elektor books

# Raspberry Pi 5 Essentials

## Program, build, and master over 60 projects with Python

TRIED & TESTED

Dogan Ibrahim

elektor
design > share > earn

# Raspberry Pi 5 Essentials

## Program, build, and master over 60 projects with Python

**Dogan Ibrahim**

elektor
design > share > earn

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444



**Declaration**

The author, editor, and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause. All the programs given in the book are Copyright of the Author and Elektor International Media. These programs may only be used for educational purposes. Written permission from the Author or Elektor must be obtained before any of these programs can be used for commercial purposes.

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

**ISBN 978-3-89576-586-5** Print
**ISBN 978-3-89576-587-2** eBook


Editor: Clemens Valens
Prepress Production: D-Vision, Julian van den Berg
Print: Ipskamp Printing, Enschede (NL)

# Contents

# Preface

The Raspberry Pi 5 is a credit-card-sized computer from Raspberry Pi that can be used in many applications, such as in audio and video media centers, as a desktop computer, in industrial controllers, robotics, and in many domestic and commercial applications. In addition to the many features found in other Raspberry Pi computers, the Raspberry Pi 5 offers Wi-Fi and Bluetooth 5.0 (with BLE support), which makes it highly desirable in remote and Internet-based control and monitoring applications.

The Raspberry Pi 5 is based on a 64-bit quad-core ARM Cortex-A76 processor running at 2.4 GHz. This implies a performance boost of two to three times compared to the Raspberry Pi 4. Raspberry Pi 5 comes with an enhanced graphic performance, using an 800 MHz VideoCore VII graphics chip. Additionally, the Raspberry Pi 5 features the RP1 southbridge chip made by Raspberry Pi. With the help of this RP1 southbridge, Raspberry Pi 5 delivers higher performance and more functionality for peripheral devices. It should now be possible to carry out many real-time operations such as audio digital signal processing, real-time digital control and monitoring, and many other real-time operations using this tiny powerhouse.

This book is about the Raspberry Pi 5 computer and its use in various control and monitoring applications. The book explains in simple terms and with many tested and working example projects how to configure the Raspberry Pi 5 computer, how to use the latest operating system (Bookworm), and how to write application programs using the popular Python programming language.

The book starts with an introduction to the Raspberry Pi 5 computer and covers the important topics of accessing the computer locally and remotely. Use of the console command language as well as accessing and using the desktop GUI have been described with working examples.

The remaining parts of the book cover many Raspberry-Pi-5-based hardware projects using components and devices such as LEDs, buzzers, LCDs, ultrasonic sensors, temperature sensors, Sense HAT, camera modules, etc. Example projects are given using Wi-Fi and Bluetooth modules to send and receive data from smartphones, from the PC, and sending real-time temperature and atmospheric pressure data to the cloud.

All the projects presented in the book have been tested and are working. Complete circuit diagrams and full program listings are given for each project, with detailed descriptions of the operation of each project. The following subheadings are used in every project wherever necessary:

- Project title
- Project description
- Block diagram
- Circuit diagram
- Program listing
- Suggestions for future work

I hope the readers find the book helpful and enjoy reading it, and use a Raspberry Pi 5 in their next new projects.

*Prof Dr. Dogan Ibrahim*
*London*

# Chapter 1 • The Raspberry Pi 5

## 1.1 Overview

The Raspberry Pi 5 is the latest credit card size computer from Raspberry Pi. In this chapter, we will look at the specifications of this new computer and compare it with the Raspberry Pi 4.

## 1.2 The Raspberry Pi 5

Raspberry Pi 4 was released in June 2019. There has been a long wait for a newer model and finally the Raspberry Pi 5 was launched in October 2023.

The Raspberry Pi 5 is claimed to have two or three times the processing power of The Raspberry Pi 4, which is already a very popular single board computer. The Raspberry Pi 5 is currently available in 4 GB and 8 GB memory capacities, but smaller memory devices may appear later. Although the Raspberry Pi 5 is the same size and shape as the Model 4B, it has a number of interesting new features such as PCIe connector, power button, built-in real-time clock and some others that we will investigate in this chapter.

The Raspberry Pi 5 is based on a 2.4 GHz Cortex-A76 ARM processor with a new south-bridge for handling the peripheral interface. A new VideoCore VII GPU is provided with 800 MHz speed. The dual camera interface is another nice feature of the Raspberry Pi 5. The microSD card interface now supports cards that work at much higher speeds.

Table 1.1 shows a comparison of the Raspberry Pi 4 and 5. Notice that both devices have dual 2 × 4kp60 HDMI display interfaces, although Pi 5 supports HDR output. The 2 × 20 pin GPIO interface is the same in both devices. The Raspberry Pi 5 additionally has two camera interfaces, a PCIe bus connector, a UART interface, an RTC clock power connector, and a fan power connector. Wi-Fi and Bluetooth are supported by both devices. The on-board power switch on Pi 5 is a useful addition and was requested by many users. Pi 5 is powered from 5 V/4 A USB-C type power supply, where Pi 4 is powered from a 3 A power supply. Pi 5 is slightly more expensive than Pi 4.

| | Raspberry Pi 4 | Raspberry Pi 5 |
|---|---|---|
| SoC | BCM2711 SoC<br>Cortex-A72 CPU at 1.8 GHz | BCM2712 SoC<br>Cortex-A76 CPU at 2.4 GHz |
| CPU | 4 core | 4 core |
| Instruction set | ARMv8-A | ARMv8-2 |
| Display | 500 MHz VideoCore Vi GPU | 800 MHz VideoCore VII GPU |
| L2 Cache | 1 MB (shared) | 2 MB |
| L3 Cache | None | 2 MB (shared) |
| RAM | 1, 2, 4, 8 GB LPDDR4 | 4, 8 GB LPDDR4X |
| SD Card | microSD | microSD (high speed SDR104 compatible) |
| GPIO | 2 × 20 pin | 2 × 20 pin |

| | | |
|---|---|---|
| USB ports | 2× USB2<br>2× USB3 | 2× USB2<br>2× USB3 |
| Networking | Gigabit Ethernet port | Gigabit Ethernet port |
| Connectors | 2-lane MIPI display port<br>2-lane MIPI CSI camera port<br>4-pole stereo audio and composite video port | 2× MIPI camera<br>2× 4-lane MIPI camera/display<br>PCIe 2.0 interface<br>UART port<br>RTC clock power port<br>Fan power port |
| Wi-Fi/Bluetooth | 802.11ac, Bluetooth 5/BLE | 802.11ac, Bluetooth 5/BLE |
| Power button | None | Yes |
| Power | 5 V, 3 A USB-C | 5 V, 4 A USB-C |
| Size | 85 × 56 mm | 85 × 56 mm |

*Table 1.1 Comparison of Raspberry Pi 5 and Raspberry Pi 4*

There are two micro-HDMI based monitor ports on both devices, with both having the same specifications.

The Ethernet port and USB ports are swapped. As a result of this, the Raspberry Pi 4 case is incompatible with the Pi 5 and a new case is required.

The camera and display connectors on the Raspberry Pi 5 are 15-pin and smaller, instead of the original 22-pin connector used on Pi 4. A ribbon cable with 22-pin on one side and 15-pin on the other side is required to connect an existing Raspberry Pi 4 camera to the Raspberry Pi 5. The Raspberry Pi 5 has two connectors, allowing two cameras or DSI displays (or a mix of either) to be connected. The PCIe connector is for fast external PCIe compatible peripherals, such as SSDs.

The new power button on the Raspberry Pi 5 could be very useful. When the device is On, pressing the button brings the shutdown (logout) menu. A safe shutdown will occur with another press of the power button.

Figure 1.1 shows the front view of the Raspberry Pi 5 with the components labelled for reference.

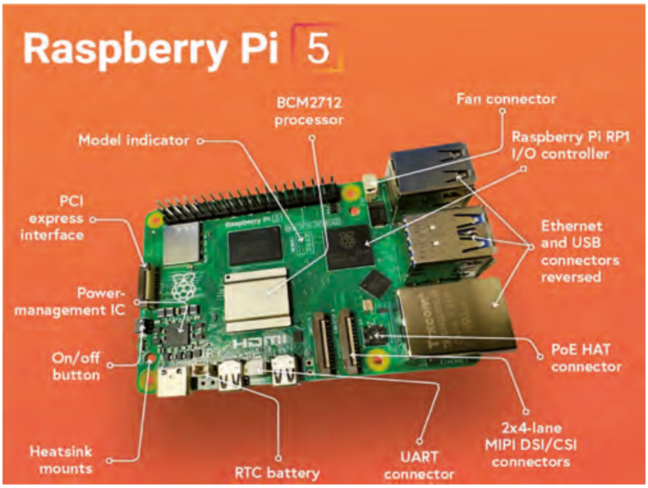



*Figure 1.1 Raspberry Pi 5*

The Raspberry Pi 5 gets rather hot, and it is recommended to use a cooler to lower the CPU temperature. Although the idle CPU temperature is around 50°C, it can go higher than 85°C under a stress test. An active cooler is available for the Raspberry Pi 5. Holes and power points are provided on the board to install and power the active cooler. Figure 1.2 shows the Raspberry Pi 5 with the active cooler installed. The active cooler cools down the SoC, RAM, and the southbridge chip. When the CPU is idle, the active cooler keeps the CPU temperature at around 40°C. The fan of the cooler operates automatically when the CPU temperature goes just above 50°C.

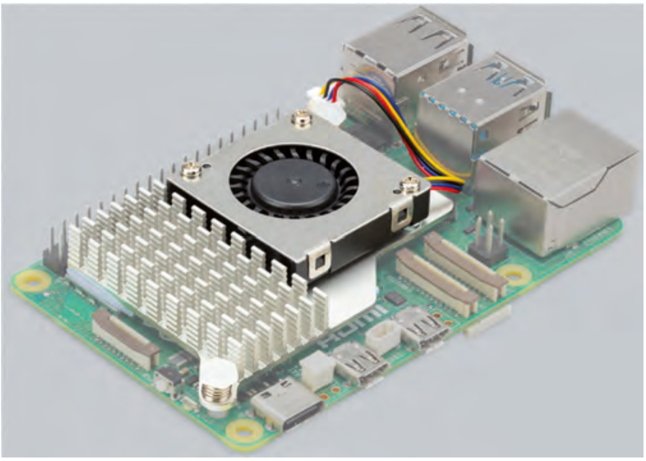


*Figure 1.2 The Raspberry Pi 5 with active cooler*

The Raspberry Pi 5 operating system (OS) is based upon Debian 12 with the code name **Bookworm**. This OS, released in July 2023, comes with a new Python interpreter (Python 3.11). This means that a Python package cannot be installed using the **pip** commands.

Another major software change is that the RPi.GPIO library (created by Ben Croston) was not available at the time of writing this book. As a result of this, all the GPIO-based Python programs in the book have been developed using the **gpiozero** library. Most third party HATs are based on RPi.GPIO and these will not work until their software is changed by their manufacturers. It is hoped that the manufacturers will change their software by the time Raspberry Pi 5 becomes officially widely available.

# Chapter 2 • Installing the Raspberry Pi 5 Operating System

## 2.1 Overview

The Raspberry Pi 5 operating system **Bookworm** is available either on a pre-installed microSD card, or you can download the operating system image on a blank microSD card. In this chapter, you will learn to install the operating system using both methods.

## 2.2 Using a pre-installed SD card

The pre-installed Raspberry Pi operating system is available on various sized microSD cards. In this section, the author used the pre-installed 32 GB microSD card supplied by Elektor. Additionally, the author used a 7-inch HDMI compatible monitor, a Raspberry Pi official keyboard, and a mouse. The author's hardware setup between the Raspberry Pi 5 and various devices is shown in Figure 2.1.

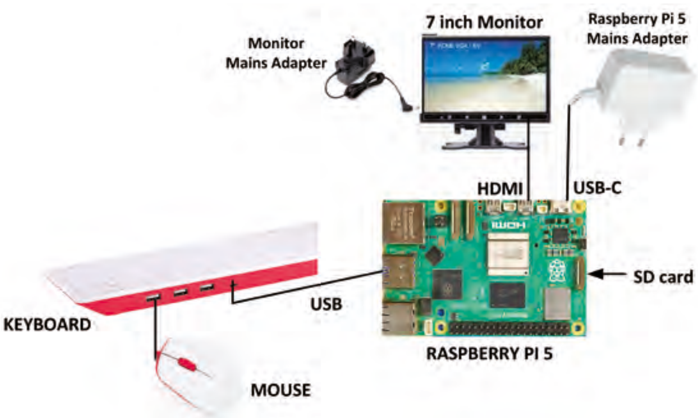



*Figure 2.1 The author's hardware setup*

The steps are as follows:

- Insert the pre-installed microSD card into your Raspberry Pi 5
- Connect all the devices as in Figure 2.1
- Connect the Raspberry Pi power adapter to the mains supply
- You should see the Raspberry Pi booting the first time and asking you various questions to set up the device, such as the username, password, Wi-Fi network name and password, any updates if necessary, etc. (see Figure 2.2 for some displays on the monitor). In this book, the username is set to **pi**.
- The Raspberry Pi will boot in Desktop mode and will display the default screen. You can press Ctrl+Alt+F1 at any time to change to the Console mode

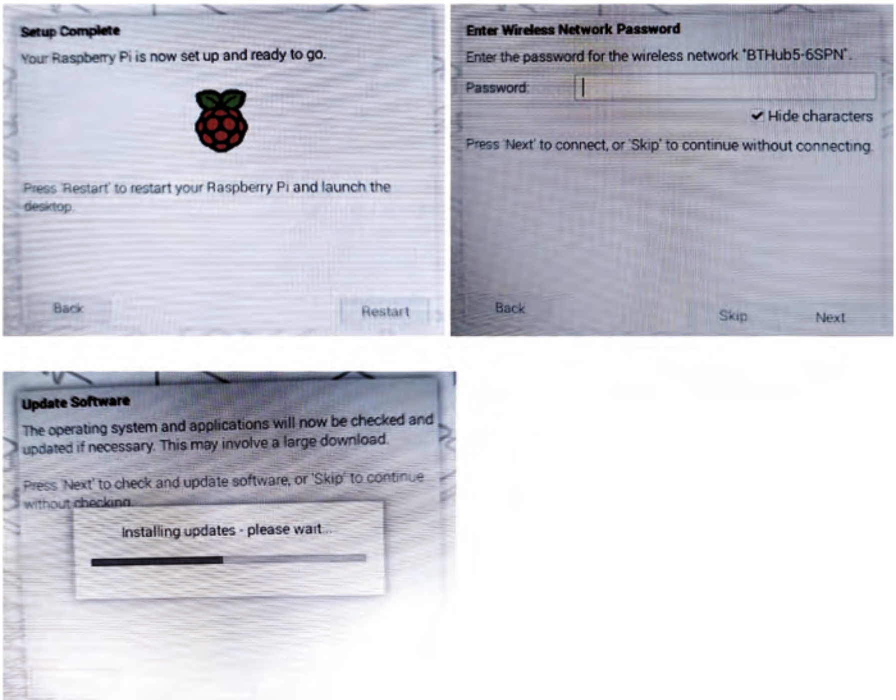



*Figure 2.2 Raspberry Pi 5 booting for the first time.*

## 2.3 Larger font in Console mode

It is probably hard to see the characters on a 7-inch monitor in console mode. You can follow the steps below to increase the font size:

- Make sure you are in the Console mode
- Enter the following command:

  pi@raspberrypi: ~ $ **sudo dpkg-reconfigure console-setup**

- Select **UTF-8** in the **Package Configuration** screen (Figure 2.3)

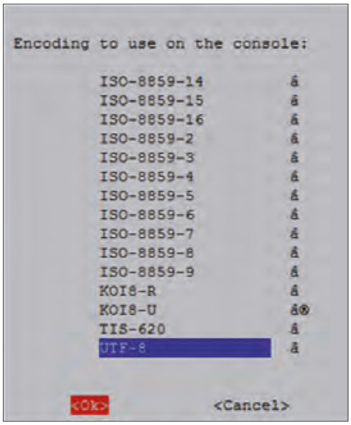



*Figure 2.3 Select UTF-8*

- Select **Guess optimal character set** (Figure 2.4)

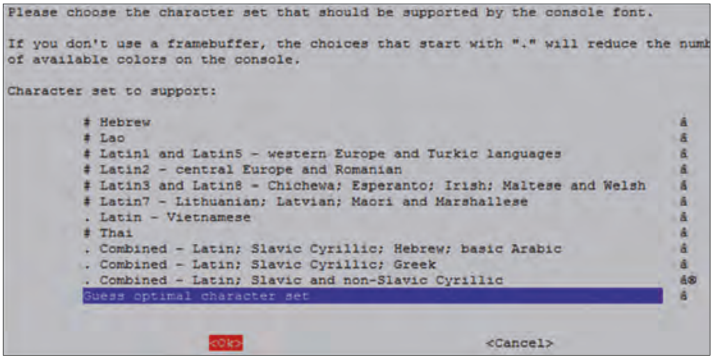



*Figure 2.4 Select Guess optimal character set*

- Select **Terminus** (Figure 2.5)

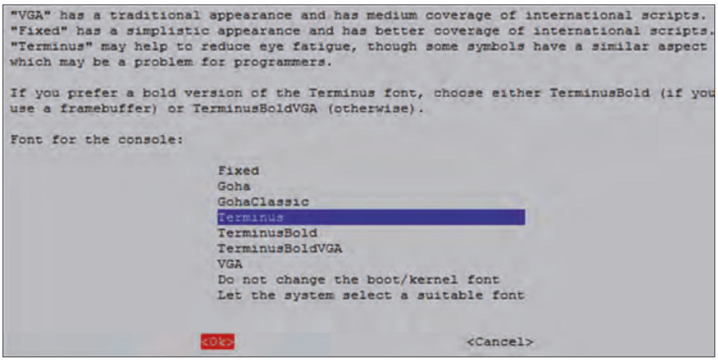



*Figure 2.5 Select Terminus*

- Select font **16x32** (Figure 2.6)

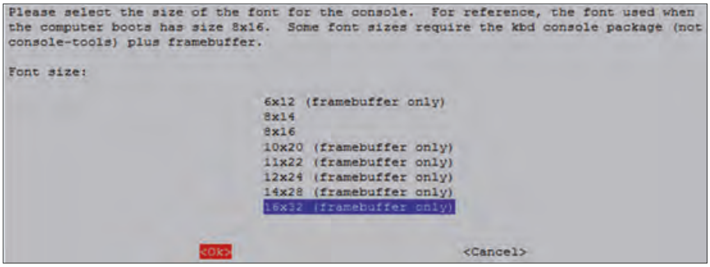



*Figure 2.6 Select font 16x32*

## 2.4 Accessing your Raspberry Pi 5 Console from your PC – the Putty program

In many applications, you may want to access your Raspberry Pi 5 from your PC. This requires enabling the SSH on your Raspberry Pi and then using a terminal emulation software on your PC. The steps to enable the SSH are as follows:

- Make sure you are in Console mode
- Type: **sudo raspi-config**
- Move down to **Interface Options**
- Highlight **SSH** and press Enter (Figure 2.7)

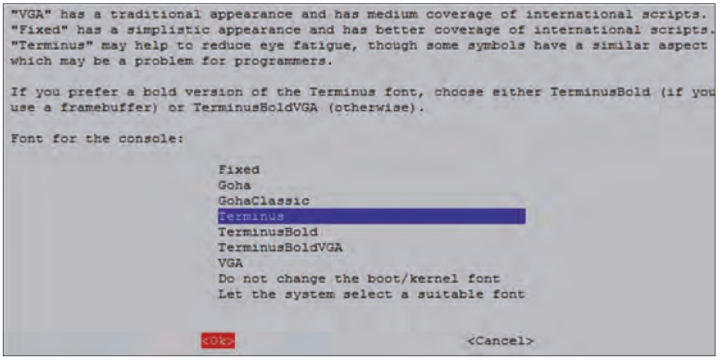



*Figure 2.7 Highlight SSH*

- Click **Yes** to enable SSH
- Click **OK**
- Move down and click **Finish**

You will now have to install a terminal emulation software on your PC. The one used by the author is the popular Putty. Download Putty from the following website:

https://www.putty.org

- Putty is a standalone program and there is no need to install it. Simply double click to run it. You should see the Putty startup screen as in Figure 2.8.

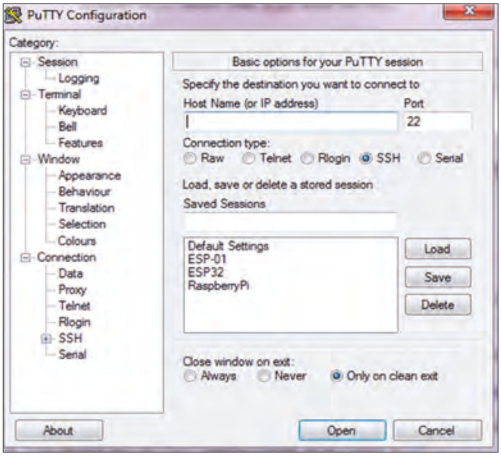
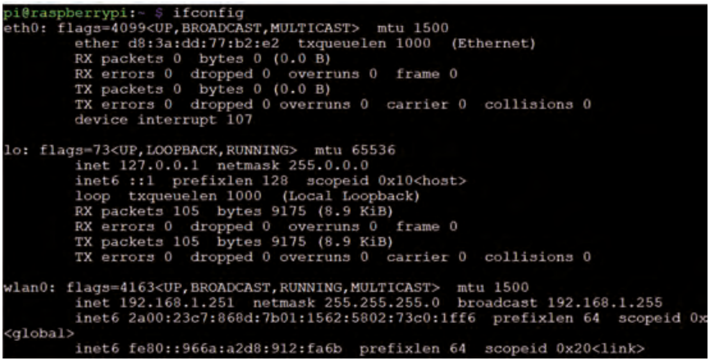



*Figure 2.8 Putty startup screen*

- Make sure that the Connection type is SSH and enter the IP address of your Raspberry Pi 5. You can obtain the IP address by entering the command **ifconfig** in console mode (Figure 2.9). In this example, the IP address was: **192.168.1.251** (see under **wlan0:**)



```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d8:3a:dd:77:b2:e2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 107

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 105  bytes 9175 (8.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 105  bytes 9175 (8.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.251  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2a00:23c7:868d:7b01:1562:5802:73c0:1ff6  prefixlen 64  scopeid 0x
<global>
        inet6 fe80::966a:a2d8:912:fa6b  prefixlen 64  scopeid 0x20<link>
```

*Figure 2.9 Command ifconfig*

- Click **Open** in Putty after entering the IP address and selecting **SSH**

- The first time you run Putty, you may get a security message. Click **Yes** to accept this security alert.

- You will then be prompted to enter the Raspberry Pi 5 username and password. You can now enter all Console-based commands through your PC.

- To change your password, enter the following command:

    pi@raspberrypi: ~ $ **passwd**

- To restart the Raspberry Pi, enter the following command:

    pi@raspberrypi: ~ $ **sudo reboot**

- To shut down the Raspberry Pi, enter the following command. Never shutdown by pulling the power cable, as this may result in the corruption or loss of files:

    pi@raspberrypi: ~ $ **sudo shutdown –h now**

### 2.4.1 Configuring Putty

By default, the **Putty** screen background is black with white characters. The author prefers a white background with black characters, with the character size set to 12 points bold. You should save your settings so that they are available next time you want to use Putty. The steps to configure Putty with these settings are given below:

- 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. MyZero) and click **Save**.
- Click **Open** to open **Putty** session with the saved configuration
- Next time you restart **Putty**, select the saved session and click **Load** followed by **Open** to start a session with the saved configuration

## 2.5 Accessing the Desktop GUI from your PC

If you are using your Raspberry Pi 5 with local keyboard, mouse, and display, you can skip this section. If, on the other hand, you want to access your Desktop remotely over the network, you will find that SSH services cannot be used. The easiest and simplest way to access your Desktop remotely from a computer is by using the VNC (Virtual Network Connection) client and server. The VNC server runs on your Pi and the VNC client runs on your computer. It is recommended to use the **tightvncserver** on your Raspberry Pi 5. The steps are:

- Enter the following command:

  pi$raspberrypi:~ $ **sudo apt-get install tightvncserver**

- Run the **tightvncserver**:

  pi$raspberrypi:~ $ **tightvncserver**

  You will be prompted to create a password for remotely accessing the Raspberry Pi desktop. You can also set up an optional read-only password. The password should be entered every time you want to access the Desktop. Enter a password and remember your password.

- Start the VNC server after reboot by the following command:

  pi$raspberrypi:~ $ **vncserver :1**

  You can optionally specify screen pixel size and colour depth in bits as follows:

  pi$raspberrypi:~ $ **vncserver :1 –geometry 1920x1080 –depth 24**

- We must now set up a VNC viewer on our laptop (or desktop) PC. There are many VNC clients available, but the recommended one which is compatible with **TightVNC** is **TightVNC** for the PC, which can be downloaded from the following link:

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

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

- Start the **TightVNC Viewer** on your PC and enter the Raspberry Pi IP address followed by ':1'. Click **Connect** to connect to your Raspberry Pi (Figure 2.10)

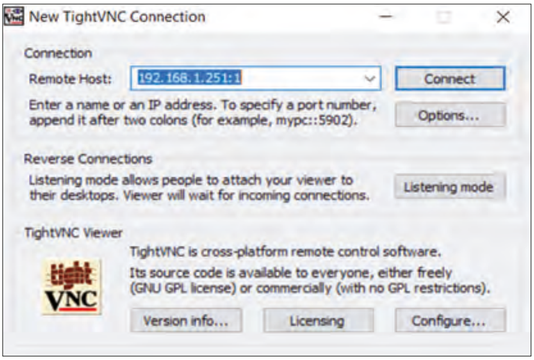



*Figure 2.10 Connect to TightVNC Viewer*

- Enter the password you have chosen earlier. You should now see the Raspberry Pi 5 Desktop displayed on your PC screen (Figure 2.11)

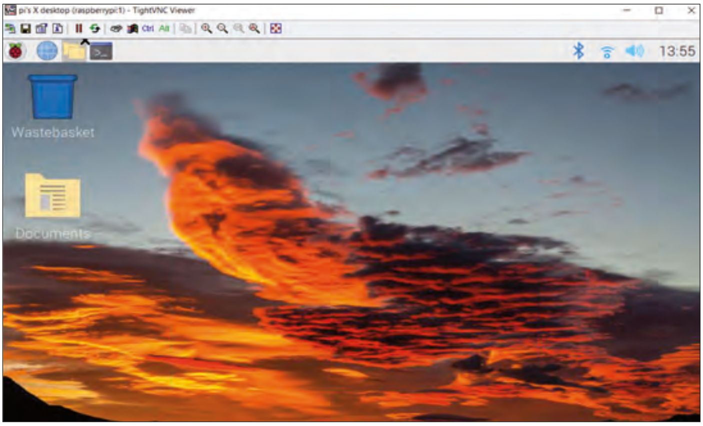



*Figure 2.11 Raspberry Pi 5 Desktop*

- The VNC server is now running on your Raspberry Pi 5 and you have access to the Desktop GUI.

## 2.6 Assigning a static IP address to your Raspberry Pi 5

When you try to access your Raspberry Pi 5 remotely over your local network, it is possible that the IP address given by your Wi-Fi router can change from time to time. This is annoying as you have to find out the new IP address allocated to your Raspberry Pi. Without knowledge of the IP address, you cannot log in using SSH or VNC.

In this section, you will learn how to fix your IP address so that it does not change between reboots. The steps are as follows:

- Log in to your Raspberry Pi 5 via Putty

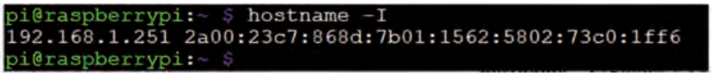
- Check whether DHCP is active on your Raspberry Pi (it should normally be active):

  pi@raspberrypi:~ $ **sudo service dhcpcd status**

  If DHCP is not active, activate it by entering the following commands:

  pi@raspberrypi:~ $ **sudo service dhcpcd start**
  pi@raspberrypi:~ $ **sudo systemctl enable dhcpcd**

- Find the IP address currently allocated to you by entering the command **ifconfig** or **hostname – I** (Figure 2.12). In this example, the IP address was: 192.168.1.251. We can use this IP address as our fixed address, since no other device on the network is currently using it.



```
pi@raspberrypi:~ $ hostname -I
192.168.1.251 2a00:23c7:868d:7b01:1562:5802:73c0:1ff6
pi@raspberrypi:~ $
```

*Figure 2.12 Find the IP address using the command hostname -I*

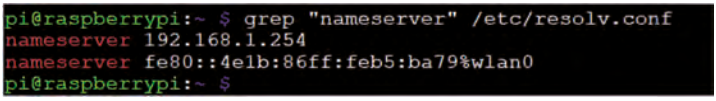
- Find the IP address of your router by entering the command **ip r** (Figure 2.13). In this example, the IP address was: 192.168.1.254

```
pi@raspberrypi:~ $ ip r
default via 192.168.1.254 dev wlan0 proto dhcp src 192.168.1.251 metric 600
192.168.1.0/24 dev wlan0 proto kernel scope link src 192.168.1.251 metric 600
pi@raspberrypi:~ $
```

*Figure 2.13 Find the IP address of your router.*

- Find the IP address of your DNS by entering the following command (Figure 2.14). This is usually the same as your router address:

  pi@raspberrypi:~ $ **grep "nameserver" /etc/resolv.conf**



```
pi@raspberrypi:~ $ grep "nameserver" /etc/resolv.conf
nameserver 192.168.1.254
nameserver fe80::4e1b:86ff:feb5:ba79%wlan0
pi@raspberrypi:~ $
```

*Figure 2.14 Find the DNS address.*

- Edit file **/etc/dhcpcd.conf** by entering the command:

  pi@raspberrypi:~ $ **nano /etc/dhcpcd.conf**

- Add the following lines to the bottom of the file (these will be different for your router). If these lines already exist, remove the comment character '#' at the beginning of the lines and change the lines as follows (you may notice that **eth0** for Ethernet is listed):

```
interface wlan0
static_routers=192.168.1.254
static domain_name_servers=192.168.1.254
          static ip_address=192.168.1.251/24
```

- Save the file by entering **CTRL + X** followed by **Y** and reboot your Raspberry Pi

- In this example, the Raspberry Pi should reboot with the static IP address: 192.168.1.251

## 2.7 Enabling Bluetooth

In this section, you will see how to enable the Bluetooth on your Raspberry Pi 5 so that it can communicate with other Bluetooth devices. The steps are given below:

- Enable the Bluetooth on your other device

- Click on the Bluetooth icon on your Raspberry Pi 5 at the top right-hand side, and select **Make Discoverable.** You should see the Bluetooth icon flashing

- Select 'raspberrypi' in the Bluetooth menu on your other device

- Accept the pairing request on your Raspberry Pi 5

- You should now see the message **Connected Successfully** on your Raspberry Pi 5 and you can exchange files between your other device and the Raspberry Pi computer.

## 2.8 Connecting the Raspberry Pi 5 to a wired network

You may want to connect your Raspberry Pi 5 to a network through an Ethernet cable. The steps are as follows:

**Step 1:** Connect a network cable between your Raspberry Pi 5 and your Wi-Fi router.

**Step 2:** Connect the keyboard, mouse and monitor to your Raspberry Pi and power up as normal

**Step 3:** Log in to the system by entering your username and password

**Step 4:** Providing your network hub supports DHCP (nearly all network routers support DHCP), you will be connected automatically to the network and will be assigned a unique IP address within your network. Note that DHCP assigns IP addresses to newly connected devices.

**Step 5:** Check to find out the IP address assigned to your Raspberry Pi 5 by the network router. Enter the command **ifconfig** as described earlier

### 2.8.1 Unable to connect to a wired network

If you find out that you are not assigned an IP address by the DHCP server, possible causes are:

- Your network cable is faulty
- The network hub does not support DHCP
- DHCP is not enabled on your Raspberry Pi, i.e. it may have been configured for a fixed IP address

In most cases, it is very unlikely that the network cable is faulty. Also, most network hubs support the DHCP protocol. If you are having problems with the network, it is possible that your Raspberry Pi is not configured to accept DHCP issued addresses. The Raspberry Pi is normally configured to accept DHCP addresses, but it is possible that you have changed the configuration somehow.

To resolve the wired network connectivity problem, follow the steps given below:

**Step 1:** find out whether your Raspberry Pi is configured for DHCP or fixed IP addresses. Enter the following command:

pi@raspberrypi ~$ **cat /etc/network/interfaces**

If your Raspberry Pi is configured to use the DHCP protocol (which is normally the default configuration), the word **dhcp** should appear at the end of the following line:

iface eth0 inet dhcp

If, on the other hand, your Raspberry Pi is configured to use static addresses, then you should see the word **static** at the end of the following line:

iface eth0 inet static

**Step 2:** To use the DHCP protocol, edit file **interfaces** (e.g. using the **nano** text editor) and change the word **static** to **dhcp**. It is recommended to make a backup copy of the file interfaces before you change it:

pi@raspberrypi ~$ **sudo cp /etc/network/interfaces /etc/network/int.bac**

You should now restart your Raspberry Pi and an IP address will probably be assigned to your device.

**Step 3:** To use static addressing, make sure that the word **static** appears as shown above. If not, edit file **interfaces** and change **dhcp** to **static**

**Step 4:** You need to edit and add the required unique IP address, subnet mask and gateway addresses to file **interfaces** as in the following example (this example assumes that the required fixed IP address is 192.168.1.251, the subnet mask used in the network is 255.255.255.0, and the gateway address is 192.168.1.1):

```
iface eth0 inet static
address 192.168.1.251
netmask 255.255.255.0
gateway 192.168.1.1
```

Save the changes and exit the editor. If you are using the **nano** editor, exit by pressing Ctrl+X, then enter Y to save the changes, and enter the filename to write to as **/etc/network/interfaces**.

Restart your Raspberry Pi 5.

## 2.9 Installing the Raspberry Pi 5 Bookworm operating system on a blank microSD card

If you have a pre-installed Raspberry Pi operating system Bookworm on a microSD card, then you can start using it as described earlier in this chapter. In this section, you will learn how to install the latest Bookworm operating system on a microSD card if you do not have a pre-installed card.

The steps are as follows:

- Insert a microSD card into your PC. You may need to use an SD card adapter
- Go to the website: https://www.raspberrypi.com/software/
- Click to download the **Raspberry Pi Imager.** At the time of writing this book, this file was called: **imager_1.7.5.exe**
- Double click to start the imager program and click to install it
- Click **Finish** to run the imager
- Click **Operating System** and select the operating system at the top of the list as: **Raspberry Pi OS (64-bit)**. See Figure 2.15

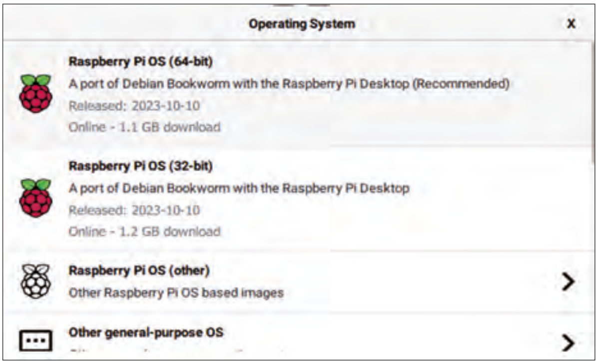



*Figure 2.15 Select the operating system*

- Click **Storage** and select the SD card storage
- Click to open the settings (gear shape)
- Click to enable SSH
- Click to enable password authentication
- Set username and password
- Click to **Configure wireless LAN**
- Click **Save**
- Click **Write** to write the operating system to the microSD card
- Wait until writing and verifying are finished (Figure 2.16)


- Remove the microSD card and insert into your Raspberry Pi 5

If you have a monitor and keyboard, you can log in to your Raspberry Pi 5 directly and start using it. Otherwise, find the IP address of your Raspberry Pi 5 (e.g. from your router, or there are many apps for smartphones, such as **who's on my wifi** that shows all the devices connected to your router with their IP addresses). Then log in to your Raspberry Pi 5 and start using it.

# Chapter 3 • Using The Console Commands

## 3.1 Overview

Raspberry Pi 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 since 1991. In general, Linux is harder to manage than some other operating systems like Windows, but offers more flexibility and configuration options. There are several popular versions of the Linux operating system, such as Debian, Ubuntu, Red Hat, Fedora and so on.

Linux commands are text-based. In this chapter, you will be looking at some of the useful Linux commands and see how you can manage your Raspberry Pi using these commands.

When you apply power to your Raspberry Pi 5, the Linux command line (or the Linux shell, or Console commands) is the first thing you see, and it is where you can enter operating system commands.

## 3.2 The Command Prompt

Assuming your username is **pi**, after you log in to Raspberry Pi 5, you will see the following prompt displayed where the system waits for you to enter a command:

pi@raspberrypi: ~$

Here, **pi** is the name of the user who is logged in.
**raspberrypi** is the name of the computer, used to identify it when connecting over the network.

**~** character indicates that you are currently in your default directory.

## 3.3 Useful Console commands

In this section, you will be learning some of the useful Console commands, where examples will be given for each command. **In this chapter, commands entered by the user are shown in bold for clarity**. Also, it is important to remind you that all the commands must be terminated by the Enter key.

### 3.3.1 System and user information

These commands are useful as they tell you information about the system. Command **cat /proc/cpuinfo** displays information about the processor (command **cat** displays the contents of a file. In this example, the contents of file **/proc/cpuinfo** is displayed). Since there are four cores in the Raspberry Pi 5, the display is in four sections. Figure 3.1 shows an example display, where only part of the display is shown here.

```
pi@raspberrypi:~ $ cat /proc/cpuinfo
processor       : 0
BogoMIPS        : 108.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp
p cpuid asimdrdm lrcpc dcpop asimddp
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x4
CPU part        : 0xd0b
CPU revision    : 1

processor       : 1
BogoMIPS        : 108.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp
p cpuid asimdrdm lrcpc dcpop asimddp
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x4
CPU part        : 0xd0b
CPU revision    : 1

processor       : 2
BogoMIPS        : 108.00
```

*Figure 3.1 Command: cat /proc/cpuinfo*

Command **uname –s** displays the operating system kernel name, which is Linux. Command **uname –a** displays complete detailed information about the kernel and the operating system. An example is shown in Figure 3.2.

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 6.1.0-rpi4-rpi-2712 #1 SMP PREEMPT Debian 1:6.1.54-1+rpt1
3-09-27) aarch64 GNU/Linux
pi@raspberrypi:~ $
```

*Figure 3.2 Command:  uname – a*

Command **cat /proc/meminfo** displays information about the memory on your Raspberry Pi. Information such as the total memory and free memory at the time of issuing the command are displayed. Figure 3.3 shows an example, where only part of the display is shown here.

```
pi@raspberrypi:~ $ cat /proc/meminfo
MemTotal:        8246848 kB
MemFree:         7792320 kB
MemAvailable:    7993952 kB
Buffers:           21552 kB
Cached:           246240 kB
SwapCached:            0 kB
Active:           280096 kB
Inactive:          64848 kB
Active(anon):      77008 kB
Inactive(anon):     4112 kB
Active(file):     203088 kB
Inactive(file):    60736 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:        102368 kB
SwapFree:         102368 kB
Zswap:                 0 kB
Zswapped:              0 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:         77232 kB
Mapped:            70880 kB
```

*Figure 3.3 Command: cat /proc/meminfo*

Command **whoami** displays the name of the current user. In this case, **pi** is displayed as the current user.

A new user can be added to your Raspberry Pi 5 using the command **useradd**. In the example in Figure 3.5, a user called **John** is added. A password for the new user can be added using the **passwd** command followed by the username. In Figure 3.4, the password for user John is set to **mypassword** (not displayed for security reasons). Notice that both the **useradd** and **passwd** are privileged commands, and the keyword **sudo** must be entered before these commands. Notice that the **–m** option creates a home directory for the new user.

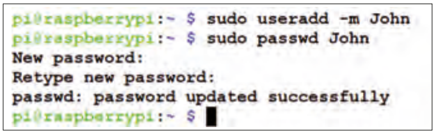



*Figure 3.4 Commands: useradd and passwd*

You can log in to the new user account by specifying the username and the password as shown in Figure 3.5. You can type command **exit** to log out from the new account.

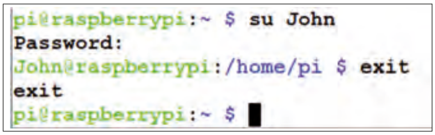



*Figure 3.5 Logging into a new account*

Command **sudo apt-get upgrade** is used to upgrade all the software packages on the system.

### 3.3.2 The Raspberry Pi 5 directory structure

The Raspberry Pi 5 directory structure consists of a single root directory, with directories and subdirectories under the root. Different types of operating system programs and application programs are stored in different directories and subdirectories.

Figure 3.6 shows part of the Raspberry Pi 5 directory structure. Notice that 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. The **home** directory contains subdirectories for each user of the system. In the example in Figure 3.7, **pi** is the subdirectory for user **pi**. In a new system, this subdirectory contains two subdirectories called **Desktop** and **python_games**.

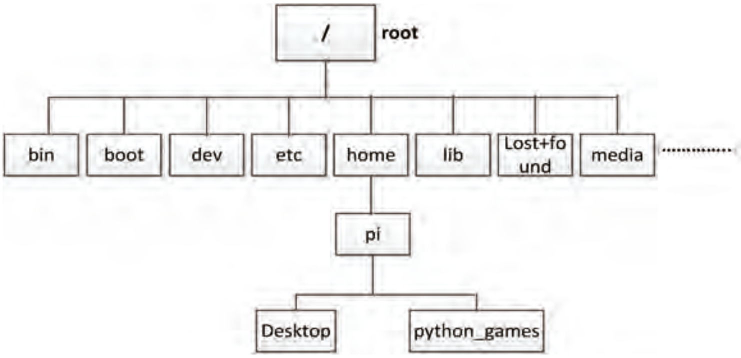



*Figure 3.6 Raspberry Pi 5 directory structure (only part of it is shown)*

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

pi@raspberrypi: ~$ **pwd**
/home/pi
pi@raspberry: ~$

To show the directory structure, enter the command **ls /** (Figure 3.7):

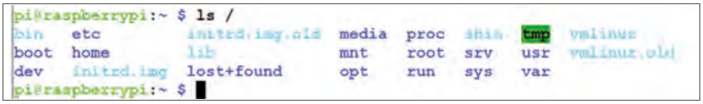



*Figure 3.7 Directory structure*

To show the subdirectories and files in your working directory, enter **ls**:

pi@raspberrypi: ~$ **ls**
Bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates
pi@raspberrypi: ~$

Notice that the subdirectories are displayed in blue colour and files in black colour.

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**
Bookshelf
Desktop
Documents
Downloads
Music

Pictures
Public
Templates
Videos
pi@raspberrypi: ~$

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

pi@raspberrypi: ~$ **ls –F**
Bookshelf/ Documents/ Music/ Public/ Videos/
Desktop/ Downloads/ Pictures/ Templates/
pi@raspberrypi: ~$

To list the results, separated by commas:

pi@raspberrypi: ~$ **ls –m**

Bookshelf, Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos

pi@raspberrypi: ~$

You can mix the arguments as shown in Figure 3.8.


*Figure 3.8 Mixing the arguments*

Subdirectories are created using the command **mkdir** followed by the name of the subdirectory (Figure 3.9)

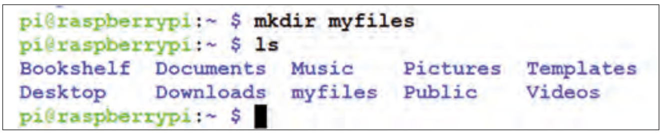



*Figure 3.9 Creating a subdirectory*

Command **find** is used to search the whole system for a file and outputs a list of all directories that contain the file. For example, the command **find / -name myfile.txt** searches the whole system for the file **myfile.txt.**

**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 below, 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 file was created is on the left-hand side of its name. Next comes the size, given in bytes. The characters at the beginning of each line are about permissions, i.e. who is allowed to use or modify the file or the directory.

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 - GROUP
- What everyone else can do – called WORLD

The first word **pi** in the example in Figure 3.10 shows who the user of the file (or directory) is, and the second word **pi** shows the group name that owns the file. In this example, both the user and the group names are **pi**.

```
pi@raspberrypi:~ $ ls -l
total 40
drwxr-xr-x 2 pi dogan-ibrahim 4096 Sep 22 08:25 Bookshelf
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Desktop
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Documents
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Downloads
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Music
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  3 11:37 myfiles
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Pictures
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Public
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Templates
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Videos
pi@raspberrypi:~ $
```

*Figure 3.10 File permissions example*

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 come 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** directory, 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. In this example, a file with the name **mytestfile.txt** was created in the home directory for demonstration purposes (See Figure 3.11). In this example, the file **mytestfile.txt** has the user read and write permissions.

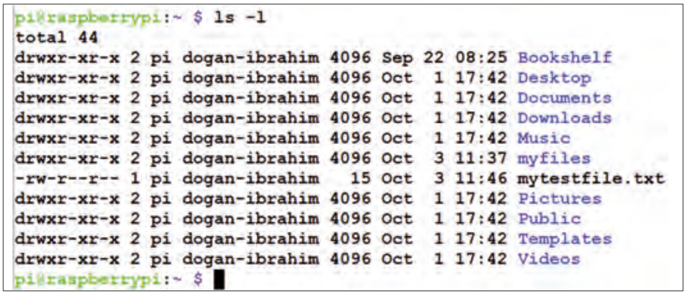
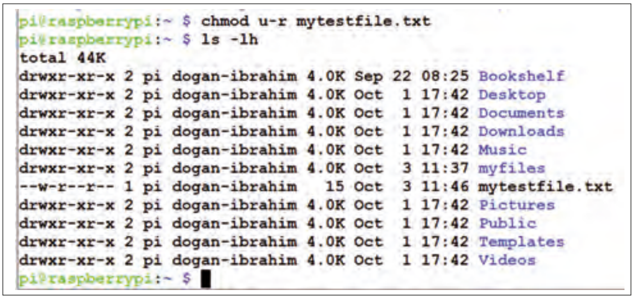



*Figure 3.11 File permissions*

We will be changing the permissions so that the user does not have read permission on this file:

pi@raspberrypi: ~$ **chmod u-r mytestfile.txt**
pi@raspberrypi: ~$ **ls –lh**

The result is shown in Figure 3.12.



```
pi@raspberrypi:~ $ chmod u-r mytestfile.txt
pi@raspberrypi:~ $ ls -lh
total 44K
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Sep 22 08:25 Bookshelf
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Desktop
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Documents
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Downloads
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Music
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  3 11:37 myfiles
--w-r--r-- 1 pi dogan-ibrahim   15 Oct  3 11:46 mytestfile.txt
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Pictures
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Public
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Templates
drwxr-xr-x 2 pi dogan-ibrahim 4.0K Oct  1 17:42 Videos
pi@raspberrypi:~ $
```

*Figure 3.12 File permissions of* **mytestfile.txt**

Notice that if you now try to display the contents of the file **mytestfile.txt** using the **cat** command, you will get an error message:

pi@raspberrypi: ~$ **cat mytestfile.txt**
cat: mytestfile.txt: Permission denied
pi@raspberrypi: ~$

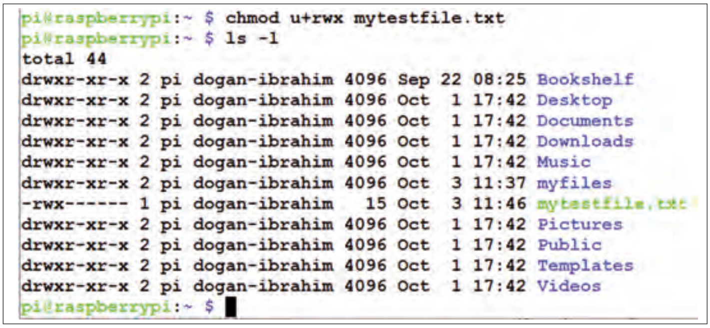
All the permissions can be removed from a file by the following command:

pi@raspberrypi: ~$ **chmod a=  mytestfile.txt**

In the following example, **rwx** user permissions are given to file **mytestfile.txt**:

pi@raspberrypi: ~$ **chmod u+rwx mytestfile.txt**

Figure 3.13 shows the new permissions of file mytestfile.txt.



```
pi@raspberrypi:~ $ chmod u+rwx mytestfile.txt
pi@raspberrypi:~ $ ls -l
total 44
drwxr-xr-x 2 pi dogan-ibrahim 4096 Sep 22 08:25 Bookshelf
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Desktop
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Documents
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Downloads
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Music
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  3 11:37 myfiles
-rwx------ 1 pi dogan-ibrahim   15 Oct  3 11:46 mytestfile.txt
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Pictures
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Public
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Templates
drwxr-xr-x 2 pi dogan-ibrahim 4096 Oct  1 17:42 Videos
pi@raspberrypi:~ $
```

*Figure 3.13 New permissions of file* **mytestfile.txt**

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 your working directory to **Music**, you can also enter the command:

pi@raspberrypi: ~$ **cd ~/Music**
pi@raspberrypi: ~/myfiles $

To go back to the default working directory, you can enter:

pi@raspberrypi: ~/Music $ **cd ~**
pi@raspberrypi: ~$

To find out more information about a file, you can use the **file** command. For example:

pi@raspberrypi: ~$ **file mytestfiile.txt**
mytestfile.txt: ASCII text
pi@raspberrypi: ~$

The **–R** argument of the command **ls** lists all the files in all the subdirectories of the current working directory. An example is given below (only part of the display is shown). Notice here in Figure 3.14 that subdirectory **Bookshelf** contains file **BeginnersGuide-4thEd-Eng_v2.pdf**

```
pi@raspberrypi:~ $ ls -R
.:
Bookshelf  Documents  Music    mytestfile.txt  Public     Videos
Desktop    Downloads  myfiles  Pictures        Templates

./Bookshelf:
BeginnersGuide-4thEd-Eng_v2.pdf

./Desktop:

./Documents:

./Downloads:
```

*Figure 3.14 Command:* **ls –R**

To display information on how to use a command, you can use the **man** command. As an example, to get help on using the **mkdir** command:

```
pi@raspberrypi: ~$ man mkdir
MKDIR(1)

NAME
        Mkdir – make directories

SYNOPSIS
```

```
            Mkdir [OPTION]…DIRECTORY…

      DESCRIPTION
            Create the DIRECTORY(ies), if they do not already exist.

            Mandatory arguments for long options are mandatory for short
options

            -m, --mode=MODE
                        Set file mode (as in chmod), not a=rwx – umask
      ---------------------------------------------------------------------------
      ---------------------------------------------------------------------------
```

Enter **q** to exit the man display.

**Help**
The **man** command usually gives several pages of information on how to use a command. You can type **q** to exit the **man** command and return to the operating system prompt.

The **less** command can be used to display a long listing one page at a time. Using the up and down arrow keys, we can move between pages. An example is given below. Type **q** to exit:

pi@raspberrypi: ~$ **man ls | less**
<display of help on using the ls command>
pi@raspberrypi: ~$

**Date and Time**
To display the current date and time, the **date** command is used.

**Copying a File**
To make a copy of a file, use the command **cp**. In the following example, a copy of the file **mytestfile.txt** is made, and the new file is given the name **test.txt**:

pi@raspberrypi: ~$ **cp mytestfile.txt test.txt**
pi@raspberrypi: ~$

**Wildcards**
You can use wildcard characters to select multiple files with similar characteristics. e.g. files having the same file-extension names. The ***** character is used to match any number of characters. Similarly, the **?** character is used to match any single character. In the example below, all the files with extensions **.txt** are listed:

pi@raspberrypi: ~$ **ls *.txt**
mytestfile.txt   test.txt
pi@raspberrypi: ~$

The wildcard characters [a-z] can be used to match any single character in the specified character range. An example is given below which matches any files that start with the letters 'o', 'p', 'q', 'r', 's', and 't', and with the **.txt** extension:

pi@raspberrypi: ~$ **ls [o-t]*.txt**
test.txt
pi@raspberrypi: ~$

**Renaming a File**
You can rename a file using the **mv** command. In the example below, the name of file **test.txt** is changed to **test2.txt**:

pi@raspberrypi: ~$ **mv test.txt test2.txt**
pi@raspberrypi: ~$

**Deleting a File**
The command **rm** can be used to remove (delete) a file. In the example below, the file **test2.txt** is deleted:

pi@raspberrypi: ~$ **rm test2.txt**
pi@raspberrypi: ~$

The argument **–v** can be used to display a message when a file is removed. Also, the **–i** argument asks for confirmation before a file is removed. In general, the two arguments are used together as **–vi**. An example is given below:

pi@raspberrypi: ~$ **rm –vi test2.txt**
rm: remove regular file 'test2.txt'? **y**
removed 'test2.txt'
pi@raspberrypi: ~$

**Sorting a file**
The command sort displays the contents of a file in ascending order. The general format of this command is:

**sort** <options> <filename>

Valid options are:

| | |
|---|---|
| **-u** | removes duplicates from the output |
| **-r** | sorts the output in descending order |
| **-o** | writes the sorted output to a file |

**Word count**
Command **wc** <filename> displays the word count in a file

**File differences**
Command **diff** <file1> <file2) displays the differences between two files line by line

**Removing a Directory**
A directory can be removed using the **rmdir** command:

pi@raspberrypi: ~$ **rmdir Music**
pi@raspberrypi: ~$

**Re-directing the Output**
The greater sign **>** can be used to redirect the output of a command to a file. For example, we can redirect the output of the **ls** command to a file called **lstest.txt**:

pi@raspberrypi: ~$ **ls > lstest.txt**
pi@raspberrypi: ~$

The **cat** command can be used to display the contents of a file:

pi@raspberrypi: ~$ **cat mytestfile.txt**
This is a file
This is line 2
pi@raspberrypi: ~$

Using two greater signs '**>>**' adds to the end of a file.

**Writing to the Screen or to a File**
The **echo** command can be used to write to the screen. It can be used to perform simple mathematical operations if the numbers and the operation are enclosed in two brackets, preceded by a $ character:

pi@raspberrypi: ~$ **echo $((5*6))**
30
pi@raspberrypi: ~$

The echo command can also be used to write a line of text to a file. An example is shown below:

pi@raspberrypi: ~$ **echo a line of text > lin.dat**
pi@raspberrypi: ~$ **cat lin.dat**
a line of text
pi@raspberrypi: ~$

**Matching a String**
The grep command can be used to match a string in a file. An example is given below, assuming that the file lin.dat contains sting a line of text. Notice that the matched word is shown in bold:

pi@raspberrypi: ~$ **grep line lin.dat**
a **line** of text
pi@raspberrypi: ~$

**Head and Tail Commands**
The head command can be used to display the first 10 lines of a file. The format of this command is as follows:

pi@raspberrypi: ~$ **head mytestfile.txt**
……………………………….
……………………………….
pi@raspberrypi: ~$

Similarly, the tail command is used to display the last 10 lines of a file. The format of this command is as follows:

pi@raspberrypi: ~$ **tail mytestfile.txt**
……………………………….
……………………………….
pi@raspberrypi: ~$

The **which** command displays the location of an executable program. For example, the location of the python program can be found as follows:

pi@raspberrypi: ~$ **which python**
/usr/bin/python
pi@raspberrypi: ~$

**Super User Commands**
Some of the commands are privileged and only the authorized persons can use them. Inserting the word **sudo** at the beginning of a command gives us the authority to use the command without having to log in as an authorized user.

**What software is installed on my Raspberry Pi 5**
To find out what software is installed on your Raspberry Pi 5, enter the following command. You should get several pages of display:

pi@raspberrypi: ~$ **dpkg –l**
…………………………..
…………………………..
pi@raspberrypi: ~$

You can also find out if a certain software package is already installed on our computer. An example is given below which checks whether software called **xpdf** (PDF reader) is installed. In this example, **xpdf** is installed and the details of this software are displayed:

pi@raspberrypi: ~$ **dpkg --s xpdf**
Package: xpdf
Status: install ok installed
Priority: optional
Section: text
Installed-Size: 395
………………………….
………………………….
pi@raspberrypi: ~$

If the software is not installed, you get a message similar to the following (assuming we are checking to see if a software package called **bbgd** is installed):

pi@raspberrypi: ~$ **dpkg –s bbgd**
dpkg-query: package 'bbgd' is not installed and no information is available
……………………………………………………………………………..
…………………………………………………………………………….
pi@raspberrypi: ~$

### 3.3.3 Resource monitoring on the Raspberry Pi 5

System monitoring is an important topic for managing usage of your Raspberry Pi. One of the most useful system monitoring commands is the top, which displays the current usage of system resources and displays which processes are running and how much memory and CPU time they are consuming.
Figure 3.15 shows a typical system resource display obtained by entering the following command (only part of the display is shown, Enter **q** to exit):

pi@raspberrypi: ~$ **top**
pi@raspberrypi: ~$

```
top - 12:14:08 up  2:54,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 136 total,   1 running, 135 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us, 33.3 sy,  0.0 ni, 66.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   8053.6 total,   7589.7 free,    247.3 used,    303.7 buff/cache
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7806.2 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1357 pi        20   0   11984   4416   2784 R   6.2   0.1   0:00.01 top
    1 root      20   0  169008  12528   8448 S   0.0   0.2   0:02.72 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
   12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftir+
   15 root      20   0       0      0      0 I   0.0   0.0   0:00.05 rcu_pre+
   16 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migrati+
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   19 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migrati+
```

*Figure 3.15 Typical system resource display*

Some of the important points in Figure 3.15 are summarized below (for lines 1 to 5 of the display):

- There are a total of 138 processes in the system
- Currently, only one process is running, 1 process is sleeping, and 0 processes are stopped
- The percentage of CPU utilization is 0.0 'us' for user applications (us)
- The percentage of CPU utilization for system applications is 0.0 (sy)
- There are no processes requiring more or less priority (ni)
- 100% of the time the CPU is idle (id)
- There are no processes waiting for I/O completion (wa)
- There are no processes waiting for hardware interrupts (hi)
- There are no processes waiting for software interrupts (si)
- There is no time reserved for a hypervisor (st)
- The total usable memory is 8053 bytes, of which 247 bytes are in use, 7589 bytes are free, and 303 bytes are used by buffers/cache
- Line 5 displays the swap space usage

The process table gives the following information for all the processes loaded to the system:

- PID: the process ID number
- USER: owner of the process
- PR: priority of the process
- NI: the nice value of the process
- VIRT: the amount of virtual memory used by the process
- RES: size of the resident memory
- SHR: shared memory the process is using
- S: process status (sleeping, running, zombie)
- %CPU: the percentage of CPU consumed
- %MEM: percentage of RAM used
- TIME+: total CPU time the task used
- COMMAND: The actual name of the command

The command **htop** is similar to the **top** command, except it has more features and is more user-friendly.

The **ps** command can be used to list all the processes used by the current user. An example is shown in Figure 3.16.

```
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
  971 pts/0    00:00:00 bash
 1372 pts/0    00:00:00 ps
pi@raspberrypi:~ $
```

*Figure 3.16 Command:* **ps**

Command **ps –ef** gives a lot more information about the processes running in the system.

**Killing a process**

There are many options for killing (or stopping) a process. A process can be killed by specifying its PID and using the following command:

pi@raspberrypi: ~$ **kill -9 <PID>**

**Disk (microSD card) usage**

The disk free command **df** can be used to display the disk usage statistics. An example is shown in Figure 3.17. option **–h** displays in human-readable form.

```
pi@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G     0  3.8G   0% /dev
tmpfs           806M  3.6M  802M   1% /run
/dev/mmcblk0p2   29G  4.6G   23G  17% /
tmpfs           4.0G   32K  4.0G   1% /dev/shm
tmpfs           5.0M   48K  5.0M   1% /run/lock
/dev/mmcblk0p1  510M   73M  438M  15% /boot/firmware
tmpfs           806M  128K  806M   1% /run/user/1000
pi@raspberrypi:~ $
```

*Figure 3.17 Command:* **df -h**

Command **free** shows how much memory is used and the amount of free memory.

### 3.3.4 Shutting Down

Although you can disconnect the power supply from your Raspberry Pi 5 when you finish working with it, it is not recommended since there are many processes running on the system, and it is possible to corrupt the file system. It is much better to shut down the system in an orderly manner.

The following command will stop all the processes and make the file system safe, and then turn off the system safely:

pi@raspberrypi: ~$ **sudo halt**

The following command stops and then restarts the system:

pi@raspberrypi: ~$ **sudo reboot**

The system can also be shut down and then restarted after a time by entering the following command. Optionally, a shutdown message can be displayed if desired:

pi@raspberrypi: ~$ **shutdown –r <time> <message>**

To shutdown at 1:55 AM:

pi@raspberrypi: ~$ **sudo shutdown -h 01:55:**

Enter the following command to shut down now:

pi@raspberrypi: ~$ **sudo shutdown now**

Broadcast message from root @raspberrypi on pts/1 (Tue 2023-10-03 12:03:00 BST)

The system will power off now!

**Note**: Raspberry Pi 5 includes a power switch at its side. When the Raspberry Pi is ON, a single press brings the shutdown/logout menu. Another press triggers a safe shutdown, which is a standby with the Raspberry Pi consuming about 1.4 W. A press of the button will start up the Raspberry Pi 5.

### 3.3.5 Networking

Some useful networking commands are:

**ifconfig**: check the IP address of your Raspberry Pi

**iwconfig**: check which network the Raspberry Pi is using. An example is shown in Figure 3.18. Here, the SSID of the Wi-Fi adapter used is BTHub5-6SPN

```
pi@raspberrypi:~ $ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:"BTHub5-6SPN"
          Mode:Managed  Frequency:5.18 GHz  Access Point: 4C:1B:86:B5:BA:7B
          Bit Rate=433.3 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=45/70  Signal level=-65 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:3  Invalid misc:0   Missed beacon:0

pi@raspberrypi:~ $
```

*Figure 3.18 Command iwconfig*

**ping**: used to test the availability of a network device. An example is shown in Figure 3.19

```
pi@raspberrypi:~ $ ping 192.168.1.251
PING 192.168.1.251 (192.168.1.251) 56(84) bytes of data.
64 bytes from 192.168.1.251: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 192.168.1.251: icmp_seq=2 ttl=64 time=0.014 ms
64 bytes from 192.168.1.251: icmp_seq=3 ttl=64 time=0.011 ms
```

*Figure 3.19 Command ping*

**wget**: this command is used to download a file from the web and saves the file in the current directory.

**hostname – I**: shows the IP address of the Raspberry Pi

The command **vcgencmd measure_temp** displays the CPU temperature as shown in Figure 3.20.

```
pi@raspberrypi:~ $ vcgencmd measure_temp
temp=49.4'C
pi@raspberrypi:~ $ 
```

*Figure 3.20 Displaying the CPU temperature*

### 3.3.6 System information and other useful commands

The **uname** command is used to display system information. This command has the following options:

| | |
|---|---|
| **-a** | Show all system information |
| **-s** | display the kernel name |
| **-n** | print the network node hostname |
| **-r** | print the kernel release |
| **-v** | print the kernel version number |
| **-m** | print the system hardware name |
| **-p** | print the processor type |
| **-i** | print the hardware platform type |
| **-o** | print the operating system type |

Some examples are shown in Figure 3.21

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 6.1.0-rpi4-rpi-2712 #1 SMP PREEMPT Debian 1:6.1.54-1+rpt1
3-09-27) aarch64 GNU/Linux
pi@raspberrypi:~ $ uname -s
Linux
pi@raspberrypi:~ $ uname -n
raspberrypi
pi@raspberrypi:~ $ uname -r
6.1.0-rpi4-rpi-2712
```

*Figure 3.21 The uname command*

If you have executed many commands and want to use some of them again, but you cannot remember the command name, you can use the **history** command. An example is shown in Figure 3.22. To execute a command from the history, enter **!** followed by the command number. For example, to execute the **ls** command again, you can enter **!6** followed by the Enter key.

```
pi@raspberrypi:~ $ history
    1  ls
    2  sudo nano /etc/default/console-setup
    3  sudo /etc/init.d/console-setup restart
    4  sudo restart
    5  sudo reboot
    6  ls
    7  cat /etc/default/console-setup
    8  sudo dpkg-reconfigure console-setup
    9  ls
   10  ls music
```

*Figure 3.22 The history command*

The **clear** command is also useful, and it is used to clear the screen.

To install a package, use the command: **sudo apt install <package_name>**

The **&** operator allows you to run any command in the background so that you can use the terminal for other tasks. This operator must be added to the end of a command.

The **&&** operator allows you to run two or more commands at the same time. For example, command1 **&&** command2

# Index

elektor books

# Raspberry Pi 5 Essentials

## Program, build, and master over 60 projects with Python

The Raspberry Pi 5 is the latest single-board computer from the Raspberry Pi Foundation. It can be used in many applications, such as in audio and video media centers, as a desktop computer, in industrial controllers, robotics, and in many domestic and commercial applications. In addition to the well-established features found in other Raspberry Pi computers, the Raspberry Pi 5 offers Wi-Fi and Bluetooth (classic and BLE), which makes it a perfect match for IoT as well as in remote and Internet-based control and monitoring applications. It is now possible to develop many real-time projects such as audio digital signal processing, real-time digital filtering, real-time digital control and monitoring, and many other real-time operations using this tiny powerhouse.

The book starts with an introduction to the Raspberry Pi 5 computer and covers the important topics of accessing the computer locally and remotely. Use of the console language commands as well as accessing and using the desktop GUI are described with working examples. The remaining parts of the book cover many Raspberry Pi 5-based hardware projects using components and devices such as

- LEDs and buzzers
- LCDs
- Ultrasonic sensors
- Temperature and atmospheric pressure sensors
- The Sense HAT
- Camera modules

Example projects are given using Wi-Fi and Bluetooth modules to send and receive data from smartphones and PCs, and sending real-time temperature and atmospheric pressure data to the cloud.

All projects given in the book have been fully tested for correct operation. Only basic programming and electronics experience are required to follow the projects. Brief descriptions, block diagrams, detailed circuit diagrams, and full Python program listings are given for all projects described. Readers can find the program listings on the Elektor Store website, www.elektor.com (search for: book title).

**Prof Dogan Ibrahim** has a BSc (Hons) degree in Electronic Engineering, an MSc degree in Automatic Control Engineering, and a PhD degree in Digital Signal Processing and Microprocessors.

Dogan has worked in many organizations and is a Fellow of the Institution of Engineering and Technology (IET) in UK as well as a Chartered Electrical Engineer. He has authored over 100 technical books and over 200 technical articles on electronics, microprocessors, microcontrollers, and related fields. Dogan is a certified Arduino professional and has many years of experience with numerous types of microprocessors and microcontrollers.

**Elektor International Media**
www.elektor.com

ISBN 978-3-89576-586-5

elektor
design > share > earn