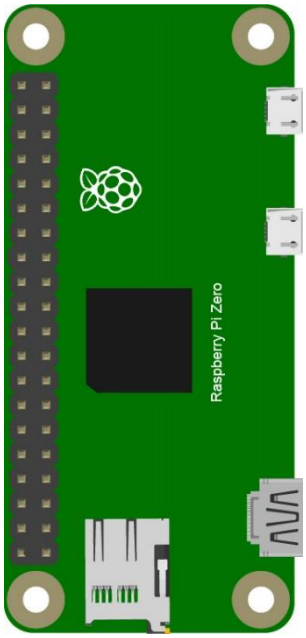
Below are the raspberry pi pictures and model pictures supported by this product. They have 40 pins.

Actual image of Raspberry Pi 4 Model B:	CAD image of Raspberry Pi 4 Model B:
	

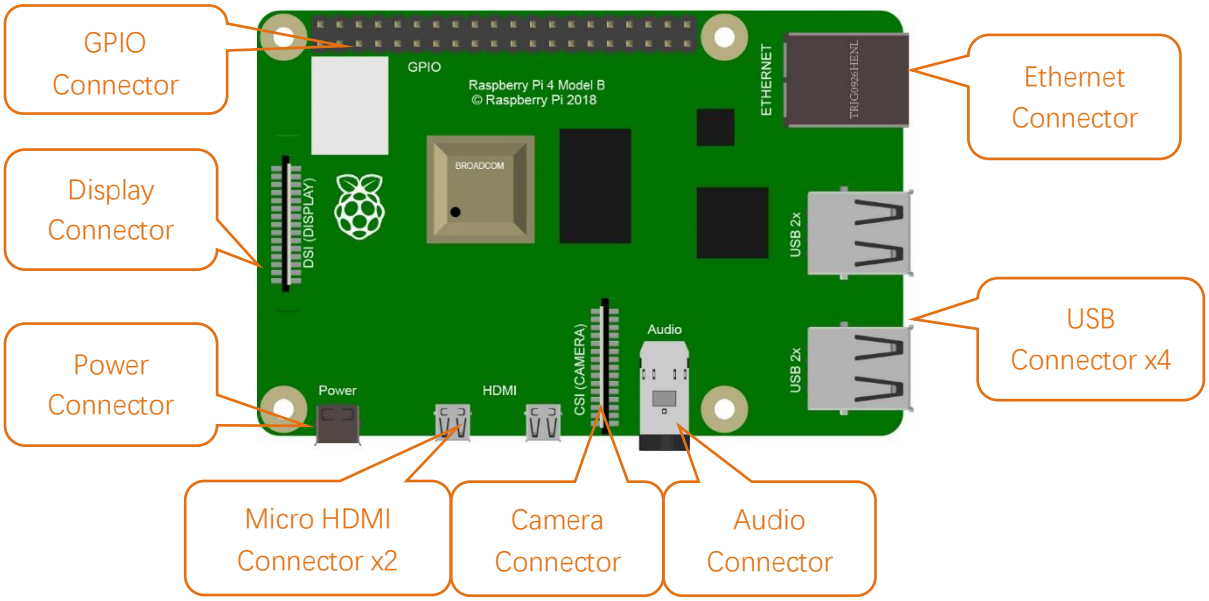
Actual image of Raspberry Pi 3 Model B+: 	CAD image of Raspberry Pi 3 Model B+: 
Actual image of Raspberry Pi 3 Model B: 	CAD image of Raspberry Pi 3 Model B: 

Actual image of Raspberry Pi 2 Model B:	CAD image of Raspberry Pi 2 Model B:
	
Actual image of Raspberry Pi 1 Model B+:	CAD image of Raspberry Pi 1 Model B+:
	

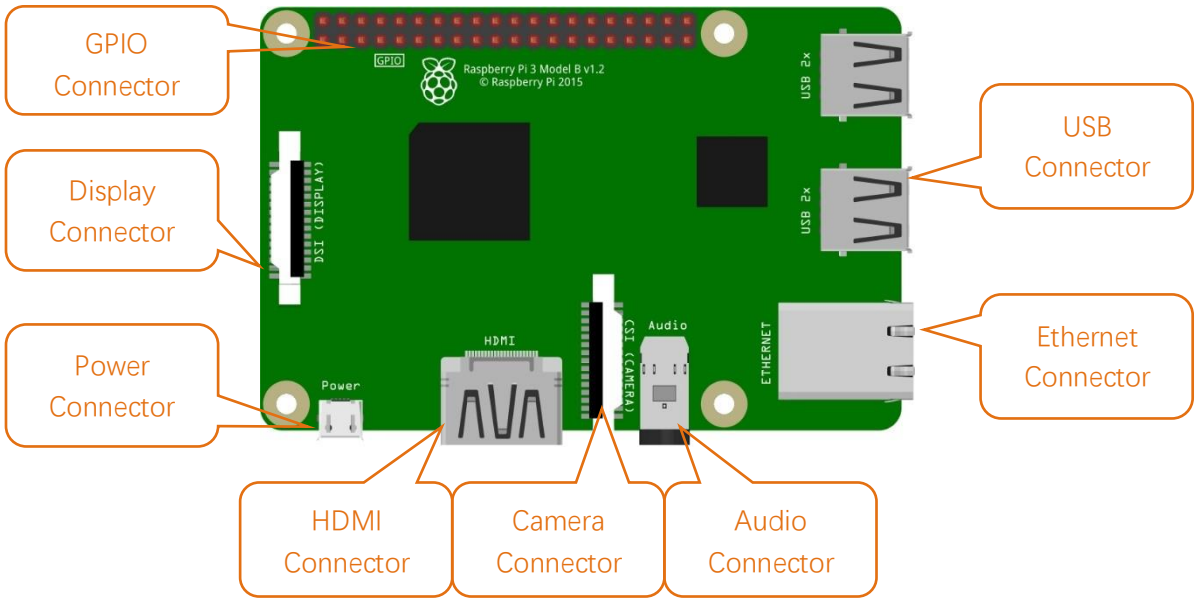
Actual image of Raspberry Pi 3 Model A+: 	CAD image of Raspberry Pi 3 Model A+: 
Actual image of Raspberry Pi 1 Model A+: 	CAD image of Raspberry Pi 1 Model A+: 

Actual image of Raspberry Pi Zero W: 	CAD image of Raspberry Pi Zero W: 
Actual image of Raspberry Pi Zero: 	CAD image of Raspberry Pi Zero: 

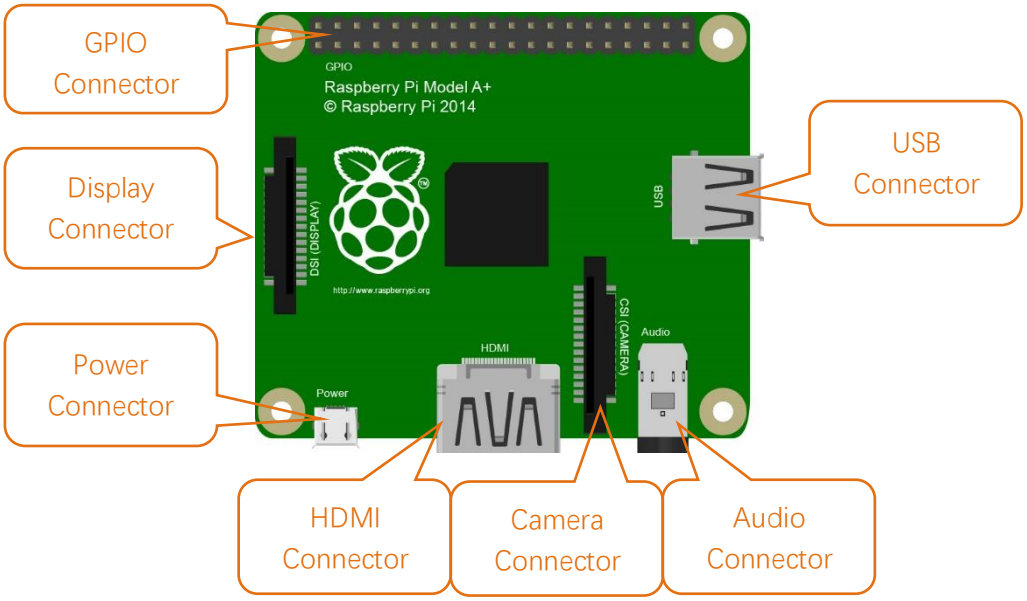
Hardware interface diagram of RPi 4B:



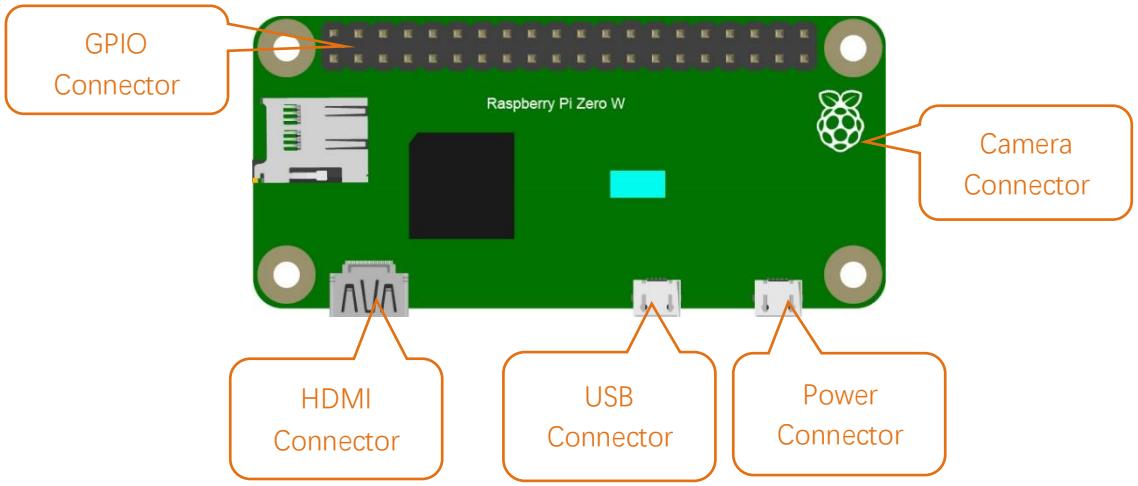
Hardware interface diagram of RPi 3B+/3B/2B/1B+:



Hardware interface diagram of RPi 3A+/A+:



Hardware interface diagram of RPi Zero/Zero W:

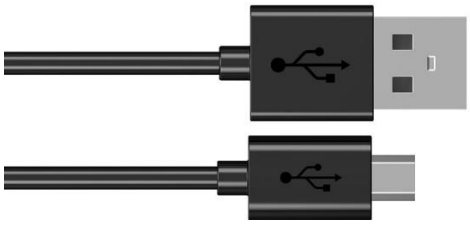


Installing an Operating System

The first step is to install an operating system on your RPi so that it can be programmed and function. If you have installed a system in your RPi, you can start from Chapter 0 Preparation.

Component List

Required Components

Any Raspberry Pi with 40 GPIO 	5V/3A Power Adapter. Note: Different versions of Raspberry Pi have different power requirements (please check the power requirements for yours on the chart in the following page.) 
Micro or Type-C USB Cable x1 	Micro SD Card (TF Card) x1, Card Reader x1 

Power requirements of various versions of Raspberry Pi are 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 an Ethernet network cable used to connect it to a WAN (Wide Area Network).

All these components are necessary for any of your projects to work. Among them, the power supply of at least 5V/2.5A, because a lack of a sufficient power supply may lead to many functional issues and even damage your RPi, we STRONGLY RECOMMEND a 5V/2.5A power supply. We also recommend using a SD Micro Card with a capacity of 16GB or more (which, functions as the RPI's "hard drive") and is used to store the operating system and necessary operational files.

Optional Components

Under normal circumstances, there are two ways to login to Raspberry Pi: 1) Using a stand-alone monitor. 2) Using a remote desktop or laptop computer monitor "sharing" the PC monitor with your RPi.

Required Accessories for Monitor

If you choose to use an independent monitor, mouse and keyboard, you also need the following accessories:
1. A display with a HDMI interface
2. A Mouse and a Keyboard with an USB interface

As to Pi Zero and Pi Zero W, you also need the following accessories:
1. A Mini-HDMI to HDMI Adapter and Cable.
2. A Micro-USB to USB-A Adapter and Cable (Micro USB OTG Cable).
3. A USB HUB.
4. USB to Ethernet Interface or USB Wi-Fi receiver.

For different Raspberry Pi Modules, the optional items may vary slightly but they all aim to convert the interfaces to Raspberry Pi standards.

<table>
<tr><th></th><th>Pi Zero</th><th>Pi A+</th><th>Pi Zero W</th><th>Pi 3A+</th><th>Pi B+/2B</th><th>Pi 3B/3B+</th><th>Pi 4B</th></tr>
<tr><td>Monitor</td><td colspan="7">Yes (All)</td></tr>
<tr><td>Mouse</td><td colspan="7">Yes (All)</td></tr>
<tr><td>Keyboard</td><td colspan="7">Yes (All)</td></tr>
<tr><td>Micro-HDMI to HDMI Adapter & Cable</td><td>Yes</td><td>No</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td>Micro-HDMI to HDMI Adapter & Cable</td><td colspan="6">No</td><td>Yes</td></tr>
<tr><td>Micro-USB to USB-A Adapter & Cable (Micro USB OTG Cable)</td><td>Yes</td><td>No</td><td>Yes</td><td colspan="4">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><td></td></tr>
<tr><td>USB to Ethernet Interface</td><td colspan="2" rowspan="2">select one from two or select two from two</td><td colspan="2" rowspan="2">optional</td><td rowspan="2">Internal Integration</td><td colspan="2" rowspan="2">Internal Integration</td></tr>
<tr><td>USB Wi-Fi Receiver</td></tr>
<tr><td></td><td colspan="2"></td><td colspan="2">Internal Integration</td><td>optional</td><td colspan="2"></td></tr>
</table>

Required Accessories for Remote Desktop

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

	Pi Zero	Pi Zero W	Pi A+	Pi 3A+	Pi B+/2B	Pi 3B/3B+/4B
Micro-USB to USB-A Adapter & Cable (Micro USB OTG Cable)	Yes	Yes	No	NO		
USB to Ethernet interface	Yes	Yes	Yes			

Raspberry Pi OS

Without Screen - Use Raspberry Pi - under Windows PC: https://youtu.be/YND0RUuP-to

With Screen - Use Raspberry Pi - under Windows PC: https://youtu.be/HEywFsFrj3I

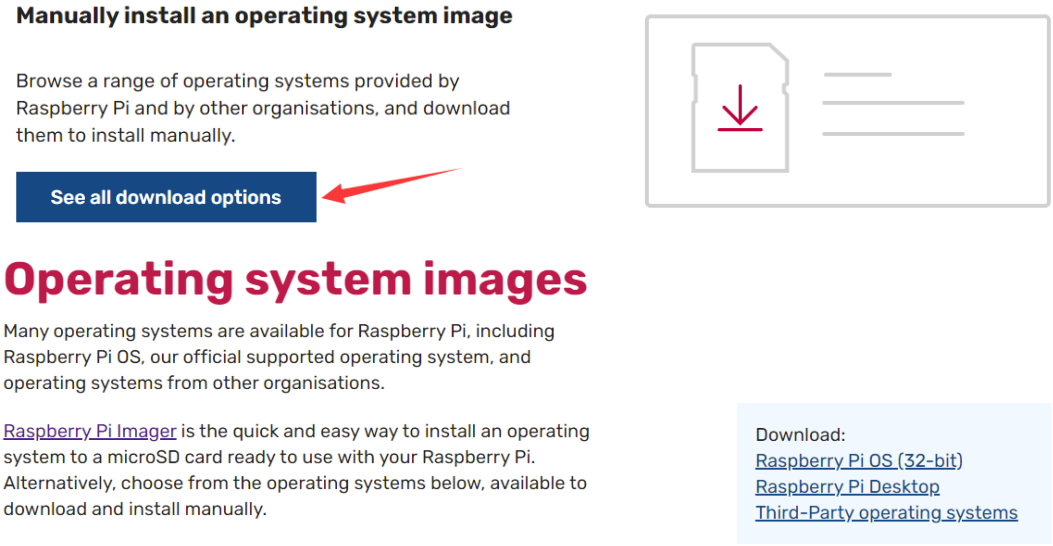
Automatically Method

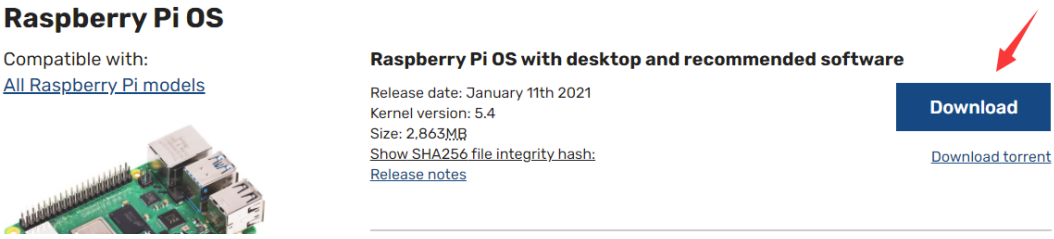
You can follow the official method to install the system for raspberry pi via visiting link below:
https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/2
In this way, the system will be downloaded **automatically** via the application.

Manually Method

After installing the Imager Tool in the **link above**. You can **also** download the system **manually** first.

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



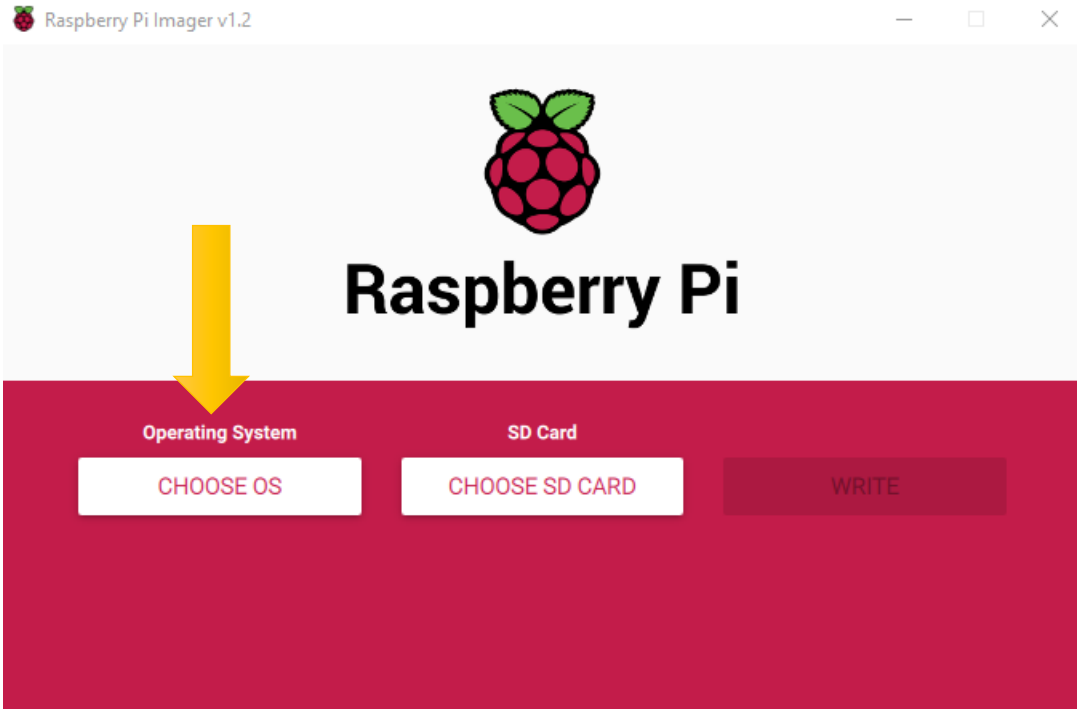


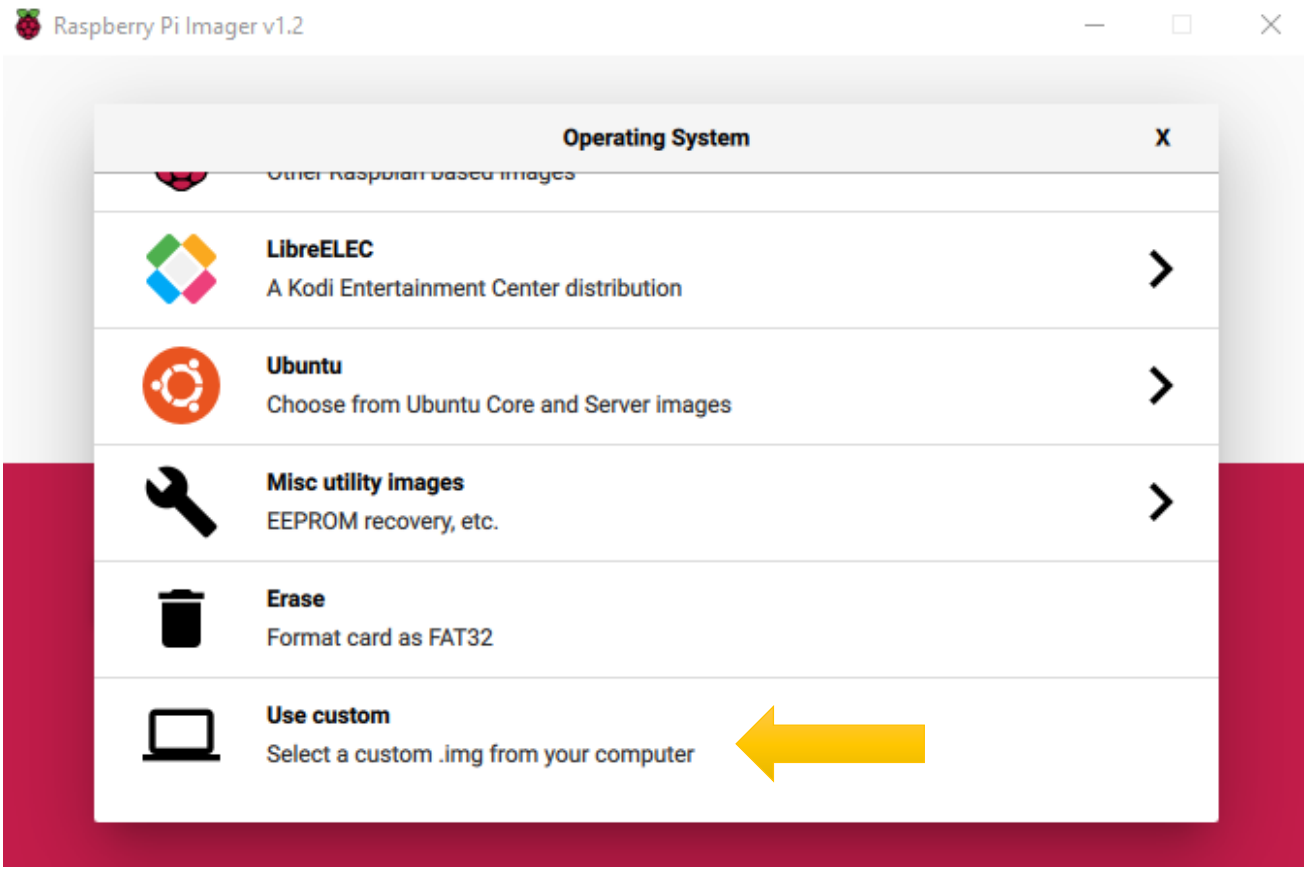
And then the zip file is downloaded.

Write System to Micro SD Card

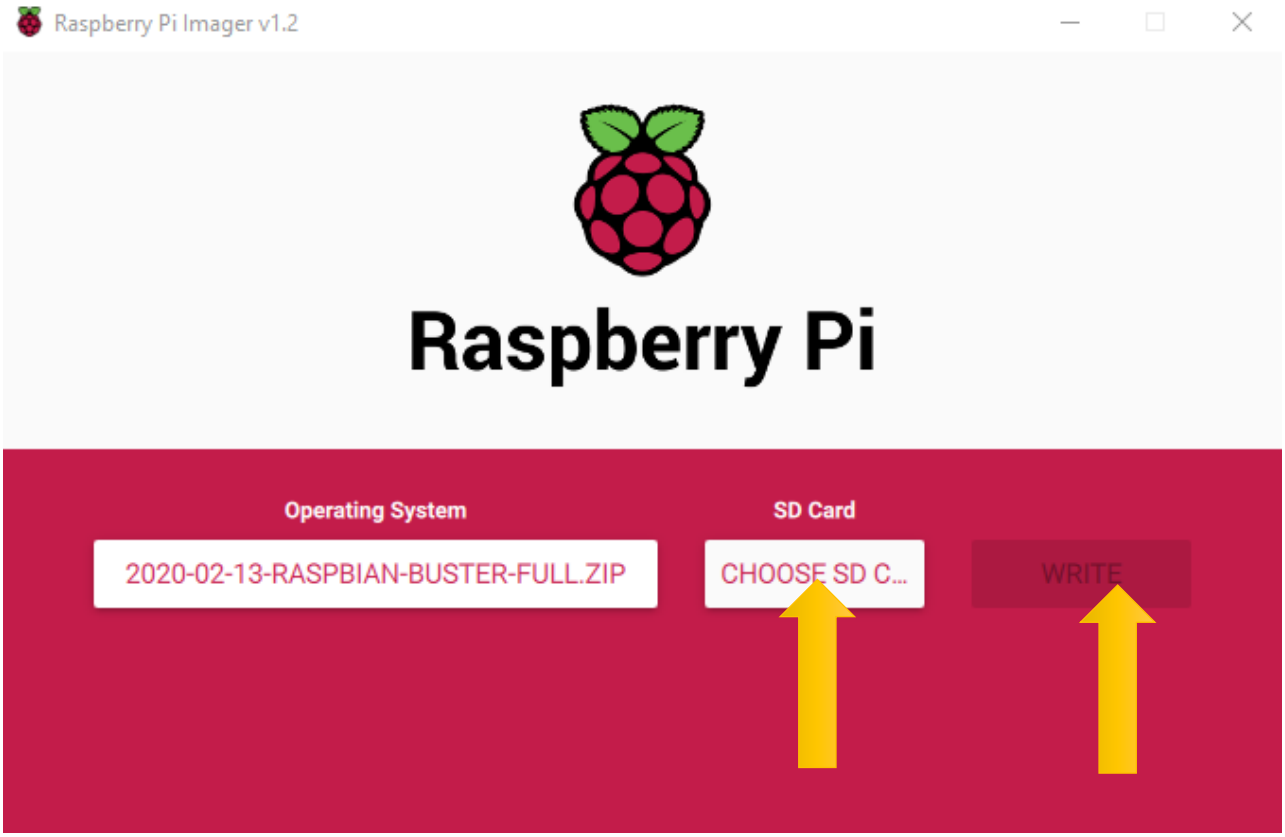
First, put your Micro **SD card** into card reader and connect it to USB port of PC.

Then open imager toll. Choose system that you just downloaded in Use custom.



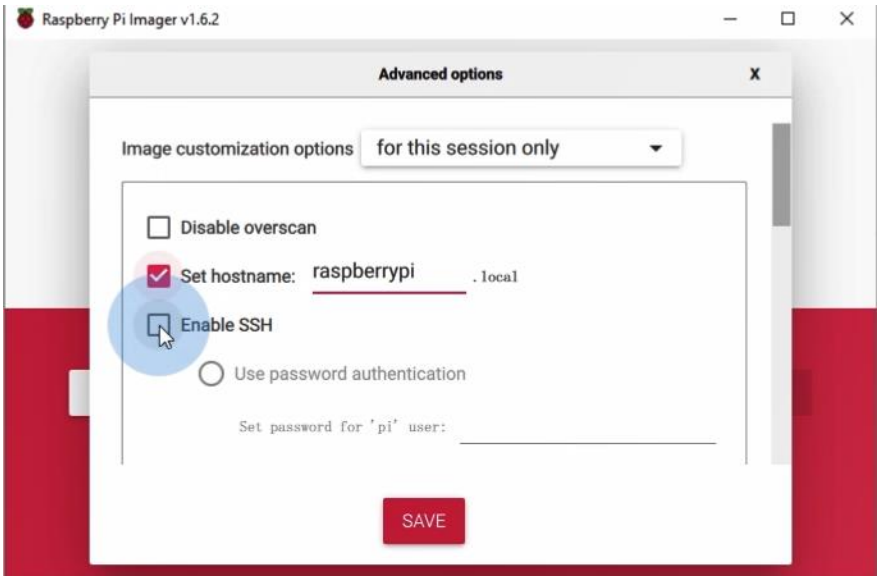


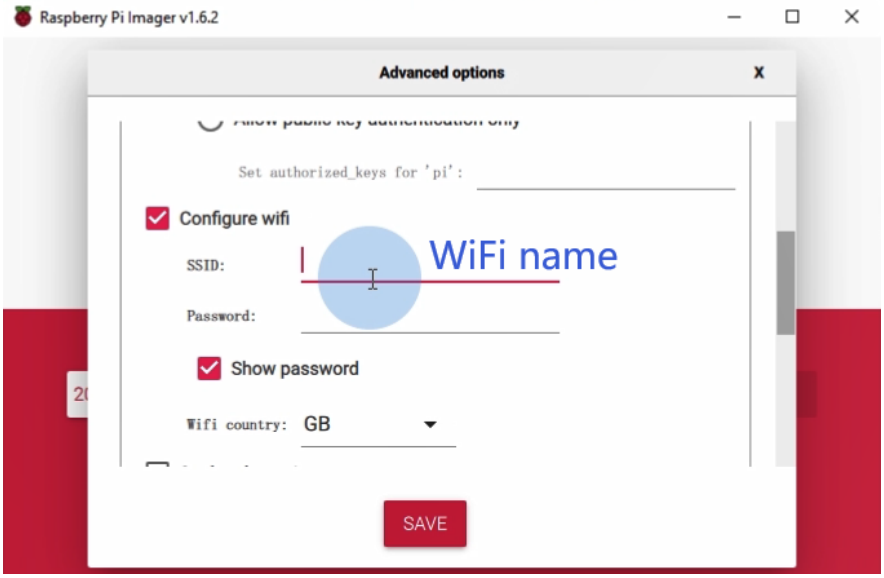
Choose the SD card. Then click "WRITE".

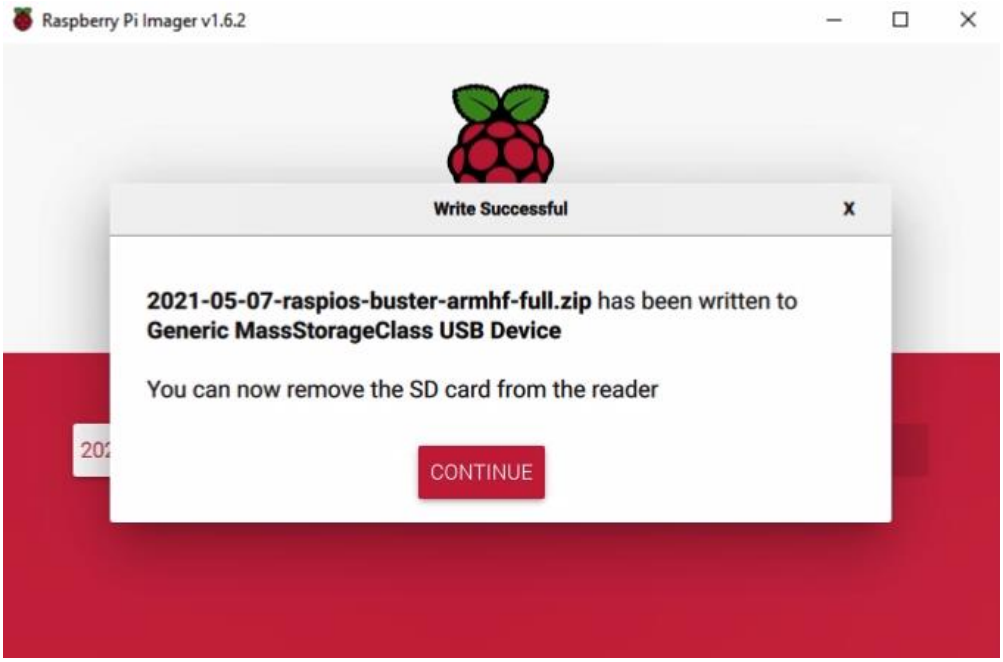


Enable ssh and configure WiFi

Press Ctrl+Shift+x to configure RPi.

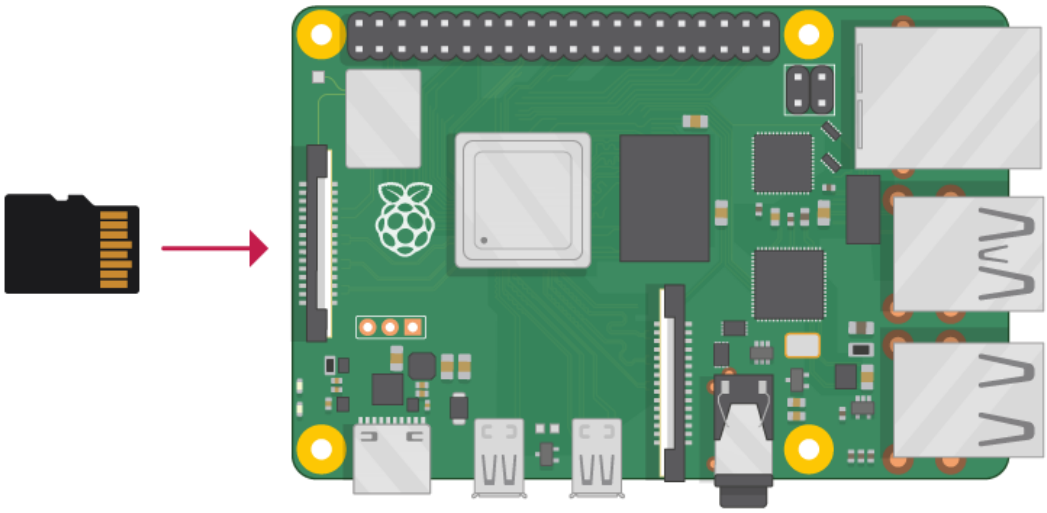






Insert SD card

Then remove SD card from card reader and insert it into Raspberry Pi.



Getting Started with Raspberry Pi

Monitor desktop

If you do not have a spare monitor, please skip to next section Remote desktop & VNC. If you have a spare monitor, please follow the steps in this section.

After the system is written successfully, take out Micro SD Card and put it into the SD card slot of RPi. Then connect your RPi to the monitor through the HDMI port, attach your mouse and keyboard through the USB ports, attach a network cable to the network port and finally, connect your power supply (making sure that it meets the specifications required by your RPi Module Version. Your RPi should start (power up). Later, after setup, you will need to enter your user name and password to login. The default user name: pi; password: raspberry. After login, you should see the following screen.

Congratulations! You have successfully installed the RASPBERRY PI OS operating system on your RPi.

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.

Remote desktop & VNC

If you have logged in Raspberry Pi via display, you can skip to VNC Viewer.

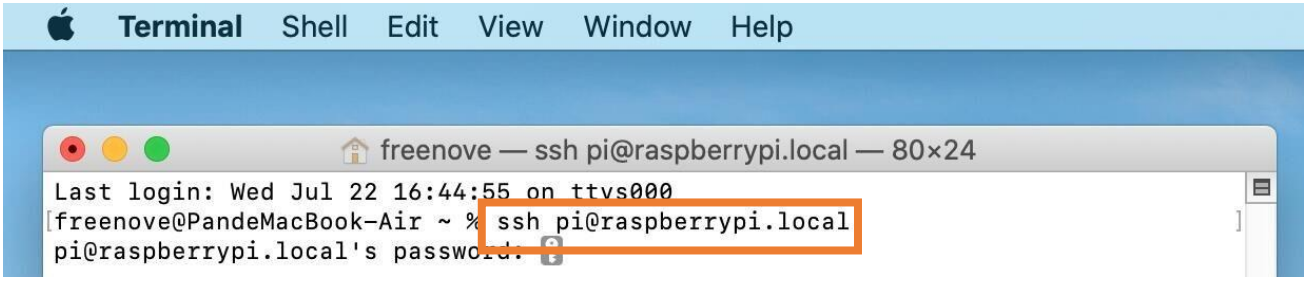
If you don't have a spare display, mouse and keyboard for your RPi, you can use a remote desktop to share a display, keyboard, and mouse with your PC. Below is how to use:
MAC OS remote desktop and Windows OS remote desktop.

MAC OS Remote Desktop

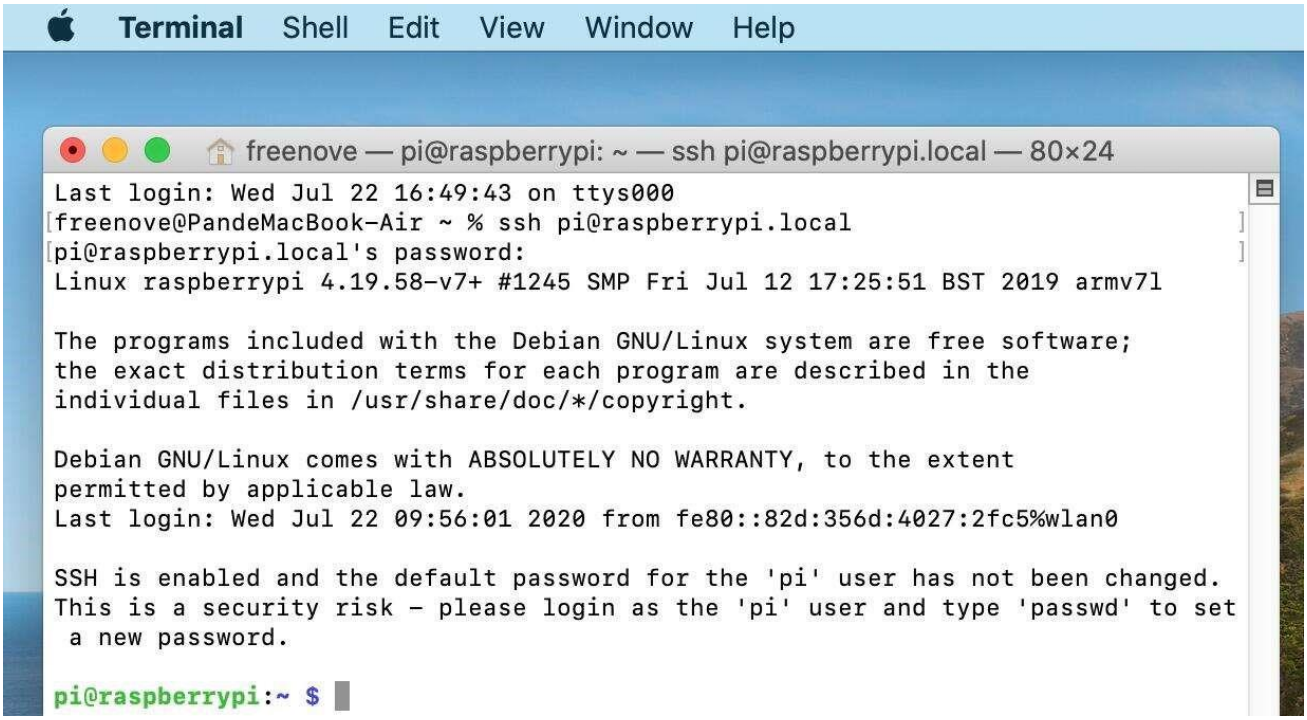
Open the terminal and type following command. **If this command doesn't work, please move to next page.**

```
ssh pi@raspberrypi.local
```

The password is **raspberry** by default, case sensitive.



You may need to type **yes** during the process.

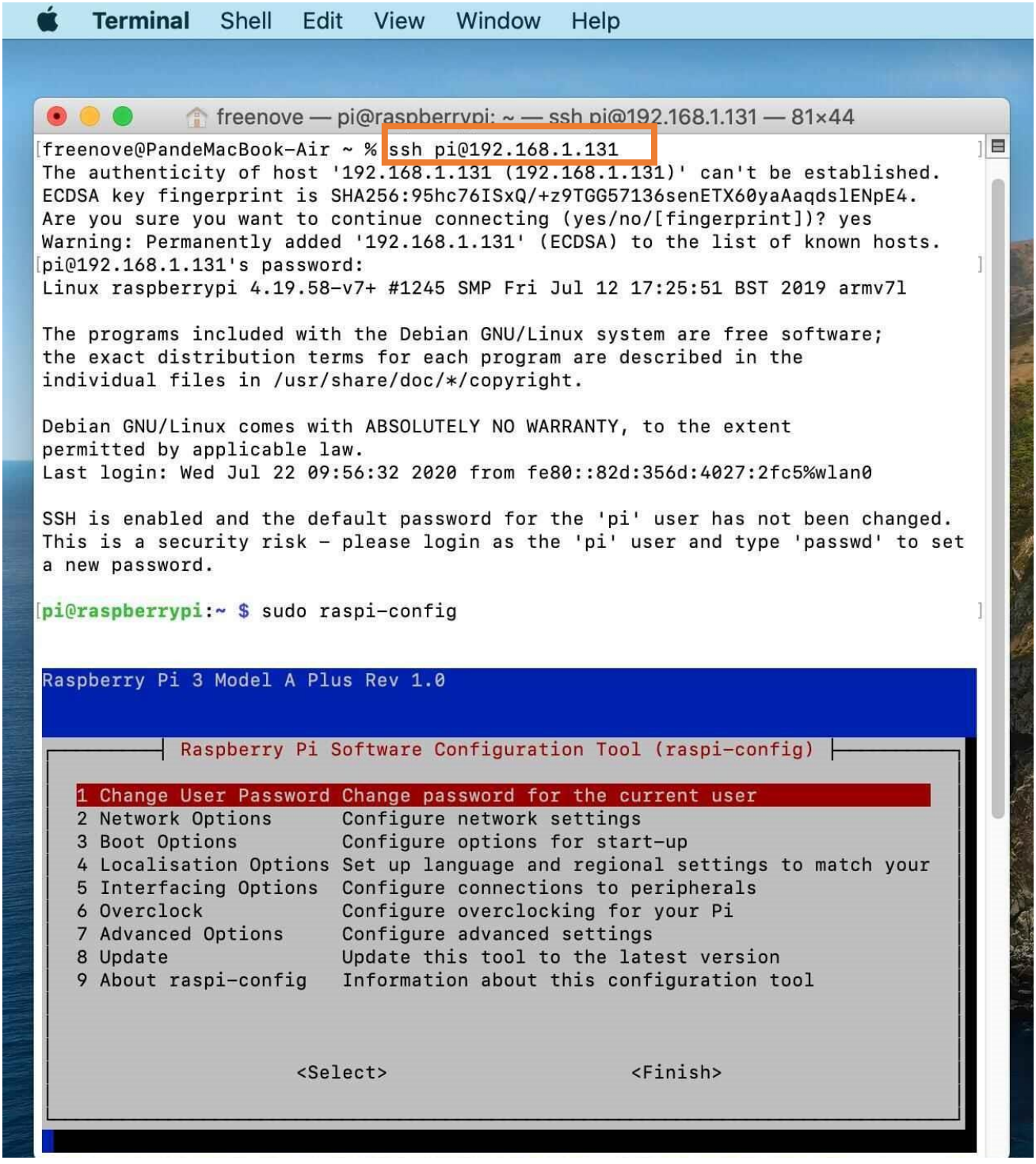


You can also use the IP address to log in Pi.
Enter **router** client to **inquiry IP address** named "raspberry pi". For example, I have inquired to **my RPi IP address, and it is "192.168.1.131".**

Open the terminal and type following command.

```
ssh pi@192.168.1.131
```

When you see pi@raspberrypi:~ $, you have logged in Pi successfully. Then you can skip to next section.



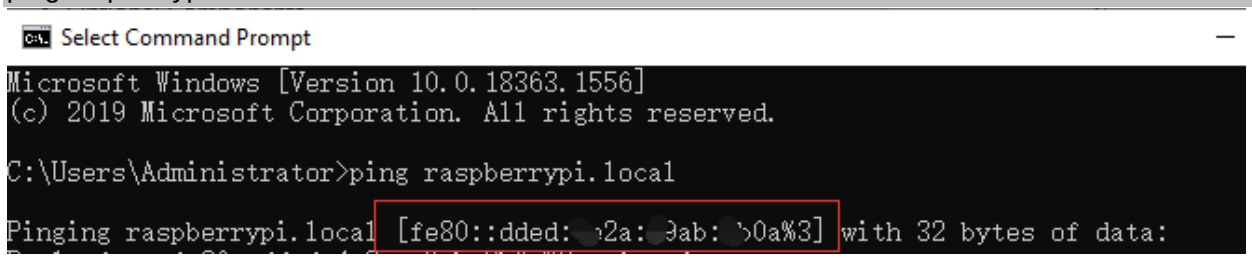
Then you can skip to VNC Viewer.

Windows OS Remote Desktop

If you are using win10, you can use follow way to login RaspberryPi without desktop.

Press **Win+R**. Enter **cmd**. Then use this command to check IP:

```
ping raspberrypi.local
```

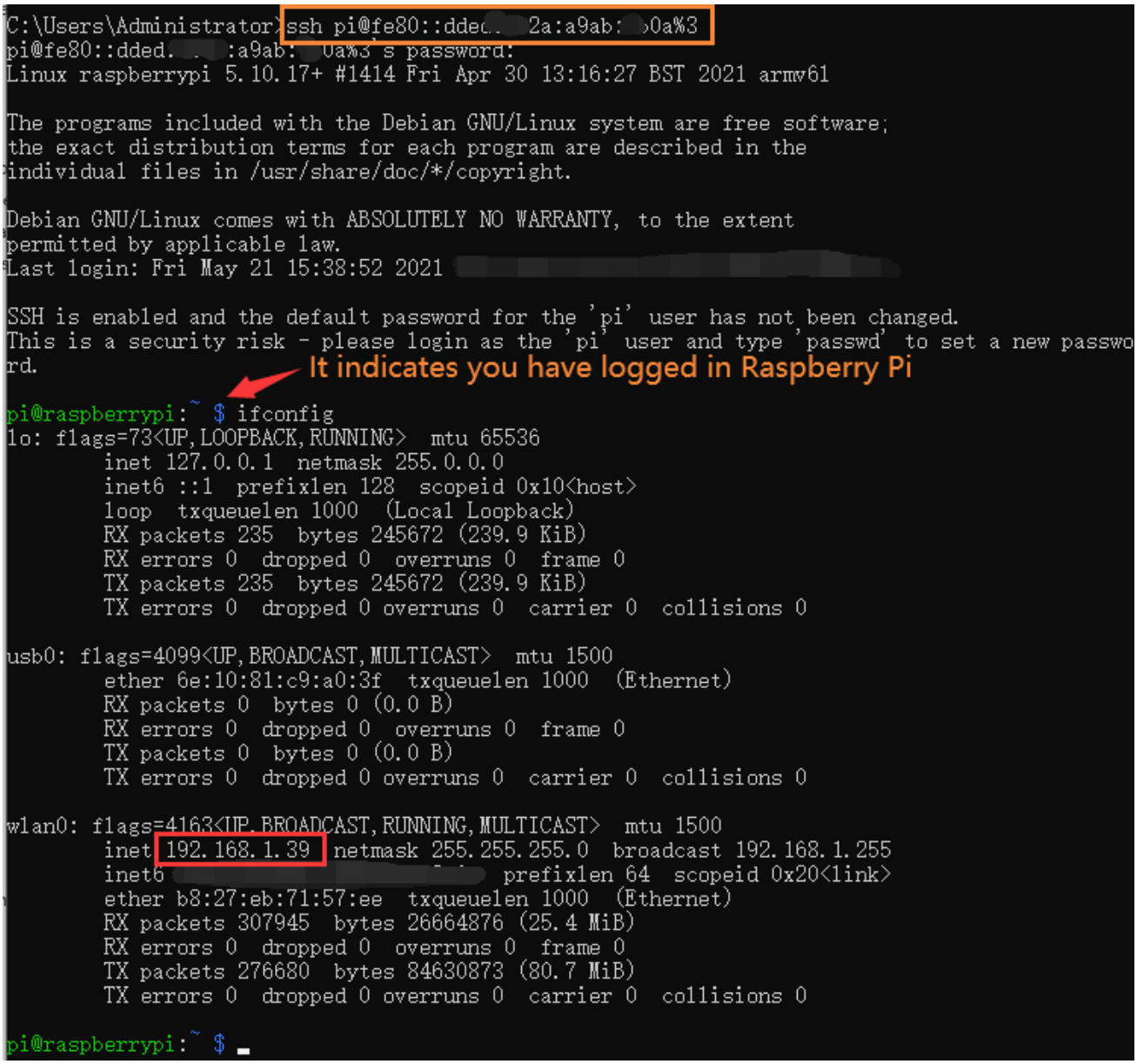


The one befind raspberrypi.local is the IPV6 address of RaspberryPi

Use following command to login Raspberry Pi.

```
ssh pi@xxxxxxxxxx(IPV6 address)
```

Enter yes not y if needed.

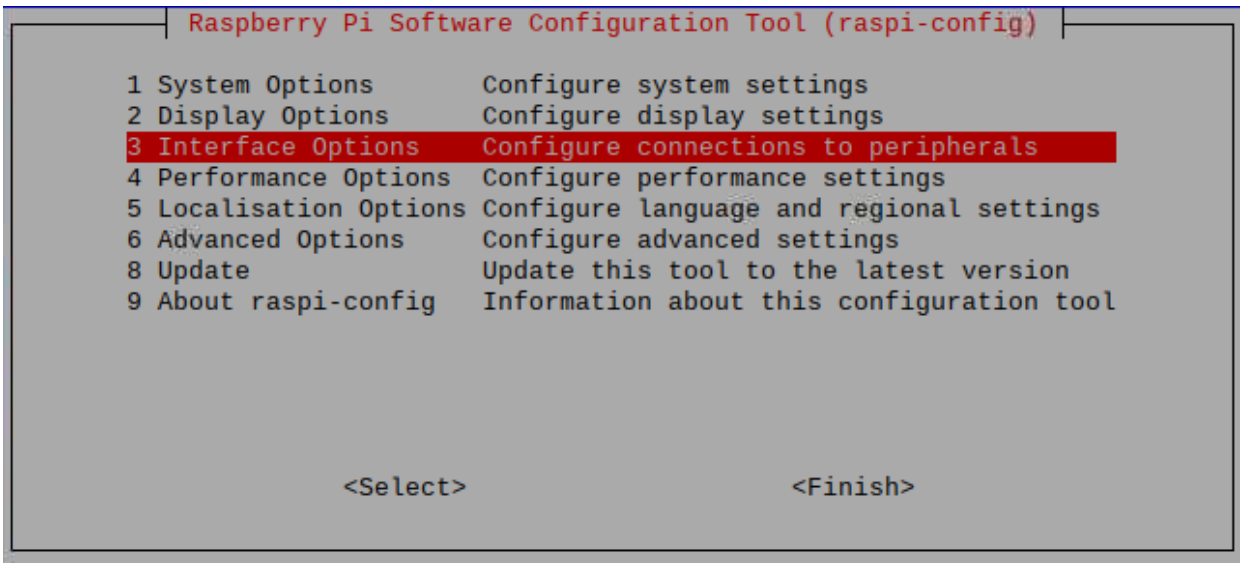


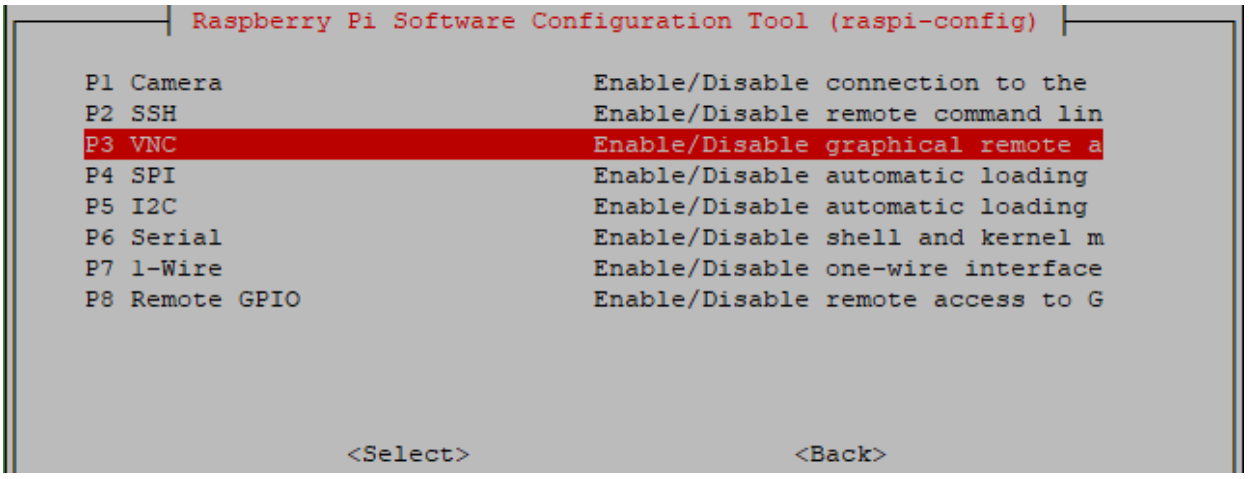
VNC Viewer & VNC

Enable VNC

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

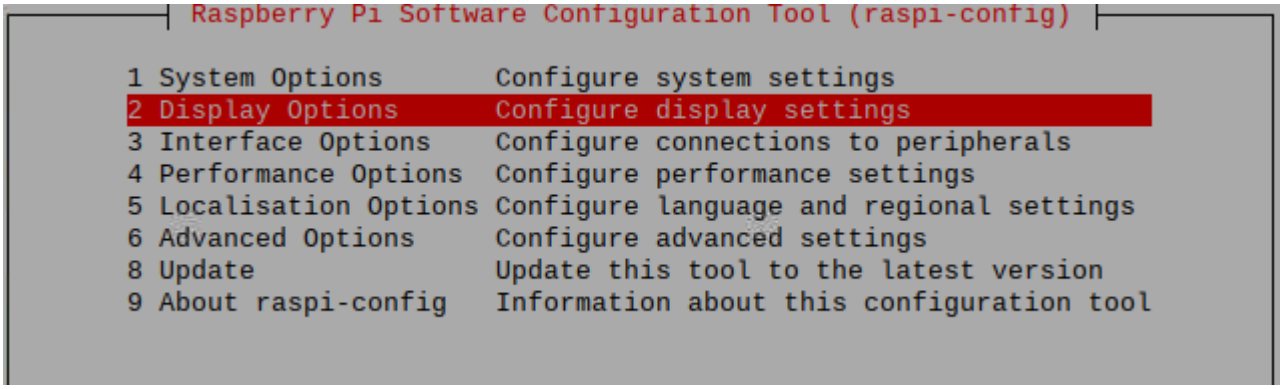
```
sudo raspi-config
```

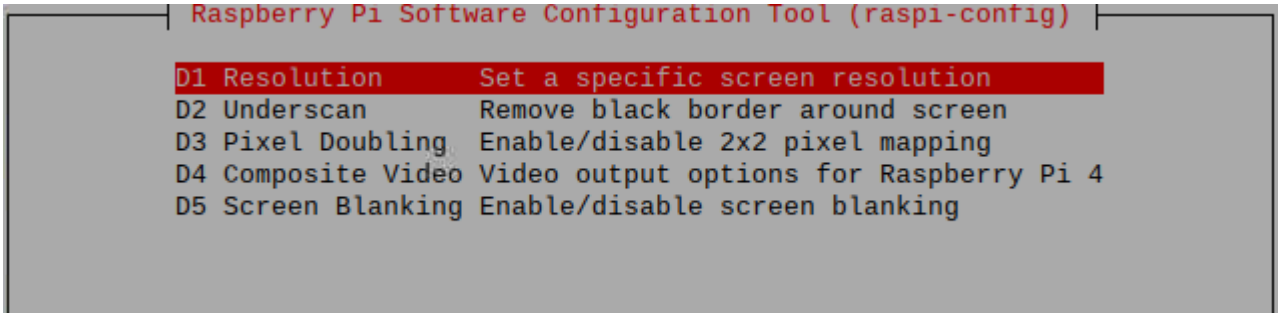


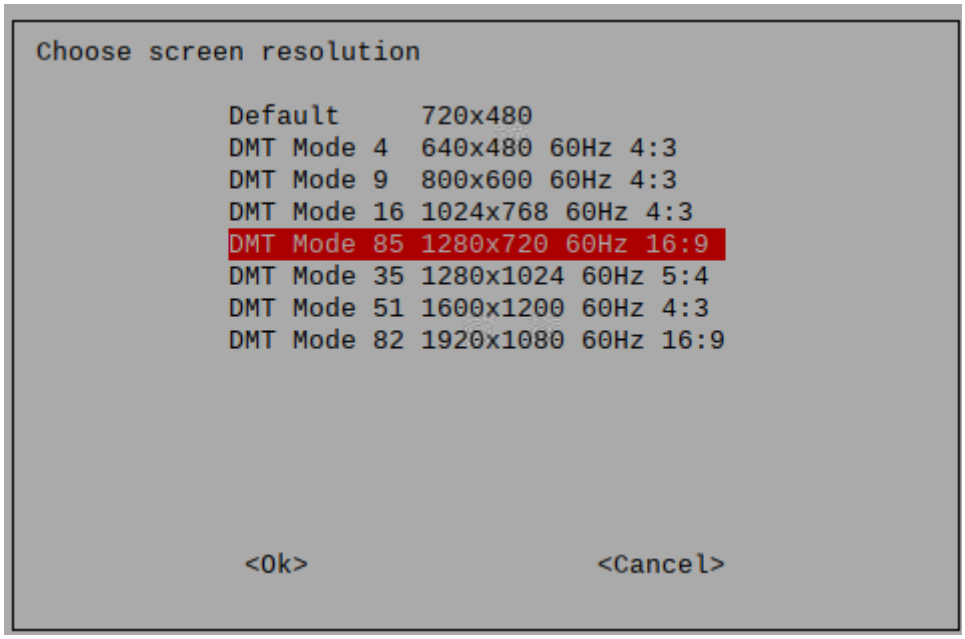


Set Resolution

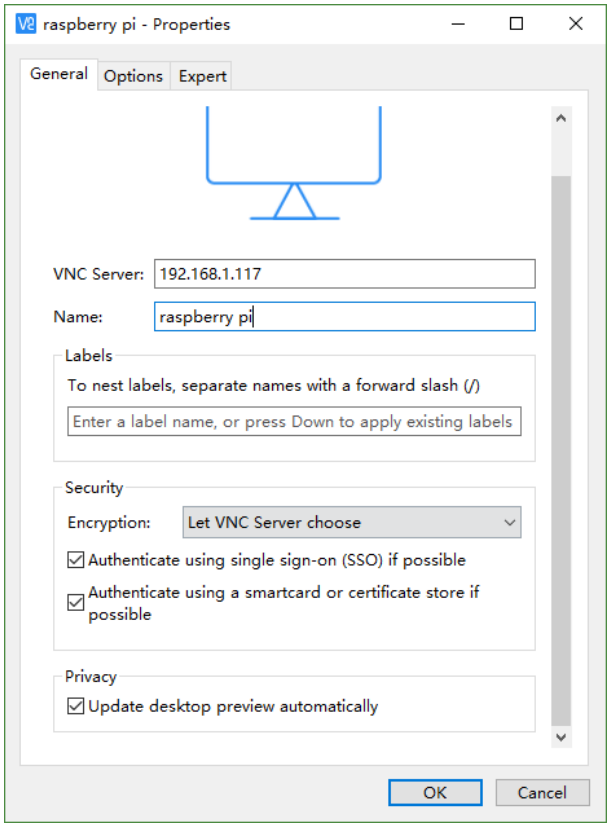
You can also set other resolutions. If you don't know what to set, you can set it as 1280x720 first.







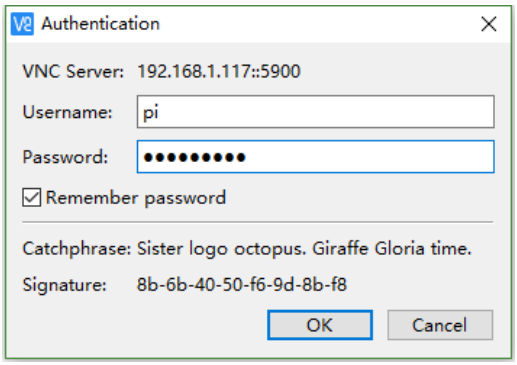
Then download and install VNC Viewer according to your computer system by click following link:
https://www.realvnc.com/en/connect/download/viewer/
After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.



Enter ip address of your Raspberry Pi and fill in a name. Then click OK.

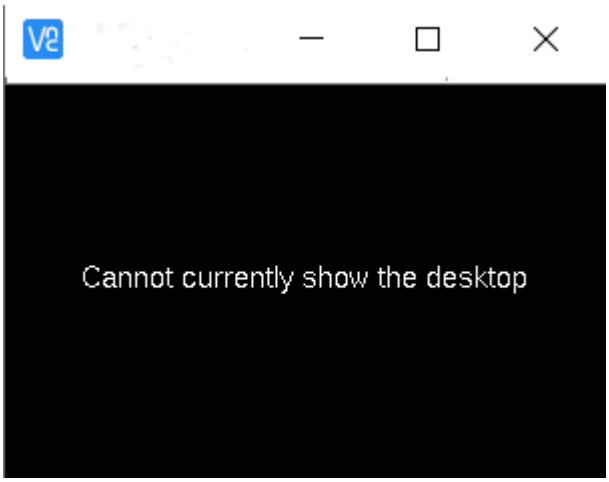
Then on the VNC Viewer panel, double-click new connection you just created,

and the following dialog box pops up.

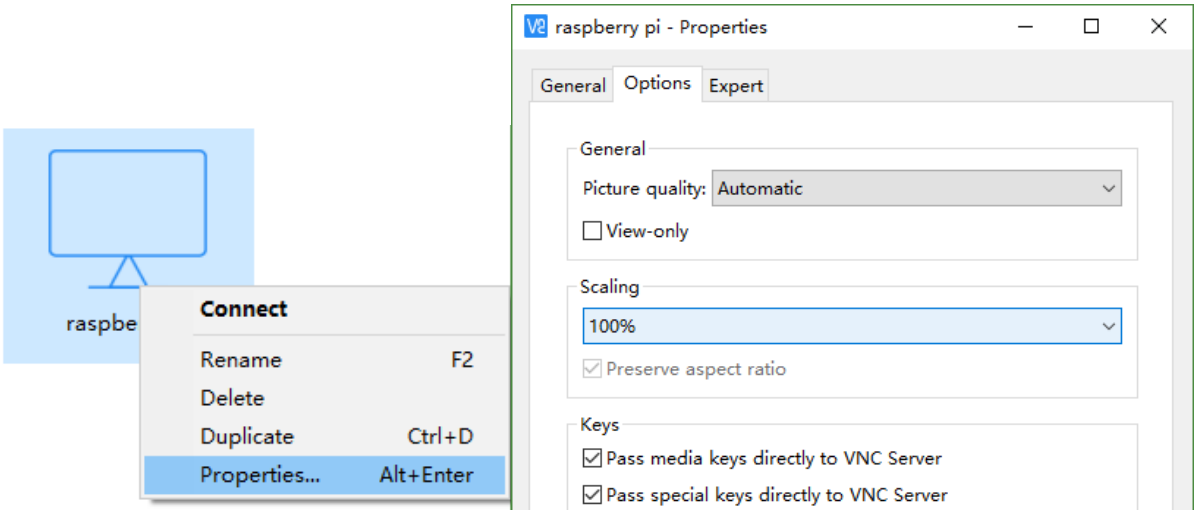


Enter username: **pi** and Password: **raspberry**. And click OK.

Here, you have logged in to Raspberry Pi successfully by using VNC Viewer
If there is black window, please set another resolution.



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.

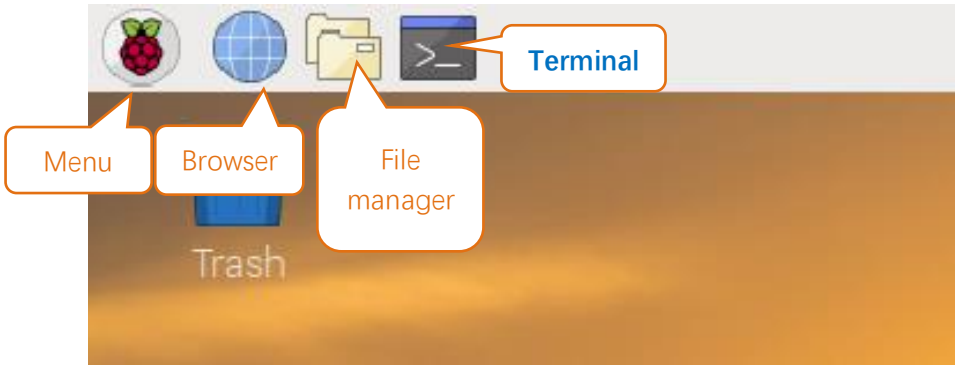
Raspberry Pi 4B/3B+/3B integrates a Wi-Fi adaptor.If you did not connect Pi to WiFi. You can connect it to wirelessly control the robot.

Chapter 0 Preparation and Get code

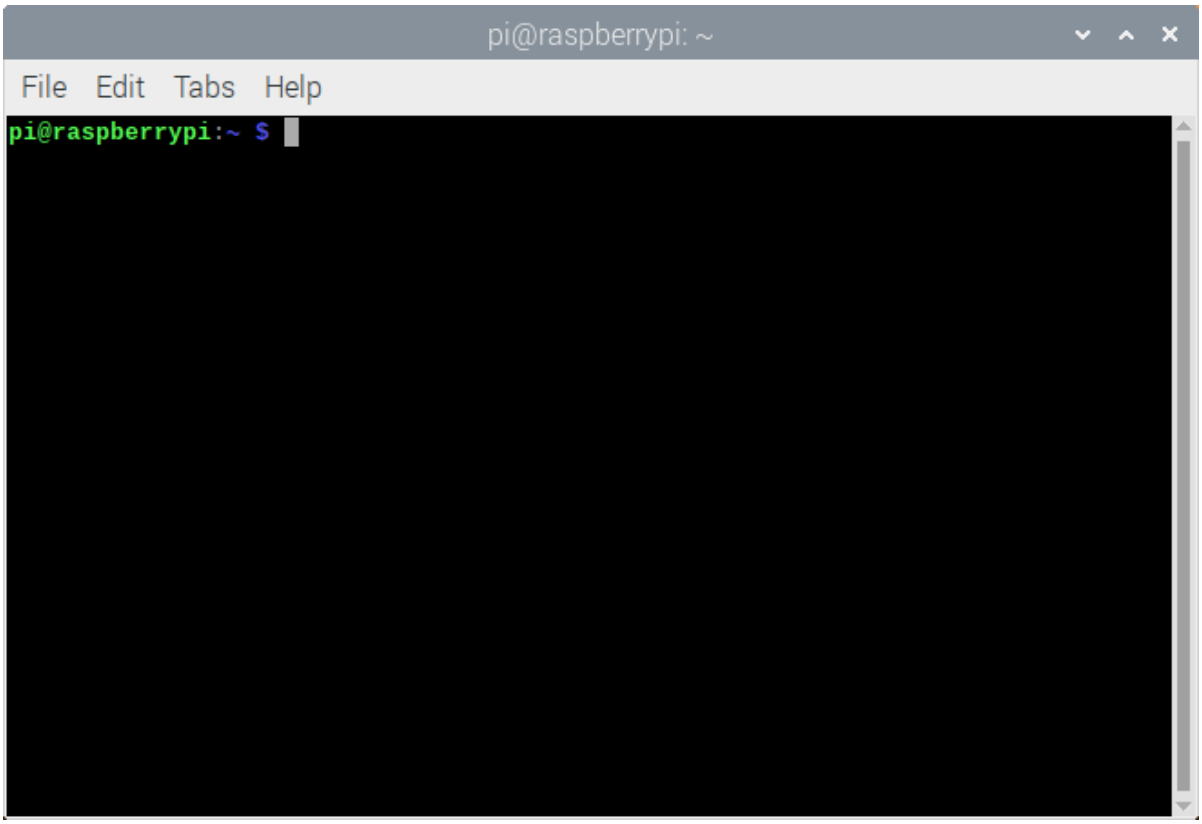
Why "Chapter 0"? Because in program code the first number is 0. We choose to follow this rule. In this chapter, we will do some necessary foundational preparation work: Start your Raspberry Pi and install some necessary libraries.

Linux Command

Raspberry Pi OS is based on the Linux Operation System. Now we will introduce you to some frequently used Linux commands and rules.
First, open the Terminal. All commands are executed in Terminal.

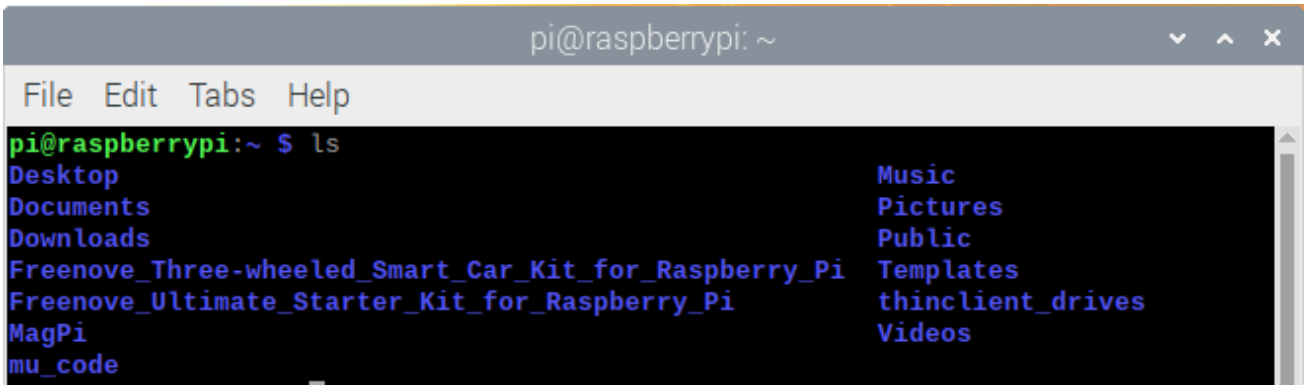


When you click the Terminal icon, following interface appears.



Note: The Linux is case sensitive.

First, type "ls" into the Terminal and press the "Enter" key. The result is shown below:



The "ls" command lists information about the files (the current directory by default).

Content between "$" and "pi@raspberrypi:" is the current working path. "~" represents the user directory, which refers to "/home/pi" here.

```
pi@raspberrypi:~ $ pwd
/home/pi
```

"cd" is used to change directory. "/" represents the root directory.

```
pi@raspberrypi:~ $ cd /usr
pi@raspberrypi:/usr $ ls
bin  games  include  lib  local  man  sbin  share  src
pi@raspberrypi:/usr $ cd ~
pi@raspberrypi:~ $
```

Later in this Tutorial, we will often change the working path. Typing commands under the wrong directory may cause errors and break the execution of further commands.

Many frequently used commands and instructions can be found in the following reference table.

Command	instruction
ls	Lists information about the FILEs (the current directory by default) and entries alphabetically.
cd	Changes directory
sudo + cmd	Executes cmd under root authority
./	Under current directory
gcc	GNU Compiler Collection
git clone URL	Use git tool to clone the contents of specified repository, and URL in the repository address.

There are many commands, which will come later. For more details about commands. You can refer to: http://www.linux-commands-examples.com

Shortcut Key

Now, we will introduce several commonly used shortcuts that are very useful in Terminal.

1. **Up and Down Arrow Keys**: Pressing "↑" (the Up key) will go backwards through the command history and pressing "↓" (the Down Key) will go forwards through the command history.

2. **Tab Key**: The Tab key can automatically complete the command/path you want to type. When there is only one eligible option, the command/path will be completely typed as soon as you press the Tab key even you only type one character of the command/path.

As shown below, under the '~' directory, you enter the Documents directory with the "cd" command. After typing "cd D", pressing the Tab key (there is no response), pressing the Tab key again then all the files/folders that begin with "D" will be listed. Continue to type the letters "oc" and then pressing the Tab key, the "Documents" is typed automatically.

```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Doc
```

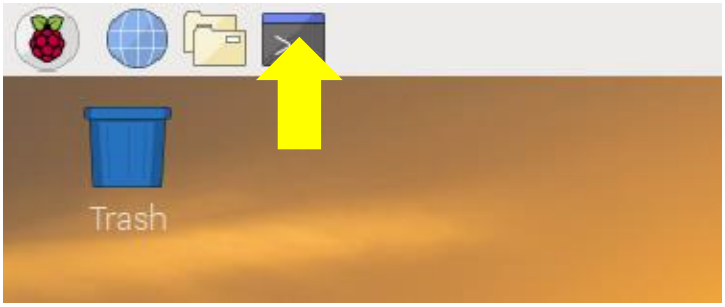
```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Documents/
```

Get the Project Code

If you have not get the code. In the pi directory of the RPi terminal, enter the following commands to get.

```
cd
git clone --depth 1 https://github.com/freenove/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi
```

(There is no need for a password. If you get some errors, please check your commands.)



After the download is completed, a new folder "Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi" is generated, which contains all of the tutorials and required code.

This folder name seems a little too long. We can simply rename it by using the following command.

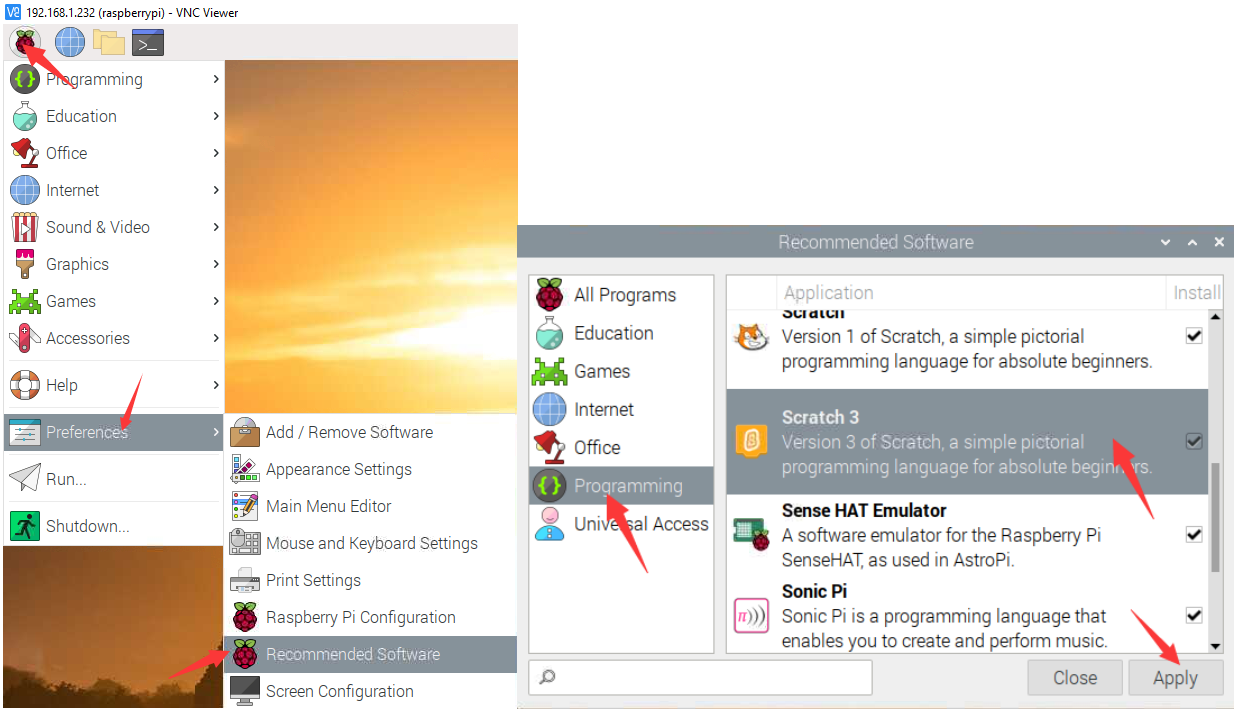
```
mv Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/ Freenove_Kit/
```

"Freenove_Kit" is now the new and much shorter folder name.
If you have no experience with scratch, we suggest that you refer to this website for basic information and knowledge:
https://scratch.mit.edu/projects/editor/?tutorial=getStarted

Install and Configure Scratch 3

Installation Steps

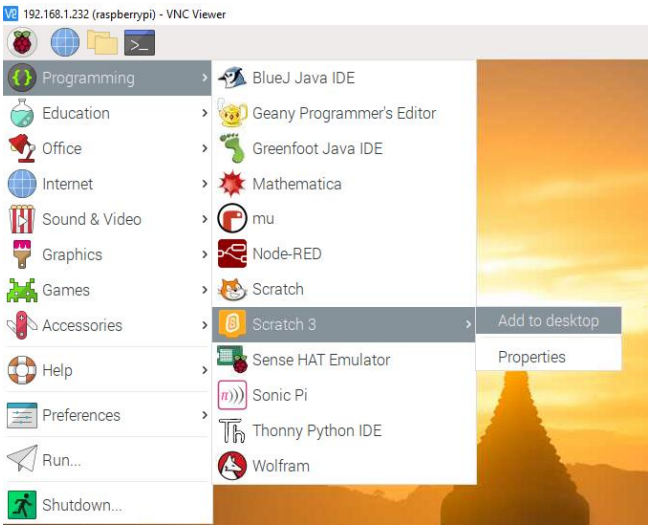
Open the **Menu**, click on **Preferences** > **Recommended Software** > **Programming** >, and then select **Scratch 3** and click on **Apply**.



We can **also** install scratch3 in the terminal.

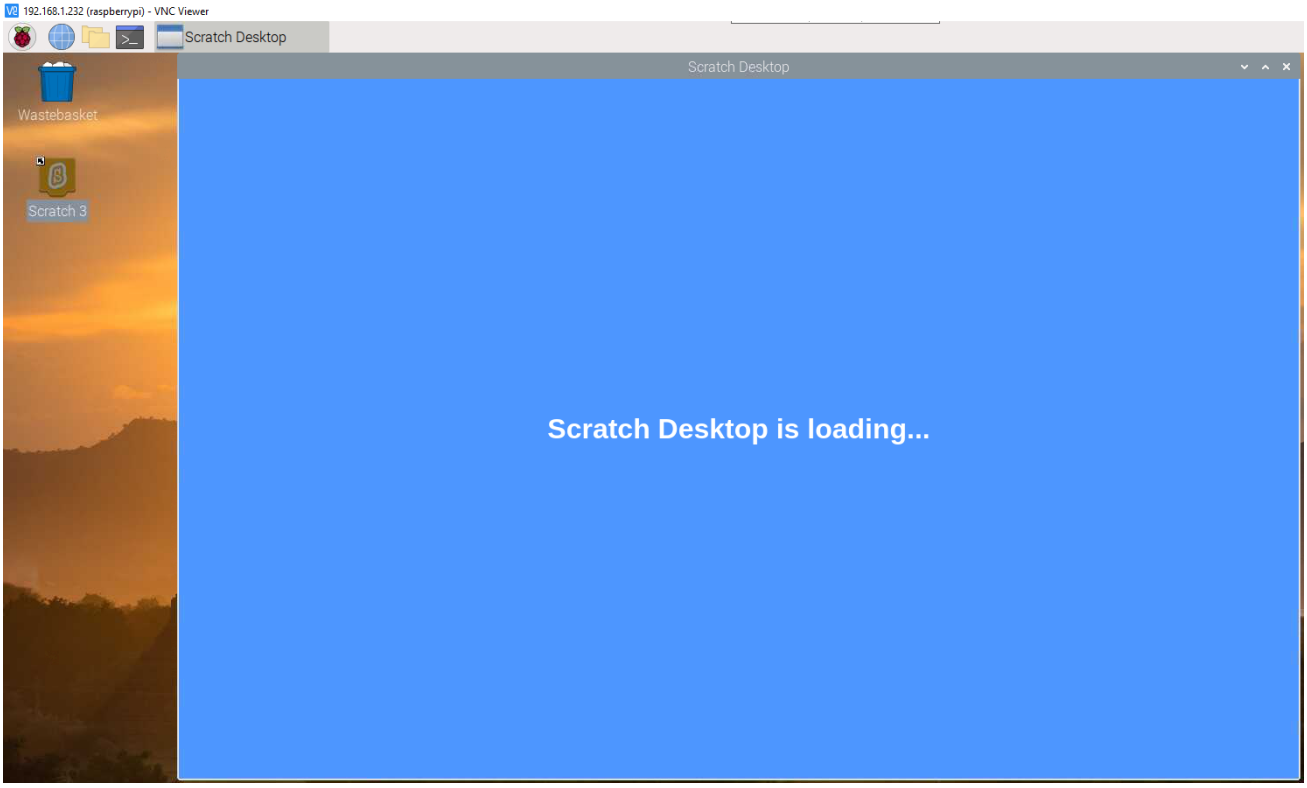
```
sudo apt-get update
sudo apt-get install scratch3
```

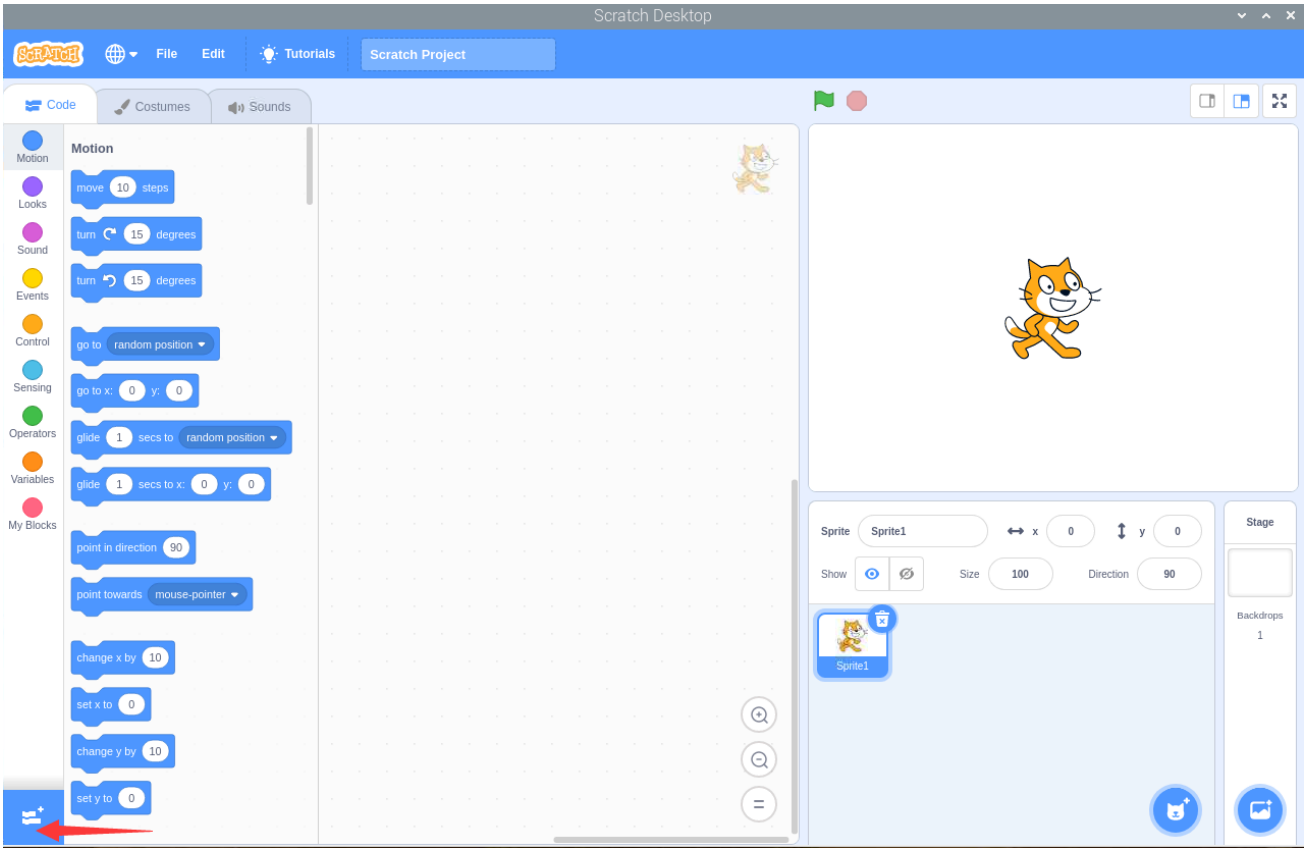
Add scratch3 to desktop.

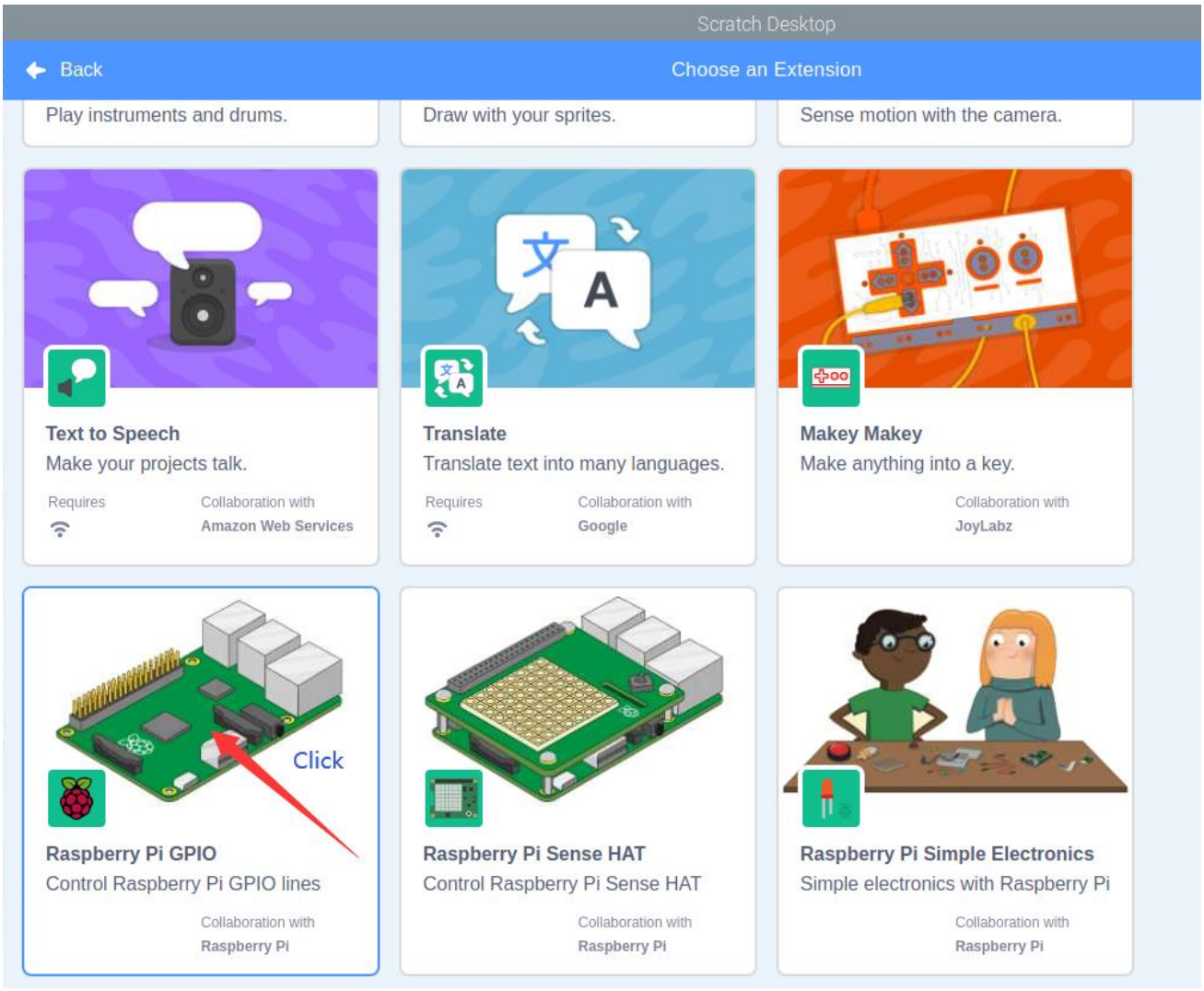


Add Raspberry Pi Library

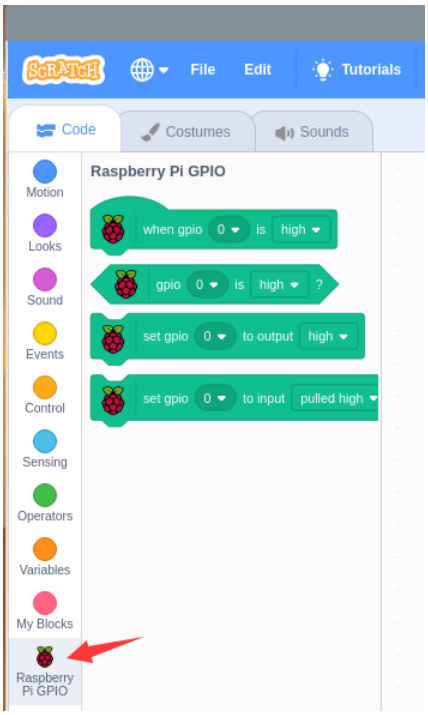
Double click scratch3, which takes some time to open.







We will learn how to use it later:



Chapter 1 LED

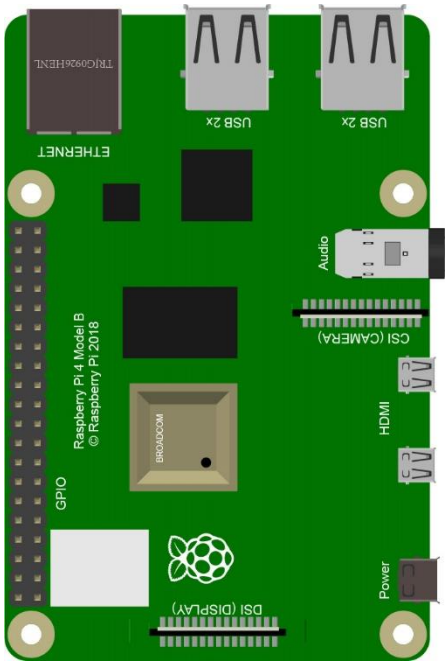
This chapter is the Start Point in the journey to build and explore RPi electronic projects. We will start with simple “Blink” project.

Project 01.0_Blink, 01.1_Keyboard_LED and 01.2 Sprite_LED

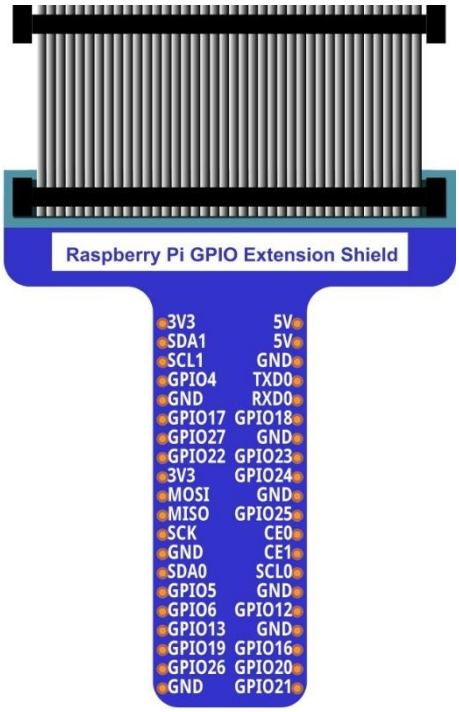
In this project, we will use RPi to control blinking a common LED.

Component List

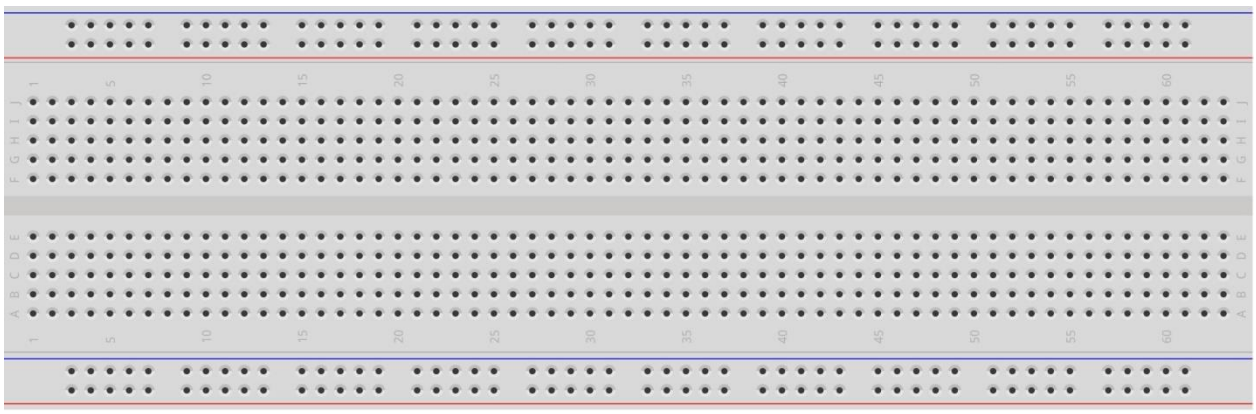
Raspberry Pi
(Recommended: Raspberry Pi 4B / 3B+ / 3B
Compatible: 3A+ / 2B / 1B+ / 1A+ / Zero W / Zero)



GPIO Extension Board & Ribbon Cable



Breadboard x1



LED x1	Resistor 220Ω x1	Jumper **Specific quantity depends on the circuit.**
		

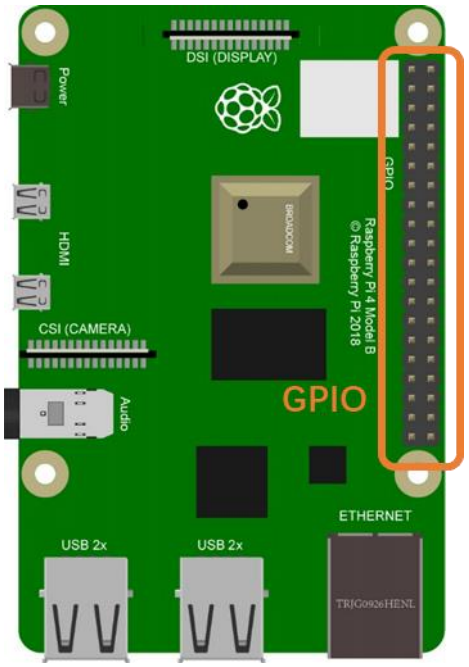
In the components list, GPIO Extension Board, Raspberry and Breadboard are necessary for each project. Later, they will be reference by text only (no images as in above).

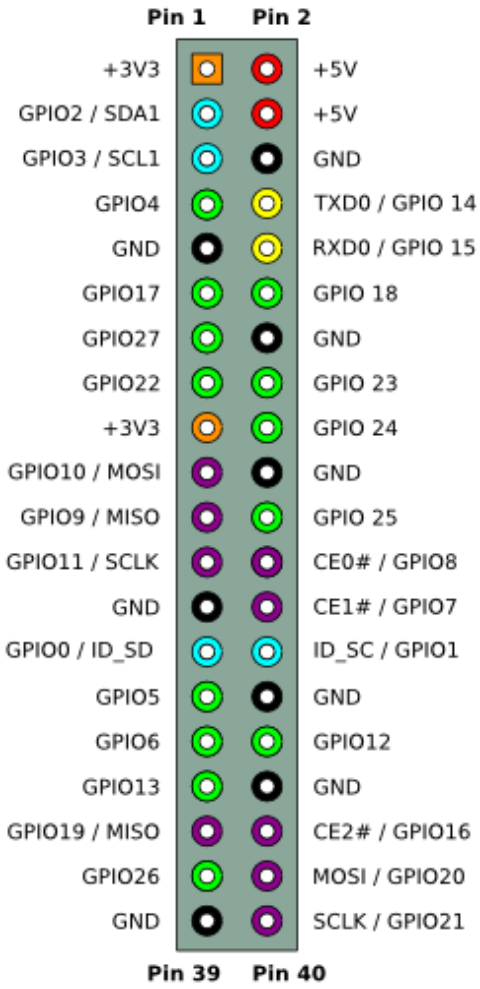
GPIO

GPIO: General Purpose Input/Output. Here we will introduce the specific function of the pins on the Raspberry Pi and how you can utilize them in all sorts of ways in your projects. Most RPi Module pins can be used as either an input or output, depending on your program and its functions.
When programming GPIO pins there are 3 different ways to reference them: GPIO Numbering, Physical Numbering and WiringPi GPIO Numbering.

BCM GPIO Numbering

The Raspberry Pi CPU uses Broadcom (BCM) processing chips BCM2835, BCM2836 or BCM2837. GPIO pin numbers are assigned by the processing chip manufacturer and are how the computer recognizes each pin. The pin numbers themselves do not make sense or have meaning as they are only a form of identification. Since their numeric values and physical locations have no specific order, there is no way to remember them, so you will need to have a printed reference or a reference board that fits over the pins.
Each pin's functional assignment is defined in the image below:

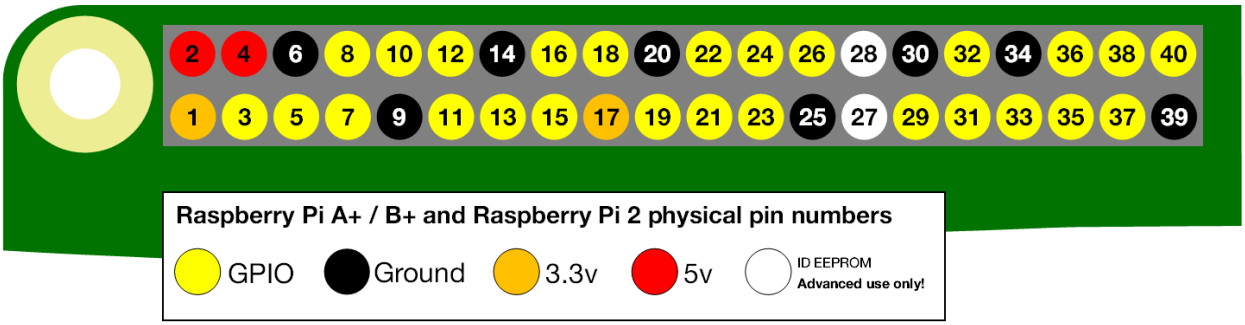




For more details about pin definition of GPIO, please refer to http://pinout.xyz/

PHYSICAL Numbering

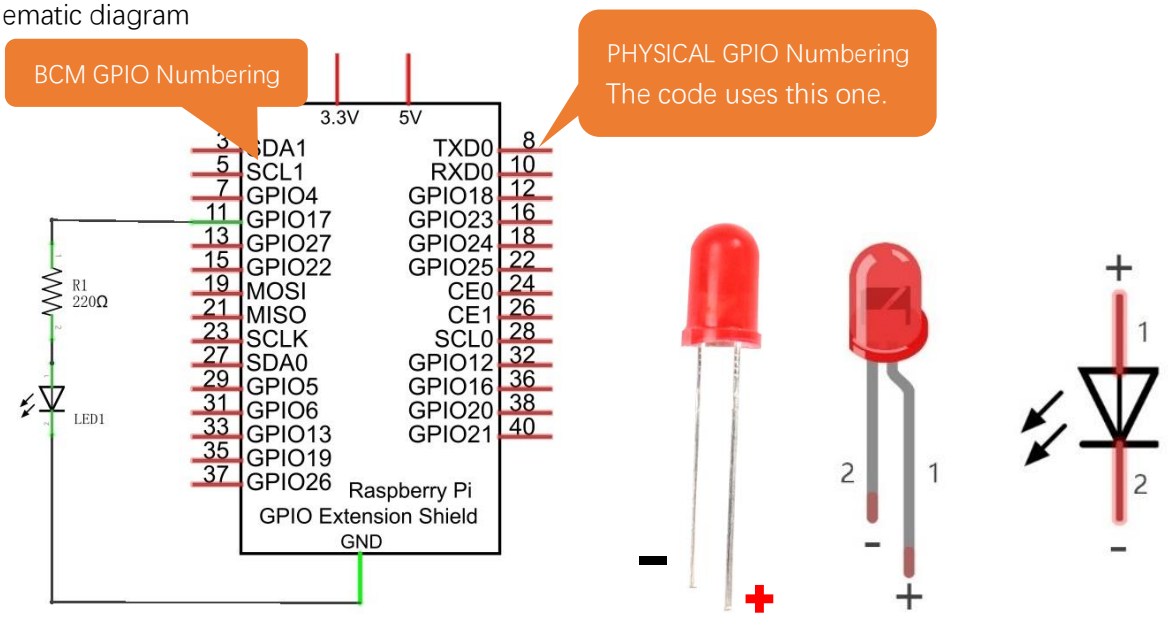
Another way to refer to the pins is by simply counting across and down from pin 1 at the top left (nearest to the SD card). This is 'Physical Numbering', as shown below:



Circuit

CAUTION: Avoid any possible short circuits (especially connecting 5V or GND, 3.3V and GND)!
WARNING: A short circuit can cause high current in your circuit, create excessive component heat and cause permanent damage to your RPi!

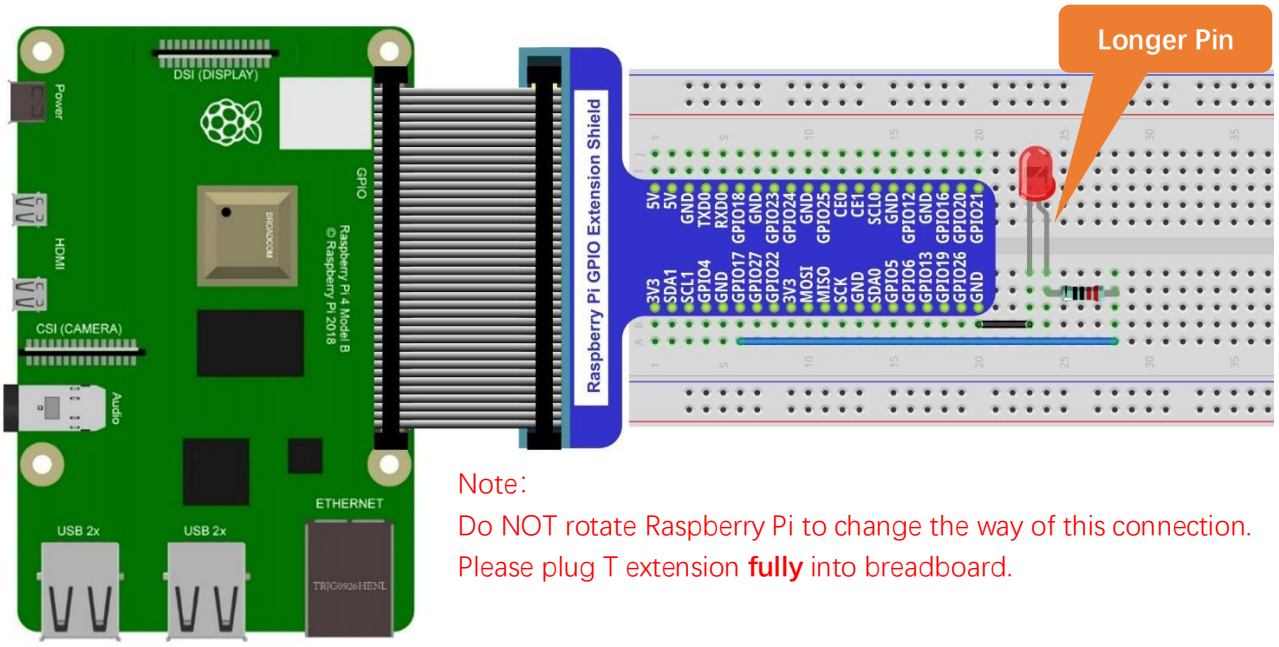
Schematic diagram



Scratch use **BCM numbering**.

BCM(Extension)	Physical		BCM(Extension)
3.3V	1	2	5V
SDA1	3	4	**5V**
SCL1	5	6	**GND**
GPIO4	7	8	GPIO14/TXD0
GND	9	10	GPIO15/RXD0
GPIO17	11	12	GPIO18
GPIO27	13	14	**GND**
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10/MOSI)	19	20	**GND**
GPIO9/MOIS	21	22	GPIO25
GPIO11/SCLK	23	24	GPIO8/CE0
GND	25	26	GPIO7CE1
GPIO0/SDA0	27	28	GPIO1/SCL0
GPIO5	29	30	**GND**
GPIO6	31	32	GPIO12
GPIO13	33	34	**GND**
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

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

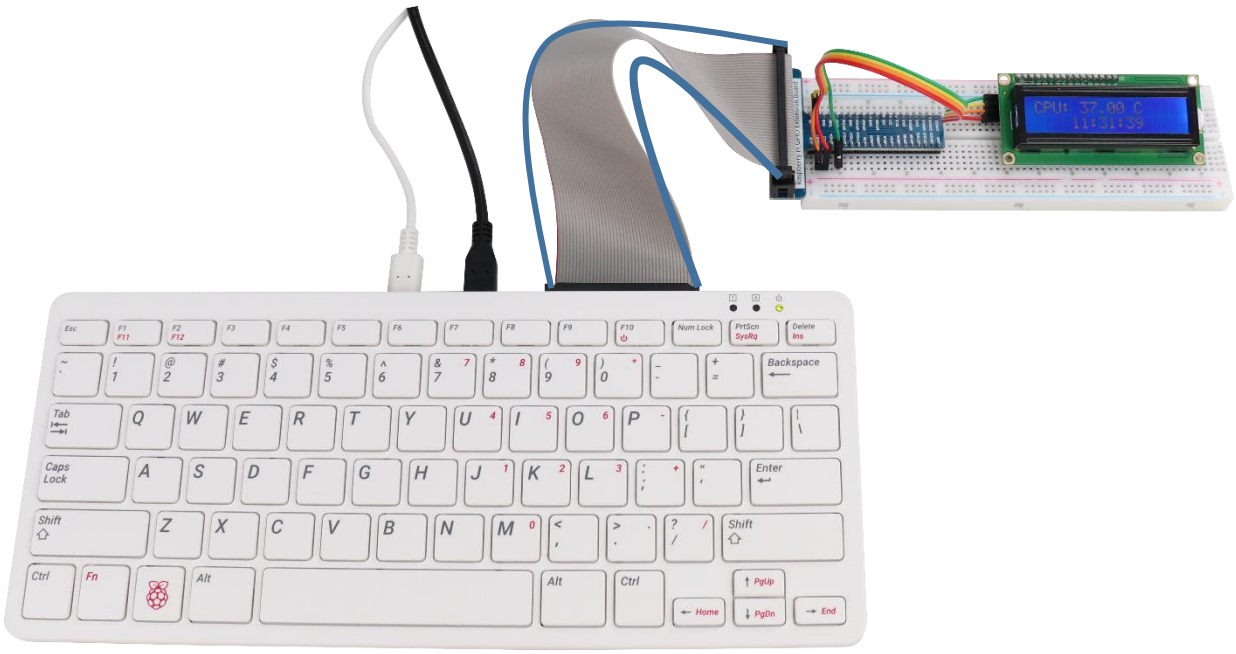


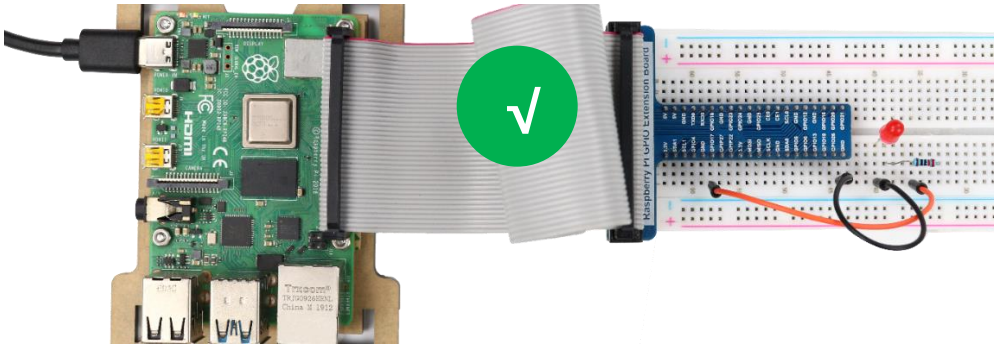
Note:

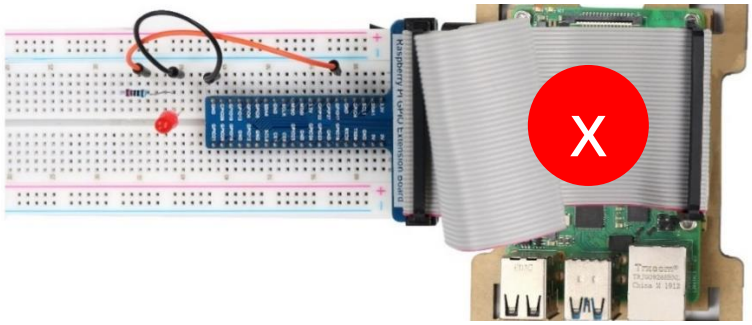
Do NOT rotate Raspberry Pi to change the way of this connection.

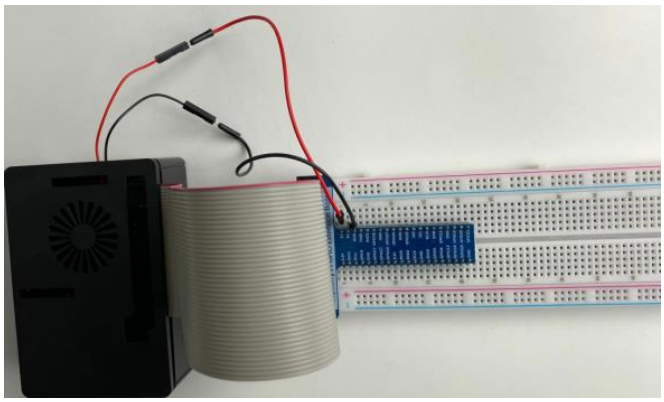
Please plug T extension **fully** into breadboard.

The connection of Raspberry Pi 400 and T extension board is as below. Don't reverse the ribbon.









If you have a fan, you can connect it to 5V GND of breadboard via jumper wires.

How to distinguish resistors?

There are only three kind of resistors in this kit.

The one with 1 red ring is 10KΩ 

The one with 2 red rings is 220Ω 

The one with 0 red ring is 1KΩ 

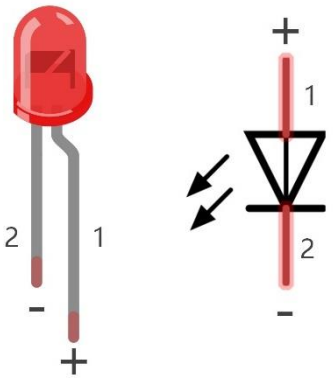
Future hardware connection diagrams will only show that part of breadboard and GPIO Extension Shield.

Component knowledge

LED

An LED is a type of diode. All diodes have two Poles and only work if current is flowing in the correct direction. An LED will only work (light up) if the longer pin (+) of LED is connected to the positive output from a power source and the shorter pin is connected to the negative (-) output, which is also referred to as Ground (GND). This type of component is known as “Polar” (think One-Way Street).

All common 2 lead diodes are the same in this respect. Diodes work only if the voltage of its positive electrode is higher than its negative electrode and there is a narrow range of operating voltage for most all common diodes of 1.9 and 3.4V. If you use much more than 3.3V the LED will be damaged and burnt out.



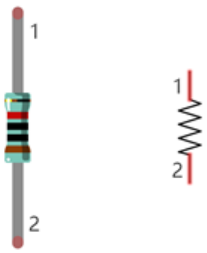
LED	Voltage	Maximum current	Recommended current
Red	1.9-2.2V	20mA	10mA
Green	2.9-3.4V	10mA	5mA
Blue	2.9-3.4V	10mA	5mA
Volt ampere characteristics conform to diode			

Note: LEDs cannot be directly connected to a power supply, which usually ends in a damaged component. A resistor with a specified resistance value must be connected in series to the LED you plan to use.

Resistor

Resistors use Ohms (Ω) as the unit of measurement of their resistance (R). 1MΩ=1000kΩ, 1kΩ=1000Ω.

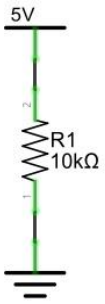
A resistor is a passive electrical component that limits or regulates the flow of current in an electronic circuit. On the left, we see a physical representation of a resistor, and the right is the symbol used to represent the presence of a resistor in a circuit diagram or schematic.



The bands of color on a resistor is a shorthand code used to identify its resistance value. For more details of resistor color codes, please refer to the card in the kit package.

With a fixed voltage, there will be less current output with greater resistance added to the circuit. The relationship between Current, Voltage and Resistance can be expressed by this formula: I=V/R known as Ohm’s Law where I = Current, V = Voltage and R = Resistance. Knowing the values of any two of these allows you to solve the value of the third.

In the following diagram, the current through R1 is: I=U/R=5V/10kΩ=0.0005A=0.5mA.

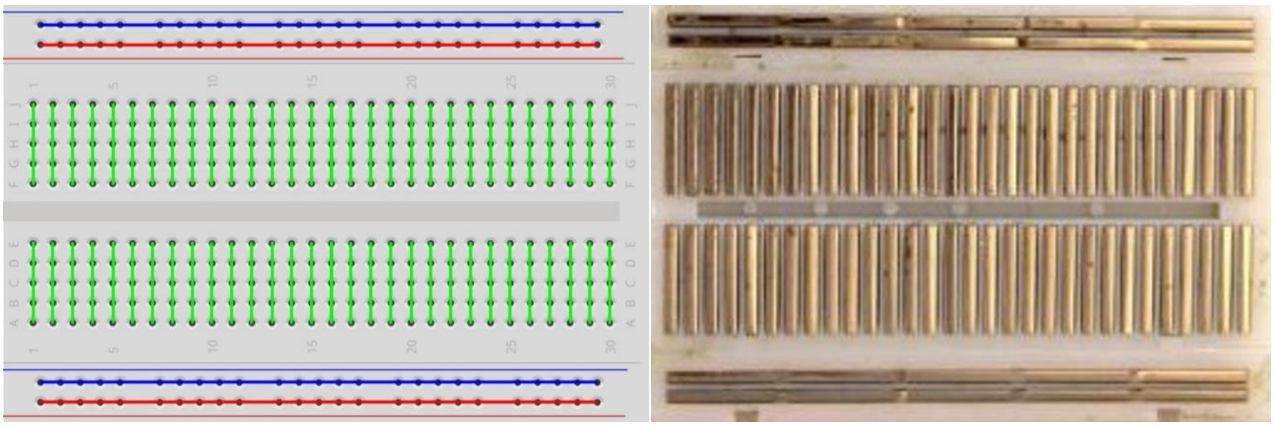


WARNING: Never connect the two poles of a power supply with anything of low resistance value (i.e. a metal object or bare wire) this is a Short and results in high current that may damage the power supply and electronic components.

Note: Unlike LEDs and Diodes, Resistors have no poles and re non-polar (it does not matter which direction you insert them into a circuit, it will work the same)

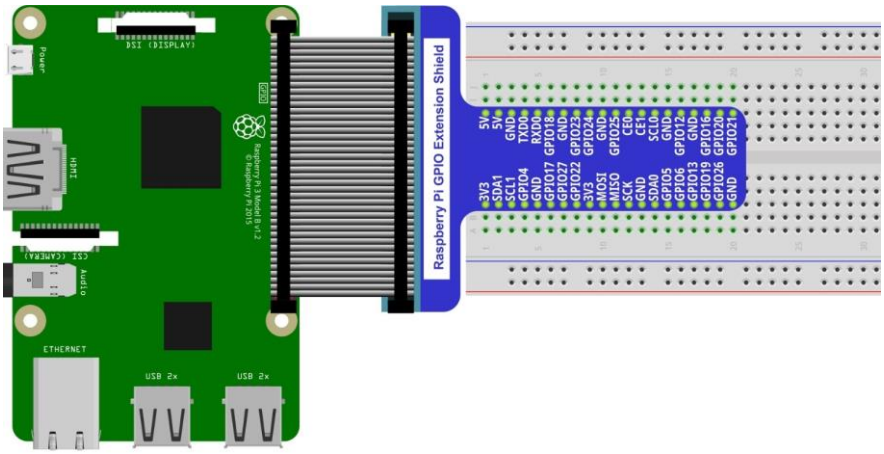
Breadboard

Here we have a small breadboard as an example of how the rows of holes (sockets) are electrically attached. The left picture shows the ways the pins have shared electrical connection and the right picture shows the actual internal metal, which connect these rows electrically.



GPIO Extension Board

GPIO board is a convenient way to connect the RPi I/O ports to the breadboard directly. The GPIO pin sequence on Extension Board is identical to the GPIO pin sequence of RPi.



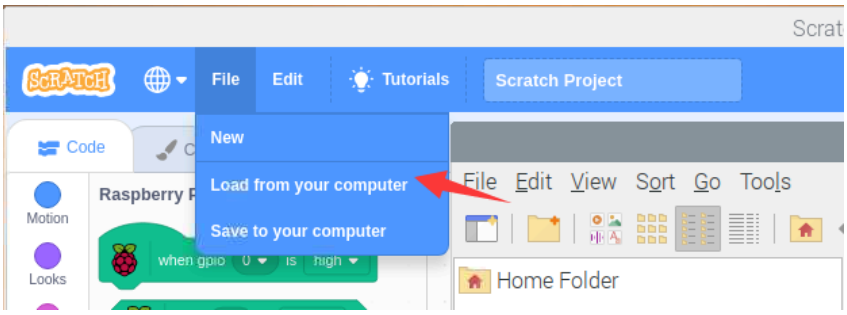
Code

According to the circuit, when the GPIO17 of RPi output level is high, the LED turns ON. Conversely, when the GPIO17 RPi output level is low, the LED turns OFF. Therefore, we can let GPIO17 repeatedly output high and low level to make the LED blink.

01.0_Blink

You can refet to chapter 0 about how to open scratch.

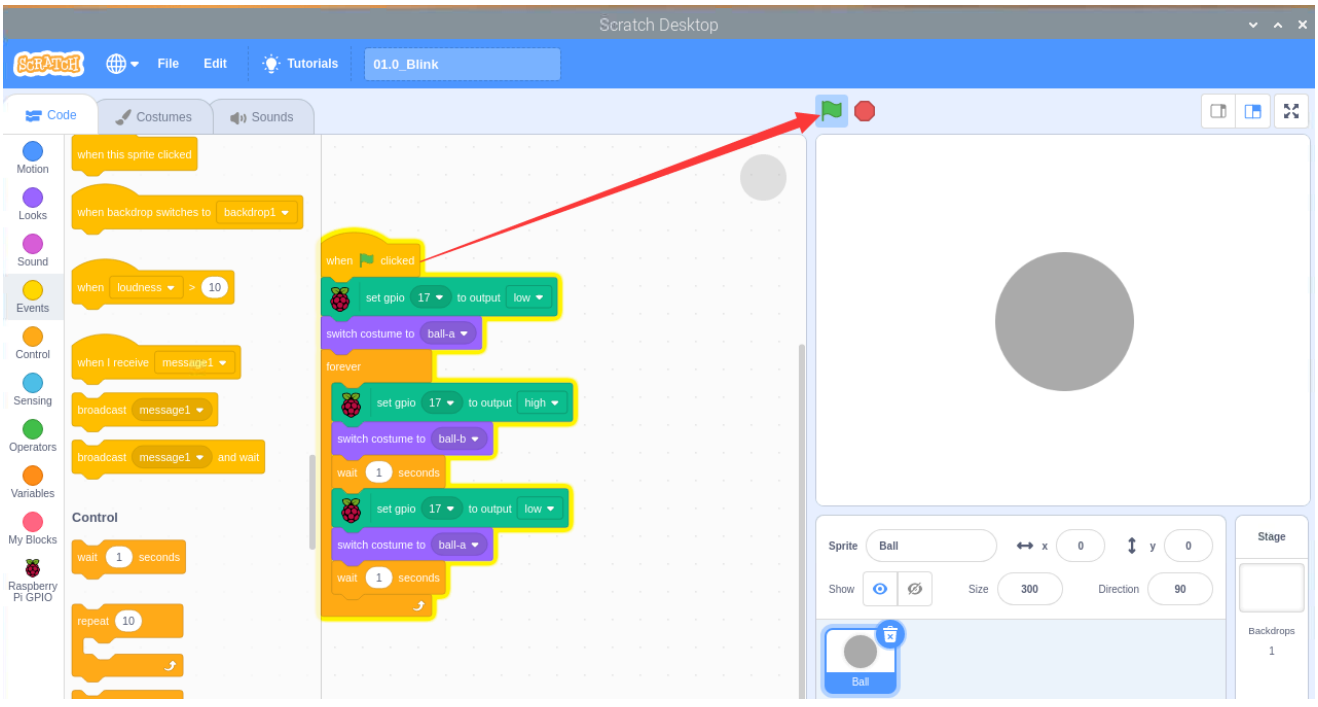
If you have any concerns, please contact us via: support@freenove.com



Load the code.

```
Freenove_Kit/Scratch3/01.0_Blink
```

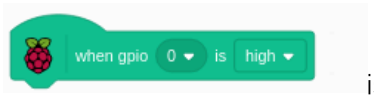
Click the green flag, then the led will keep blinking until you click stop.



Generallya, we make a program starting from 

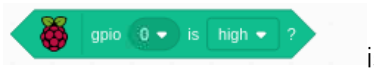
, which you can click on the right.

There are four modules in scratch for Raspberry Pi.

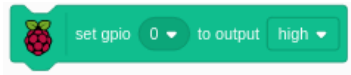


is a kind of event like 

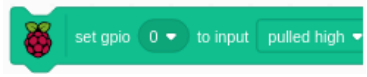
.



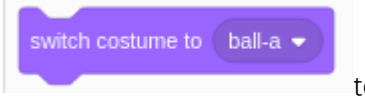
is a condition for judgment, which usually works with if.



is to set a GPIO to output hight level or low level.



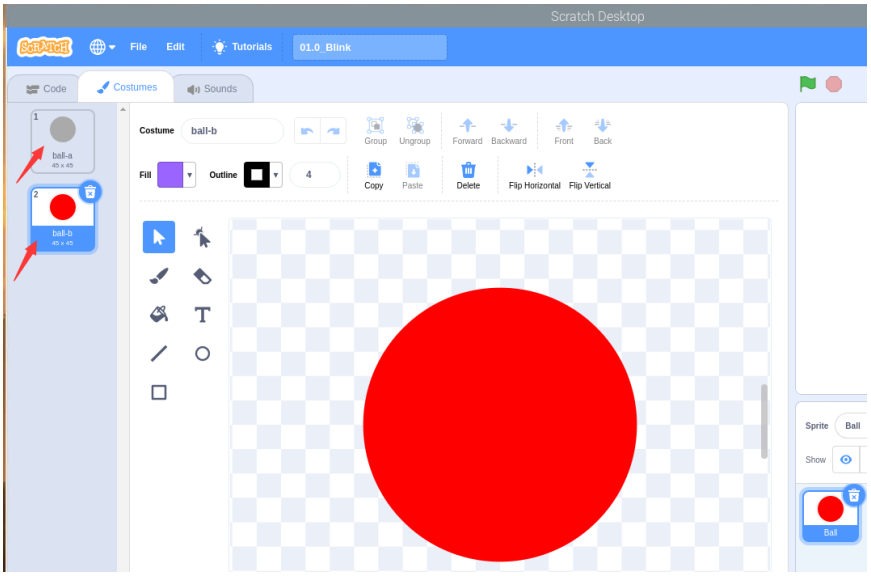
is use to set a GPIO to input mode.

In the code, we use 

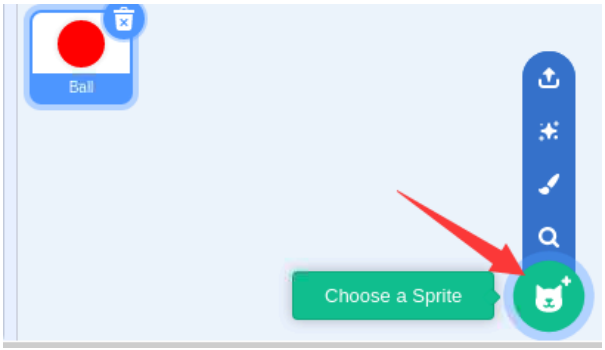
to change the color of ball.

is similar to delay or sleep.

Click constumes, we will see there are two ball costumes. We can also define more costumes. We will define three costumes in RGB section.



Click this we can choose to use other sprites or upload a sprite.



01.1_Keyboard_LED

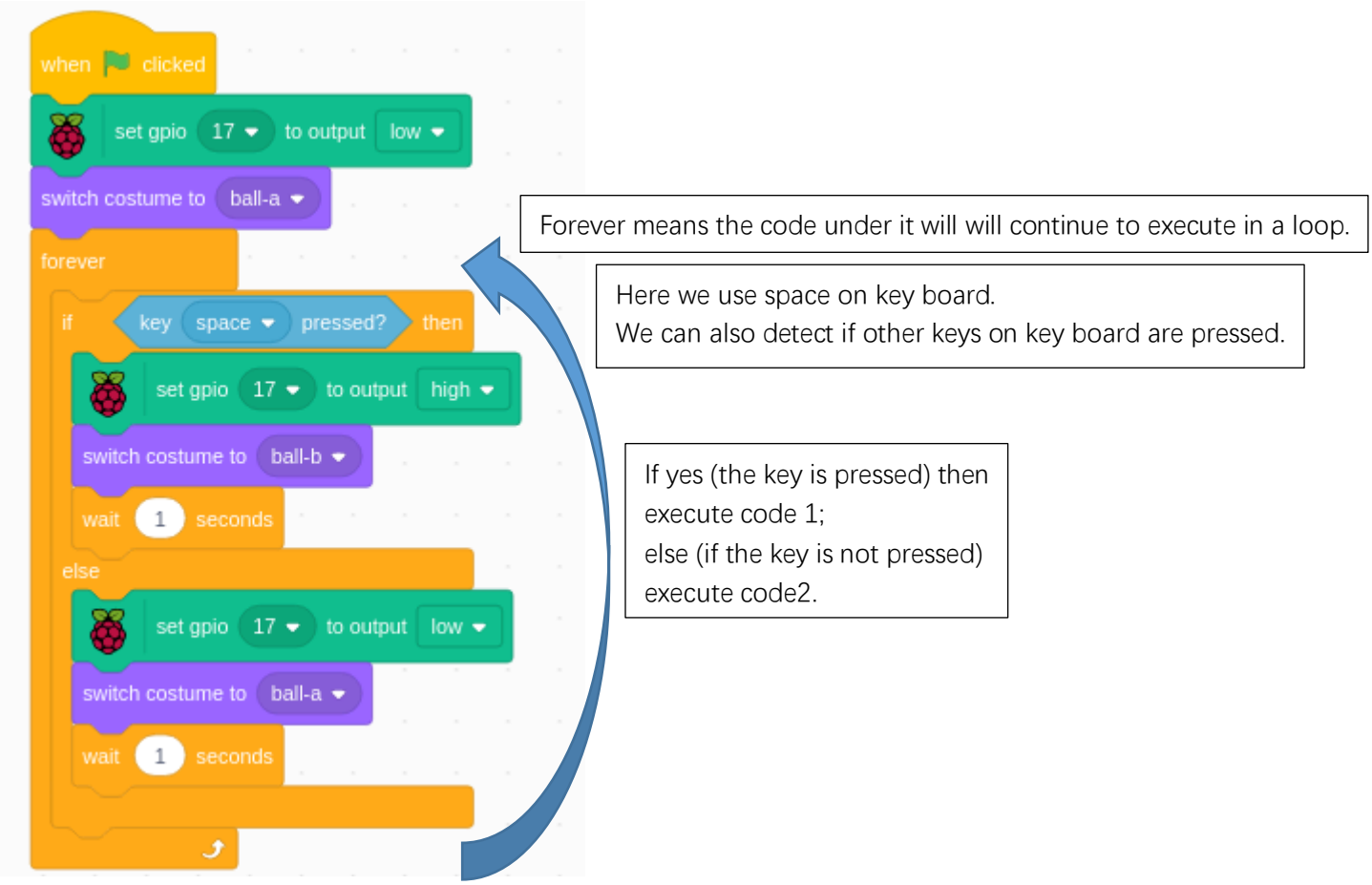
The circuit is the same as previous section.

If you have any concerns, please contact us via: support@freenove.com

Load the code to scratch3.

```
cd ~/Freenove_Kit/Code/Scratch3/01.0_Keyboard_LED.sb3
```

Click the green flag. Then when you press space key on your keyboard, the LED will be turned on. Otherwise, the LED will be turned off.



01.2_Sprite_LED

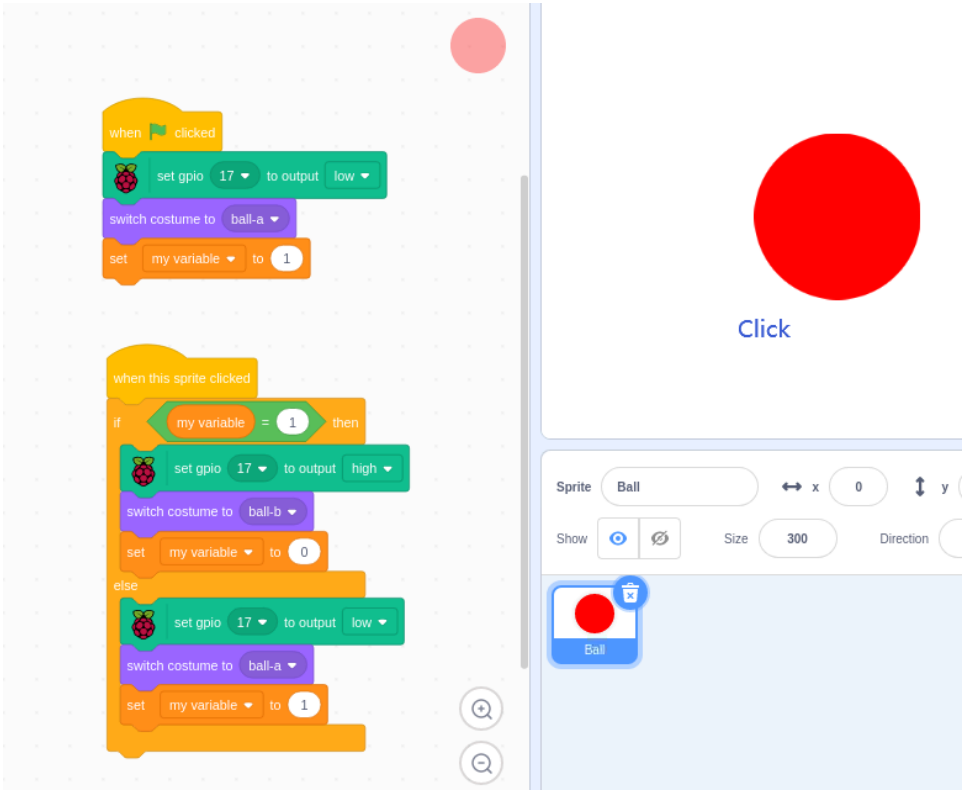
The circuit is the same as previous section.

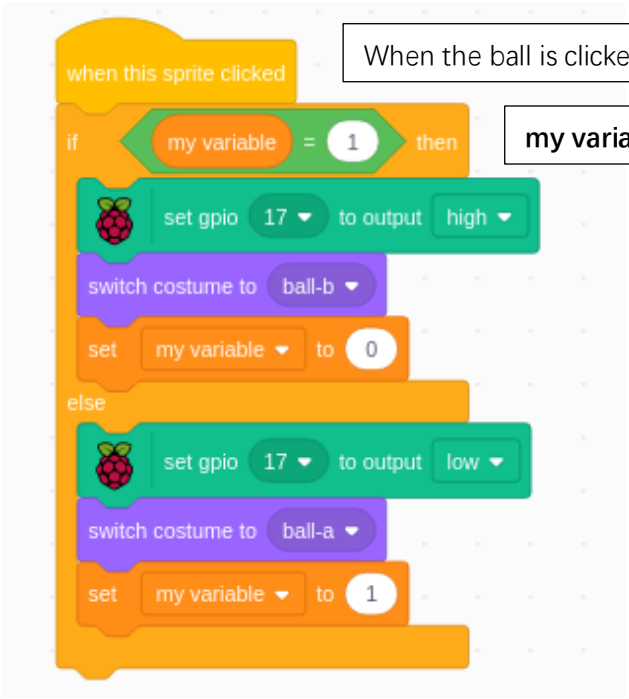
If you have any concerns, please contact us via: support@freenove.com

Load the code to scratch3.

```
cd ~/Freenove_Kit/Code/Scratch3/01.2_Sprite_LED.sb3
```

Click the green flag. **Then click the ball**. The LED will be turned on or turned off.





When the ball is clicked.

my variable will be changed to 1, 0, 1, 0, 1, 0……

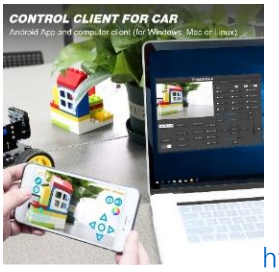
So every time we click the ball the LED status will be changed.

Freenove Car, Robot and other products for Raspberry Pi

We also have car and robot kits for Raspberry Pi. You can visit our website for details.

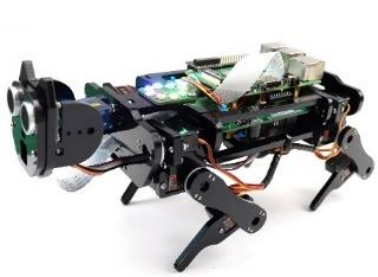
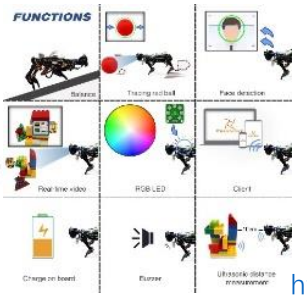
https://www.amazon.com/freenove

FNK0043 **Freenove 4WD Smart Car Kit for Raspberry Pi**

https://www.youtube.com/watch?v=4Zv0GZUQjZc

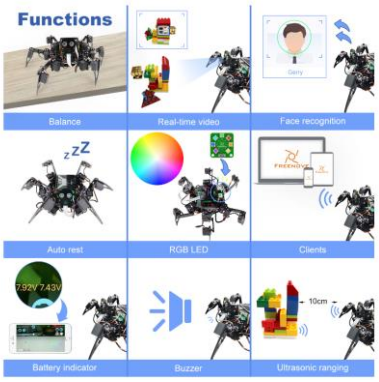
FNK0050 **Freenove Robot Dog Kit for Raspberry Pi**

https://www.youtube.com/watch?v=7BmlZ8_R9d4

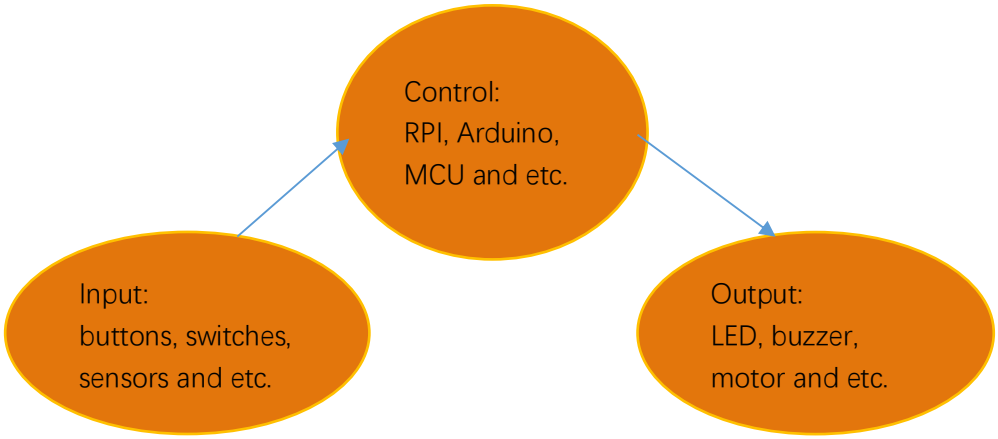
FNK0052 **Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi**

https://youtu.be/LvghnJ2DNZ0

Chapter 2 Buttons & LEDs

Usually, there are three essential parts in a complete automatic control device: INPUT, OUTPUT, and CONTROL. In last section, the LED module was the output part and RPI was the control part. In practical applications, we not only make LEDs flash, but also make a device sense the surrounding environment, receive instructions and then take the appropriate action such as turn on LEDs, make a buzzer beep and so on.



Next, we will build a simple control system to control an LED through a push button switch.

Project 02.0_Electronic_Button_LED and 02.1_Keyboard_Button_LED

Component List

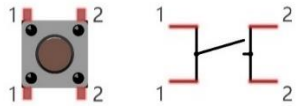
Raspberry Pi (with 40 GPIO) x1 GPIO Extension Board & Wire x1 Breadboard x1	LED x1 	Resistor 220Ω x1 	Resistor 10kΩ x2 	Push Button Switch x1 
Jumper Wire 				

Please Note: In the code “button” represents switch action.

Component knowledge

Push Button Switch

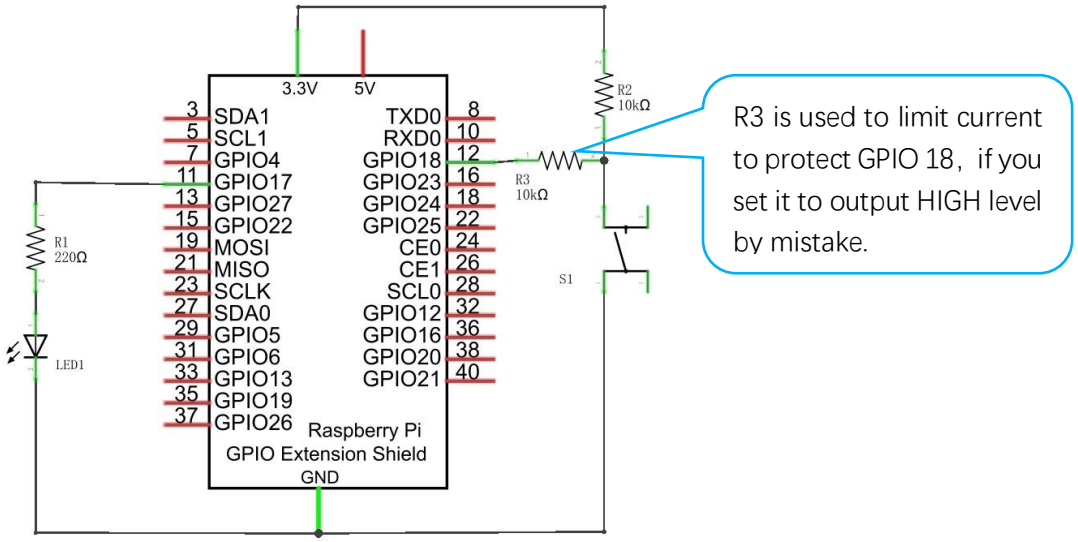
This type of Push Button Switch has 4 pins (2 Pole Switch). Two pins on the left are connected, and both left and right sides are the same per the illustration:



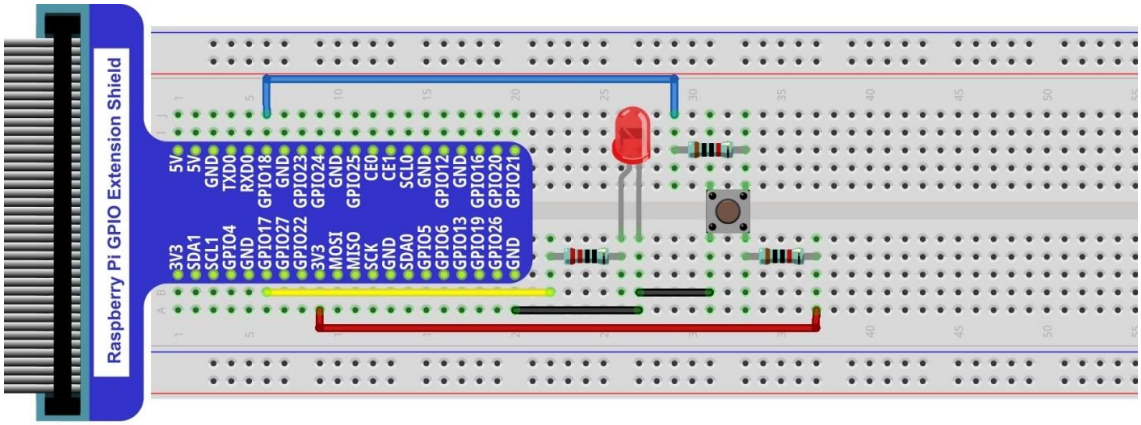
When the button on the switch is pressed, the circuit is completed (your project is Powered ON).

Circuit

Schematic diagram



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

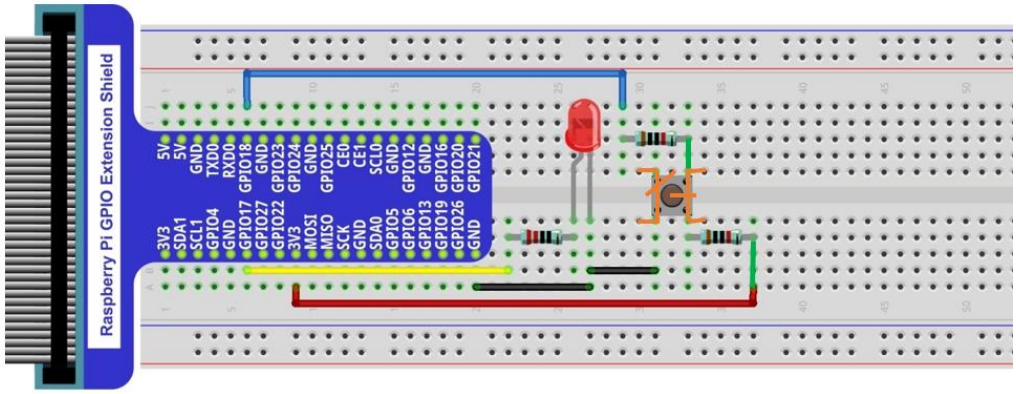


There are two kinds of push button switch in this kit.

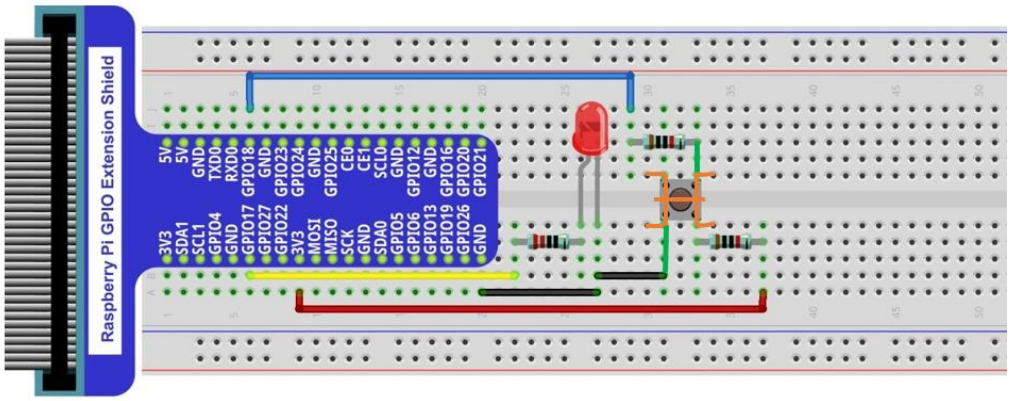
The smaller push button switches are contained in a plastic bag.

This is how it works.

When button switch is released:



When button switch is pressed:



Code

This project is designed for learning how to use Push Button Switch to control an LED. We first need to read the state of switch, and then determine whether to turn the LED ON in accordance to the state of the switch.

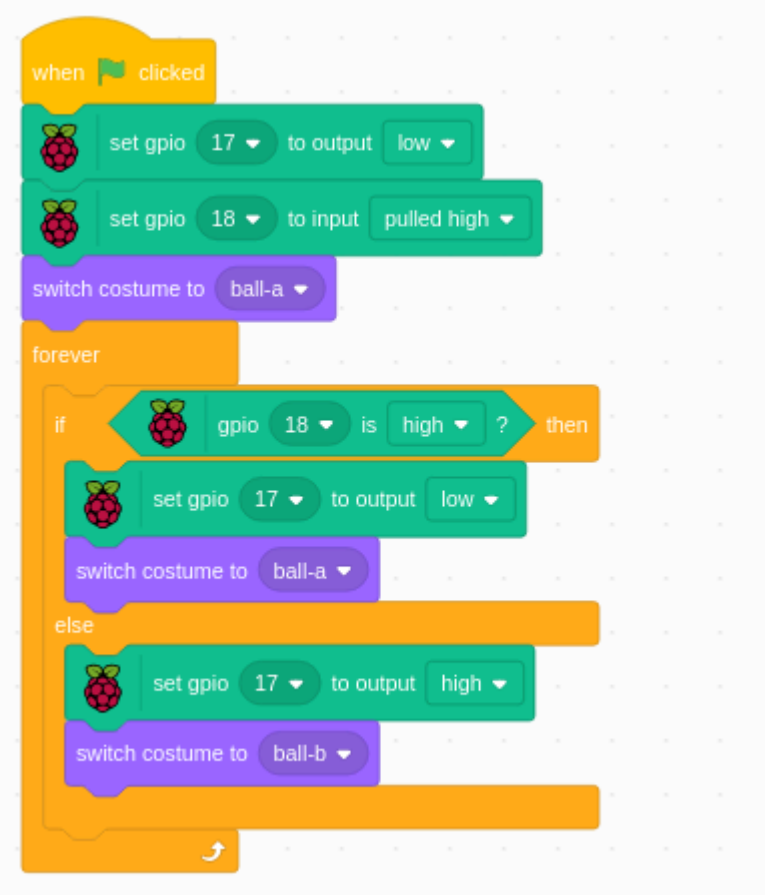
02.0_Electronic_Button_LED

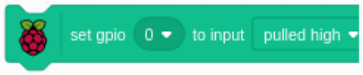
If you have any concerns, please contact us via: support@freenove.com

Load the code to scratch3.

```
Freenove_Kit/Code/Scratch3/02.0_Electronic_Button_LED.sb3
```

Click the green flag. Then when you press **button swtich**, the LED will be turned on. Otherwise, the LED will be turned off.



Here 

is to set a GPIO to input mode, which is used to detect button swtich condition.

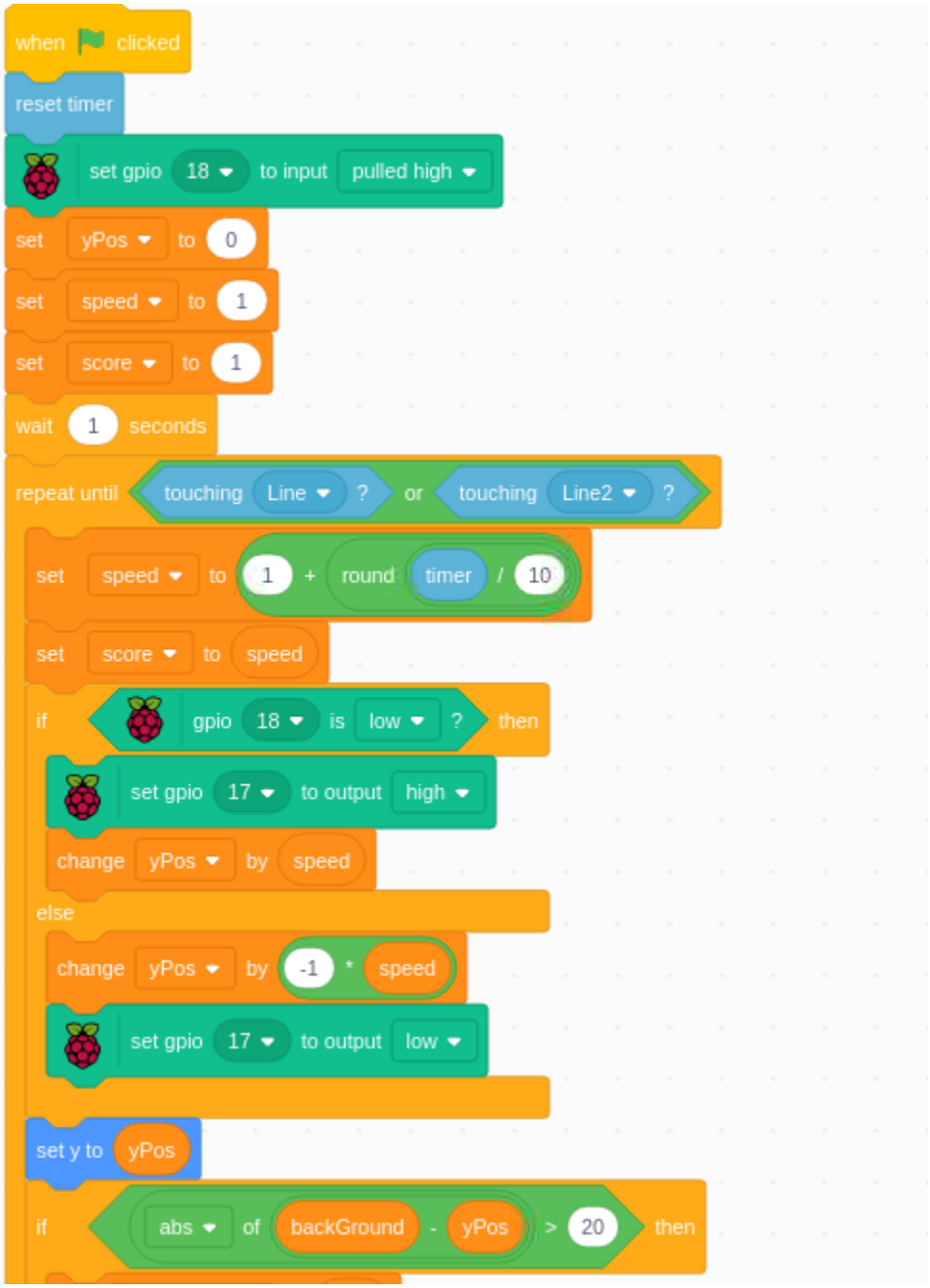
02.1_Floating_Ballon

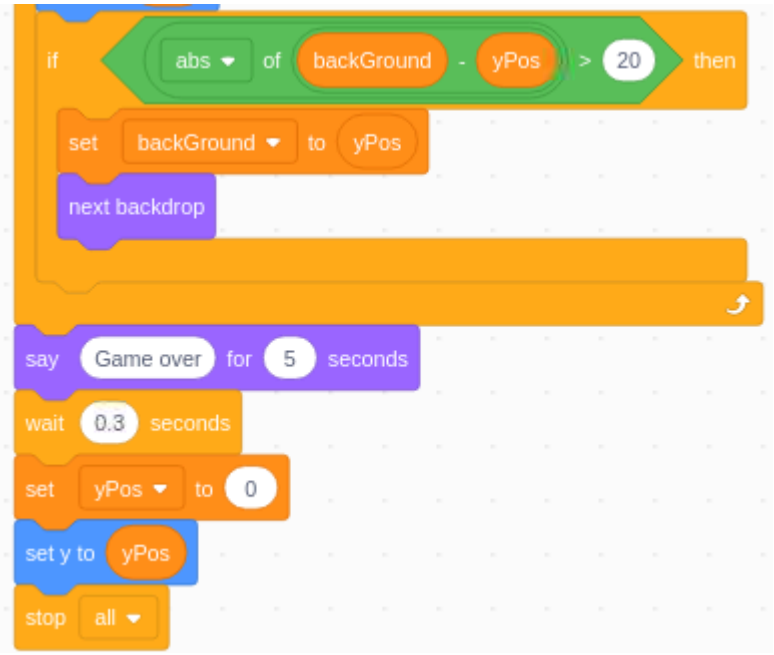
The circuit is the same as previous section.

Load the code to scratch3.

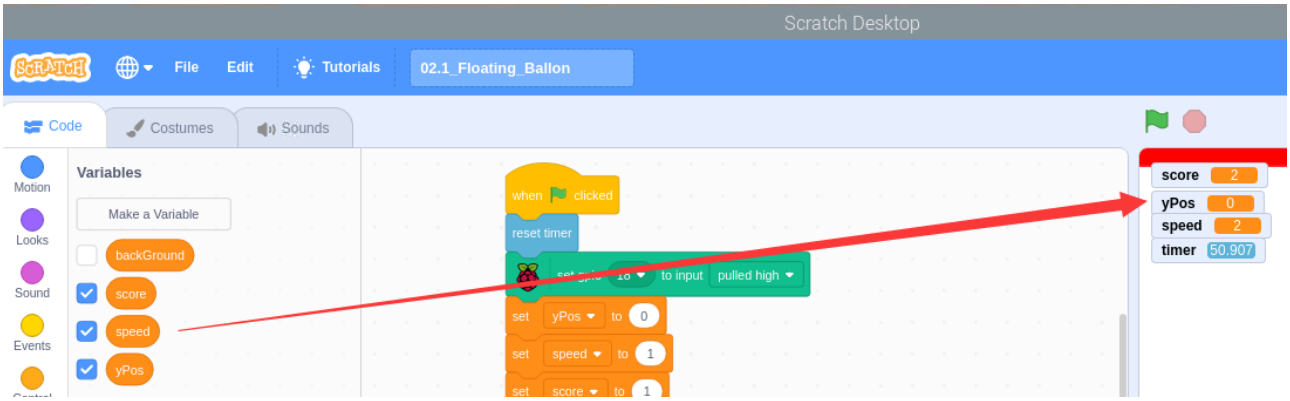
Freenove_Kit/Code/Scratch3/02.1_Floating_Ballon.sb3

Click the green flag. Then when you press button switch, the ballon will rise. Otherwise, the ballon will fall. When the ballon is too high or too low, touching the edges, game is over. The falling speed will increase over time.

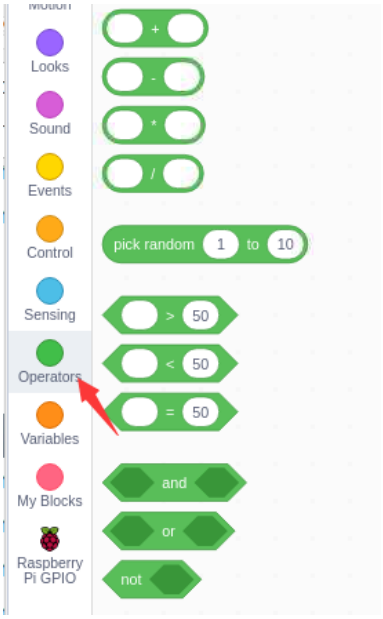




If we select variables in the left, they will be shown on the right.



Operators are used in this code.



Chapter 3 LED Bar Graph

We have learned how to control one LED to blink. Next, we will learn how to control a number of LEDs.

Project 03.0_LED_Bar_Graph

Component List

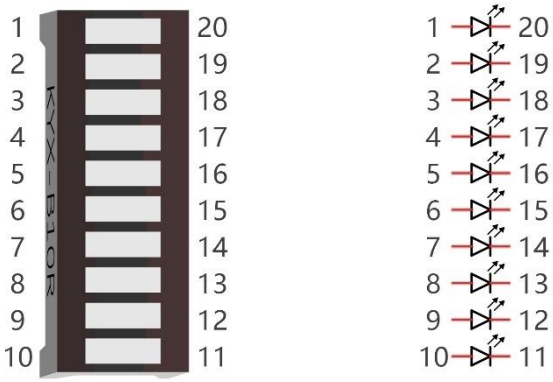
Raspberry Pi (with 40 GPIO) x1 GPIO Extension Board & Ribbon Cable x1 Breadboard x1	Bar Graph LED x1 	Resistor 220Ω x10 
Jumper Wire x 1 		

Component knowledge

Let us learn about the basic features of these components to use and understand them better.

Bar Graph LED

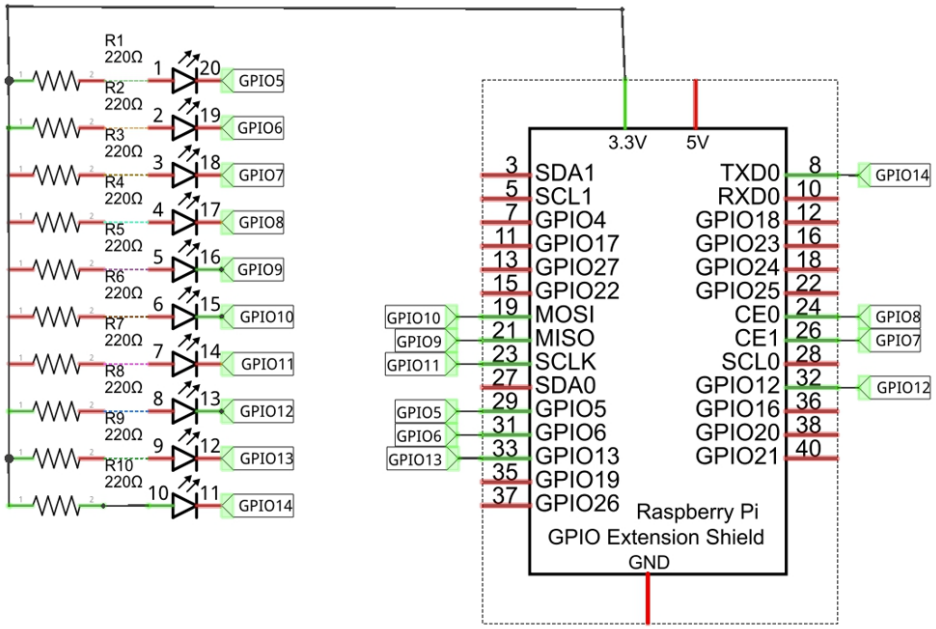
A Bar Graph LED has 10 LEDs integrated into one compact component. The two rows of pins at its bottom are paired to identify each LED like the single LED used earlier.



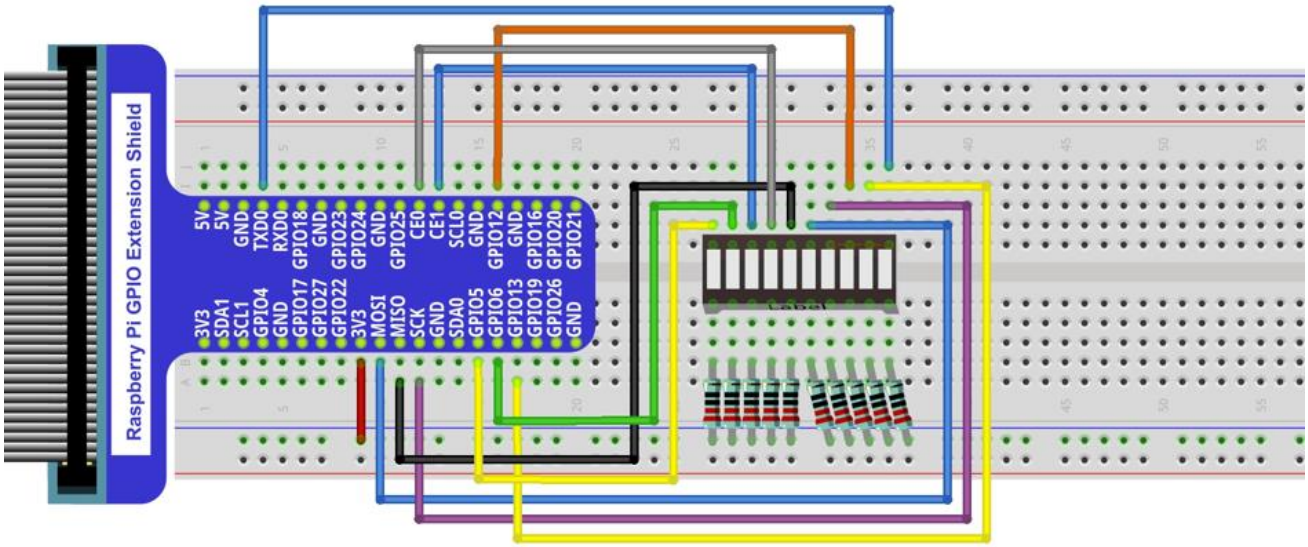
Circuit

A reference system of labels is used in the circuit diagram below. Pins with the same network label are connected together.

Schematic diagram



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



If LEDbar doesn't work, rotate LEDbar 180° to try. The label is random.

In this circuit, the cathodes of the LEDs are connected to the GPIO, which is different from the previous circuit. The LEDs turn ON when the GPIO output is low level in the program.

Code

This project is designed to make a flowing water lamp, which are these actions: First turn LED #1 ON, then

turn it OFF. Then turn LED #2 ON, and then turn it OFF... and repeat the same to all 10 LEDs until the last LED is turns OFF. This process is repeated to achieve the "movements" of flowing water.

03.0_LED_Bar_Graph

First, observe the project result, and then learn about the code in detail.

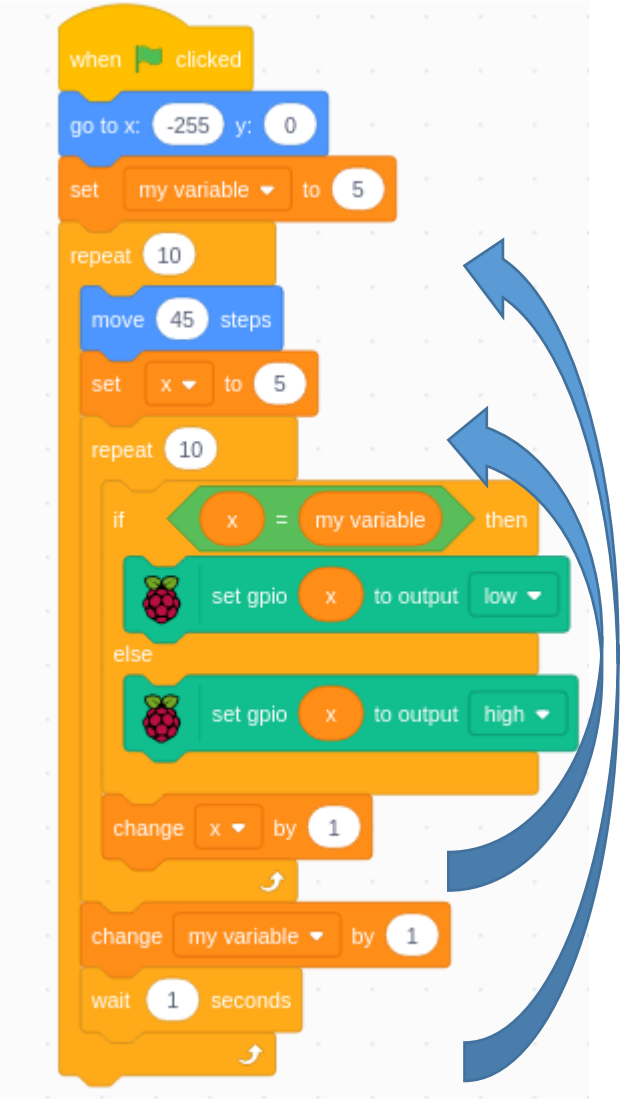
If you have any concerns, please contact us via: support@freenove.com

The circuit is same as previous section.

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/03.0_LED_Bar_Graph.sb3

Click the green flag. Then the sprite will move from number 0 to 9.

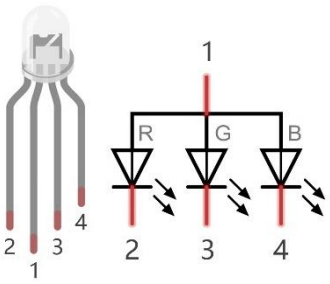


Here are two repeat functions.

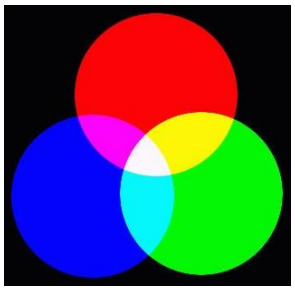
Chapter 4 RGB LED

In this chapter, we will learn how to control an RGB LED.

An RGB LED has 3 LEDs integrated into one LED component. It can respectively emit Red, Green and Blue light. In order to do this, it requires 4 pins (this is also how you identify it). The long pin (1) is the common which is the Anode (+) or positive lead, the other 3 are the Cathodes (-) or negative leads. A rendering of a RGB LED and its electronic symbol are shown below. We can make RGB LED emit various colors of light and brightness by controlling the 3 Cathodes (2, 3 & 4) of the RGB LED



Red, Green, and Blue light are called 3 Primary Colors when discussing light (Note: for pigments such as paints, the 3 Primary Colors are Red, Blue and Yellow). When you combine these three Primary Colors of light with varied brightness, they can produce almost any color of visible light. Computer screens, single pixels of cell phone screens, neon lamps, etc. can all produce millions of colors due to phenomenon.



RGB

If we use a three 8 bit PWM to control the RGB LED, in theory, we can create $2^8*2^8*2^8$=16777216 (16 million) colors through different combinations of RGB light brightness.

Next, we will use RGB LED to make a multicolored LED.

Project 04.0_RGBLED and 04.1_Keyboard_RGBLED

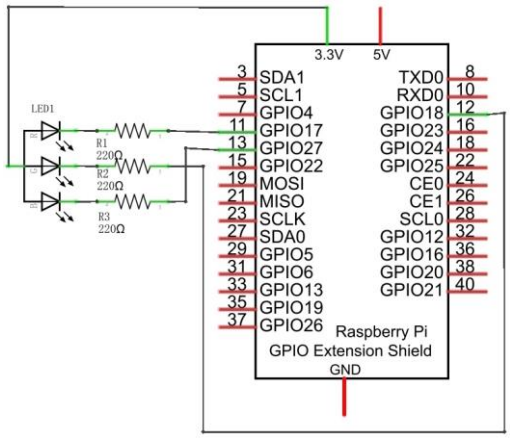
In this project, we will make a multicolored LED, which we can program the RGB LED to automatically change colors.

Component List

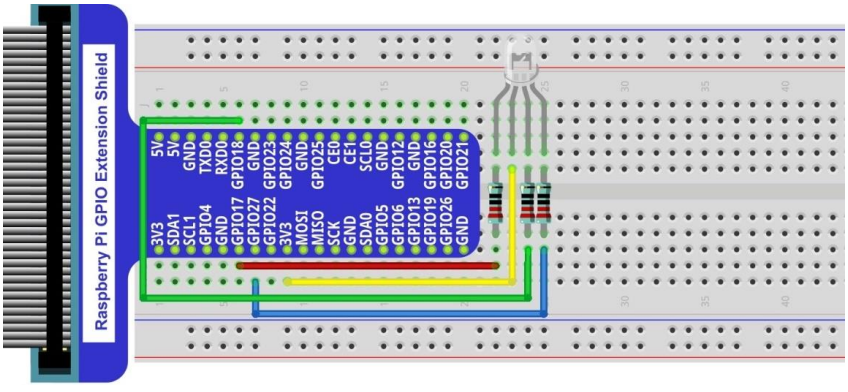
Raspberry Pi (with 40 GPIO) x1 GPIO Extension Board & Wire x1 Breadboard x1	RGB LED x1 	Resistor 220Ω x3 
Jumper Wire 		

Circuit

Schematic diagram



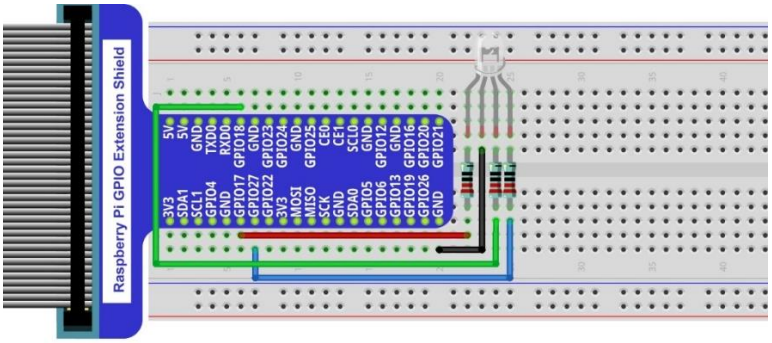
Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



In this kit, the RGB LED is **Common anode**. The **voltage difference** between LED will make it work. There is no visible GND. The GPIO ports can also receive current while in output mode.

If circuit above doesn't work, the RGB LED may be common cathode. Please try following wiring.

There is no need to modify code for random color.



Code

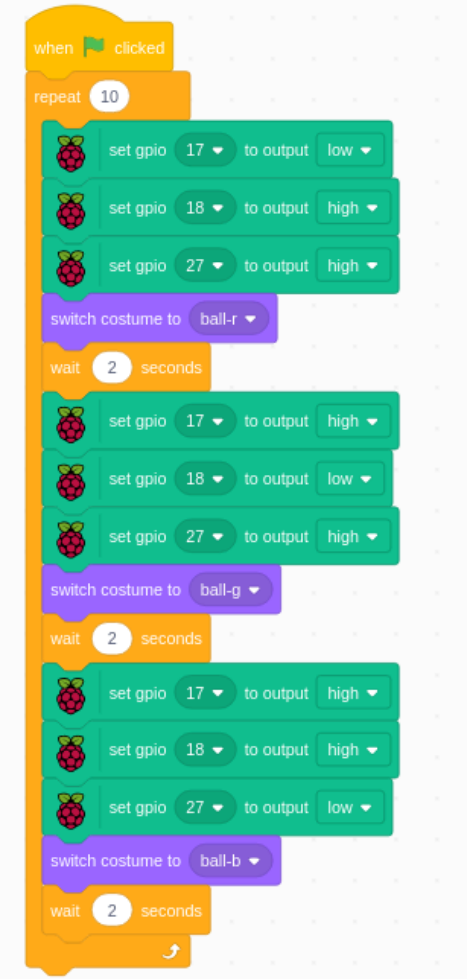
04.0_RGBLED

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/04.0_RGBLED.sb3

Click the green flag. Then the LED and the ball will show red, green and blue color in turn.

The code is for common anode RGB LED. One low level and two high.



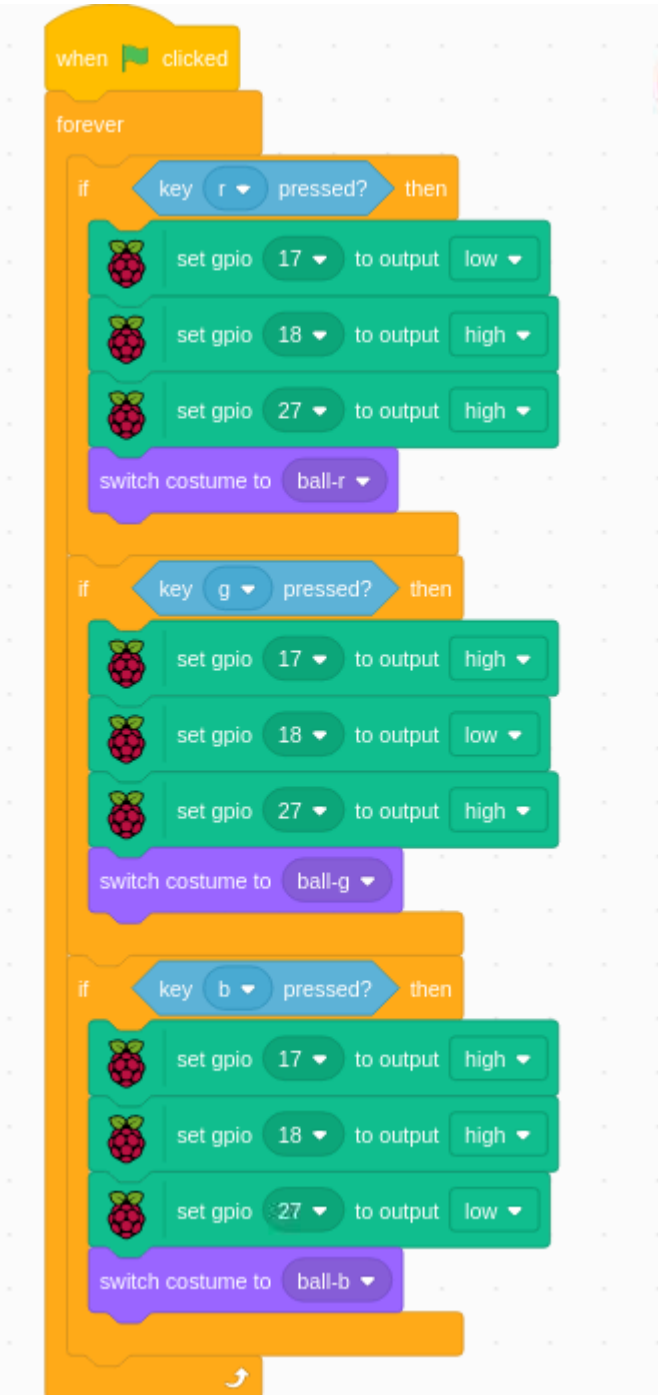
04.1_Keyboard_RGBLED

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/04.1_Keyboard_RGBLED.sb3

Click the green flag. Then click “r”, “g” and “b” randomly and observe the color change.

“r”, “g” and “b” on key board will be used and detect.



Chapter 5 Two Electronic Button Switch

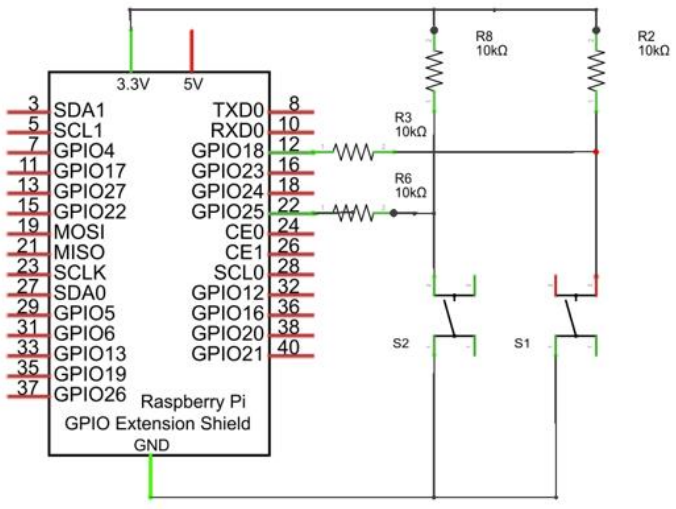
Project 05.0_Two_Electronic_Button_LED, 05.1_Squash0 and 5.2_Jump

Components

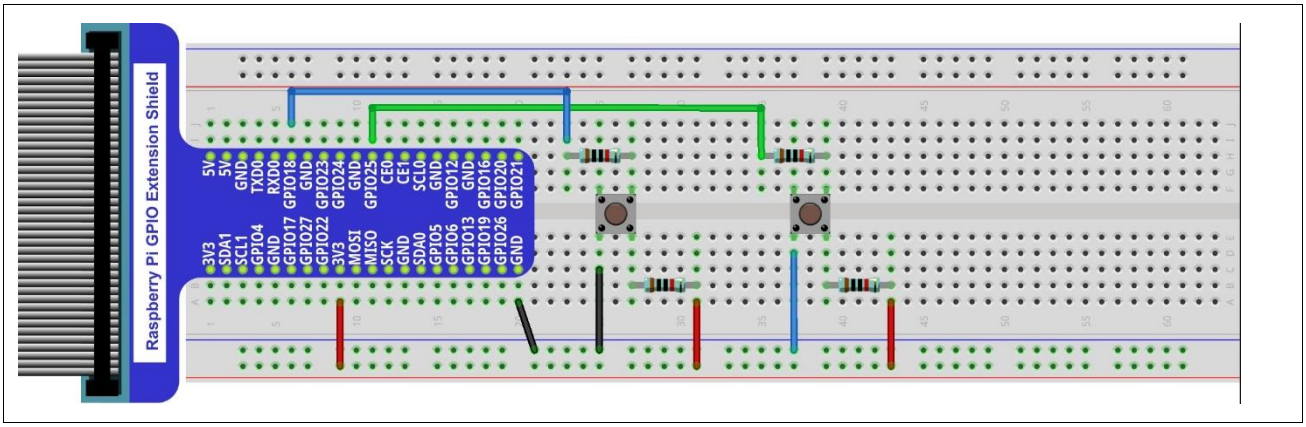
Raspberry Pi (with 40 GPIO) x1 GPIO Extension Board & Wire x1 Breadboard x1	Resistor 10kΩ x4 	Push Button Switch x2 
Jumper Wire 		

Circuit

Schematic diagram



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



Code

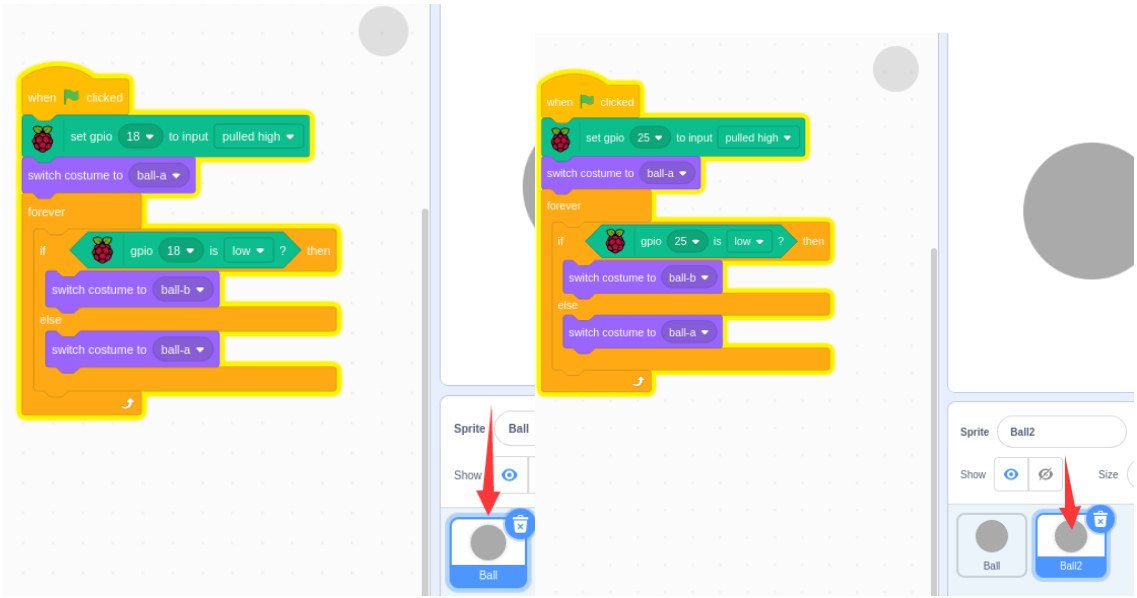
05.0_Two_Electronic_Button_LED

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/05.0_Two_Electronic_Button_LED.sb3

Click the green flag. Then press two button switches to observe the change.

There are two balls. We need write two sections of code for two GPIOs.



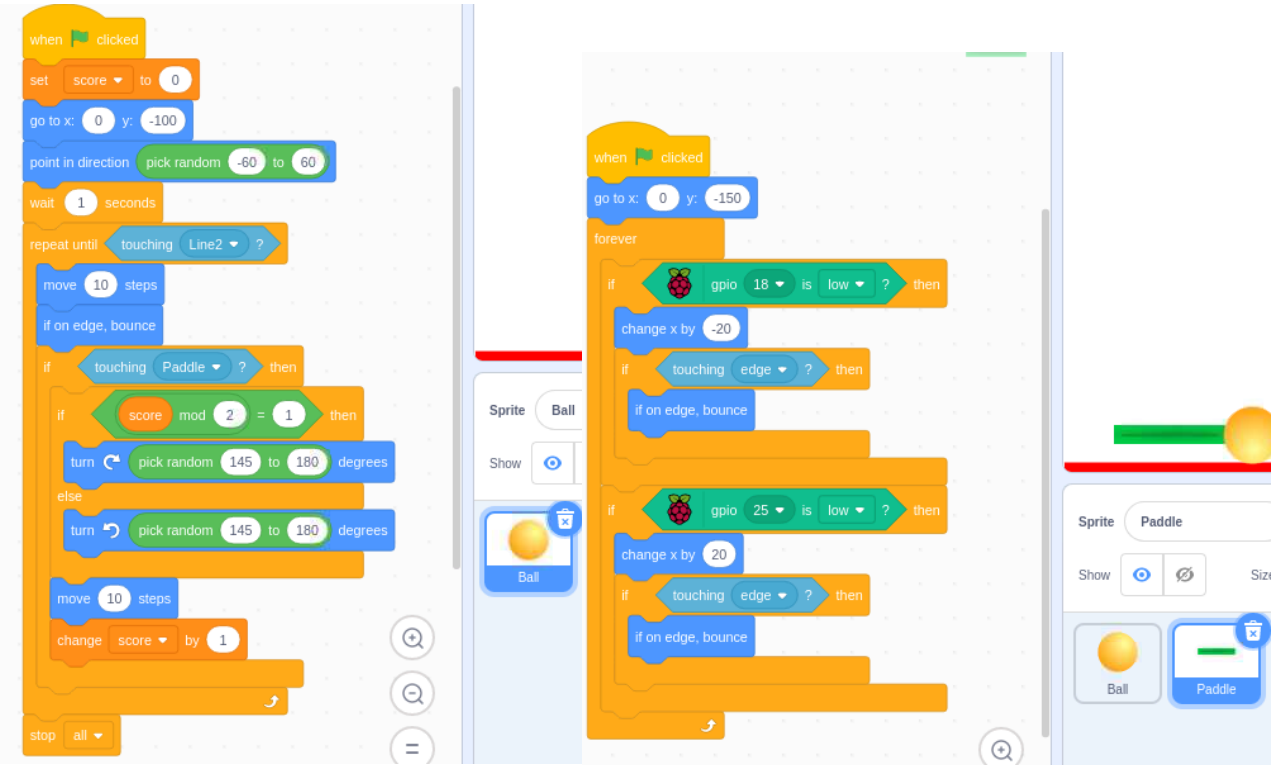
05.1_Squash

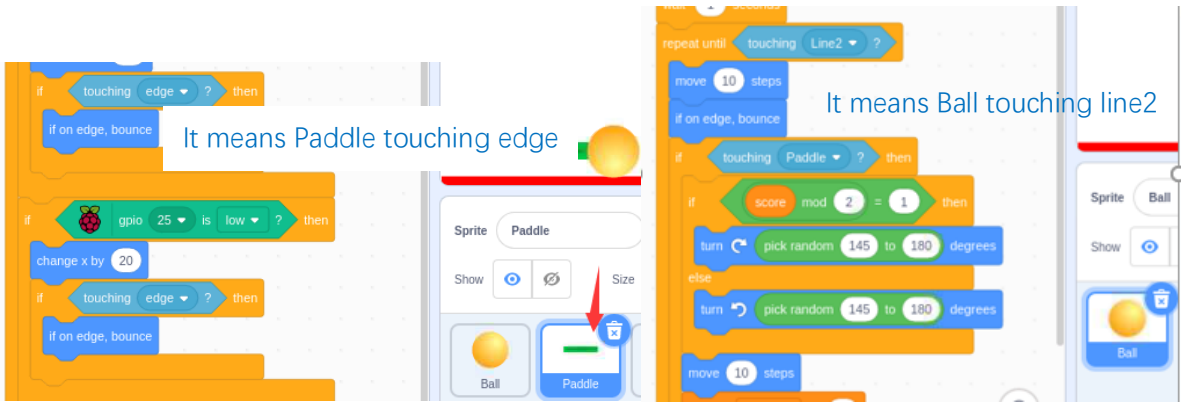
Load the code to scratch3.

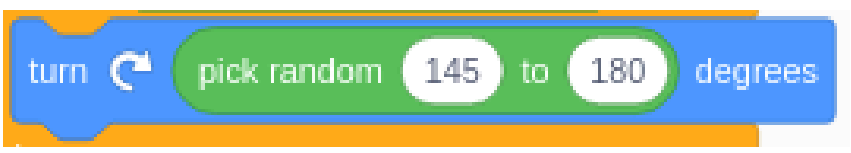
Freenove_Kit/Code/Scratch3/05.1_Squash.sb3

Click the green flag. Press the two button swtiches to move the green paddle.

There are two sprites moving, so we need to write two sections of code. One is for the ball, the other for the paddle.







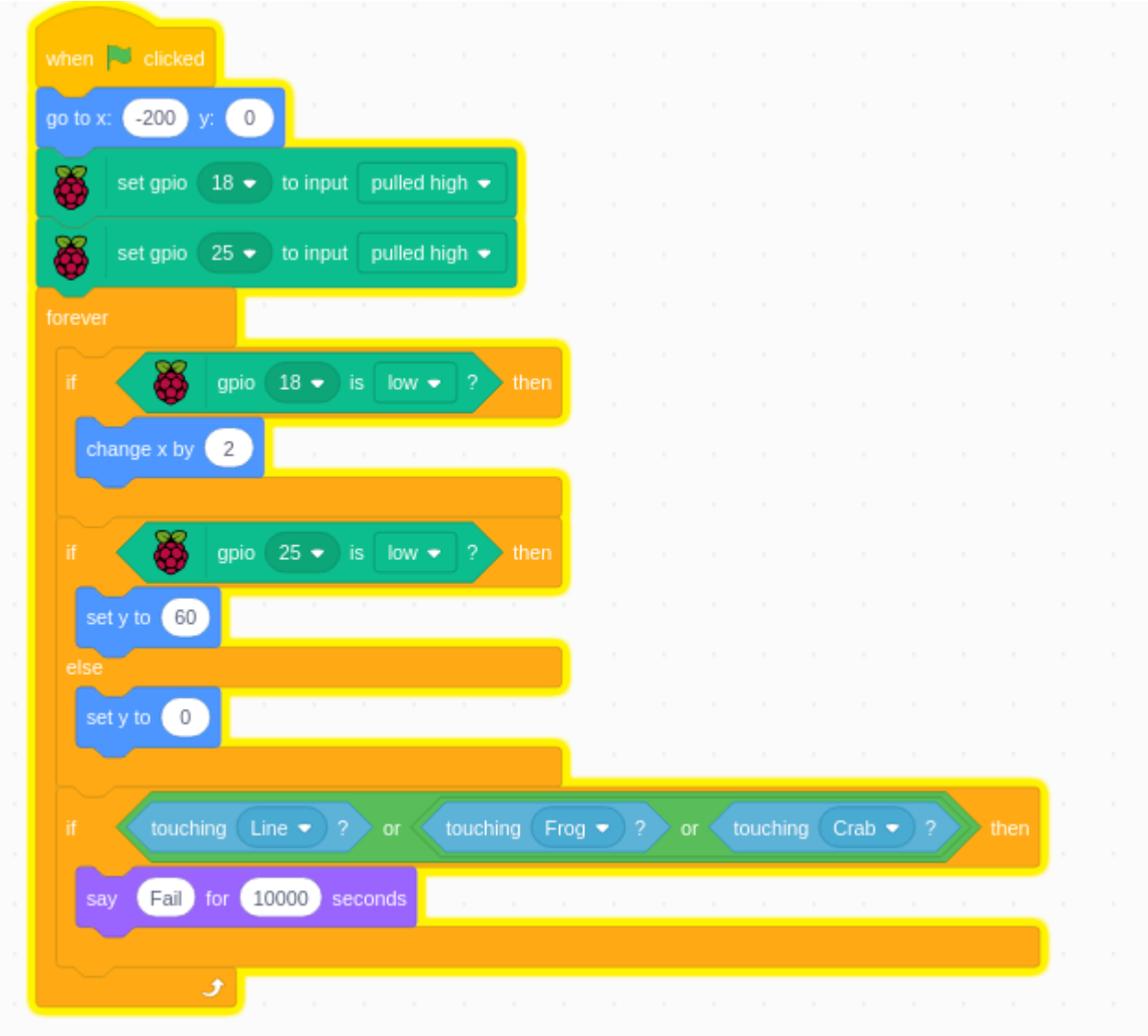
Pick random “range”. It can ne any one of 145~180, like: 145 150 180 146.

05.2_Jump

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/05.2_Jump.sb3

Click the green flag. Press the two button swtiches to make the sprite move without touching the other items.



A or B or C. When one of them is true, the whole result would be true. It means the sprite1 cannot touch any other items.

Chapter 6 Buzzer

In this chapter, we will learn about buzzers and the sounds they make. And in our next project, we will use an active buzzer to make a doorbell and a passive buzzer to make an alarm.

Project 06.0_Buzzer

We will make a doorbell with this functionality: when the Push Button Switch is pressed the buzzer sounds and when the button is released, the buzzer stops. This is a momentary switch function.

Component List

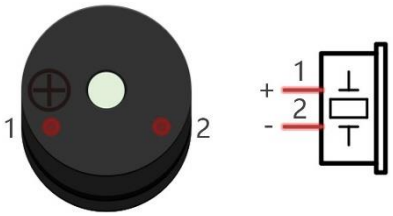
<table>
<tr><td colspan="2">Raspberry Pi (with 40 GPIO) x1
GPIO Extension Board & Ribbon Cable x1
Breadboard x1</td><td colspan="3">Jumper Wire
  </td></tr>
<tr><td>NPN transistorx1
(S8050)
  </td><td>Active buzzer x1
  </td><td>Push Button Switch x1
  </td><td>Resistor 1kΩ x1
  </td><td>Resistor 10kΩ x2
  </td></tr>
</table>

Component knowledge

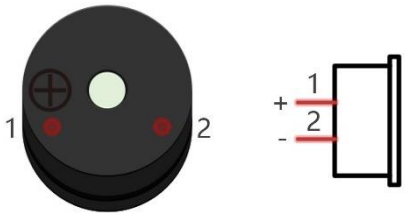
Buzzer

A buzzer is an audio component. They are widely used in electronic devices such as calculators, electronic alarm clocks, automobile fault indicators, etc. There are both active and passive types of buzzers. Active buzzers have oscillator inside, these will sound as long as power is supplied. Passive buzzers require an external oscillator signal (generally using PWM with different frequencies) to make a sound.

Active buzzer



Passive buzzer



Active buzzers are easier to use. Generally, they only make a specific sound frequency. Passive buzzers require an external circuit to make sounds, but passive buzzers can be controlled to make sounds of various frequencies. The resonant frequency of the passive buzzer in this Kit is 2kHz, which means the passive buzzer is the loudest when its resonant frequency is 2kHz.

How to identify active and passive buzzer?

1. As a rule, there is a label on an active buzzer covering the hole where sound is emitted, but there are exceptions to this rule.
2. Active buzzers are more complex than passive buzzers in their manufacture. There are many circuits and crystal oscillator elements inside active buzzers; all of this is usually protected with a waterproof coating (and a housing) exposing only its pins from the underside. On the other hand, passive buzzers do not have protective coatings on their underside. From the pin holes, view of a passive buzzer, you can see the circuit board, coils, and a permanent magnet (all or any combination of these components depending on the model.

Active buzzer bottom

Passive buzzer bottom

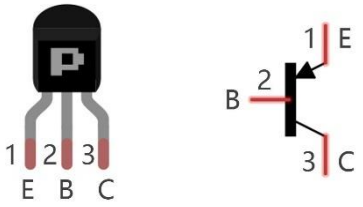
Transistors

A transistor is required in this project due to the buzzer's current being so great that GPIO of RPi's output capability cannot meet the power requirement necessary for operation. A NPN transistor is needed here to

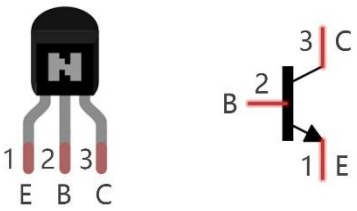
amplify the current.

Transistors, full name: semiconductor transistor, is a semiconductor device that controls current (think of a transistor as an electronic “amplifying or switching device”. Transistors can be used to amplify weak signals, or to work as a switch. Transistors have three electrodes (PINs): base (b), collector (c) and emitter (e). When there is current passing between "be" then "ce" will have a several-fold current increase (transistor magnification), in this configuration the transistor acts as an amplifier. When current produced by "be" exceeds a certain value, "ce" will limit the current output. at this point the transistor is working in its saturation region and acts like a switch. Transistors are available as two types as shown below: PNP and NPN,

PNP transistor



NPN transistor



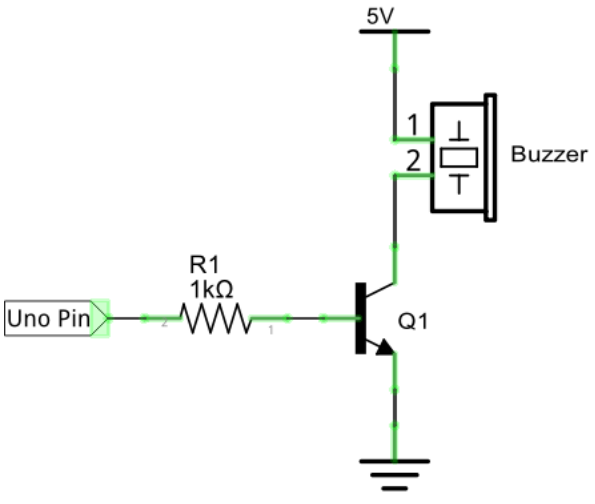
In our kit, the PNP transistor is marked with 8550, and the NPN transistor is marked with 8050.

Thanks to the transistor's characteristics, they are often used as switches in digital circuits. As micro-controllers output current capacity is very weak, we will use a transistor to amplify its current in order to drive components requiring higher current.

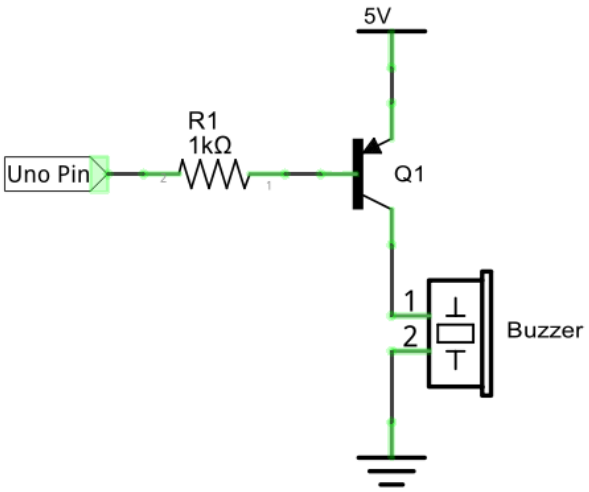
When we use a NPN transistor to drive a buzzer, we often use the following method. If GPIO outputs high level, current will flow through R1 (Resistor 1), the transistor conducts current and the buzzer will make sounds. If GPIO outputs low level, no current will flow through R1, the transistor will not conduct currentand buzzer will remain silent (no sounds).

When we use a PNP transistor to drive a buzzer, we often use the following method. If GPIO outputs low level, current will flow through R1. The transistor conducts current and the buzzer will make sounds. If GPIO outputs high level, no current flows through R1, the transistor will not conduct current and buzzer will remain silent (no sounds). Below are the circuit schematics for both a NPN and PNP transistor to power a buzzer.

NPN transistor to drive buzzer

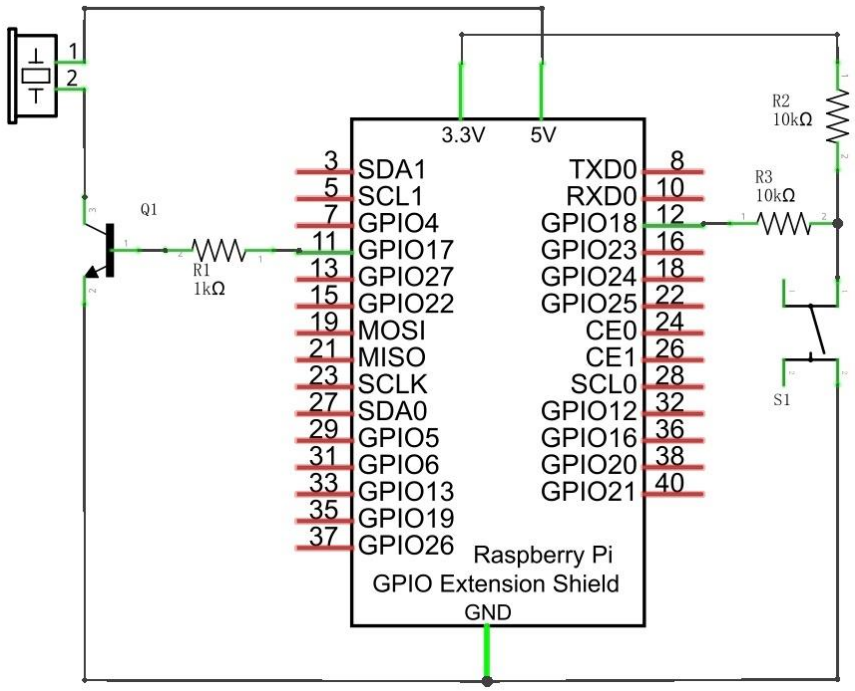


PNP transistor to drive buzzer

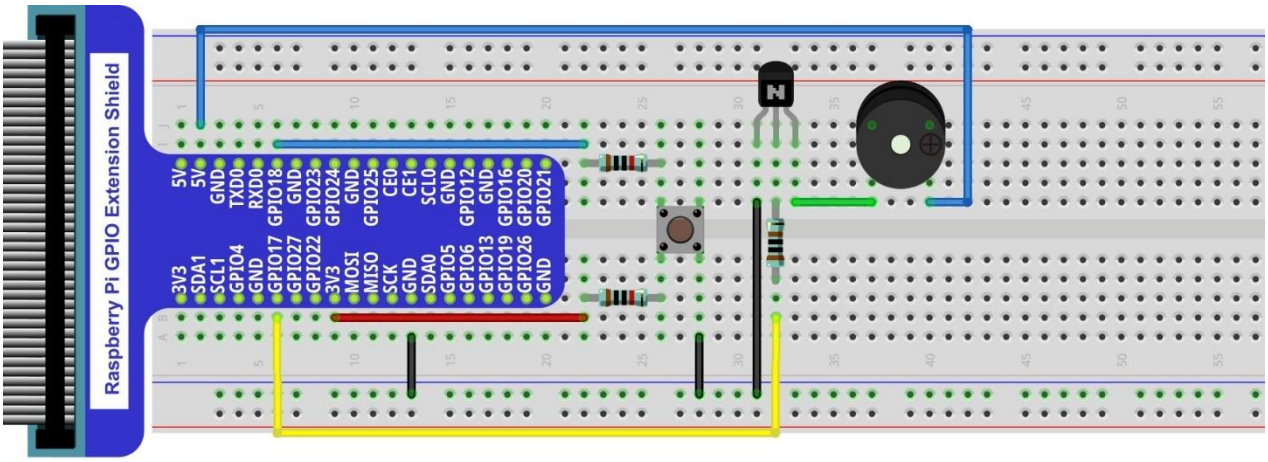


Circuit

Schematic diagram with RPi GPIO Extension Shield



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



Note: in this circuit, the power supply for the buzzer is 5V, and pull-up resistor of the push button switch is connected to the 3.3V power feed. Actually, the buzzer can work when connected to the 3.3V power feed but this will produce a weak sound from the buzzer (not very loud).

Code

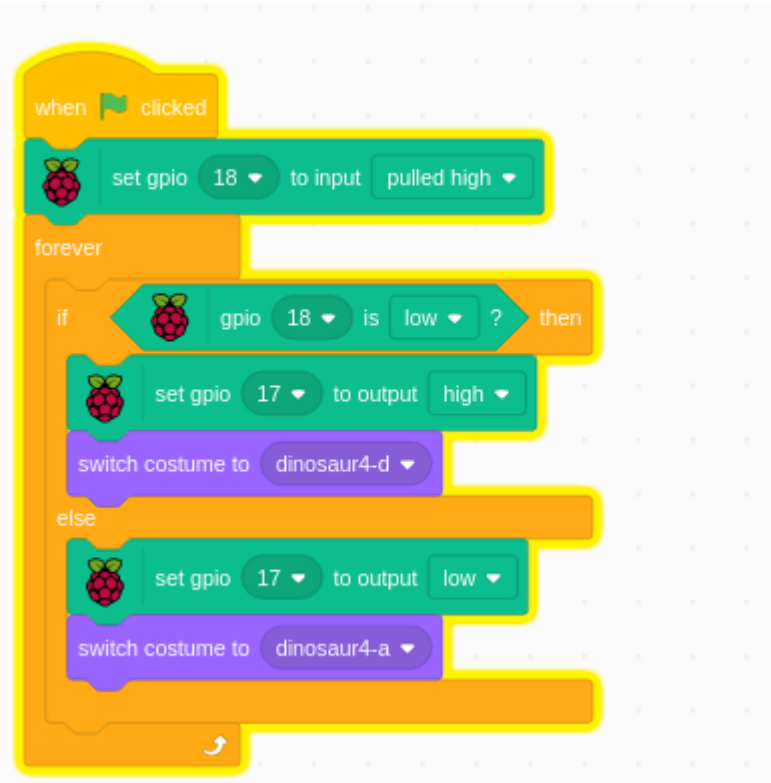
06.0_Buzzer

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/06.0_Buzzer.sb3

Click the green flag. Press the button swtich and the buzzer will beep.

The principle is the same to using a button switch to control an LED.



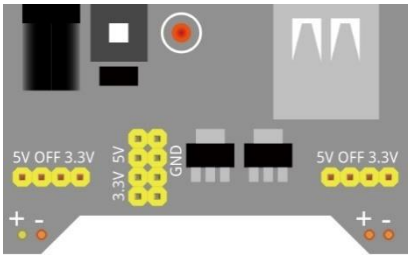
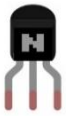
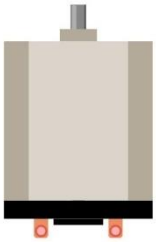
Here we use active buzzer. For use of passive buzzer, please refer to Tuttoiral.pdf

Chapter 07.0_Motor

In this chapter, we will learn a kind of special switch module, Relay Module. We can use small current to control big current via this module.

Project 07.0_Motor

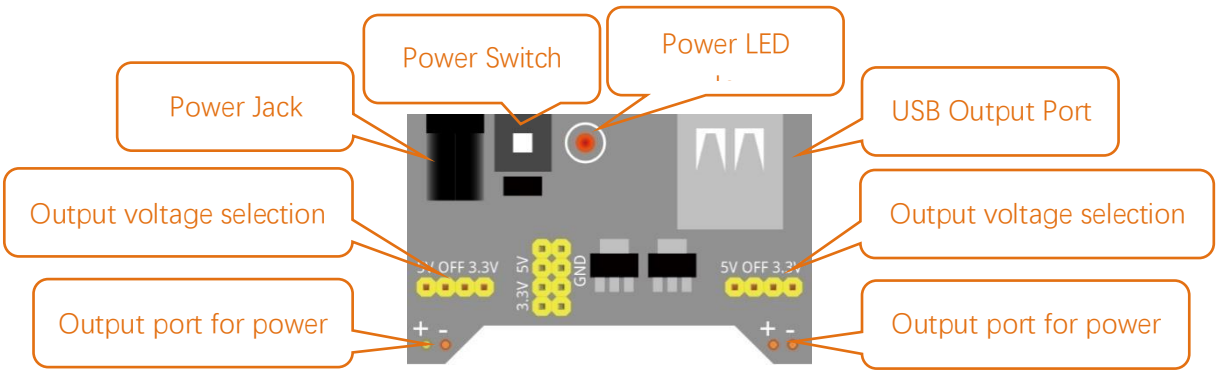
Component List

<table>
<tr><td colspan="3">Raspberry Pi (with 40 GPIO) x1
GPIO Expansion Board & Ribbon Cable x1
Breadboard x1</td><td colspan="3">Jumper Wire x11
  </td></tr>
<tr><td colspan="6">9V battery (prepared by yourself) & battery line
  </td></tr>
<tr><td colspan="3">Breadboard Power module x1
  </td><td>Resistor 10kΩ x2
  </td><td>Resistor 1kΩ x1
  </td><td>Resistor 220Ω x1
  </td></tr>
<tr><td>NPN transistor x1
  </td><td>Relay x1
  </td><td>Motor x1
  </td><td>Push button x1
  </td><td>LED x1
  </td><td>Diode x1
  </td></tr>
</table>

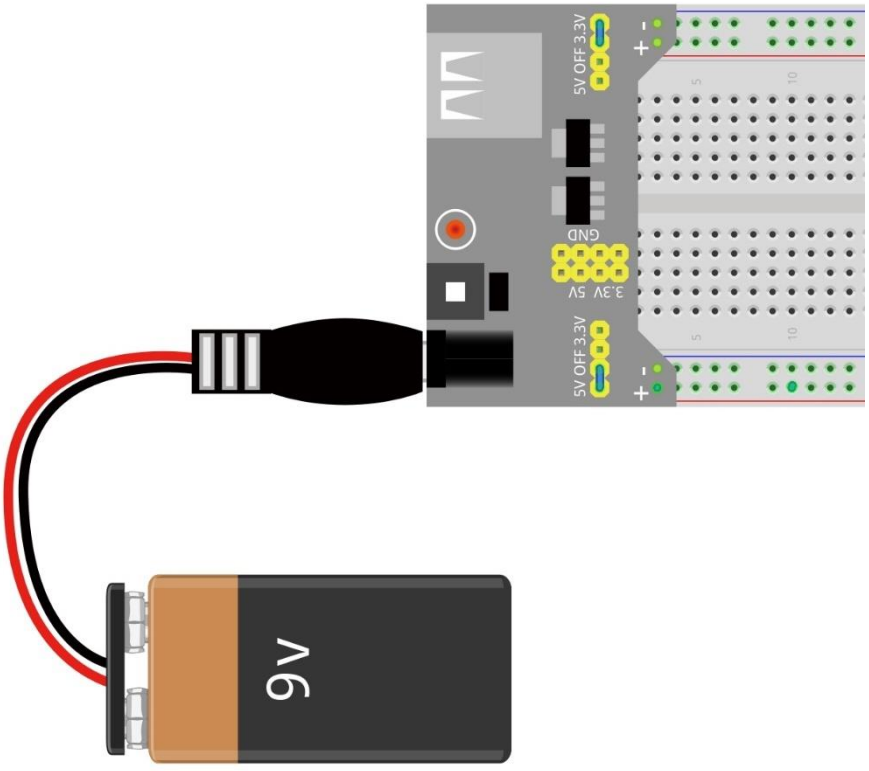
Component knowledge

Breadboard Power Module

Breadboard Power Module is an independent circuit board, which can provide independent 5V or 3.3V power to the breadboard when building circuits. It also has built-in power protection to avoid damaging your RPi module. The schematic diagram below identifies the important features of this Power Module:

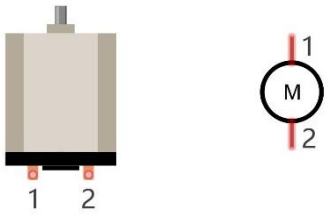


Here is an acceptable connection between Breadboard Power Module and Breadboard using a 9V battery and the provided power harness:



DC Motor

DC Motor is a device that converts electrical energy into mechanical energy. DC Motors consist of two major parts, a Stator and the Rotor. The stationary part of a DC Motor is the Stator and the part that Rotates is the Rotor. The Stator is usually part of the outer case of motor (if it is simply a pair of permanent magnets), and it has terminals to connect to the power if it is made up of electromagnet coils. Most Hobby DC Motors only use Permanent Magnets for the Stator Field. The Rotor is usually the shaft of motor with 3 or more electromagnets connected to a commutator where the brushes (via the terminals 1 & 2 below) supply electrical power, which can drive other mechanical devices. The diagram below shows a small DC Motor with two terminal pins.

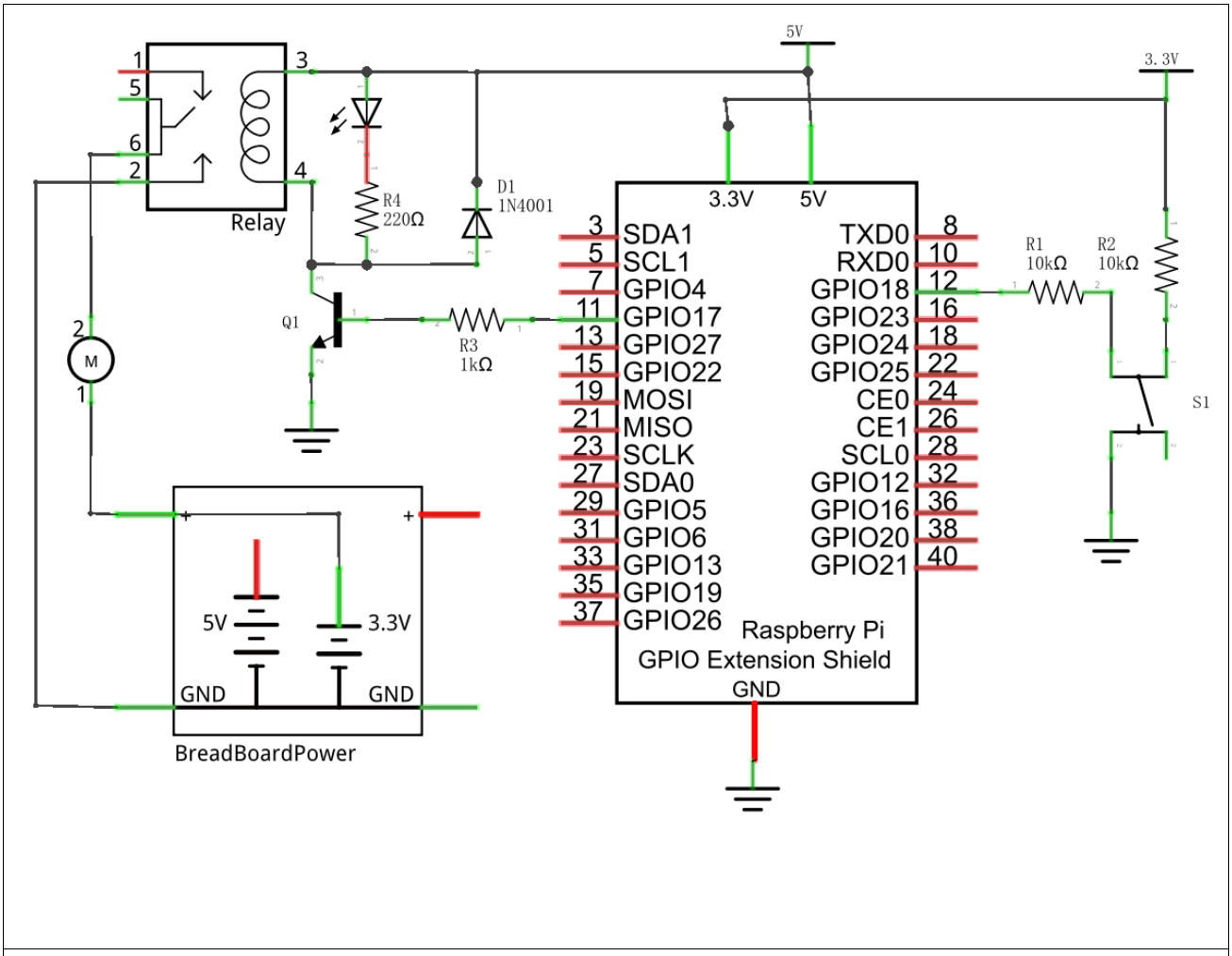


When a DC Motor is connected to a power supply, it will rotate in one direction. If you reverse the polarity of the power supply, the DC Motor will rotate in opposite direction. This is important to note.

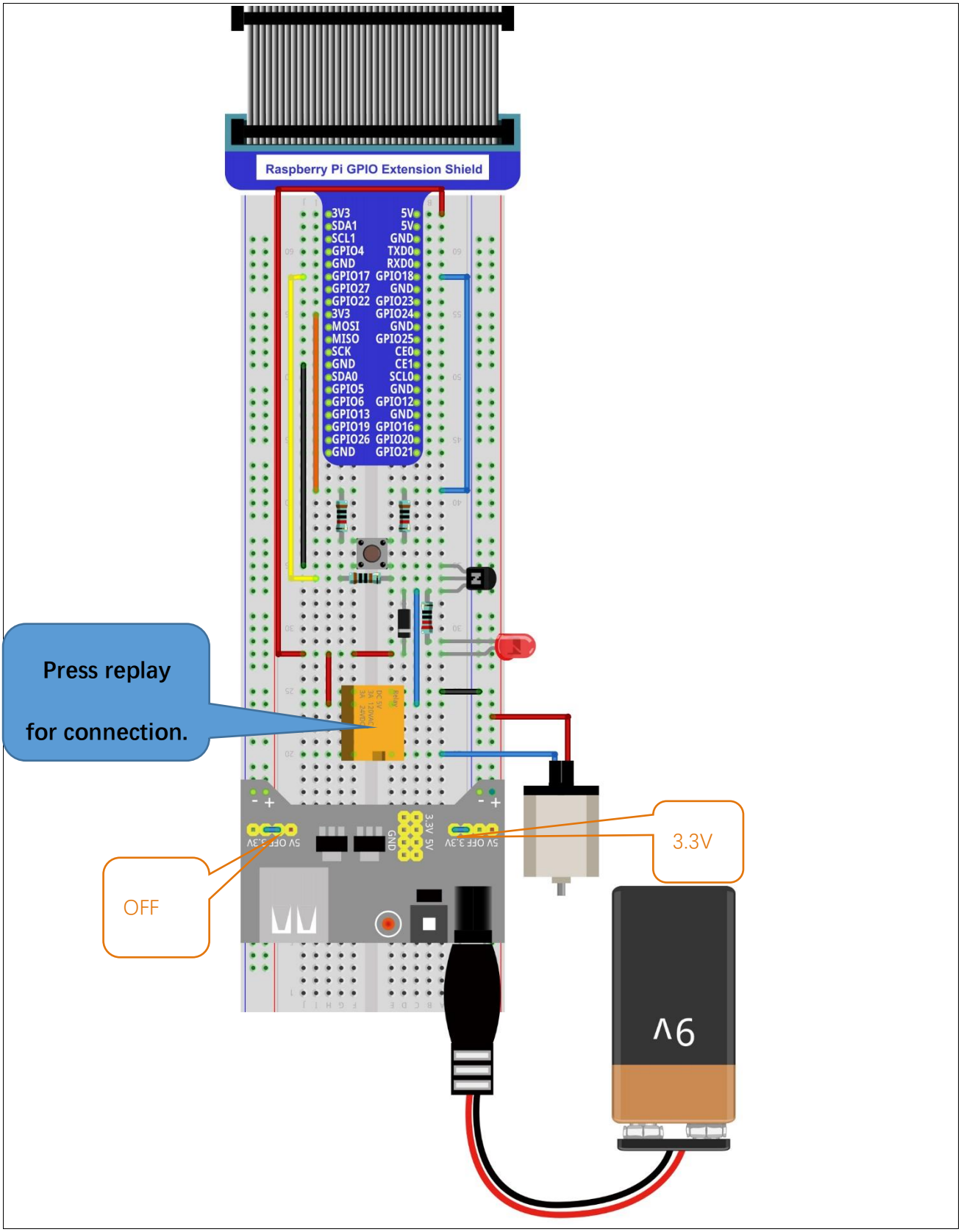
Circuit

Use caution with the power supply voltage needed for the components in this circuit. The Relay requires a power supply voltage of 5V, and the DC Motor only requires 3.3V. Additionally, there is an LED present, which acts as an indicator (ON or OFF) for the status of the Relay's active status.

Schematic diagram



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


Raspberry Pi GPIO Extension Shield
3V3 5V
SDA1 5V
SCL1 GND
GPIO4 TXD0
GND RXD0
GPIO17 GPIO18
GPIO27 GND
GPIO22 GPIO23
3V3 GPIO24
MOSI GND
MISO GPIO25
SCK CE0
GND CE1
SDA0 SCL0
GPIO5 GND
GPIO6 GPIO12
GPIO13 GND
GPIO19 GPIO16
GPIO26 GPIO20
GND GPIO21
Press replay
for connection.
OFF
3.3V
9V

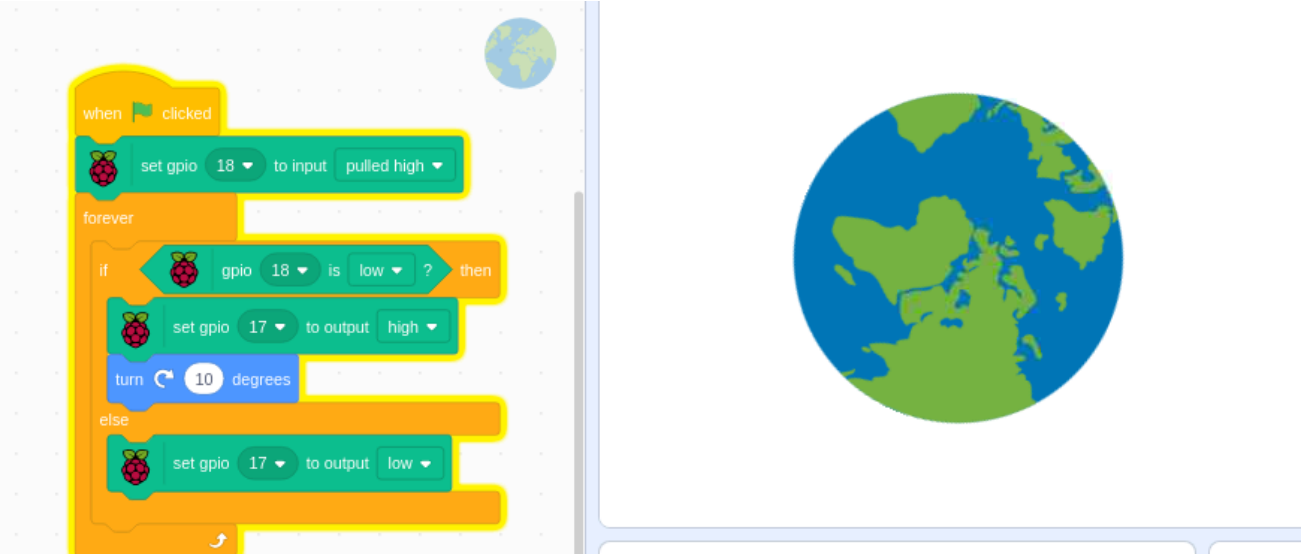
Code

07.0_Motor

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/07.0_Motor.sb3

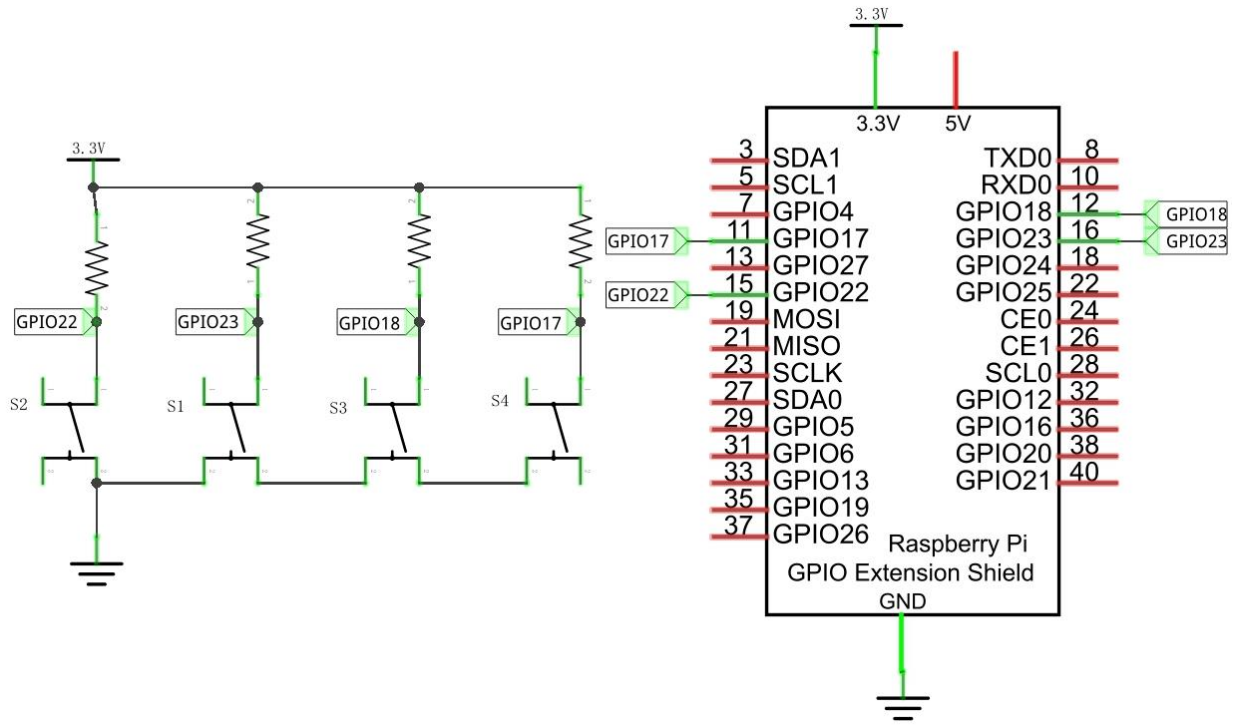
Click the green flag. Press the button swtich and the motor will rotate or stop.



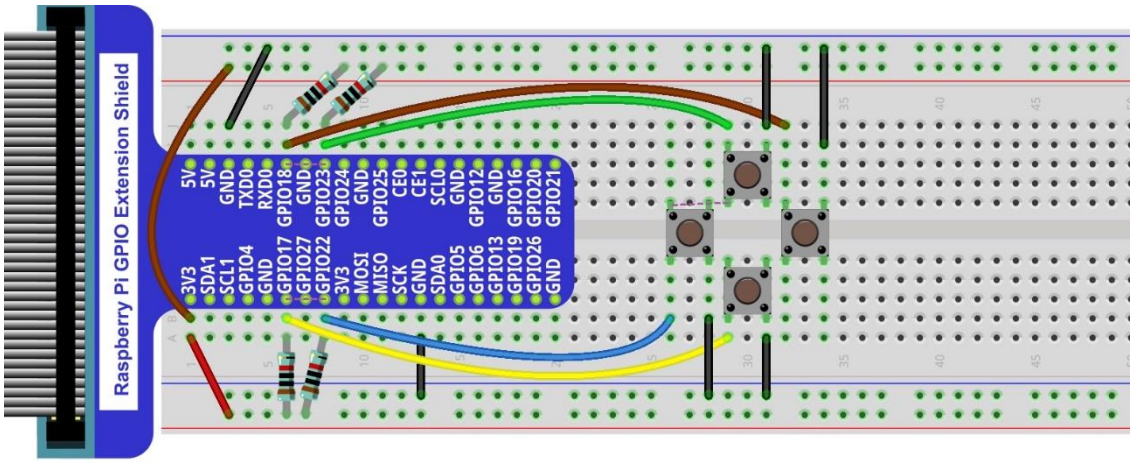
Chapter 08 Four button swithches games.

Project 08.0_Pick_Apple and 08.1_Fish

Schematic diagram



Hardware connection



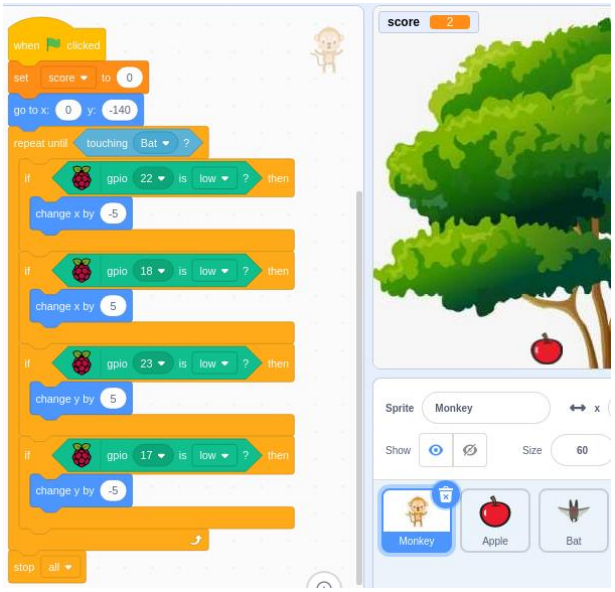
08.0_PicK_Apple

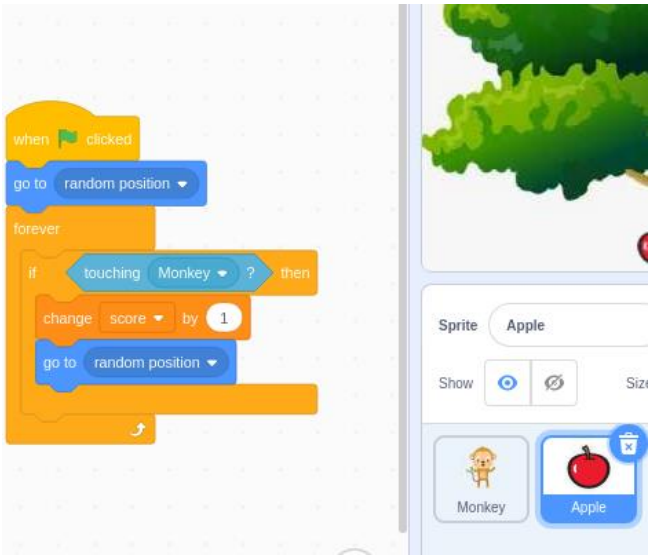
Load the code to scratch3.

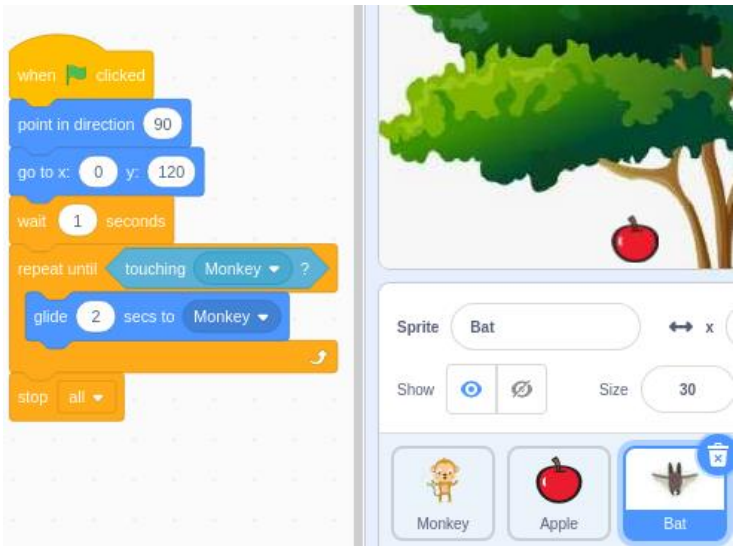
Freenove_Kit/Code/Scratch3/08.0_PicK_Apple.sb3

Click the green flag to control the monkey to pick apple, while avoiding touching the bat.

There are there items moving, so there are three sections of code.







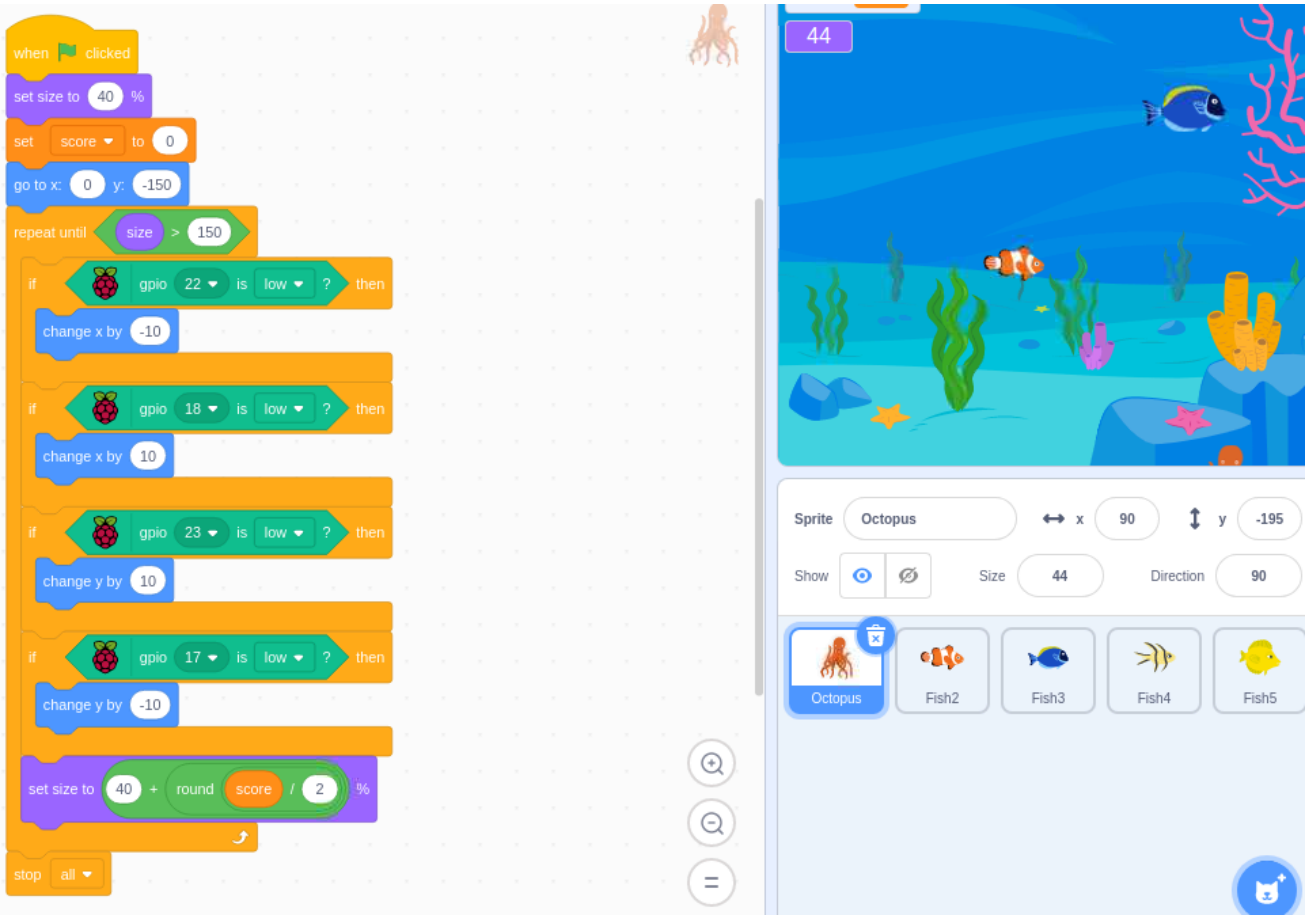
08.1_Fish

Load the code to scratch3.

Freenove_Kit/Code/Scratch3/08.1_Fish.sb3

Click the green flag. Control the octopus to approach the fish.

There is one octopus and four fishes. All of them will move, so there are five sections of code.



What's Next?

THANK YOU for participating in this learning experience! If you have completed all of the projects successfully you can consider yourself a Raspberry Pi Master.

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: **support@freenove.com**
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you are interested in processing, you can study the Processing.pdf in the unzipped folder.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.