elektor books

C Programming on Raspberry Pi

Develop innovative hardware-based projects in C

Dogan Ibrahim

elektor
design > share > sell

C Programming on Raspberry Pi

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: 978-3-89576-431-8

● EISBN: 978-3-89576-432-5

● EPUB: 978-3-89576-433-2

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

To my wife Nadire, my daughter Alev, and my son Ahmet, for their love and wisdom.

• Preface

The Raspberry Pi 4 is the latest credit-card sized computer that can be used in many applications, such as audiovisual media centers, desktop computers, industrial control, robotics, and many more 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.

The Raspberry Pi has traditionally been programmed using Python. Although Python is a very powerful language, many programmers may not be familiar with using it. The C language is probably the most commonly used programming languages. All embedded microcontrollers can be programmed using the C language these days. The C language is taught in all technical colleges and universities - almost all engineering students are familiar with the use of this language in their projects.

This book is about using C with Raspberry Pi to develop various hardware-based projects. Two of the most popular C libraries, wiringPi and pigpio are used.

The book starts with an introduction to the C language and most students and newcomers will find this chapter invaluable. Many projects are provided in the book, including using Wi-Fi and Bluetooth to establish communication with smartphones.

The book includes many sensors and hardware-based projects. Both wiringPi and pigpio libraries are used in all projects. Complete program listings are given with full explanations. All projects given 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

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

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

Prof Dr Dogan Ibrahim
January 2021
London.

Table of Contents

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. Notice the installation process given below applies to all Raspberry Pi models unless otherwise specified.

1.2 • Raspbian Buster installation steps on Raspberry Pi 4

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

The steps to install the Raspbian Buster operating system 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 this book, the file was called: **2020-02-13-raspbian-buster-full.img**. You may have to use the Windows 7Zip software to unzip the download due to some features not being supported by older zip software.

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

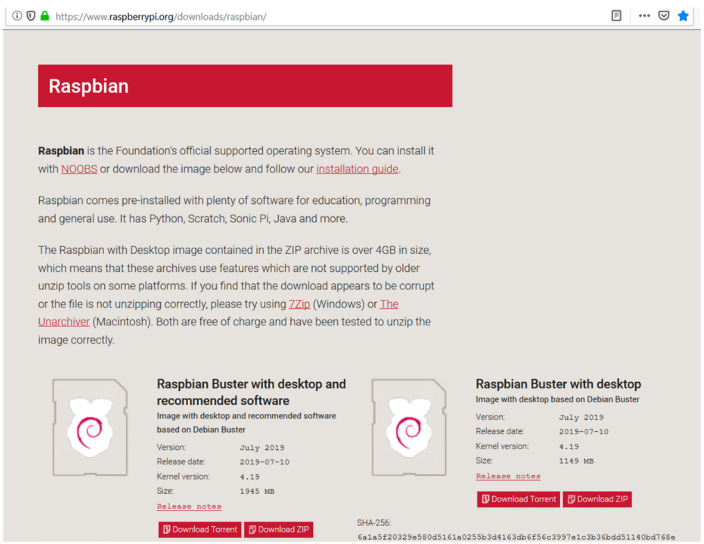


Figure 1.1 Raspbian Buster download page

- Put the blank micro SD card into the card slot of your computer. You may need an adapter to do this.
- Download Etcher to your PC to flash the disk image. The link is (see Figure 1.2):

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

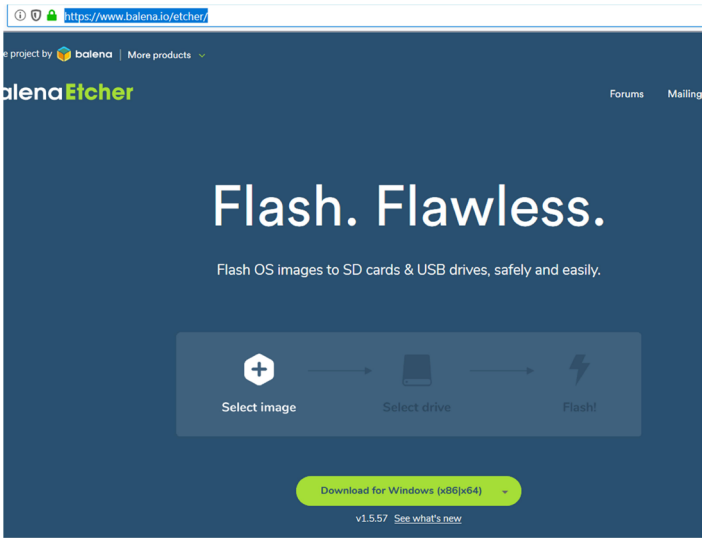


Figure 1.2 Download Etcher

- Double click to open Etcher and then click **Select image**. Select the Raspbian Buster file you 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.

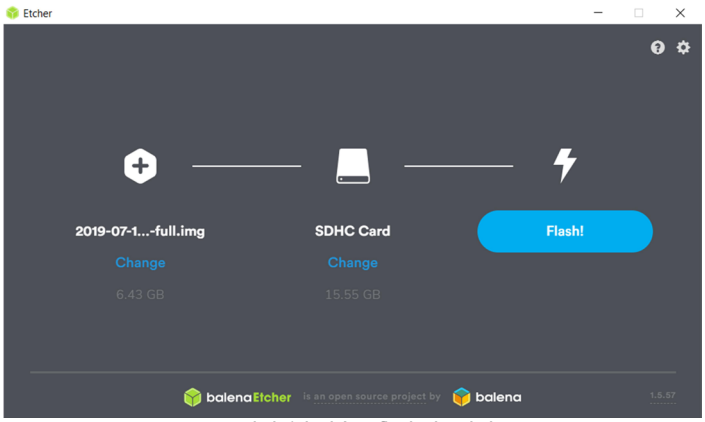


Figure 1.3 Click 'Flash' to flash the disk image

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

Using 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 while, you will be prompted to enter the login details. The default values are username: **pi**, password: **raspberry**.

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

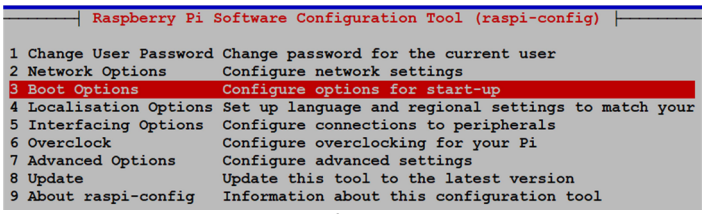


Figure 1.4 Select Boot Options

- 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 either to access it remotely from a PC or to use the internet. If your Raspberry Pi is equipped with an ethernet port (e.g. Raspberry Pi 2/3/4), you can directly connect 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 Wi-Fi and access it remotely.

You will need to enable SSH. The steps are as follows:

- Start the configuration tool:

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

- Move down to **Interface Options** and 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 Wi-Fi. Note the IP address allocated automatically to your Raspberry Pi.
- You can now remotely access your Raspberry Pi using terminal emulation software, such as **Putty** (see Section 1.4 and 1.5).

1.3 ● Using networked connection

If you do not have a suitable monitor and keyboard to directly connect to your Raspberry Pi, you will have to use a networked connection and remotely access your Raspberry Pi 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 as follows:

- 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 **Notepad++**.
- Click **Edit -> EOL Conversion -> UNIX/OSX Format**.
- Create a new empty file with the **Notepad++** and save it to the boot folder of the SD card with the name ssh(without any extension), where 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 including loader.bin, start.elf, kernel.img, etc.
- 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 it up.
- Find out the IP address allocated to your Raspberry Pi by accessing your Wi-Fi router. Alternatively, install **Advanced IP Scanner** on your PC, which is available at 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)

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

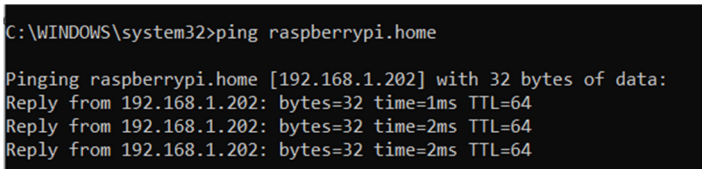


Figure 1.6 Using ping to find the Raspberry Pi IP address

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

Connection using Wi-Fi: This is the preferred method to access your Raspberry Pi and is the one used by the author. Here, as described in Chapter 1, the Raspberry Pi can be placed anywhere you like within the range of the Wi-Fi router and is easily accessed from 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/

- Insert the SD card back to your PC and start **Notepad++**.
- Click **Edit -> EOL Conversion -> UNIX/OSX Format**
- Create a new empty file with **Notepad++** and save it to the boot folder of the SD card with the name **ssh**(without any extension), where 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 the **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 on your SD card with the name: **wpa_supplicant.conf**.
- Insert the SD card back into your Raspberry Pi and power-up the device.
- Use **Advanced Ip Scanner** or one of the 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 **sudoraspi-config** from the command line to enable VNC, I2C, and SPI as they are useful interface tools that can be used in your future GPIO based work.

1.4 ● Remote access

It is much easier to remotely access the Raspberry Pi over the internet: for example 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 in 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 in a terminal session:

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

Go to the configuration menu and select **Interface Options**. Go down to **P3 VNC** and enable VNC. Click **<Finish>** to exit the menu. At this stage you may want to shut down 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 doesn't need to be installed and can be stored in any folder of your choice and run from there.

Putty can be downloaded from the following web site:

https://www.putty.org/

Simply double click to run it and the Putty startup screen will be displayed. Click **SSH** and enter the Raspberry Pi IP address, then click **Open** (see Figure 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.

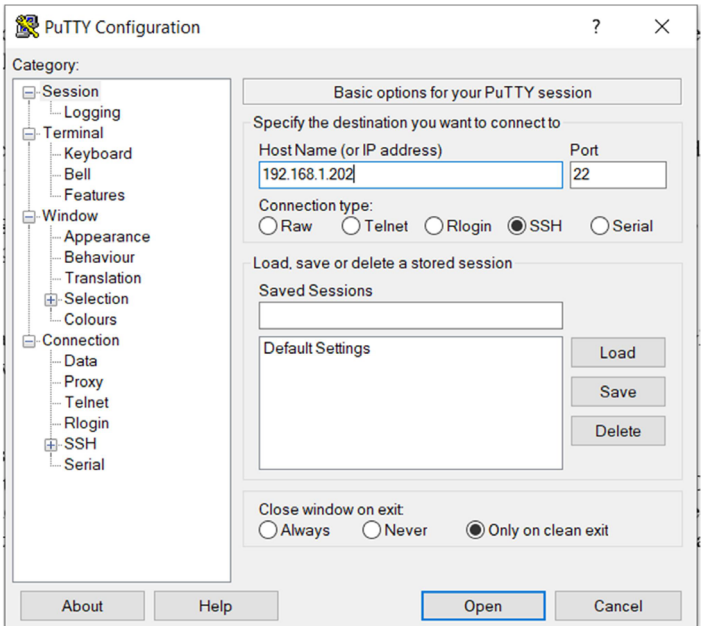


Figure 1.7 Putty startup screen

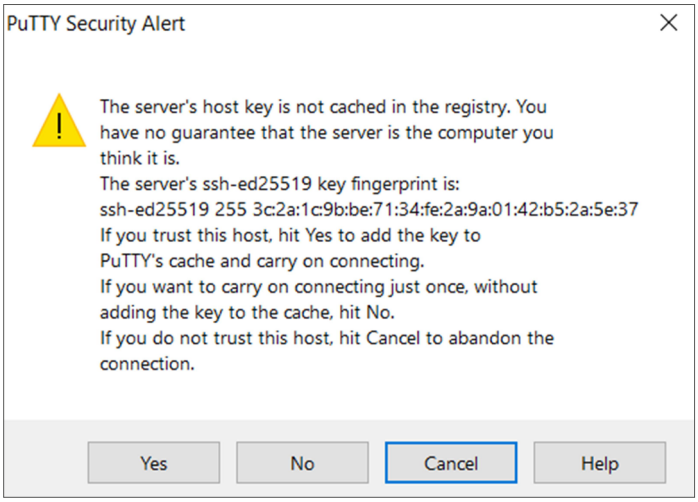


Figure 1.8 Click Yes to accept

You will be prompted to enter the username and password. Notice the default username and password are:

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

You now have a terminal connection with the Raspberry Pi and can type in commands, including the **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 although not graphical programs.

1.5.1 ● Configuring Putty

By default, the Putty screen background 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 the Putty with these settings are given below. Notice that in this example these settings are saved with the name **RPI4** so that they can be recalled whenever the 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 re-start the Putty, select the saved session and click **Load** followed by **Open** to start a session with the saved configuration.

1.6 • Remote access of the Desktop

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

Installing and using VNC

VNC consists of two parts: VNC Server and the VNC Viewer. VNC Server runs on the Raspberry Pi, and the 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 given below:

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

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

- Download and install **TightVNC** for 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.

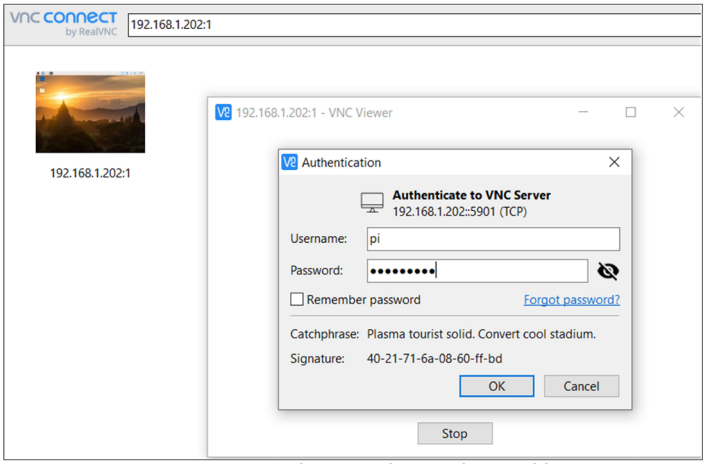


Figure 1.9 Start TightVNC and enter the IP address

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

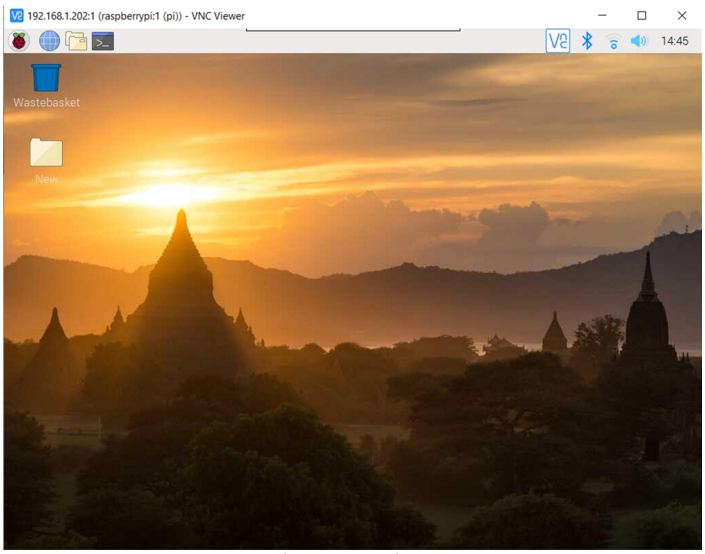


Figure 1.10 Raspberry Pi Desktop on a PC screen

1.7 ● Static IP address

When we are using 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 before we log in.

We can give our Raspberry Pi a static IP address so that every time it starts, the same IP

address is allocated from 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 logging in to the Wi-Fi router or by displaying the devices on our network using an app on a smartphone.

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

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

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

You should see the 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
```

- We now 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 from 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.

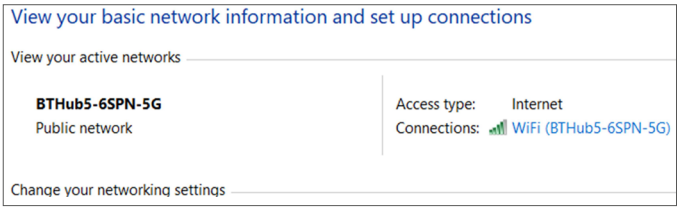
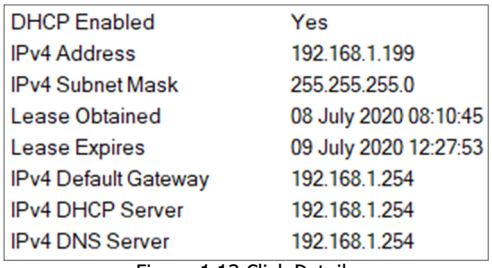


Figure 1.12 Click Internet

- 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**, follow the instructions given here (**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
staticdomain_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 chose a static IP address of 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. You have to make sure 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.

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

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

- Reboot your Raspberry Pi and 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, and also how to start using the Raspberry Pi remotely. The instructions provided apply to all versions of Raspberry Pi. Additionally, setting a static IP address for your Raspberry Pi is demonstrated.

In the next chapter, we will look at various 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 learn how to develop programs using Raspberry Pi 4. We will develop a very simple example project and learn how to use Python and C. Although Python is not the topic of this book, it will be used in a simple project so that readers can compare Python with C programming.

2.2 ● The nano text editor

A text editor is a very useful tool for creating program source files. Raspberry Pi supports several text editors including **vi**, **nano**, etc. In this section, we will introduce **nano** which is normally run from the command line.

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

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

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

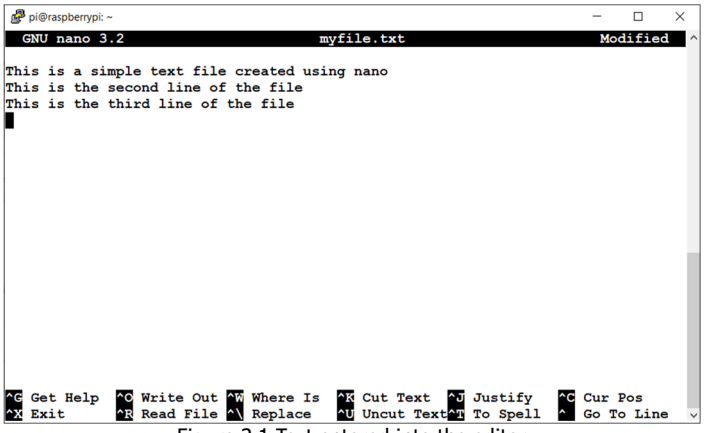


Figure 2.1 Text entered into the editor

- 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 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** and type **simple** and press **Enter** (see Figure 2.2). You should see the cursor moving to the start of the word **simple**. Delete **simple** and change it to **difficult**.

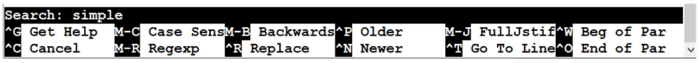


Figure 2.2 Search for word simple

- Let us replace the word **third** with **fourth**: press **Ctrl+** and type **third**, and then **fourth** when **Replace with:** is displayed. Press **Enter**. The message **Replace this instance?** will be displayed. Type **Y**. You should see that the word **third** is replaced with **fourth**.
- Let us delete the second line of text: Move the cursor to the second line and enter **Ctrl+K**. You should see all the text on the second line is deleted.
- To recall the line just deleted, 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.

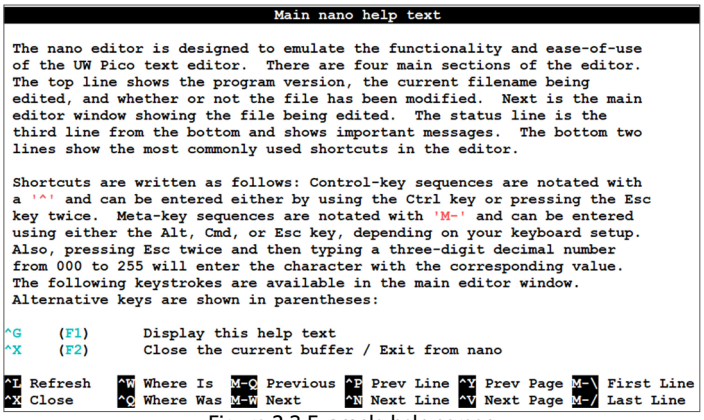


Figure 2.3 Example help screen

- 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** shortcuts are given below:

Ctrl+W: Search for a word.

Ctrl+V: Move to the next page.

Ctrl+Y: Move to the previous page.

Ctrl+K: Cut the current row of text.

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 nano.

Ctrl+-: Go to a specified line and column position.

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

Ctrl+X: Exit nano.

2.3 ● Example project

In this chapter, we will develop a simple project using both Python and C which will display the message **Hello From Raspberry Pi 4** on your PC screen. This project aims to show how a project can be created and then run on Raspberry Pi.

2.4 ● Creating and running a Python program on Raspberry Pi

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

Method 1 – Interactively from command mode

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

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

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

- The text required 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 Python program interactively

Method 2 – Create a Python program in command mode

In this method, we will log in to our Raspberry Pi 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 **hello.py**. This figure also shows how to run the file using Python 3. Notice 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

Using this method, we will log in to our Raspberry Pi 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 **Thonny** which is used to create, debug, and run Python 3 programs.

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

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

- Click **Applications menu**, then **Programming**, and select **Thonny Python IDE** as shown in Figure 2.7

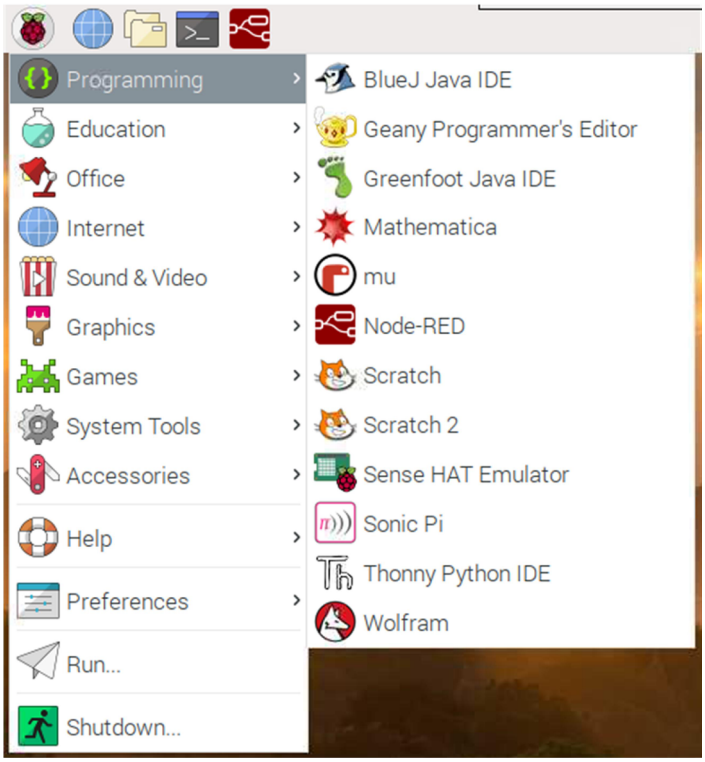


Figure 2.7 Select Thonny Python IDE

- The Thonny startup screen will be displayed as shown in Figure 2.8. The screen has 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 in 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, open an existing file, close, save, or print a file. Menu option **Edit** is used to undo, cut, paste, select, find, replace, and so on. Option **View** is used to enable us 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, upload a current script as main script, and so on. Menu option **Tools** is used to manage packages, manage plug-ins, configure Thonny, and so on. Finally, the **Help** menu option is used to get help on using Thonny.

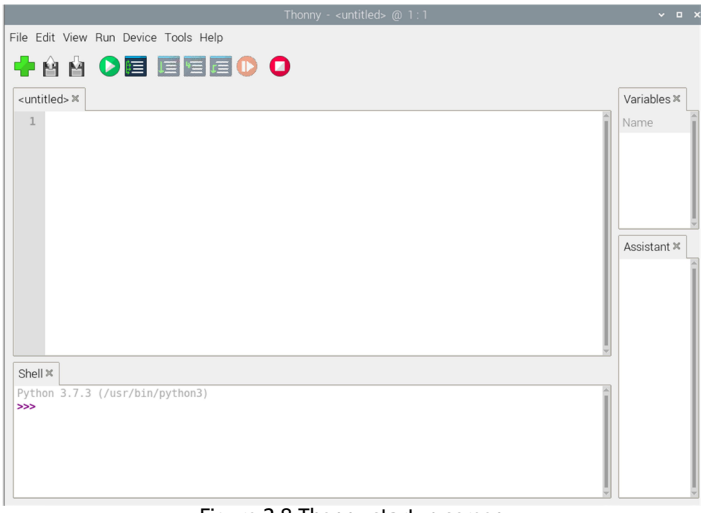


Figure 2.8 Thonny startup screen

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

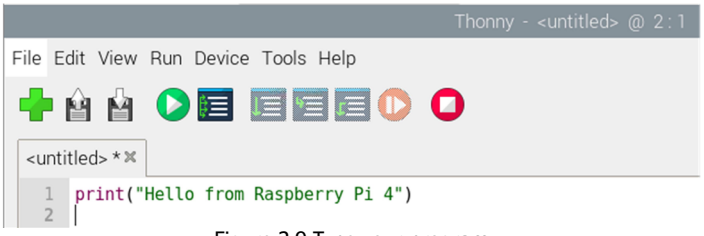


Figure 2.9 Type your program

- Click **File** and save your program by giving it a name. You do not have to specify the file extension as this is automatically added 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 to debug 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 and you should see the output of the program displayed on the lower part of the screen.
- While in Debug mode, you can also **Resume** (the orange and white icon) the program so it continues normally, or **Stop and Restart** (the red and white icon) the program from the beginning.

Method 4 – Using the Mu editor

In this method, we will use **Mu** to create and run our Python example program. **Mu** is a very easy to use Python editor and Integrated Development Environment (IDE) for beginners.

Mu is installed by default with the Raspbian Buster operating system and can be started by clicking the Applications Menu, followed by **Programming** and then **Mu** (see Figure 2.11).

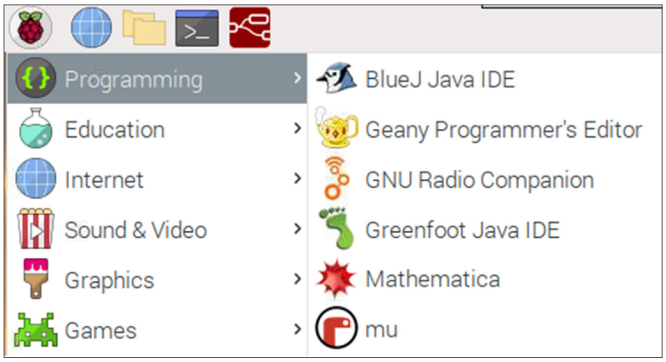


Figure 2.11 Starting Mu

If not available, Mu can be installed using the following steps:

- Click **Applications Menu ->Preferences ->Recommended Software** (Figure 2.12).