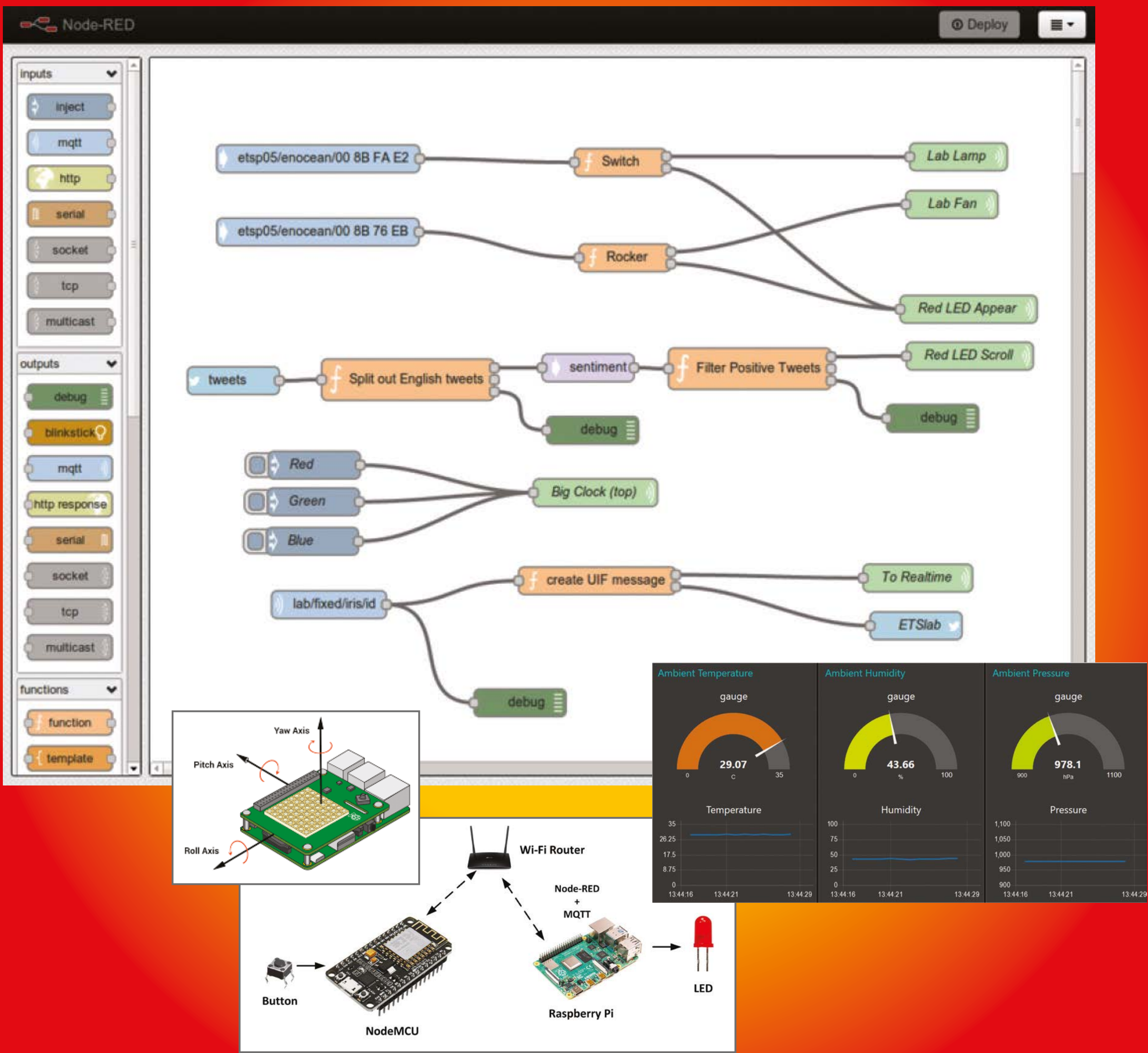
Design IoT Projects with Raspberry Pi, Arduino and ESP32

Programming with Node-RED



elektor

Dogan Ibrahim

Programming with Node-RED

Design IoT Projects with Raspberry Pi, Arduino and ESP32

Dogan Ibrahim

elektor

LEARN DESIGN SHARE

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
78 York Street, London W1H 1DP, UK
Phone: (+44) (0)20 7692 8344

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978-1-907920-88-2
EISBN 978-3-89576-364-9
EPUB 978-3-89576-365-6

Prepress Production: D-Vision, Julian van den Berg
First published in the United Kingdom 2020

Preface

It is becoming important for microcontroller users to adapt to new technologies quickly and learn the architecture and use of high-performance 32-bit microcontrollers. Several manufacturers offer 32-bit microcontrollers as general-purpose processors in embedded applications. ARM architecture is currently one of the most frequently used 32-bit microcontroller architectures in mobile devices, such as in mobile phones, iPads, games consoles, and in many other portable devices.

Raspberry Pi is based on ARM architecture and it is currently one of the most commonly used single-board computers used by students, engineers, and hobbyists. There are hundreds of Raspberry Pi based projects available on the internet

Arduino is also a very popular microcontroller development board. Although it is based on 8-bit architecture, it is widely used as it is supported by large numbers of software libraries, making it easy to develop projects in relatively short times.

Another popular microcontroller is ESP32. It is widely sold as a development board known as the ESP32 DevKitC. The reason why ESP32 is very popular is that it has onboard Wi-Fi and Bluetooth capability, many digital and analog input ports, and built-in timers. Additionally, its power consumption is very low and it has a processor that can be put into sleep mode, which consumes extremely low current.

The Internet of Things (IoT) is becoming a major application area of embedded systems. As a result, more people are becoming interested in learning about embedded design and programming. In addition, we can see that more technical colleges and universities are moving away from legacy 8-bit and 16-bit microcontrollers and introducing 32-bit embedded microcontrollers into their curriculums. Some IoT applications demand precision, high processing power, and very low power consumption.

Node-RED is an open-source visual editor for wiring the Internet of Things produced by IBM. Node-RED comes with a large number of nodes to handle a variety of tasks. The required nodes are selected and joined together to perform a particular task. Node-RED is based on flow type programming where nodes are configured and joined together to form an application program. There are nodes for doing very complex tasks, including web access, Twitter, E-mail, HTTP, Bluetooth, MQTT, controlling GPIO ports, etc. The nice thing about Node-RED is that the programmer does not need to learn how to write complex programs. For example, an email can be sent by joining a few nodes together and writing a few lines of code.

The aim of this book is to teach how Node-RED can be used in projects. The main processor used in the majority of projects in this book is Raspberry Pi 4. Chapters are included to show how Node-RED can be used with Arduino Uno, ESP32 DevKitC, and the ESP8266 NodeMCU microcontroller development boards.

Many example projects are given in the book. All projects have been fully tested and were working at the time of writing this book. Users can select flow programs of the projects from the book website and configure them to suit their own applications. The operation of each flow program is fully described in the book.

I hope you enjoy reading this book and are then able to use Node-RED happily in your future projects.

Prof Dr. Dogan Ibrahim
London, 2020

Chapter 1 • Raspberry Pi 4

1.1 Overview

Raspberry Pi has recently become one of the most popular and most powerful single-board computers used by students, hobbyists, and professional engineers. Raspberry Pi 4 is the latest and the most powerful version of Raspberry Pi. In this chapter, we will be looking at the basic specifications and requirements of the Raspberry Pi 4 computer. What is included in this chapter can easily be applied to other models in the Raspberry Pi family.

1.2 Parts of the Raspberry Pi 4

Just like its earlier versions, Raspberry Pi 4 is a single-board computer having the following basic specifications:

- 1.5GHz 64-bit quad-core CPU
- 1GB, 2GB, or 4GB RAM
- 2.4GHZ and 5.0GHz IEEE 802.11ac Wi-Fi
- Bluetooth 5.0 BLE
- Gigabit Ethernet
- 2 x USB 3.0, 2 x USB 2.0 and 1 x USB-C ports
- 2 x micro-HDMI ports for dual display, supporting up to 4K resolution
- DSI display and CSI camera ports
- micro SD card slot for the operating system and data storage
- 4-pole stereo audio and composite video port
- 40-pin GPIO header
- Power over Ethernet (PoE) enabled with the PoE HAT
- OpenGL ES 3.0 graphics

Figure 1 shows the Raspberry Pi 4 board with its major components identified.

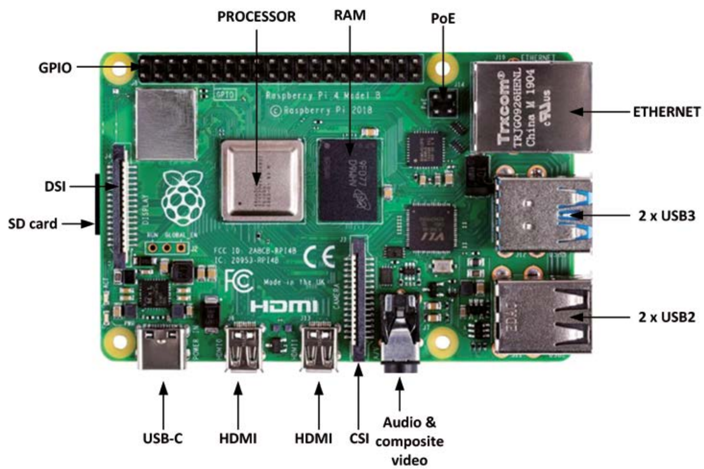


Figure 1.1 Raspberry Pi 4 board

A brief description of the various components on the board is given below:

Processor: the processor is enclosed in a metal cap and it is based on Broadcom BCM2711B0, which consists of a Cortex A-72 core, operating at 1.5GHz.

RAM: There are 3 versions of Raspberry Pi 4 depending on the amount of DDR4 RAM required: 1GB, 2GB, and 4GB.

USB Ports: Raspberry Pi 4 includes 2 x USB 3.0, 2 x USB 2.0, and 1 x USB-C ports. USB 3.0 data transfer rate is 4,800 Mbps (megabits per second), while USB 2.0 can transfer at up to 480Mbps, i.e. 10 times slower than the USB 2.0. The USB-C port enables the board to be connected to a suitable power source.

Ethernet: The Ethernet port enables the board to be connected directly to an Ethernet port on a router. The port supports Gigabit connections (125Mbps).

HDMI: Two micro HDMI ports are provided that support up to 4K screen resolutions. HDMI adapters can be used to interface the board to standard size HDMI devices.

GPIO: A 40-pin header is provided as the GPIO (General Purpose Input Output). This is compatible with the earlier GPIO ports.

Audio and Video Port: A 3.5mm jack type socket is provided for stereo audio and composite video interface. Headphones can be connected to this port. External amplifier devices will be required to connect speakers to this port. This port also supports composite video, enabling TVs, projectors, and other composite video compatible display devices to be connected to the port.

CSI Port: This is the camera port (Camera Serial Interface), allowing a compatible camera to be connected to the Raspberry Pi.

DSI Port: This is the display port (Display Serial Interface), allowing a compatible display (e.g. 7 inch Raspberry Pi display) to be connected to the Raspberry Pi.

PoE Port: This is a 4-pin header, allowing the Raspberry Pi to receive power from a network connection.

Micro SD Card: This card is mounted at the cardholder placed at the bottom of the board and it holds the operating system software as well as the operating system and user data. Requirements of the Raspberry Pi 4

As listed below, a number of external devices are required before the Raspberry Pi can be used:

- Power supply
- Micro SD card

- Operating system software
- USB keyboard and mouse
- A micro HDMI cable to receive sound and video signals
- HDMI compatible display or TV (you may also need to have micro HDMI to DVI-D or VGA adapters. A 3.5mm TRRS type cable and plug will be required if you will be using an old TV with composite video)

Power Supply: A 5V 3A power supply with a USB-C type connector is required. You may either purchase the official Raspberry Pi 4 power supply (Figure 1.2) or use a USB-C adapter to provide power from an external source.

Figure 1.2 Official Raspberry Pi 4 power supply

Micro SD Card: It is recommended to use a micro SD card with a capacity of at least 8GB, although higher capacity (e.g. 16GB or 32GB) is better as there will be room to grow in the future. A Class 10 (or faster) card is recommended.

Operating System: You can purchase the operating system pre-loaded on a micro SD card, known as NOOBS (New Out Of Box Software) which requires minimum configuration before it is fully functional. The alternative is to purchase a blank micro SD card and upload the operating system on this card. The steps to prepare a new micro SD card with the operating system is given in the next Chapter.

USB Keyboard and Mouse: You can either use a wireless or wired keyboard and mouse pair. If using a wired pair, you should connect the keyboard to one of the USB ports and the mouse to another USB port. If using a wireless keyboard and mouse, you should connect the wireless dongle to one of the USB ports.

Display: A standard HDMI compatible display monitor with a micro HDMI to standard HDMI adapter can be used. Alternatively, a VGA type display monitor with a micro HDMI to VGA adapter or DVI-D adapter can be used. If you have an old TV with a composite video interface, then you can connect it to the Raspberry Pi 3.5mm port with a TRRS type connector. You may also consider purchasing additional parts, such as a case, CPU fan, and so on. The case is very useful as it protects your Raspberry Pi electronics. The working temperature of the CPU can go as high as 80 degrees Centigrade. Using a fan (see Figure 1.3) makes the CPU more efficient as it can lower its temperature by about 50%.

Figure 1.3 Raspberry Pi 4 CPU fan (www.seeedstudio.com)

1.3.1 Setup option 1

As shown in Figure 1.4, in this option various devices are connected directly to the Raspberry Pi 4. Depending on what type of display monitor we have, we can use an HDMI display, VGA monitor, DVI-D monitor, or TV. Notice that depending on the external USB devices used, you can use either the USB 2.0 or the USB 3.0 ports.

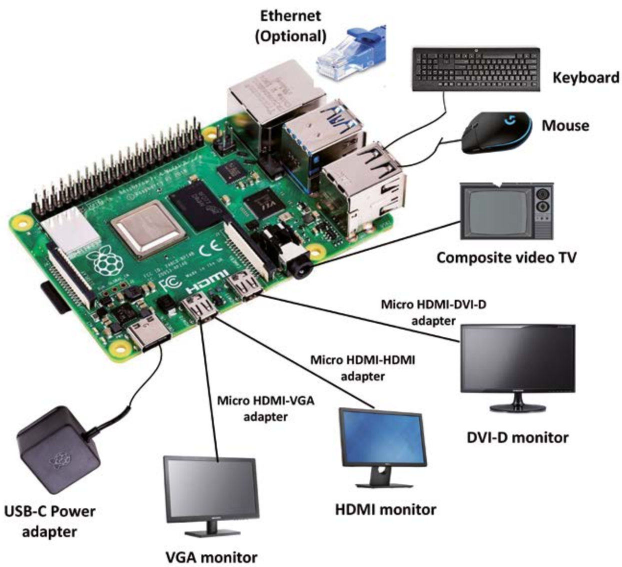


Figure 1.4 Raspberry Pi 4 setup - option 1

1.3.2 Setup option 2

In this option, shown in Figure 1.5, a powered hub is used to connect the USB devices.

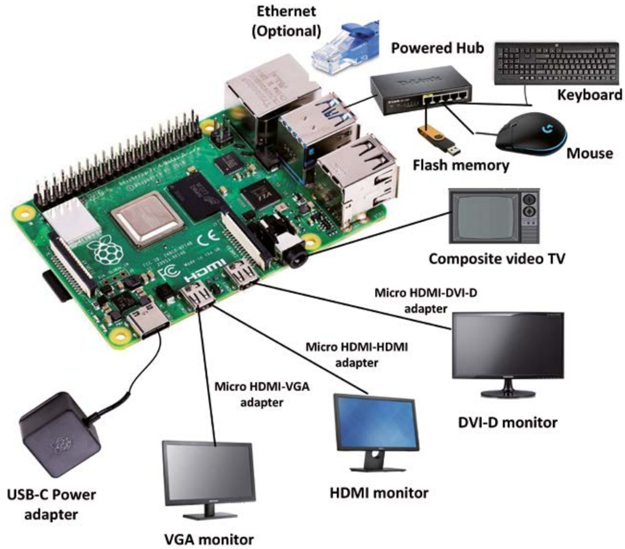


Figure 1.5 Raspberry Pi 4 setup – option 2

Summary

In this chapter, we have learned about the main parts and their functions of the Raspberry Pi 4 board. In addition, we have learned how to configure Raspberry Pi 4.

In the next chapter, we will be learning how to install the latest Raspberry Pi operating system, Raspbian Buster, on a new blank micro SD card.

Chapter 2 • Installing the Raspberry Pi operating system

2.1 Overview

In the last chapter, we had a look at some of the hardware features of Raspberry Pi 4 and learned how to set up the hardware using various external devices. In this chapter, we will be learning how to install the latest Raspberry Pi operating system, Raspbian Buster, on an SD card.

2.2 Raspbian Buster installation steps on Raspberry Pi 4

Raspbian Buster is the latest operating system of Raspberry Pi 4. This section gives the steps for installing this operating system on a new blank SD card, ready to use with your Raspberry Pi 4. You will need a micro SD card with a capacity of at least 8GB (16 GB is preferable) before installing the new operating system on it.

The steps to install Raspbian Buster are as follows:

- Download the Buster image to a folder on your PC (e.g. C:\RPIBuster) from the following link by clicking the Download ZIP under section **Raspbian Buster with desktop and recommended software** (see Figure 2.1). At the time of writing this book the file was called: **2019-07-10-raspbian-buster-full.img**. You may have to use the Windows 7Zip software to unzip the download since some of the features are not supported by older unzip software.

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

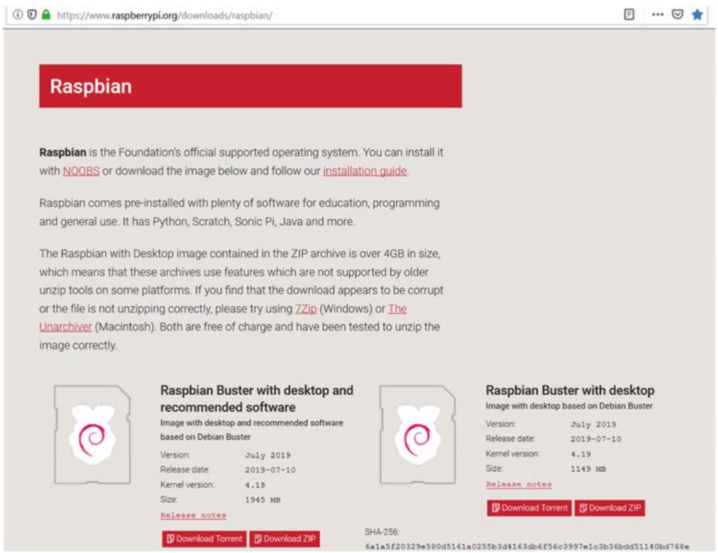


Figure 2.1 Raspbian Buster download page

- Put your blank micro SD card into the card slot on your computer. You may need to use an adapter to do this.

- Download the Etcher program to your PC to flash the disk image. The link is (see Figure 2.2):

  https://www.balena.io/etcher/

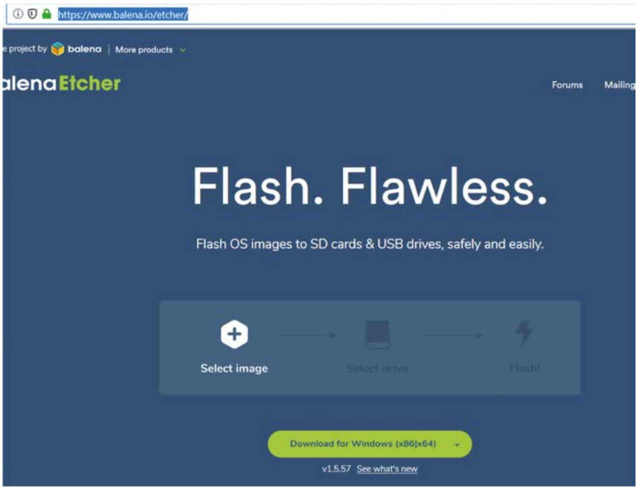


Figure 2.2 Download Etcher

- Double click to Open Etcher, and click **Select image**. Select the Raspbian Buster file you just downloaded.

- Click **Select target** and select the micro SD card.

- Click **Flash** (see Figure 2.3). This may take several minutes, wait until it is finished. The program will then validate and unmount the micro SD card. You can remove your micro SD card after it is unmounted.

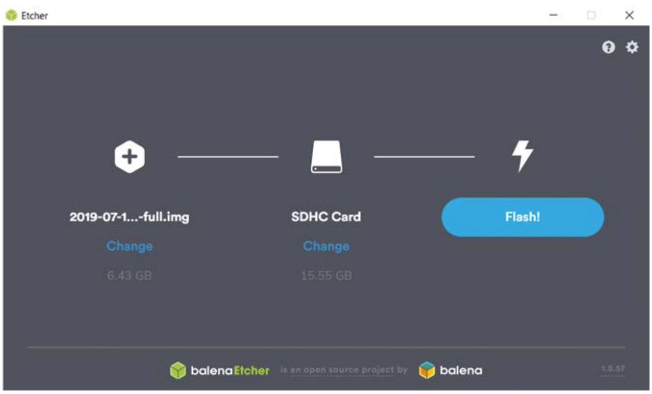


Figure 2.3 Click Flash to flash the disk image

- You are now ready to use your micro SD card on your Raspberry Pi 4.

- Connect your Raspberry Pi 4 to a HDMI monitor (you may need to use an adapter cable for mini HDMI to standard HDMI conversion), connect a USB keyboard, and power up the Raspberry Pi.
- You will see the startup menu displayed on the monitor. Click Next to get started.
- Select the Wi-Fi network and enter the password of your Wi-Fi router.
- Click on the Wi-Fi icon at the top right-hand side of the screen and note the Wireless IP address of your Raspberry Pi (notice that this IP address is not static and it can change next time you power-up your Raspberry Pi).
- You should now be ready to use your Raspberry Pi 4 (see Desktop in Figure 2.4).

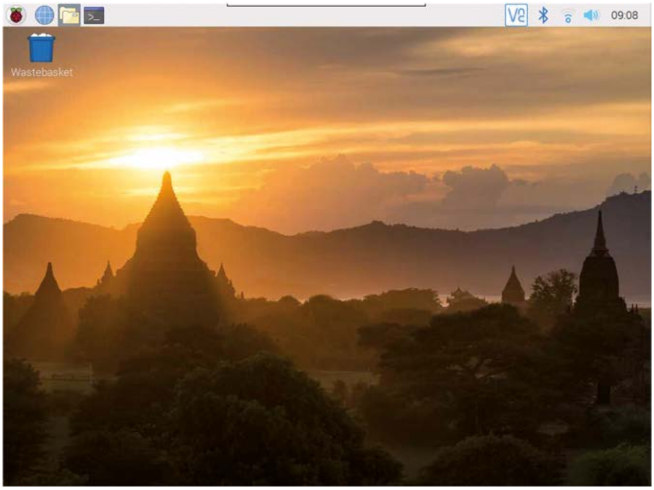


Figure 2.4 Raspberry Pi 4 desktop

Notice that the IP address of your Raspberry Pi can also be seen in your router. You can also get the IP address of your Raspberry Pi using your mobile phone. There are several programs free of charge that can be installed on your mobile phone that will show you the IP addresses of all the devices connected to your router. In this section, the use of the Android **apps** called **Who's On My Wi-Fi – Network Scanner** *by Magdalm* is used to show how the IP address of your Raspberry Pi can be displayed. Running this program will display the Raspberry Pi Wireless IP address under the heading **Raspberry Pi Trading Ltd**. In addition to the IP address, other parameters such as MAC address, gateway address, IP mask, etc are all displayed by this program.

2.3 Remote access

It is much easier to access the Raspberry Pi remotely over the Internet, for example using a PC rather than connecting a keyboard, mouse, and display to it. Before being able to access your Raspberry Pi remotely, we have to enable the SSH and the VNC by entering the following command at a terminal session:

```
pi$raspberrypi:~ $ sudo raspi-config
```

Go to the configuration menu and select **Interface Options**. Go down to **P2 SSH** (see Figure 2.5) and enable SSH. Click **<Finish>** to exit the menu.

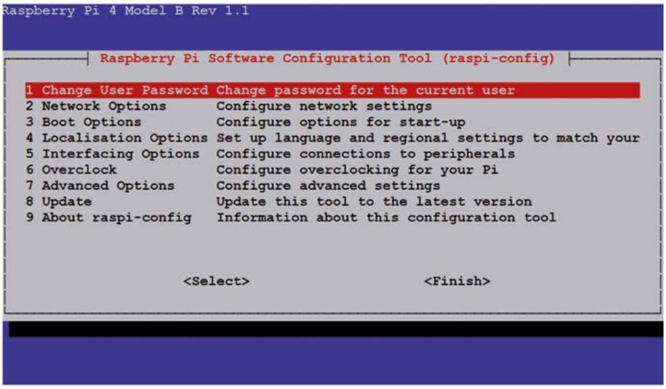


Figure 2.5 Enable SSH

You should also enable VNC so that your Raspberry Pi can be accessed graphically over the Internet. This can be done by entering the following command at a terminal session:

```
pi$raspberrypi:~ $ sudo raspi-config
```

Go to the configuration menu and select **Interface Options**. Go down to **P3 VNC** and enable VNC. Click **<Finish>** to exit the menu. At this stage, you may want to shut down your Raspberry Pi by clicking the **Applications Menu** on Desktop and selecting the **Shutdown** option.

2.4 Using Putty

Putty is a communications program that is used to create a connection between your PC and Raspberry Pi. This connection uses a secure protocol called SSH (Secure Shell). Putty doesn't need to be installed as it can just be stored in any folder of your choice and run from there.

Putty can be downloaded from the following web site:

https://www.putty.org/

Simply double click to run it and the Putty startup screen will be displayed. Click **SSH** and enter the Raspberry Pi IP address, then click **Open** (see Figure 2.6). The message shown in Figure 2.7 will be displayed the first time you access your Raspberry Pi. Click **Yes** to accept this security alert.

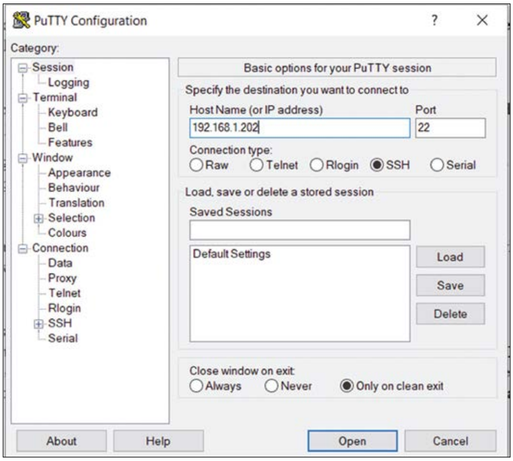


Figure 2.6 Putty startup screen

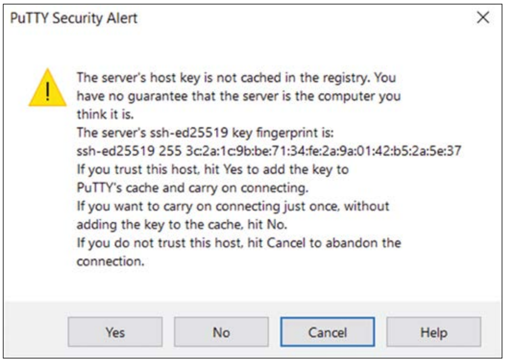


Figure 2.7 Click Yes to accept

You will be prompted to enter the username and password. Notice that the default user-name and password are:

username: **pi**
password: **raspberry**

You now have a terminal connection with the Raspberry Pi and you can type in commands, including sudo commands. You can use the cursor keys to scroll up and down through the commands you've previously entered in the same session. You can also run programs although not graphical programs.

2.4.1 Configuring Putty

By default, the Putty screen background is black with white foreground characters. In this book, we used a white background with black foreground characters, with the character size set to 12 points bold. The steps to configure Putty with these settings are given below. Notice that in this example the settings are saved with the name **RPI4** so that they can be recalled whenever Putty is re-started:

- Restart Putty
- Select **SSH** and enter the Raspberry Pi IP address
- Click **Colours** under **Window**
- Set the **Default Foreground** and **Default Bold Foreground** colours to black (Red:0, Green:0, Blue:0)
- Set the **Default Background** and **Default Bold Background** to white (Red:255, Green:255, Blue:255)
- Set the **Cursor Text** and **Cursor Colour** to black (Red:0, Green:0, Blue:0)
- Select **Appearance** under **Window** and click **Change** in **Font settings**. Set the font to **Bold 12**.
- Select **Session** and give a name to the session (e.g. RPI4) and click **Save**.
- Click **Open** to open the Putty session with the saved configuration
- Next time you re-start Putty, select the saved session and click **Load** followed by **Open** to start a session with the saved configuration

2.5 Remote access of the desktop

You can control your Raspberry Pi via Putty, and run programs on it from your Windows PC. This, however, will not work with graphical programs because Windows doesn't know how to represent the display. As a result of this, for example, we cannot run any graphical programs in the Desktop mode. We can get around this problem using some extra software. Two popular programs used for this purpose are VNC (Virtual Network Connection), and Xming. Here, we will be learning how to use VNC.

Installing and using VNC

VNC consists of two parts: VNC Server and VNC Viewer. VNC Server runs on the Raspberry Pi, and VNC Viewer runs on the PC. VNC server is already installed on your Raspberry Pi. You can start the server by entering the following command in the command mode:

```
pi$raspberrypi:~ $ vncserver :1
```

The steps to install and use VNC Viewer on your PC are given below:

- There are many VNC Viewers available, but the recommended one is TightVNC which can be downloaded from the following web site:

  https://www.tightvnc.com/download.php

- Download and install **TightVNC** software for your PC. You will have to choose a password during the installation.

- Start **TightVNC Viewer** on your PC and enter the Raspberry Pi IP address (see Figure 2.8) followed by :1. Click **Connect** to connect to your Raspberry Pi.

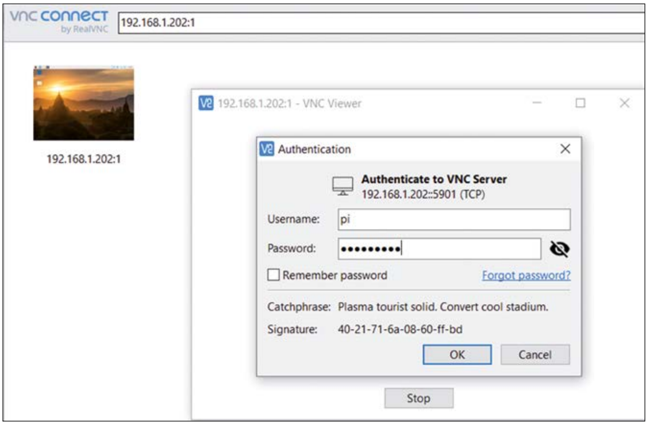


Figure 2.8 Start the TightVNC and enter the IP address

Figure 2.9 shows the Raspberry Pi Desktop displayed on the PC screen.

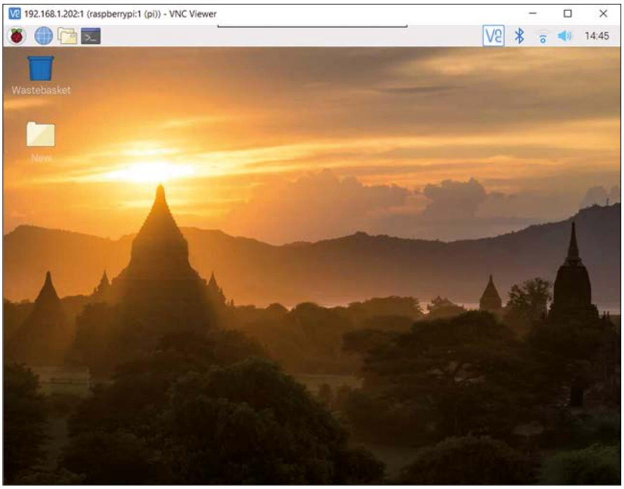


Figure 2.9 Raspberry Pi Desktop on the PC screen]

2.6 Summary

In this chapter, we have learned how to install the latest Raspberry Pi operating system on an SD card. Now that the software has been installed and our Raspberry Pi is working, in the next chapter, we will look at some of the important commands of the Raspberry Pi operating system.

Chapter 3 • Using the command line

3.1 Overview

Raspberry Pi operating system is based on a version of the Linux operating system. Linux is one of the most popular operating systems in use today. Linux is very similar to other operating systems, such as Windows and Unix. Linux is an open operating system based on Unix and has been developed collaboratively by many companies and universities since 1991. In general, Linux is harder to manage than some other operating systems like Windows but offers more flexibility and wider configuration options. There are several popular versions of the Linux operating system such as Debian, Ubuntu, Red Hat, Fedora and so on. In this chapter we shall be looking at some of the commonly used Raspberry Pi commands that we can enter from the command line. The commands entered by the user are shown in bold for clarity.

3.2 The Raspberry Pi directory structure

The Raspberry Pi directory structure consists of a single root directory, with directories and subdirectories under the root. Different operating system programs, applications, and user data are stored in different directories and subdirectories.

The root directory is identified by the **"/"** symbol. Under the root, we have directories named such as **bin**, **boot**, **dev**, **etc**, **home**, **lib**, **lost+found**, **media**, **mnt**, **opt**, **proc**, and many more. The important directory as far as the users are concerned is the **home** directory which contains subdirectories for each user of the system. The full path to the home directory is **/home/pi**. We can move to our home directory from any other directory by entering the command **cd ~**

Some useful directory commands are given below. Command **pwd** displays the user home directory:

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

To show the directory structure, enter the command **ls /** as shown in Figure 3.1.

```
pi@raspberrypi:~ $ ls /
bin   dev  home  lost+found  mnt  proc  run   srv  tmp  var
boot  etc  lib   media       opt  root  sbin  sys  usr
pi@raspberrypi:~ $
```

Figure 3.1 List the directory structure

To show the subdirectories and files in our home directory, enter **ls** as shown in Figure 3.2.

```
pi@raspberrypi:~ $ ls
Desktop    Downloads  Music     Public     Videos
Documents  MagPi      Pictures  Templates
pi@raspberrypi:~ $
```

Figure 3.2 List of files in our home directory

The ls command can take a number of arguments. Some examples are given below:
To display the subdirectories and files in a single row:

```
pi@raspberrypi:~ $ ls -1
```

To display the file type, enter the following command. Note that directories have a "/" after their names, executable files have a "*" character after their names:

```
pi@raspberrypi:~ $ ls -F
```

To list the results separated by commas:

```
pi@raspberrypi:~ $ ls -m
```

We can mix the arguments as in the following example:

```
pi@raspberrypi:~ $ ls -m -F
```

Subdirectories are created using command **mkdir** followed by the name of the subdirectory. In the following example, subdirectory **myfiles** is created in our working directory (see Figure 3.3).

```
pi@raspberrypi:~ $ mkdir myfiles
pi@raspberrypi:~ $ ls
Desktop    Downloads  Music    Pictures  Templates
Documents  MagPi      myfiles  Public    Videos
pi@raspberrypi:~ $
```

Figure 3.3 Creating a subdirectory

Use command **rmdir** followed by the subdirectory name to remove a subdirectory.

3.3 File permissions

One of the important arguments used with the **ls** command is **"-l"** (lower case letter l) which displays the file permissions, file sizes, and when they were last modified. In the example in Figure 3.4, each line relates to one directory or file. Reading from right to left, the name of the directory or the file is on the right-hand side. The date the directory or the file was created is on the left-hand side of its name. Next comes the size in bytes. The characters at the beginning of each line are about the permissions. i.e. who is allowed to use or modify the file or the subdirectory.

```
pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Desktop
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Documents
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Downloads
drwxr-xr-x 2 pi pi 4096 Jun 20 17:55 MagPi
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Music
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Pictures
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Public
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Templates
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Videos
pi@raspberrypi:~ $
```

Figure 3.4 File permissions example

The permissions are divided into 3 categories:

- What the user (or owner, or creator) can do – called USER
- What the group owner (people in the same group) can do – called GROUP
- What everyone else can do – called WORLD

The first **pi** in the example shows who the user of the file (or subdirectory) is, and the second word **pi** shows the group name that owns the file (or subdirectory). In this example, both the user and the group names are **pi**.

The permissions can be analysed by breaking down the characters into four chunks for File type, User, Group, World. The first character for a file is "-", and for a directory, it is "d". Next comes the permissions for the User, Group, and World. The permissions are as follows:

- **Read permission (r):** the permission to open and read a file or to list a directory
- **Write permission (w):** the permission to modify a file, or to delete or create a file in a directory
- **Execute permission (x):** the permission to execute the file (applies to executable files), or to enter a directory

The three letters **rwx** are used as a group and if there is no permission assigned then a "-" character is used.

As an example, considering the **Music** subdirectory, we have the following permission codes:

drwxr-xr-x which translates to:

d:	it is a directory
rwx:	user (owner) can read, write, and execute
r-x:	group can read and execute, but cannot write (e.g. create or delete)
r-x:	world (everyone else) can read and execute, but cannot write

The **chmod** command is used to change the file permissions. Before going into details of how to change the permissions, let us look and see what arguments are available in **chmod** for changing the file permissions.

The available arguments for changing file permissions are given below. We can use these arguments to add/remove permissions or to explicitly set permissions. It is important to realize that if we explicitly set permissions then any unspecified permissions in the command will be revoked:

u:	user (or owner)
g:	group
o:	other (world)
a:	all
+:	add
-:	remove
=:	set
r:	read
w:	write
x:	execute

To change the permissions of a file we type the **chmod** command, followed by one of the letters **u**, **g**, **o**, or **a** to select the people, followed by the **+ -** or **=** to select the type of change, and finally followed by the filename. An example is given below. In this example, subdirectory **Music** has the user read and write permissions. We will be changing the permissions so that the user does not have read permission on this file:

```
pi@raspberrypi ~$ chmod -u -r Music
pi@raspberrypi ~$ ls -l
```

The result is shown in Figure 3.5.

```
pi@raspberrypi:~ $ chmod -u -r Music
pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Desktop
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Documents
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Downloads
drwxr-xr-x 2 pi pi 4096 Jun 20 17:55 MagPi
d--------- 2 pi pi 4096 Jun 20 18:20 Music
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Pictures
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Public
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Templates
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Videos
pi@raspberrypi:~ $
```

Figure 3.5 File permissions of subdirectory Music

In the following example, **rwx** user permissions are given to subdirectory **Music**:

```
pi@raspberrypi ~$ chmod u+rwx Music
```

Figure 3.6 shows the new permissions of subdirectory Music.

```
pi@raspberrypi:~ $ chmod u+rwx Music
pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Desktop
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Documents
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Downloads
drwxr-xr-x 2 pi pi 4096 Jun 20 17:55 MagPi
drwx------ 2 pi pi 4096 Jun 20 18:20 Music
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Pictures
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Public
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Templates
drwxr-xr-x 2 pi pi 4096 Jun 20 18:20 Videos
pi@raspberrypi:~ $
```

Figure 3.6 New permissions of subdirectory Music

To change our working directory the command **cd** is used. In the following example we change our working directory to **Music**:

```
pi@raspberrypi ~$ cd /home/pi/Music
pi@raspberrypi ~/Music $
```

to go up one directory level, i.e. to our default working directory:

```
pi@raspberrypi ~/Music $ cd..

pi@raspberrypi ~$
```

to change our working directory to **Music**, we can also enter the command:

```
pi@raspberrypi ~$ cd ~/Music
pi@raspberrypi ~/myfiles $
```

to go back to the default working directory, we can enter:

```
pi@raspberrypi ~/Music $ cd ~
pi@raspberrypi ~$
```

to find out more information about a file we can use the file command. For example:

```
pi@raspberrypi ~$ file Music
Music: directory
pi@raspberrypi ~$
```