

Lesson 1 Introduction to Raspberry Pi

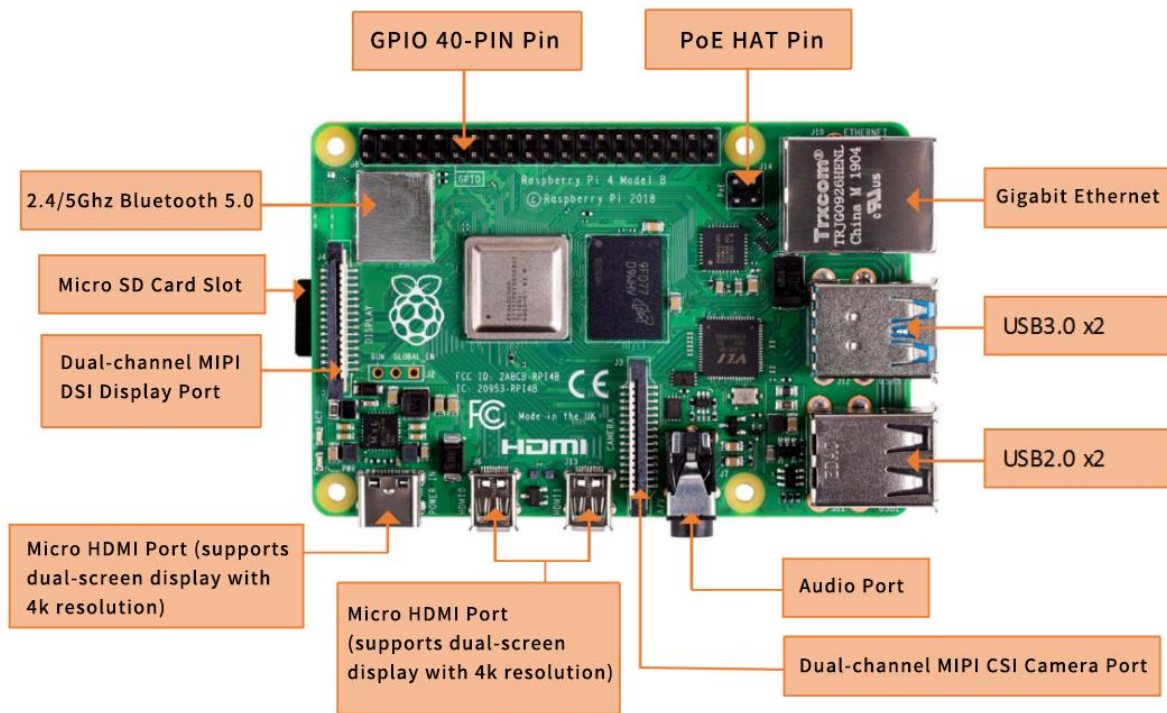
The Raspberry Pi is not included in the product kit. Recommended Raspberry Pi version: 3B, 3B+, 4B.

1.1 Overview of Raspberry Pi

Raspberry Pi (Raspberry Pi, RasPi/RPi) is developed by the British charity organization "Raspberry Pi Foundation", based on ARM microcomputer motherboard, only the size of a credit card, but has the basic functions of a personal computer. The original purpose of the Foundation's development of the Raspberry Pi was to improve the teaching level of the school's computer science and related disciplines, and cultivate the youth's computer programming interest and ability. Nowadays, most people use the Raspberry Pi for embedded development, which is mostly used in the Internet of Things, smart home, and artificial intelligence.

1.2 Raspberry Pi Motherboard

In our lessons, we will use the Raspberry Pi 4 motherboard. Let's take a look at the structure of the Raspberry Pi 4 motherboard as shown below:



The Raspberry Pi 4 motherboards mainly includes the following 11 parts/ports:

1. GPIO 40-PIN GPIO

The General-Purpose Input Output (GPIO) is designed as a slot with two rows of pins on the board. It can be used to connect various peripheral electronic devices and sensors to control or monitor them through input/output level signals. For example, you can use GPIO to control the speed of a DC motor or read the distance measured of an ultrasonic sensor. These functions of GPIO differ the Raspberry Pi from common computer motherboards because it gives developers the freedom to operate manually. We will further introduce GPIO in the subsequent chapters and use them extensively.

2. Gigabit Ethernet port

The Ethernet interface allows the Raspberry Pi to connect to the computer network in a wired manner, which enables easy access to the Internet or logging in to the Raspberry Pi remotely. The Raspberry Pi's Ethernet interface is implemented by using a USB bus through which data is transferred. Most models of Raspberry Pi include an Ethernet interface.

3. Micro HDMI port

High-definition multimedia interface (HDMI) is a fully digital video and sound transmission interface used to transmit uncompressed audio and video signals. By connecting it to a display (or TV) equipped with an HDMI interface, you can display media data of the Raspberry Pi. The HDMI interface can transmit video and audio signals at the same time, meaning you don't need to connect speakers to the audio interface of the Raspberry Pi – unless you really want to play audio through the audio interface, then you may need to modify the operating system configuration accordingly.

4. USB2.0/3.0 port

The Universal Serial Bus (USB) interface is the most common interface on a computer. You can use it to connect devices such as keyboards, mice, USB flash drives, and wireless network cards. When the number of USB ports is not enough, we can also extend by a USB hub.

5. Audio port

Audio interface (3.5mm headphone jack). When HDMI connection is not used, you can connect speakers or headphones via the standard 3.5mm headphone jack to play audio. At the same time, the interface also integrates a composite video interface with a composite audio-visual output function, which is generally used to connect old models of TV, which is rarely used now.

6. MIPI CSI camera port

The CSI interface can be used to connect the CSI camera to the Raspberry Pi via a ribbon cable for easy video recording and image capture. Compared with the USB camera, this camera module has better performance.

7. USB-C 5V/3A power supply port

The Micro USB power supply interface is one of the most used supply methods of the Raspberry Pi. The rated voltage is 5V. The standard current requirements of different Raspberry Pi models are slightly different. For example: the 1B type only needs 700mA, while the 3B+ type requires 2.5A. The chargers of many Android mobile phones can provide flawless power to the Raspberry Pi. The current required for Raspberry Pi also depends on the external device connected. It is recommended to do the

calculation before connecting. Choose a power supply with suitable current (power) for the Raspberry Pi. When the external device demands a large power, an extra power supply should be adopted.

8. Micro SD card slot

The SD card slot is located on the back of the Raspberry Pi motherboard. The SD/MicroSD card is an essential storage of the Raspberry Pi and usually used for operating system installation and data storage. Use an SD card above 2GB. For better experience, it is recommended to equip your Raspberry Pi with a large-capacity (above 16G), high-speed (Class10 or above) SD card.

9. Bluetooth port

The Bluetooth function allows the Raspberry Pi to connect with Bluetooth-enabled devices (such as a mouse, keyboard, and gaming console).

10. PoE HAT port

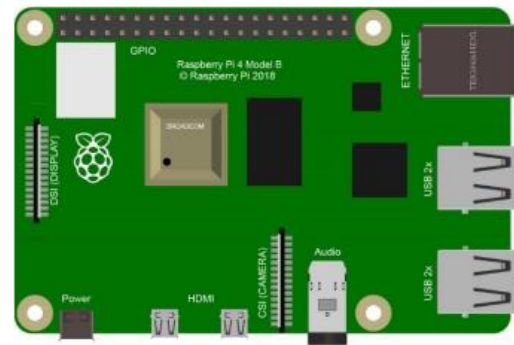
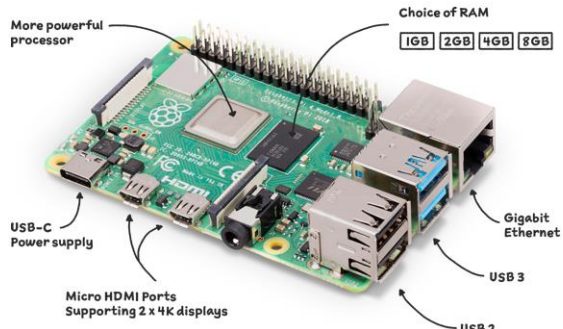
Active Ethernet (Power Over Ethernet, PoE) refers to a technology that uses Ethernet for power transmission. Besides the basic Micro USB and GPIO power supply, the Raspberry Pi 3B+ type adds a new method of power supply over Ethernet. User can connect a network cable to supply power to the Raspberry Pi without the need of configuring an additional power supply, which is convenient for certain application scenarios.

11. MIPI DSI display port

You can connect the LCD display to the Raspberry Pi, which is generally used for embedded product development. Under normal circumstances, the HDMI interface can already meet the demand.

1.3 Below are the Raspberry Pi pictures and model pictures supported by this product:

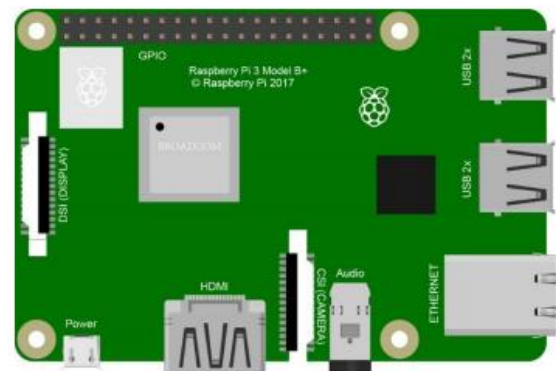
Raspberry Pi 4 Model B:	Model diagram
-------------------------	---------------



Raspberry Pi 3 Model B+:



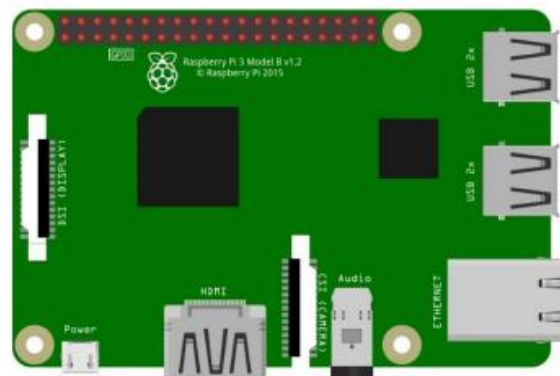
Model diagram



Raspberry Pi 3 Model B:



Model diagram



1.4 Operating System

The Raspberry Pi supports a variety of operating systems, mainly based on Linux and Windows, and most of them can be found on the official website of the Raspberry Pi Foundation (www.raspberrypi.org). Two representative operating systems will be introduced here.

1. Raspbian

Raspbian is the official operating system of the Raspberry Pi Foundation. It is customized based on Debian GNU/Linux and can run on all versions of the Raspberry Pi motherboard. According to the experience, Raspbian and Raspberry Pi combine all the best: stable operation, powerful function, easy to use, being able to meet various application needs, etc. Therefore, it is strongly recommended to use Raspbian as the preferred operating system for Raspberry Pi. In the following chapters, we will further introduce the use of Raspbian in detail and develop various applications on it.

2. Windows 10 IoT Core

Windows 10 IoT Core is an operating system specifically created by Microsoft for the Internet of Things ecosystem. Windows 10 IoT Core is the core version of the Windows 10 IoT operating system. It has relatively simple functions and can run on type 2B or above models of Raspberry Pi. The installation and use of Windows 10 IoT Core will not be described in detail here. For more information, visit Microsoft's website.

Apart from the two systems above, there are several more that support the Raspberry Pi, including Ubuntu MATE, OSMC, LibreELEC, PiNet, RISC OS, etc. As for which one to choose, it depends on what you want to use Raspberry for. If you want to use it as a computer or for electronic project development, then Raspbian is a very good choice. If you plan to use it as a media center, you can consider using OSMC or LibreELEC.

1.5 Programming Language

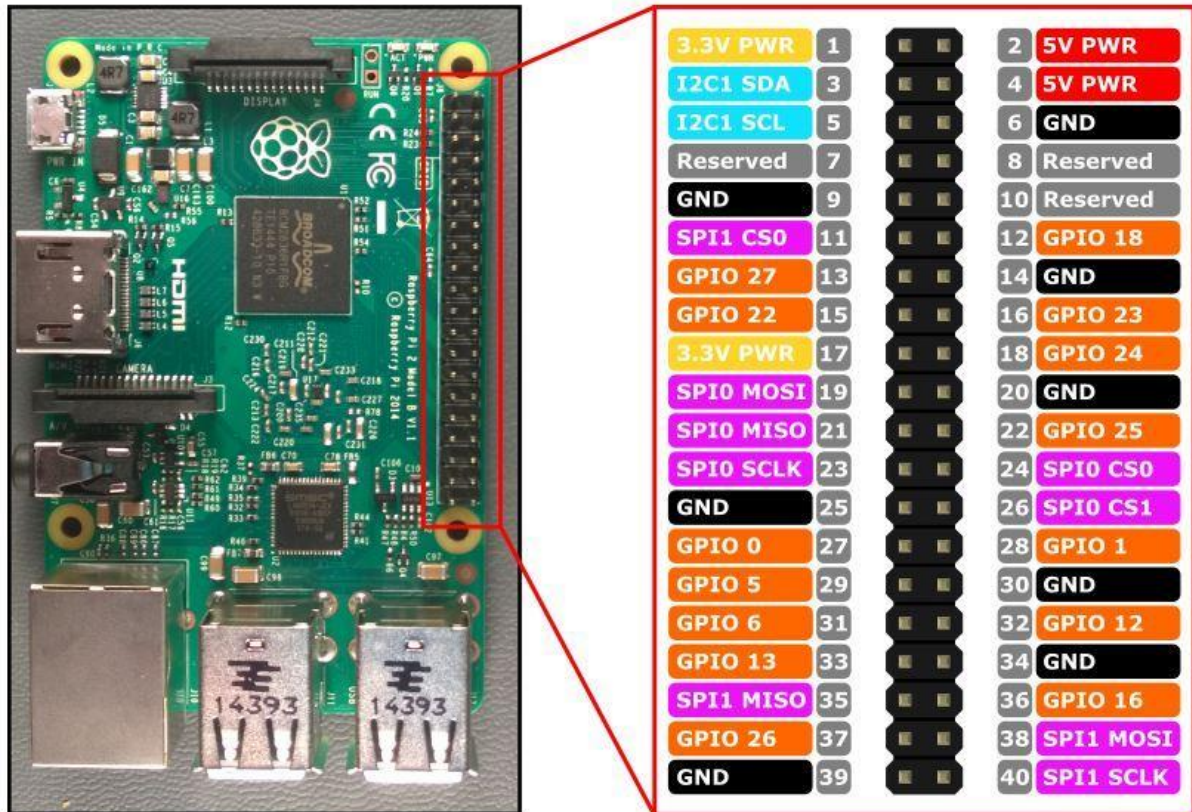
For the Raspberry Pi, there are many programming languages available. In fact, any language that can be compiled for the ARM architecture (such as the C language) can be used for the Raspberry Pi. The

most popular language should be Python. In fact, the Pi in the name of the Raspberry Pi was inspired by the word Python.

Python is an interpretive, object-oriented, and dynamic data type high-level programming language with powerful functions, good compatibility, and high reliability. Python programs are easy to write and read. At present, there are two major versions of Python: Python 2 and Python 3. Both versions are updated and well maintained, but people still have disputes about which to use. You can visit Python's official website (www.python.org) to learn more about them. In the following tutorials we will mainly use Python 3 for development introduction. In addition, because of the Raspberry Pi's splendid compatibility, the program we wrote on the 3B+ model can be run on the Zero W model with little modification.

1.6 Introduction to GPIO

GPIO (General Purpose I/O Ports) are general-purpose input/output ports. In layman's terms, they are some pins with two rows of pins. They can be used to output high and low levels or to read the state of the pins – whether it is high or low. User can communicate with the hardware through the GPIO port (such as UART), control the hardware (such as LED, buzzer, etc.), read the working status signal of the hardware (such as interrupt signal), etc.



Introduction of GPIO pins

GPIO pin comparison table

Raspberry Pi 40Pin Pin Comparison Table

wiringPi Encoding	BCM Encoding	Function Name	BOARD Encoding of Physical Pins		Function Name	BCM Encoding	wiringPi Encoding
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

Form description:

1. Three naming (coding) methods for Raspberry Pi pins

Three ways to name the Raspberry Pi pins:

- The WiringPi number is the pin number of the functional wiring (such as TXD, PWM0, etc.).

- The BCM number is the Broadcom pin number, also known as GPIO.
- The physical number is the number corresponding to the physical location of the pin on the Raspberry Pi motherboard (1 ~40).

2. 3.3V/5V pin and GND pin

3.3V/5V pin and GND pin are commonly known as power and ground pins. The power and ground pins allow your Raspberry Pi to power some external components, such as LED lights. It should be noted that before using these pins to power any external modules or components, care should be taken. Excessive operating current or peak voltage may damage the Raspberry Pi. Do not use voltages greater than 5V!

3. SDA and SCL pins

The SDA and SCL pins constitute the I2C interface. I2C is a simple, bidirectional two-wire synchronous serial bus developed by Philips. It only requires two wires to transfer information between devices connected to the bus. The Raspberry Pi can control multiple sensors and components through the I2C interface. Their communication is done through SDA (data pin) and SCL (clock speed pin). Each slave device has a unique address, allowing rapid communication with many devices. The ID_EEPROM pin is also an I2C protocol, which is used to communicate with HATs.

4. SCLK, MOSI and MISO pins

SCLK, MOSI and MISO pins form the SPI interface. SPI is a serial peripheral interface, used to control components with a master-slave relationship, and works in a slave-in, master-out and master-in-slave manner. The SPI on the Raspberry Pi consists of SCLK, MOSI, and MISO interfaces, and SCLK is used for controlling data speed, MOSI sends data from the Raspberry Pi to the connected device, while MISO does the opposite.

5. TXD and RXD pins

TXD and RXD form a UART interface. TXD is a pin to send data, and RXD is a pin to receive data. A friend who uses Arduino must have heard of UART or Serial. The Universal Asynchronous

Receiver/Transmitter interface is used to connect the Arduino to the computer for which it is programmed. It is also used for communication between other devices and the RX and TX pins. If the Raspberry Pi has a serial terminal enabled in raspi-config, you can use these pins to control the Raspberry Pi through a computer or directly to control the Arduino.