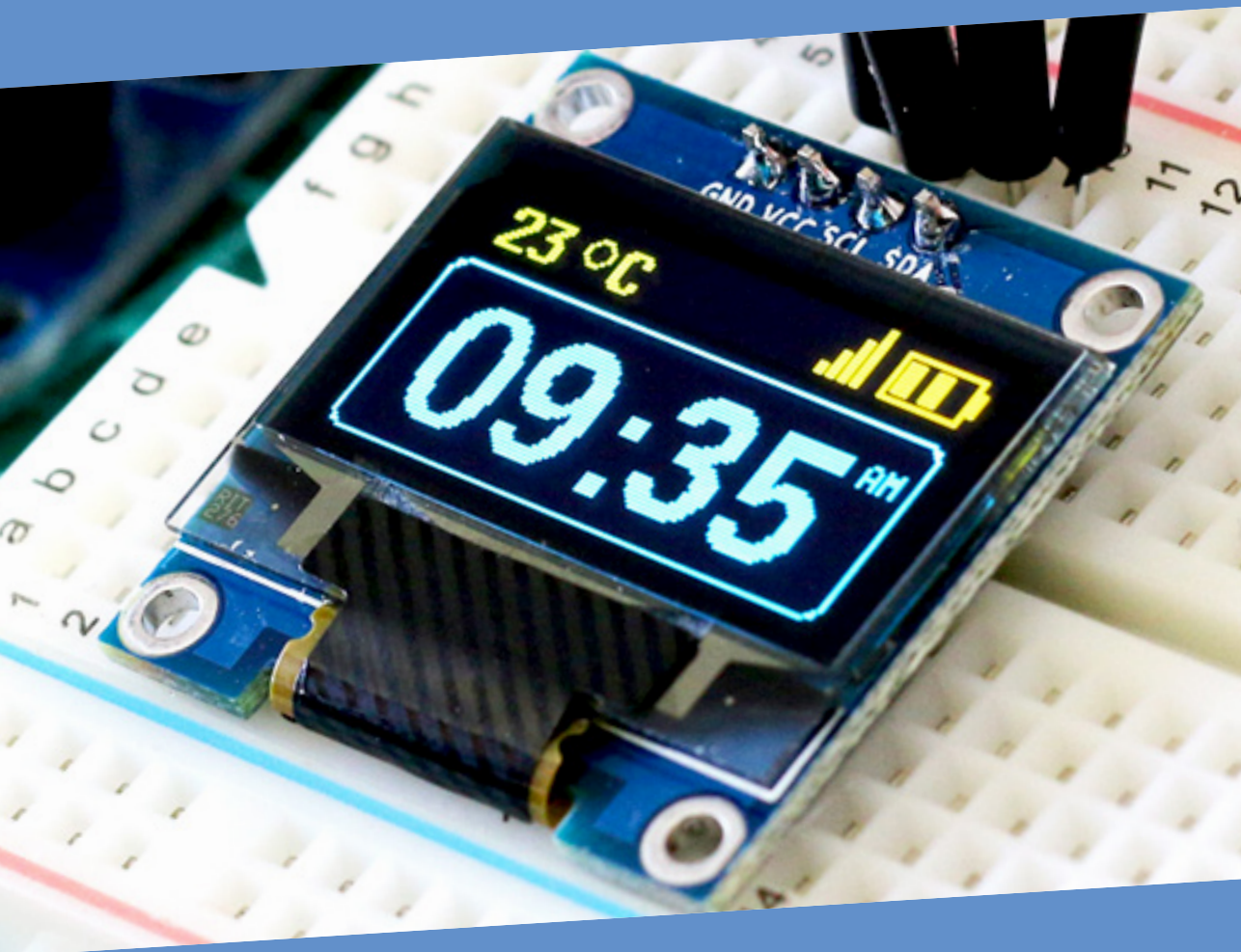
elektor books

Using Displays in Raspberry Pi Projects

Learn to program displays and GUIs with Python



Dogan Ibrahim

elektor
design > share > sell

Using Displays in Raspberry Pi Projects

Dogan Ibrahim

elektor
design > share > sell

an Elektor Publication

● 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

First published in the United Kingdom 2021

● British Library Cataloguing in Publication Data
Catalogue record for this book is available from the British Library

● ISBN: 978-3-89576-421-9

● EISBN: 978-3-89576-422-6

● EPUB: 978-3-89576-423-3

Prepress production: DMC ¦ daverid.com
Printed in the Netherlands by Wilco

• Declaration

The author and publisher have used their best efforts to ensure the correctness of the information contained in this book. They do not assume, or 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.

Table of Contents

• Preface

The Raspberry Pi 4 is the latest credit-card sized computer and can be used in many applications, such as in audio and video media centres, as a desktop computer, in industrial controllers, robotics, and in many other domestic and commercial applications. In addition to the many features found in other versions of Raspberry Pi, the Pi 4 also offers Wi-Fi and Bluetooth, making it highly desirable in remote and internet based control and monitoring applications.

This book is about Raspberry Pi 4 display projects. The book starts by explaining how to install the latest Raspbian operating system on an SD card, and how to configure and use GPIOs. The core of the book explains in simple terms and with tested and working example projects, how various types of display devices can be used in Raspberry Pi projects. Example projects provided use simple LEDs, bar graph LEDs, matrix LEDs, bit map LEDs, multi-digit 7-segment LEDs, LED strips, LCDs, OLEDs, E-paper displays, TFT displays, 7 inch touch-screen display modules, and many more.

One unique feature of this book is that it covers almost all types of display that readers will need to use in their Raspberry Pi based projects. The operation of each project is fully given, including block diagrams, circuit diagrams, and commented full program listings. It is therefore an easy task to convert the given projects to run on other popular platforms, such as Arduino or PIC microcontrollers.

All projects provided in the book have been fully tested and work. The following sub-headings are used in the projects where applicable:

- Project title
- Project description
- Aim of the project
- Block diagram
- Circuit diagram
- Program listing

Program listings of all Raspberry Pi projects developed in this book are available on the Elektor website of this book. Readers can download and use these programs in their projects. Alternatively, they can modify the programs to suit to their own applications.

I hope readers find this book helpful and enjoy reading it.

Prof Dr Dogan Ibrahim
December, 2020
London.

Chapter 1 • Installing the Operating System on Raspberry Pi

1.1 • Overview

In this chapter, we will learn how to install the latest operating system **(Raspbian Buster)** on the Raspberry Pi 4. We will also learn the different ways that Python can be used to develop applications. The installation process provided below applies to all Raspberry Pi models unless otherwise specified.

1.2 • Raspbian Buster installation steps on Raspberry Pi 4

Raspbian Buster is an established operating system for the Raspberry Pi. This section provides the steps for installing Raspbian on a new blank SD card. You will need a micro SD card with a capacity of at least 8 GB (16 GB is even better) 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 1.1). At the time of writing, the file was called: **2020-02-13-raspbian-buster-full.img**. You may have to use 7Zip to unzip the download since some of the features are not supported by older zip software.

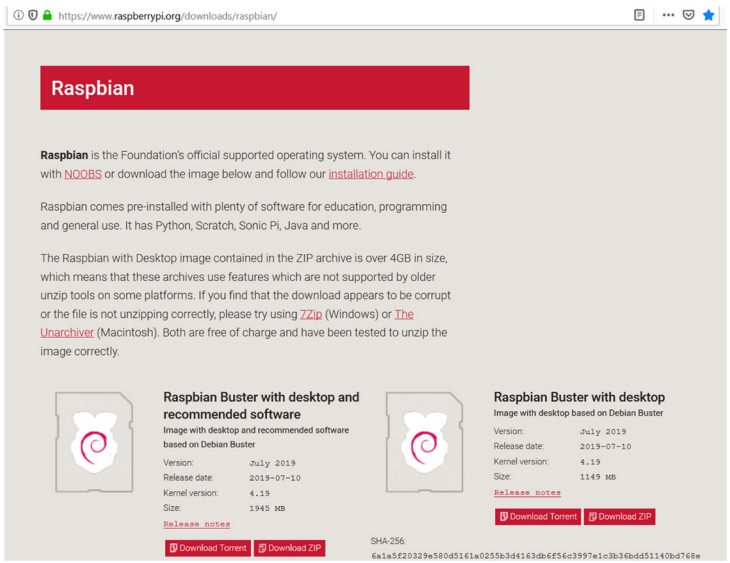


Figure 1.1 Raspbian Buster download page

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

- 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 1.2):

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

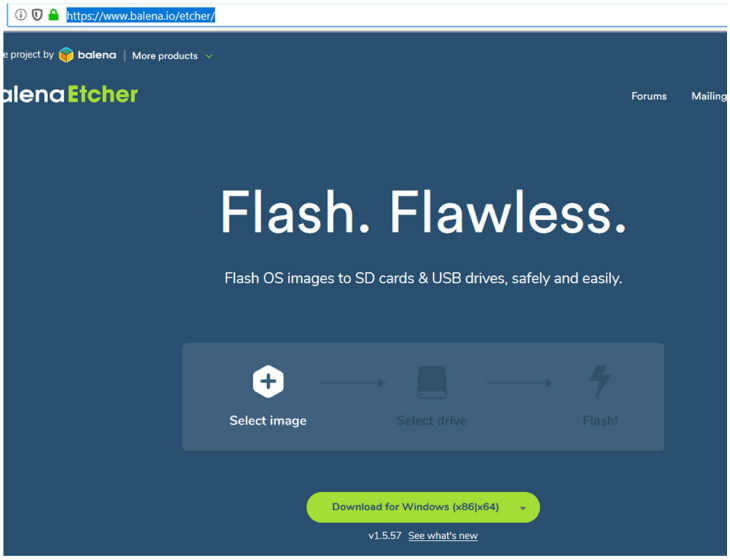


Figure 1.2 Download Etcher

- Double click to Open Etcher, and click **Select image**. Select the Raspbian Buster file you just downloaded and unzipped.
- Click **Select target** and select the micro SD card.
- Click **Flash** (see Figure 1.3). 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.

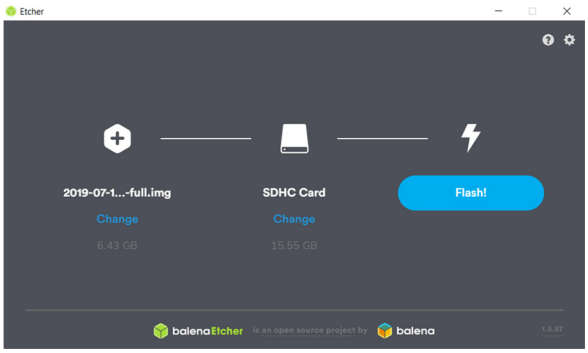


Figure 1.3 Click Flash to flash the disk image

Your micro SD card has now been loaded with the Raspberry Pi operating system. The various options are as follows:

Using a direct connection

If you are making a direct connection to your Raspberry Pi using a monitor and keyboard, just insert the SD card into the card slot and power-up your Raspberry Pi. After a short time, you will be prompted to enter the login details. The default values are username: pi, and password: raspberry.

You can now start using your Raspberry Pi in either command or desktop mode. If you are in command mode, enter the following to start GUI mode:

```
pi@raspberrypi:~ $ startx
```

If you want to boot in GUI mode by default, the steps are:

- Start the configuration tool:

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

- Move down to **Boot Options** and press Enter to select (Figure 1.4).

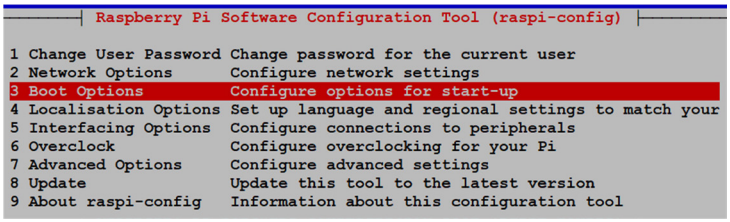


Figure 1.4 Select Boot Options

- Select **Desktop / CLI** and then select **Desktop Autologin** to boot automatically into GUI mode.
- Click **OK** and accept to reboot the system. The system will be in GUI mode next time it reboots.
- You can change your selections to boot in command mode if you wish by selecting **Console in Boot Options.**

You may now want to connect your Raspberry Pi to the internet to remotely access it from a PC or use the Internet. If your Raspberry Pi is equipped with an Ethernet port (e.g. Raspberry Pi 2/3/4), you can connect directly to your Wi-Fi router using an Ethernet cable. You can find the IP address of your connection by entering the command: **ifconfig** in command mode.

Alternatively, you may want to connect your Raspberry Pi to your Wi-Fi and access it

remotely. First, you will have to enable the **SSH**. The steps are:

- Start the configuration tool:

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

- Move down to **Interface Options** and then select **SSH** and enable it
- If you are in GUI mode, click the Wi-Fi icon at the top right hand of the screen and enable the Wi-Fi. Note the IP address allocated automatically to your Raspberry Pi.
- You can now access your Raspberry Pi remotely using a terminal emulation software, such as **Putty** (see Section 1.4 and 1.5)

1.3 ● Using a networked connection

If you don't have a suitable monitor and keyboard to connect directly to your Raspberry Pi, you will have to use a networked connection and access your Raspberry Pi remotely using a PC. There are two options: **connection using an Ethernet cable**, and **connection over Wi-Fi**.

Connection using an Ethernet cable: The steps are:

- Install **Notepad++** on your PC from the following web site:

  https://notepad-plus-plus.org/downloads/v7.8.5/

- Insert the SD card back to your PC and start the **Notepad++** software
- Click **Edit -> EOL Conversion -> UNIX/OSX Format**
- Create a new empty file with the **Notepad++** and save it in the boot folder of the SD card with the name **ssh** (without any extension), where this file will enable the SSH to be used to access your Raspberry Pi remotely. On Windows, this is the only folder you will see which contains items like: loader.bin, start.elf, kernel.img etc.
- Re-insert the SD card back into your Raspberry Pi
- Connect your Raspberry Pi to one of the ports of your Wi-Fi router through an Ethernet cable and power up
- Find out the IP address allocated to your Raspberry Pi by accessing your Wi-Fi router. Alternatively, install the **Advanced IP Scanner** program on your PC, available using the following link:

  https://www.advanced-ip-scanner.com

- Run the software and look for your Raspberry Pi. You do not have to install the software to run it. Click **Run portable version**, and then **Scan**. As shown in Figure 1.5, the IP address of the author's Raspberry Pi was 191.168.1.202

Figure 1.5 IP Address of the Raspberry Pi

- You can now use Putty to log in to your Raspberry Pi (see Section 1.4 and 1.5)

You can find the IP address of your Raspberry Pi by entering the command prompt on your PC with administrator privilege (by right-clicking and run as an administrator) and then entering the command: **ping raspberrypi.home** as shown in Figure 1.6

```
C:\WINDOWS\system32>ping raspberrypi.home

Pinging raspberrypi.home [192.168.1.202] with 32 bytes of data:
Reply from 192.168.1.202: bytes=32 time=1ms TTL=64
Reply from 192.168.1.202: bytes=32 time=2ms TTL=64
Reply from 192.168.1.202: bytes=32 time=2ms TTL=64
```

Figure 1.6 Using ping to find the Raspberry Pi IP address

It is also possible to find the IP address of your Raspberry Pi using your smartphone. There are many apps that can be used to find out who is using your Wi-Fi router. e.g. **Who's On My Wi-Fi – Network Scanner by Magdalm**.

Connection using Wi-Fi: This is perhaps the preferred method to access your Raspberry Pi. This is the method used by the author. Here, as described in Chapter 1, the Raspberry Pi can be placed anywhere within the reach of the Wi-Fi router and can be easily accessed via your PC using Putty (see Section 1.4 and 1.5).

The steps are:

- Install **Notepad++** on your PC from the following web site:

  https://notepad-plus-plus.org/downloads/v7.8.5/

- Re-insert the SD card into your PC and start **Notepad++**
- Click **Edit -> EOL Conversion -> UNIX/OSX Format**
- Create a new empty file with the **Notepad++** and save it in the boot folder of the SD card with the name ssh (without any extension). This file will enable SSH to be used to remotely access your Raspberry Pi. In Windows this is the only folder you will see which contains items like: loader.bin, start.elf, kernel.img etc.
- Enter the following statements into a blank file (replace **MySSID** and **MyPassword** with the details of your own Wi-Fi router):

```
country=GB
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
        scan_ssid=1
        ssid="MySSID"
        psk="MyPassword"
}
```

- Copy the file (save) to the boot folder of your SD card with the name: **wpa_supplicant.conf**
- Re-insert the SD card into your Raspberry Pi and power-up the device
- Use the **Advanced Ip Scanner** or one of the other methods described earlier to find out the IP address of your Raspberry Pi.
- Log in to your Raspberry Pi remotely using **Putty** on your PC (see Section 1.3 and 1.4)
- After logging in, you are advised to change your password for security reasons. You should also run **sudo raspi-config** from the command line to enable VNC, I2C, and SPI – these are useful interface tools which can be used in your future GPIO based work.

1.4 ● Remote access

It is much easier to access the Raspberry Pi remotely over the Internet, for example by using a PC rather than connecting a keyboard, mouse, and display to it. Before being able to remotely access the Raspberry Pi, we have to enable SSH by entering the following command on a terminal session (if you have followed the steps given earlier, SSH is already enabled and you can skip the following command):

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

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

You should also enable VNC so the Raspberry Pi desktop can be accessed graphically over the Internet. This can be done by entering the following command on 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 shut down or restart your Raspberry Pi by entering one of the following commands in command mode:

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

or

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

1.5 ● Using Putty

Putty is a communications program used to create a connection between your PC and Raspberry Pi. This connection uses a secure protocol called SSH (Secure Shell). Putty doesnt need to be installed and can be stored and run from any folder of your choosing.

Putty can be downloaded from the following web site:

https://www.putty.org/

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 1.7). The message shown in Figure 1.8 will be displayed the first time you access the Raspberry Pi. Click **Yes** to accept this security alert.

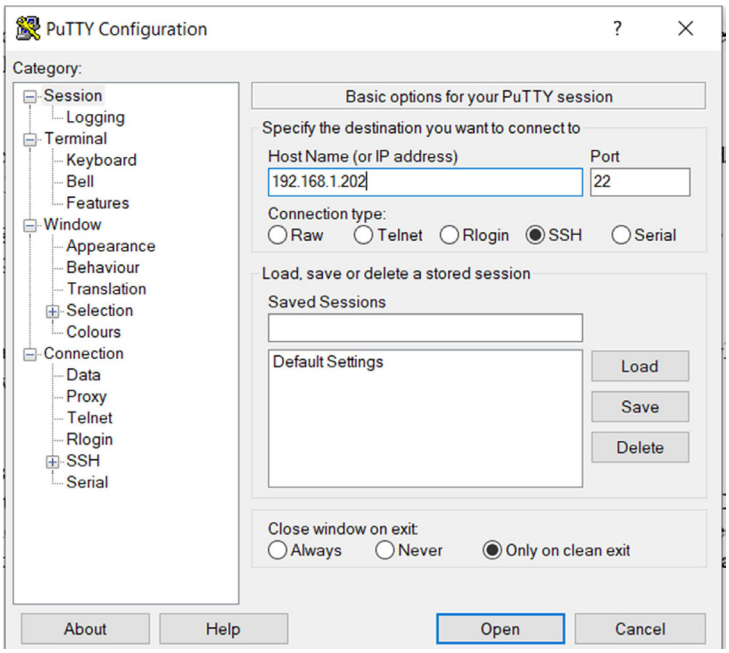


Figure 1.7 Putty startup screen

You will be prompted to enter a username and password. The default username and password are:

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

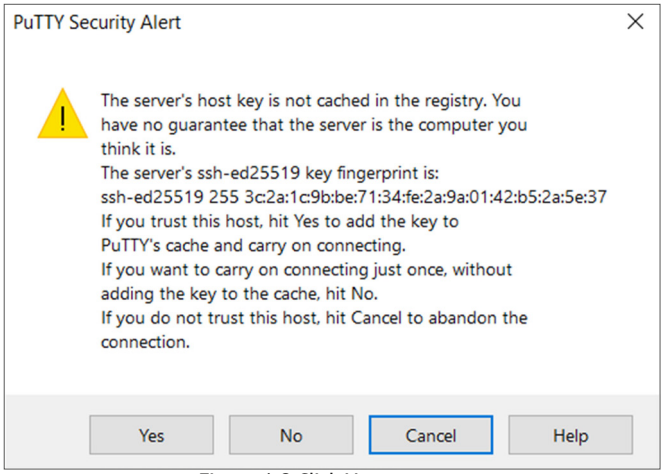


Figure 1.8 Click Yes to accept

You now have a terminal connection to the Raspberry Pi. You can type in commands, including **sudo** privileged administrative commands.

To change your password, enter the following command:

passwd

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 (not graphical programs).

1.5.1 ● Configuring Putty

By default, the Putty background screen is black with white foreground characters. The author prefers to have white background with black foreground characters, with the character size set to 12 points bold. The steps to configure Putty with these settings are provided below. In this example these settings are saved with the name **RPI4** and can therefore be recalled whenever Putty is restarted:

- 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 11**.
- 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 restart Putty, select the saved session and click **Load** followed by **Open** to start a session with the saved configuration.

1.6 ● Desktop remote access

You can control your Raspberry Pi via Putty, and run programs on it from your Windows PC. This will not work with graphical programs because Windows does not know how to represent the display. As a result of this, for example, we cannot run any graphical programs in Desktop mode. We can get around this problem using some extra software. Two popular software suites used for this purpose are: VNC (Virtual Network Connection), and Xming. Here, we will learn about 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 and is enabled as described in Section 1.3 using **raspi-config**.

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

- There are many VNC Viewers available. The recommended one is TightVNC which can be downloaded from the following website:

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

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

- Enter the following command:

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

- Start **TightVNC Viewer** on your PC and enter the Raspberry Pi IP address (see Figure 1.9) followed by :1. Click **Connect** to connect to your Raspberry Pi.

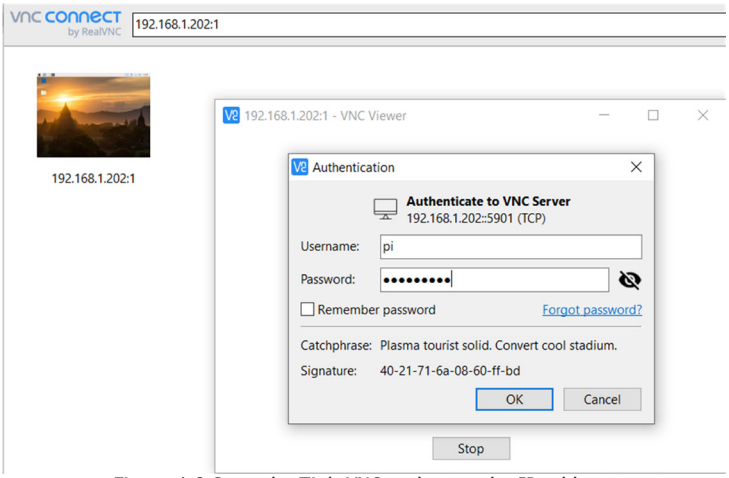


Figure 1.9 Start the TightVNC and enter the IP address

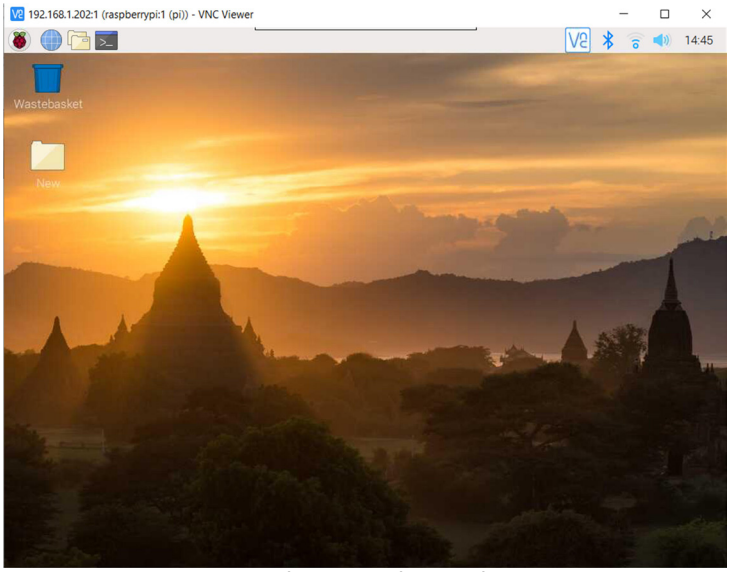


Figure 1.10 Raspberry Pi Desktop on the PC screen

Figure 1.10 shows the Raspberry Pi Desktop displayed on the PC screen.

1.7 ● Static IP address

When we use the Raspberry Pi with a Wi-Fi router the IP address is automatically allocated by the router. It is possible that every time we start the Raspberry Pi the Wi-Fi router will give the Pi another IP address. This makes it difficult to log in as we have to find the new IP address.

We can give our Raspberry Pi a static IP address so that it is allocated the same ip address by the Wi-Fi router. The IP address is given by the DHCP protocol running on the Wi-Fi router.

Before setting a static IP address, we have to decide what this address will be and also make sure that no other devices on our network use this address. We can check this by either logging in to our Wi-Fi router or by displaying the devices on our network using an app on our smart phone.

The steps to assign a static IP address are as follows:

- First, check that **dhcpcd** service is active by entering the following command:

```
pi@raspbberrypi:~ $ sudo service dhcpcd status
```

You should see **text active: (running)** displayed as shown in Figure 1.11 (only part of the display is shown). Enter **Ctrl+C** to exit from the display.

```
pi@raspberrypi:~ $ sudo service dhcpcd status
● dhcpcd.service - dhcpcd on all interfaces
   Loaded: loaded (/lib/systemd/system/dhcpcd.service; enabled; vendor pres
   Active: active (running) since Thu 2020-06-18 23:06:12 BST; 2 weeks 5 da
  Process: 375 ExecStart=/usr/lib/dhcpcd5/dhcpcd -q -b (code=exited, status
 Main PID: 416 (dhcpcd)
    Tasks: 2 (limit: 4035)
   Memory: 4.5M
```

Figure 1.11 Check DHCP running

- If **dhcpcd** is not running, enter the following commands to activate it:

```
pi@raspbberrypi:~ $  sudo service dhcpcd start
pi@raspbberrypi:~ $  sudo systemctl enable dhcpcd
```

- Now, we need to find the IP address (Default Gateway) and the Domain Name Server address of our router. This can easily be obtained either from our Wi-Fi router, or PC.

The steps to obtain these addresses on a PC are:

- Go to **Control Panel** on your Windows 10 PC.
- Click **Network and Sharing Centre.**
- Click **Internet** as shown in Figure 1.12.

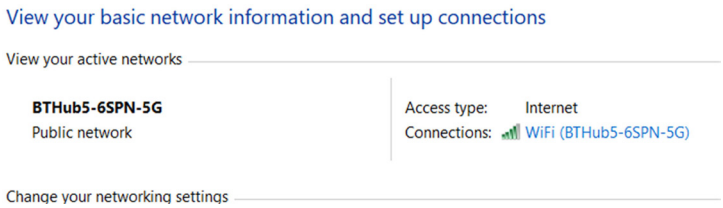


Figure 1.12 Click Internet

- Click **Details**. You will see a screen similar to the one shown in Figure 1.13 where you can see the Default Gateway and DNS server addresses. In this example they are both: 191.168.1.254.

DHCP Enabled	Yes
IPv4 Address	192.168.1.199
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	08 July 2020 08:10:45
Lease Expires	09 July 2020 12:27:53
IPv4 Default Gateway	192.168.1.254
IPv4 DHCP Server	192.168.1.254
IPv4 DNS Server	192.168.1.254

Figure 1.13 Click Details

- You will have to edit the following file: **/etc/dhcpcd.conf** using a text editor such as **nano**. Although you may not be familiar with **nano**, just follow the instructions (**nano** is described in a later Chapter).

```
pi@raspbberrypi:~ $  sudo nano /etc/dhcpcd.conf
```

- Go to the end of the file using the down arrow key and enter the following lines:

```
interface wlan0
static ip_address=191.168.1.120/24
static routers=191.168.1.254
static domain_name_servers=191.168.1.254

interface eth0
static ip_address=191.168.1.120/24
static routers=191.168.1.254
static domain_name_servers=191.168.1.254
```

In this example, we have selected the static IP address as 191.168.1.120 after making sure there are no other devices on our network with the same IP address. The suffix /24 is an abbreviation of the subnet mask 255.255.255.0 and you have to make sure that you only change the last digit of the IP address. i.e. choose an address in the form 191.168.1.x. **wlan0** is for the Wi-Fi link, and **eth0** is for the Ethernet link.

- Save the file by entering **Ctrl+X**, followed by **Y.**
- Display the file on your screen to make sure that the changes you have made are correct. Enter the command:

```
pi@raspberrypi:~ $ cat /etc/dhcpcd.conf
```

- Reboot your Raspberry Pi. It should come up with the IP address set as required.

1.8 ● Summary

In this chapter we learned how to install the latest Raspberry Pi operating system on an SD card. We also learned how to use the Raspberry Pi remotely. The instructions given here are applicable to all versions of Raspberry Pi. Additionally, setting the static IP address of your Raspberry Pi is demonstrated.

In the next chapter, we will look at some of the Raspberry Pi program development tools.

Chapter 2 ● Raspberry Pi Program Development

2.1 ● Overview

In the last chapter we learned how to install Raspbian Buster on a Raspberry Pi SD card. In this chapter, we will look at how to develop programs using Raspberry Pi 4. We will be using the Python 3 programming language in this book. Although Raspberry Pi 4 is used by the author, other models of Raspberry Pi can also be used so long as Python 3 is installed. Notice the characters entered by the user are in bold for clarity.

2.2 ● The nano text editor

A text editor is a useful tool for creating program files. Raspberry Pi supports a number of text editors such as **vi**, **nano** etc. In this section, we will introduce the simple to use **nano** text editor which is normally run from the command line.

As an example, suppose that we wish to create a text file called myfile.txt and insert the following lines into this file:

This is a simple text file created using nano
This is the second line of the file
This is the third line of the file

The steps are as follows:

- Start the nano editor.

```
pi@raspberrypi:~ $ nano myfile.txt
```

- Enter the above text into the file (see Figure 2.1). You should see a number of control codes at the bottom of the screen.

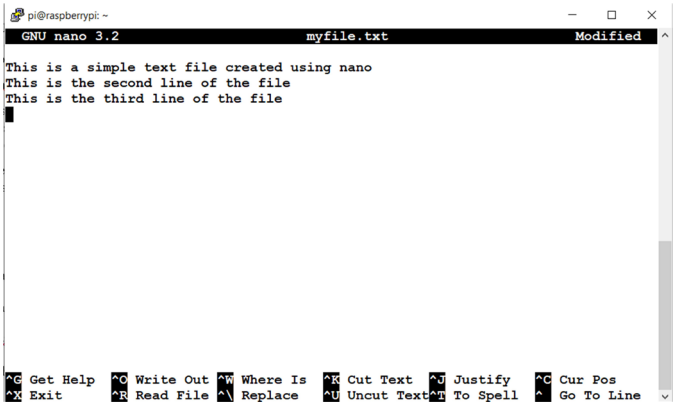


Figure 2.1 Text entered into the editor

- Enter **Ctrl+X** followed by the letter **Y** to save the file. You should now see the file listed in your directory if you enter the command:

```
pi@raspberrypi:~ $ ls myfile.txt
myfile.txt
pi@raspberrypi:~ $
```

- Let us now edit the file we just created in order to learn some of the editor commands. Restart **nano** as above by specifying the filename.
- Let us search for text starting with the word **simple**: press **Ctrl+W**, type **simple** and press **Enter** (see Figure 2.2). You should see the cursor moving to the start of word **simple**. Delete **simple** and change it to **difficult.**

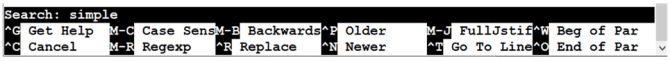


Figure 2.2 Search for word simple

- Let us replace the word **third** with fourth: press **Ctrl+** and type **third**, and then type **fourth** when **Replace with**: is displayed. Press **Enter**. The message **Replace this instance?** will be displayed. Type **Y**. You should see that word **third** is replaced with word fourth.
- Let us delete the second line of text: Move the cursor to the second line and enter **Ctrl+K**. You should see that all the text in the second line is deleted.
- To recall the deleted line, enter **Ctrl+U.**
- To get help on using nano, enter **Ctrl+G**. An example help screen is shown in Figure 2.3. Enter **Ctrl+N** to display the next page, and **Ctrl+P** to display the previous page. Enter **Ctrl+X** to close the help screen.

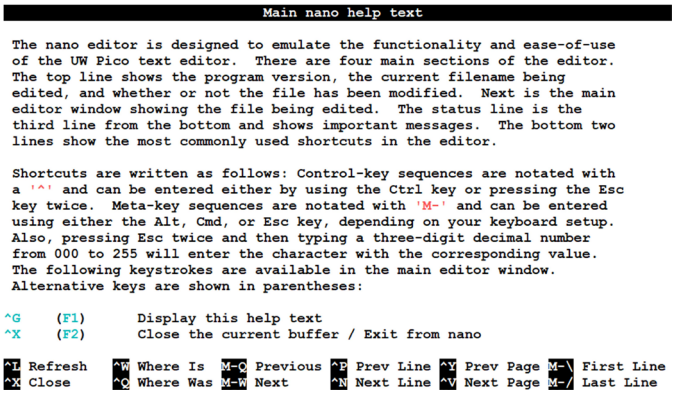


Figure 2.3 Example help screen

- Enter **Ctrl+X** followed by **Y** to save and exit the editor.
- The contents of the edited file are shown in Figure 2.4.

```
pi@raspberrypi:~ $ cat myfile.txt
This is a difficult text file created using nano
This is the second line of the file
This is the fourth line of the file

pi@raspberrypi:~ $
```

Figure 2.4 Contents of the edited file

As a summary, some of the more useful nano editor shortcuts are given below:

Ctrl+W: Search for a word

Ctrl+V: Move to next page

Ctrl+Y: Move to previous page

Ctrl+K: Cut the current row of txt

Ctrl+R: Read file

Ctrl+U: Paste the text you previously cut

Ctrl+J: Justify

Ctrl+: Search and replace text

Ctrl+C: Display current column and row position

Ctrl+G: Get detailed help on using the nano

Ctrl+-: Go to specified line and column position

Ctrl+O: Save (write out) the file currently open

Ctrl+X: Exit nano

2.3 ● Creating and running a Python program

We will be programming our Raspberry Pi 4 using Python 3. It is worthwhile to look at the creation and running of a simple Python program on our Raspberry Pi computer. In this section we will display the message **Hello From Raspberry Pi 4** on our PC screen.

As described below, there are three methods we can employ to create and run Python programs on our Raspberry Pi 4:

Method 1 – Interactively from command mode

In this method, we will log in to our Raspberry Pi 4 remotely using SSH and then create and run our program interactively in command mode. This method is excellent for small programs. The steps are as follows:

- Log in to the Raspberry Pi 4 using SSH (or through a monitor and keyboard directly connected).
- On the command prompt, enter **python3**. You should see the Python command mode which is identified by three characters: >>>
- Type the program:

```
print ("Hello From Raspberry Pi 4")
```

- The required text will be displayed interactively on the screen as shown in Figure 2.5. Enter **Ctrl+Z** to exit Python.

```
pi@raspberrypi:~ $ python3
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("HEllo from Raspberry Pi 4")
HEllo from Raspberry Pi 4
>>>
```

Figure 2.5 Running a program interactively

Method 2 – Create a Python program in command mode

In this method, we will log in to our Raspberrry Pi 4 using SSH as before and then create a Python file. A Python file is simply a text file with the extension **.py**. We can use a text editor, e.g. **nano** to create our file. In this example a file called **hello.py** is created using nano. Figure 2.6 shows the contents of file **hello.py**. This figure also shows how to run the file using Python 3. Notice that the program is run by entering the command:

```
pi@raspberrypi:~ $ python3 hello.py
```

```
pi@raspberrypi:~ $ cat hello.py
print("Hello from Raspberry 4")

pi@raspberrypi:~ $ python3 hello.py
Hello from Raspberry 4
pi@raspberrypi:~ $
```

Figure 2.6 Creating and running a Python program

Method 3 – Create a Python program in Desktop GUI mode

In this method, we will log in to our Raspberry Pi 4 in desktop mode using **VNCViewer** (if we do not have a monitor directly connected) and create and run our program in GUI mode. We will be using a program called **Thonny** which is used to create, debug, and run Python

3 programs. **Thonny** is an easy to use tool that is only available for Python 3. The nice thing about **Thonny** is that it correctly formats code while it is entered from the terminal. For example, all the statements in the body of a **while** loop are correctly automatically indented.

The steps to use **Thonny** are provided below:

- Click the **Applications menu**, then **Programming**. Select **Thonny Python IDE** as shown in Figure 2.7

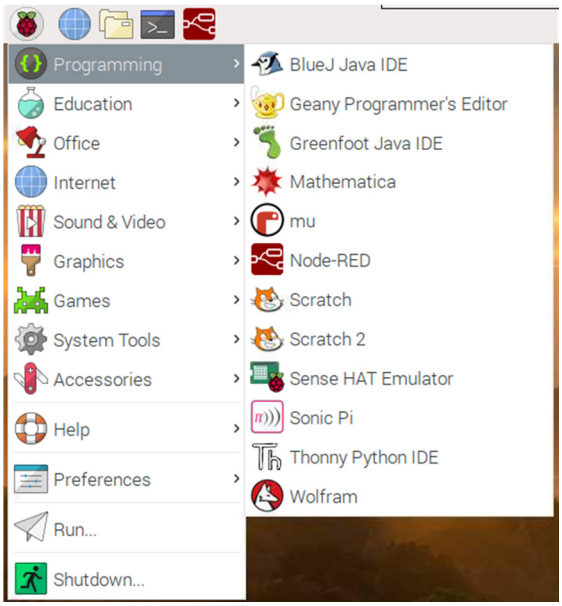


Figure 2.7 Select Thonny Python IDE

- The Thonny startup screen will be displayed as shown in Figure 2.8. The screen is in two parts: The program is written in the upper part. The lower part is the **shell** where the results of the program are displayed. We can also run Python 3 commands interactively on the lower part of the screen. In the upper part we have the usual menu items found in most GUI type displays. Menu option **File** is used to create a new file, to open an existing file, to close, save, or print a file. Menu option **Edit** is used to undo, cut, paste, select, find and replace and so on. Option **View** is used to to view files, heap, notes, stack, variables and so on. Menu option **Run** is used to run or debug a program. Menu option **Device** is used to soft reboot, to upload current script as main script and so on. Menu option **Tools** is used to manage packages, manage plug-ins, to configure Thonny, and so on. Finally, the **Help** menu option is used to get help on using the Thonny.

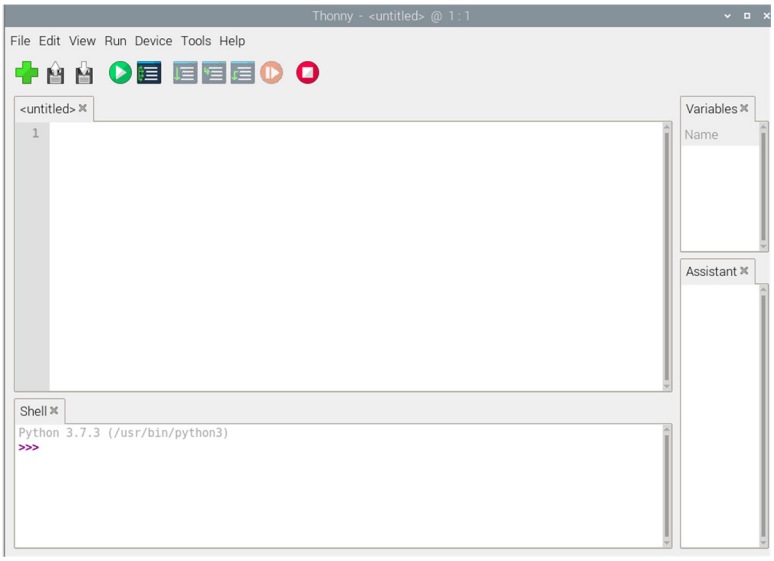


Figure 2.8 Thonny startup screen

- Type your program in the upper part as shown in Figure 2.9.

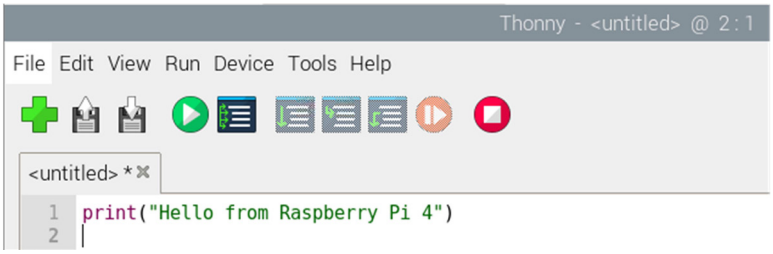


Figure 2.9 Type your program

- Click **File** and save your program by giving it a name. You do not have to specify the file extension as this is added automatically by Thonny.
- Click **Run** and you should see the program output on the lower part of the screen as shown in Figure 2.10.

Figure 2.10 Output of the program

Thonny provides the option of debugging a program, where we can single step through a program and display the variables as the program is stepped through. As an example, let us debug the program given in Figure 2.9. The steps are:

- Click **Run** and then **Debug current script** (nicer).
- You should see the current program line highlighted in yellow.
- We now have the options of: **Step over**, **Step into**, and **Step out.**
- Clicking **Step over** will step through the program lines as we see them on the screen. Click this button and you should see the output of the program displayed on the lower.
- While in Debug mode, you can also **Resume** (the orange and white icon) the program so that it continues normally, or **Stop and Restart** (the red and white icon) the program from the beginning.

Which Method?

The choice of a method depends upon the size and complexity of the program we wish to develop. Small programs can be run interactively without creating a program file. Larger programs can be created as Python files either by using **nano** in command mode, or **Thonny** to create them in Desktop GUI mode. The author has chosen to use **nano** during the development of all programs in this book, unless specified otherwise.

2.4 ● Summary

In this chapter we learned how to develop Python 3 programs using several methods. The choice of method depends entirely on the user.

In the next chapter, we will look at General Purpose Input Output (GPIO) and develop some simple programs to illustrate how GPIO can be accessed using Python programs.

Chapter 3 • GPIO

3.1 • Overview

In the last chapter we looked at Raspberry Pi program development tools and learned how to create and run a very simple program. In this chapter, we will determine the details of the GPIO (General Purpose Input Output) header connector and how to interface simple devices to GPIO.

3.2 • The Raspberry Pi 4 GPIO connector

Before going into detail of the hardware interface, it is worthwhile to look at the Raspberry Pi 4 GPIO connector. This is a 40-pin, dual-in-line 2.54mm connector as shown in Figure 3.1. Other recent Raspberry Pi models have similar connectors with almost the same pin configuration.

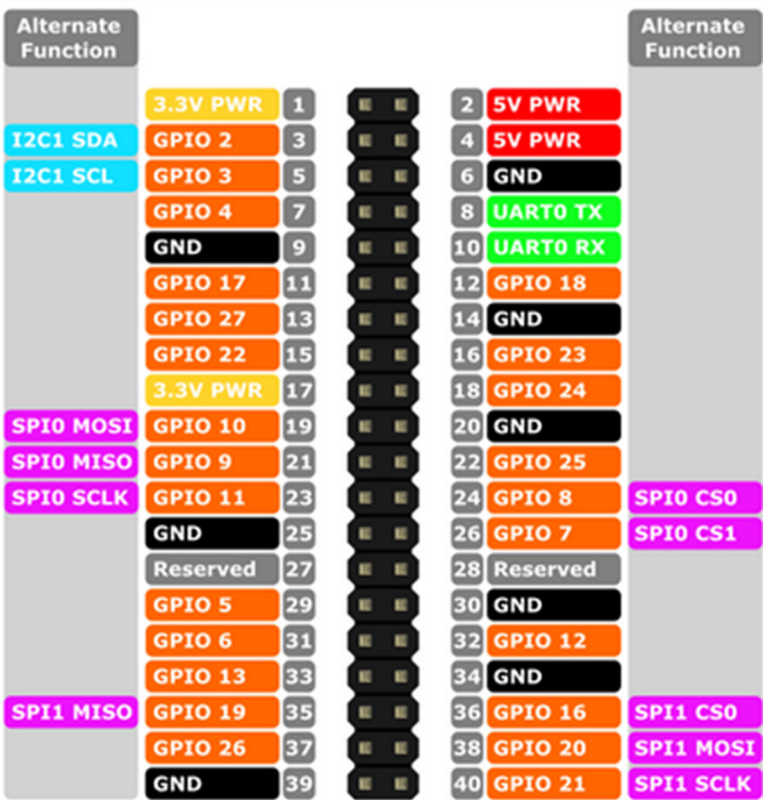


Figure 3.1 Raspberry Pi 4 GPIO connector

When the GPIO connector is on the far side of the board, the pins on the bottom, starting from the left of the connector are numbered as 1, 3, 5, 7, and so on, while the ones on the