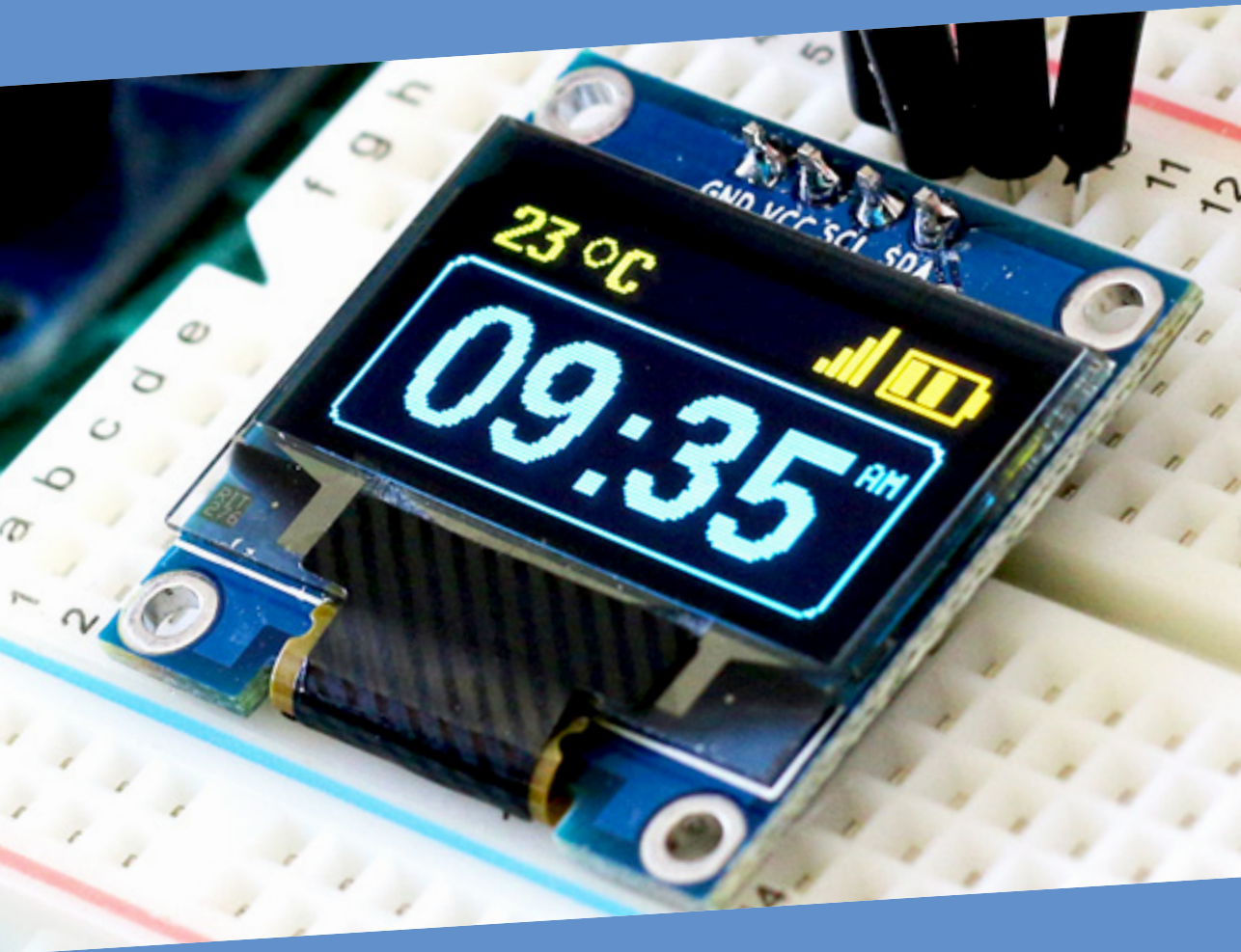


# Using Displays in Raspberry Pi Projects

Learn to program displays and GUIs with Python



Dogan Ibrahim



---

# Using Displays in Raspberry Pi Projects



Dogan Ibrahim



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

© Elektor International Media BV 2021

First published in the United Kingdom 2021

● All rights reserved. No part of this book may be reproduced in any material form, including photocopying, or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication, without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publishers. The publishers 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.

● 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](http://daverid.com)

Printed in the Netherlands by Wilco



Elektor is part of EIM, the world's leading source of essential technical information and electronics products for pro engineers, electronics designers, and the companies seeking to engage them. Each day, our international team develops and delivers high-quality content - via a variety of media channels (e.g., magazines, video, digital media, and social media) in several languages - relating to electronics design and DIY electronics. [www.elektor.com](http://www.elektor.com)

**• 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

• Declaration . . . . .	5
• Preface . . . . .	11
<b>Chapter 1 • Installing the Operating System on Raspberry Pi . . . . .</b>	<b>12</b>
1.1 • Overview . . . . .	12
1.2 • Raspbian Buster installation steps on Raspberry Pi 4 . . . . .	12
1.3 • Using a networked connection . . . . .	15
1.4 • Remote access . . . . .	17
1.5 • Using Putty . . . . .	18
1.5.1 • Configuring Putty . . . . .	19
1.6 • Desktop remote access . . . . .	20
1.7 • Static IP address . . . . .	21
1.8 • Summary . . . . .	24
<b>Chapter 2 • Raspberry Pi Program Development . . . . .</b>	<b>25</b>
2.1 • Overview . . . . .	25
2.2 • The nano text editor . . . . .	25
2.3 • Creating and running a Python program . . . . .	27
2.4 • Summary . . . . .	31
<b>Chapter 3 • GPIO . . . . .</b>	<b>32</b>
3.1 • Overview . . . . .	32
3.2 • The Raspberry Pi 4 GPIO connector . . . . .	32
3.3 • Interfacing to the GPIO . . . . .	34
3.3.1 • Loads requiring small currents . . . . .	34
3.3.2 • Loads requiring higher currents . . . . .	35
3.3.3 • Using relays . . . . .	36
3.4 • The GPIO library . . . . .	36
3.4.1 • Pin numbering . . . . .	37
3.4.2 • Channel (I/O port pin) configuration . . . . .	37
3.5 • Raspberry Pi project development cycle . . . . .	40
3.5.1 • The hardware . . . . .	40
3.5.2 • The software . . . . .	41
3.6 • Summary . . . . .	41
<b>Chapter 4 • LED Projects . . . . .</b>	<b>42</b>
4.1 • Overview . . . . .	42
4.2 • Project 1 – Alternate flashing red, green, and blue LEDs . . . . .	42
4.3 • Running a program automatically at startup time . . . . .	44

4.4 ● Scheduling a program to run at specified times . . . . .	44
4.5 ● Project 2 – Binary up counting LEDs . . . . .	51
4.6 ● Project 3 - Random flashing Christmas lights . . . . .	53
4.7 ● Project 4 – Lucky day of the week. . . . .	53
4.8 ● Project 5 - LED bargraph . . . . .	55
4.9 ● Project 6 – Using shift registers in LED displays. . . . .	56
4.10 ● Project 7 – The BarGraph click board – Counting up in binary . . . . .	59
4.11 ● Project 8 – The BarGraph click board – Bar graph . . . . .	63
4.12 ● Project 9 – 2 digit multiplexed 7-Segment LED display seconds counter. . . . .	64
4.13 ● Project 10 – 2 digit 7-segment temperature display . . . . .	71
4.14 ● Project 11 – 4 digit 7-segment display seconds counter . . . . .	73
4.15 ● Project 12 – Using the UT-M 7-SEG R Click board – 2 digit up counter . . . . .	76
4.16 ● Project 13 – MAX7219 based 8 Digit 7-Segment LED – 8 digit up counter . . . . .	81
4.17 ● Project 14 – Using a dot matrix display – Dot Matrix R Click . . . . .	88
4.18 ● Using a 8x32 dot matrix display . . . . .	101
4.19 ● Project 15 – Matrix display - Displaying letters . . . . .	103
4.20 ● Project 16 – Matrix display - Drawing a rectangle box with letters. . . . .	105
4.21 ● Project 17 – Matrix display - Scrolling text . . . . .	105
4.22 ● Project 18 – Matrix display - Scrolling printable ASCII characters in small font . . . . .	105
4.23 ● Using LED strips. . . . .	106
4.24 ● Project 19 – LED strip: displaying different colours. . . . .	110
4.25 ● Project 20 – LED strip: displaying random colours (all LEDs have same colours) . . . . .	111
4.26 ● Project 21 – LED strip: displaying random colours (LEDs have different colours) . . . . .	111
4.27 ● Summary . . . . .	111
<b>Chapter 5 ● Using Liquid Crystal Displays (LCDs) . . . . .</b>	<b>112</b>
5.1 ● Overview . . . . .	112
5.2 ● HD44780 LCD module. . . . .	112
5.3 ● Installing the Python library for parallel LCDs . . . . .	114
5.4 ● The library functions. . . . .	114
5.5 ● Project 1 – Parallel LCD – Seconds Counter . . . . .	115
5.6 ● Project 2 – Parallel LCD – Read text from the keyboard and display on LCD . . . . .	116
5.7 ● Project 3 – Parallel LCD – Scrolling text read from the keyboard. . . . .	116
5.8 ● Project 4 – Parallel LCD – Displaying temperature and humidity . . . . .	117
5.9 ● Project 5 – Parallel LCD – Display the current date and time on the LCD . . . . .	119
5.10 ● Using I <sup>2</sup> C LCDs. . . . .	119
5.11 ● Project 6 – I <sup>2</sup> C LCD – Read text from the keyboard and display on LCD . . . . .	120
5.12 ● Project 7 – I <sup>2</sup> C LCD – Display the current date and time on the LCD . . . . .	122
5.13 ● Project 8 – I <sup>2</sup> C LCD – Displaying temperature and humidity . . . . .	122
5.14 ● Summary . . . . .	124
<b>Chapter 6 ● Using Organic Light Emitting Diode Displays (OLED) . . . . .</b>	<b>125</b>
6.1 ● Overview . . . . .	125
6.2 ● Using OLED displays. . . . .	126
6.3 ● Project 1 – Displaying pixels at the four corners of the display. . . . .	128
6.4 ● Project 2 – Displaying text . . . . .	129
6.5 ● Project 3 – Displaying Shapes . . . . .	130

---

6.6 ● Project 4 – Creating and displaying a bitmap . . . . .	132
6.7 ● Summary . . . . .	134
<b>Chapter 7 ● Using e-paper Displays . . . . .</b>	<b>135</b>
7.1 ● Overview . . . . .	135
7.2 ● How do e-paper displays work? . . . . .	135
7.3 ● Use of the e-paper displays . . . . .	136
7.4 ● Advantages and disadvantages of the e-paper displays . . . . .	137
7.5 ● Comparing the e-paper displays with LCDs . . . . .	137
7.6 ● Coloured e-paper displays . . . . .	138
7.7 ● Project 1 – Using an e-paper – displaying rectangle with text inside . . . . .	138
7.8 ● Project 2 – Using an e-paper – displaying an image . . . . .	145
7.9 ● Summary . . . . .	146
<b>Chapter 8 ● Plotting Graphs . . . . .</b>	<b>147</b>
8.1 ● Overview . . . . .	147
8.2 ● Plotting in Python . . . . .	147
8.2.1 ● Graph of a quadratic function . . . . .	147
8.2.2 ● Drawing multiple graphs . . . . .	149
8.3 ● Project - Real-Time graph of the temperature and humidity . . . . .	152
8.4 ● Summary . . . . .	154
<b>Chapter 9 ● Using Thin-Film-Transistor (TFT) Displays . . . . .</b>	<b>155</b>
9.1 ● Overview . . . . .	155
9.2 ● The 1.8 inch TFT display . . . . .	155
9.3 ● Project 1 – Drawing a green colour rectangle with blue colour text inside . . . . .	157
9.4 ● Project 2 – Displaying the temperature in blue or red colour . . . . .	159
9.5 ● Project 3 – Displaying thermometer image and temperature . . . . .	161
9.6 ● Summary . . . . .	163
<b>Chapter 10 ● Using the 7-inch Raspberry Pi Touch Screen . . . . .</b>	<b>164</b>
10.1 ● Overview . . . . .	164
10.2 ● Installing the display . . . . .	165
10.3 ● Project 1 – Drawing graphics . . . . .	166
10.4 ● Project 2 – Taking selfie pictures using a camera and 7-inch display . . . . .	167
10.5 ● The Tkinter graphical interface software . . . . .	169
10.5.1 ● Label . . . . .	170
10.5.2 ● Button . . . . .	172
10.5.3 ● Entry . . . . .	175
10.5.4 ● Grid . . . . .	175
10.5.5 ● Radio button . . . . .	176
10.5.6 ● Checkbox . . . . .	178
10.5.7 ● Dialogs . . . . .	178



10.5.8 ● Scale (slider) . . . . .	179
10.5.9 ● Menu . . . . .	180
10.5.10 ● Binding to events . . . . .	181
10.6 ● Using the ttk module . . . . .	182
10.7 ● Project 3 – GUI program to convert Degrees Centigrade to Degrees Fahrenheit	183
10.8 ● Project 4 – GUI program to display the ambient temperature and humidity . .	184
<b>● Appendix . . . . .</b>	<b>186</b>
A.1 ● Program: RGB.py . . . . .	186
A.2 ● Modified program: RGB2.py . . . . .	187
A.3 ● Program: LEDCounter.py . . . . .	188
A.4 ● Program: RandomLEDS.py . . . . .	190
A.5 ● Program: LuckyDay.py . . . . .	192
A.6 ● Program: BarLED.py . . . . .	194
A.7 ● Program: ShiftLED.py . . . . .	196
A.8 ● Modified Program: ShiftLED2.py . . . . .	198
A.9 ● Program: BarClick.py . . . . .	200
A.10 ● Program: BarClickGraph.py . . . . .	202
A.11 ● Program: SevenCount.py . . . . .	205
A.12 ● Program: SevenCount2.py . . . . .	208
A.13 ● Program: dht11.py . . . . .	211
A.14 ● Program: SevenCount4.py . . . . .	214
A.15 ● Program: UTM7SEG.py . . . . .	217
A.16 ● Program: MAX7219DISP.py . . . . .	220
A.17 ● Library: MAX7219 . . . . .	223
A.18 ● Program: MAX7219TEST.py . . . . .	226
A.19 ● Program: DotMatrix.py . . . . .	227
A.20 ● Modified Program: DotMatrix2.py . . . . .	230
A.21 ● Library: Matrix.py . . . . .	233
A.22 ● Font: Fonts.py . . . . .	235
A.23 ● Program: MatrixTEST.py . . . . .	236
A.24 ● ASCII fonts used in the program (ASCII.py) . . . . .	237
A.25 ● Library: MatrixAscii.py . . . . .	240
A.26 ● Program: MatrixAsciiTest.py . . . . .	242
A.27 ● Program: MatrixLetters.py . . . . .	243
A.28 ● Program: MatrixRectangle.py . . . . .	244
A.29 ● Program: MatrixScroll.py . . . . .	245
A.30 ● Program: MatrixTinyFont.py . . . . .	246
A.31 ● Program: LEDStripScan.py . . . . .	247
A.32 ● Program: LEDStripColours.py . . . . .	249
A.33 ● Program: LEDStripColours2.py . . . . .	251
A.34 ● Program: LCDCounter.py . . . . .	253
A.35 ● Program: LCDKeyboard.py . . . . .	255
A.36 ● Program: LCDScroll.py . . . . .	256
A.37 ● Program: LCDDHT11.py . . . . .	257
A.38 ● Program: LCDDTIME.py . . . . .	259

A.39 ● Program: LCDI2CKeyboard.py . . . . .	260
A.40 ● Program: LCDI2CTIME.py . . . . .	261
A.41 ● Program: LCDI2CDHT11.py . . . . .	262
A.42 ● Program: OLEDCorners.py . . . . .	263
A.43 ● Program: OLEDText.py . . . . .	264
A.44 ● Program: OLEDRect.py . . . . .	265
A.45 ● Program: OLEDShape1.py . . . . .	266
A.46 ● Program: OLEDShape2.py . . . . .	267
A.47 ● Program: OLEDBitmap.py . . . . .	268
A.48 ● Program: EPAPERtext.py . . . . .	269
A.49 ● Program: EPAPERimg.py . . . . .	270
A.50 ● Program: graph.py . . . . .	271
A.51 ● Program: TFT1.py . . . . .	273
A.52 ● Program: TFTtemperature.py . . . . .	275
A.53 ● Program: TFTimgtemperature.py . . . . .	277
A.54 ● Program: graphs7inch.py . . . . .	279
A.55 ● Program: camdisp.py . . . . .	280
A.56 ● Program: camdisp2.py . . . . .	281
A.57 ● Program: gui1.py . . . . .	283
A.58 ● Program: gui2.py . . . . .	284
A.59 ● Program: gui3.py . . . . .	285
A.60 ● Program: gui5.py . . . . .	286
A.61 ● Program: gui6.py . . . . .	287
A.62 ● Program: gui7.py . . . . .	288
A.63 ● Program: gui8.py . . . . .	289
A.64 ● Program: gui9.py . . . . .	290
A.65 ● Program: gui10.py . . . . .	291
A.66 ● Program: gui11.py . . . . .	292
A.67 ● Program: gui12.py . . . . .	293
A.68 ● Program: gui13.py . . . . .	294
A.69 ● Program: gui14.py . . . . .	295
A.70 ● Program: CTOF.py . . . . .	296
A.71 ● Program: TH7display.py . . . . .	297
<b>● Index</b> . . . . .	<b>299</b>

## • 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.

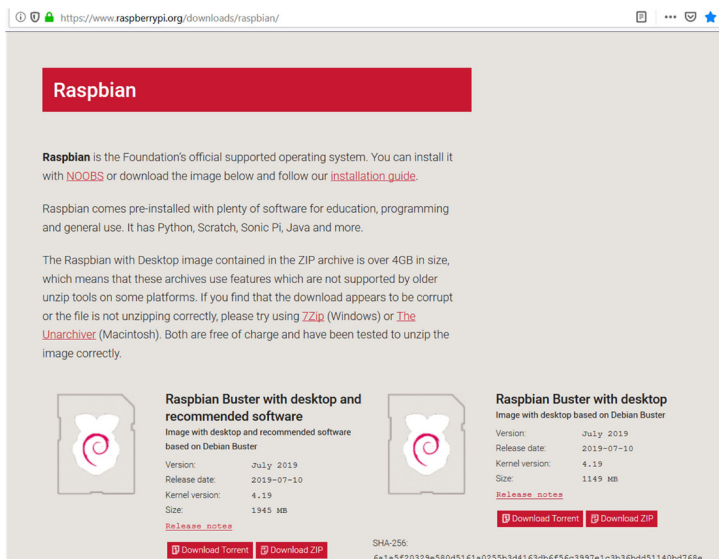


Figure 1.1 Raspbian Buster download page

<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/>

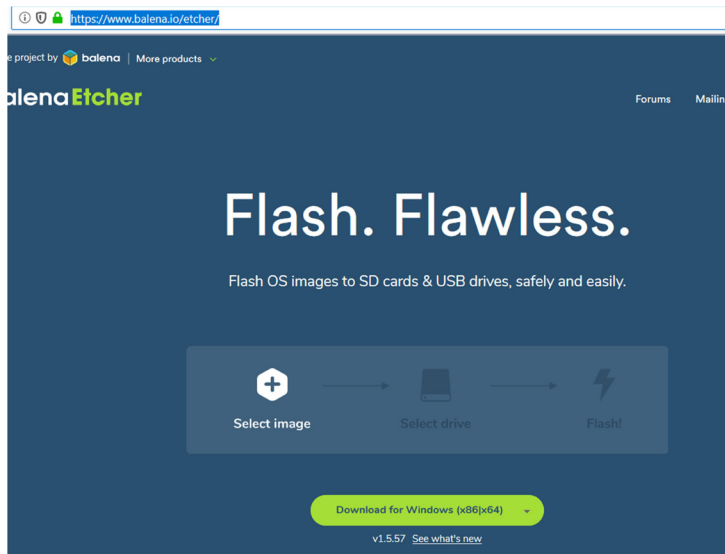


Figure 1.2 Download Etcher

- 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.

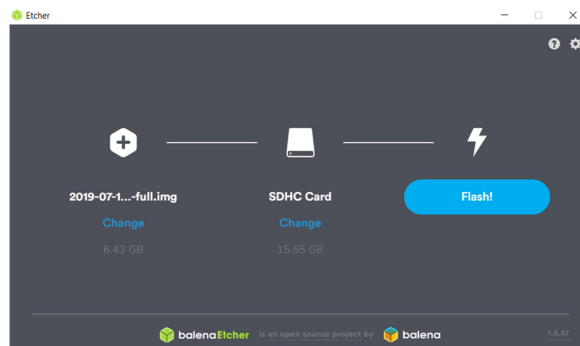


Figure 1.3 Click Flash to flash the disk image

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).

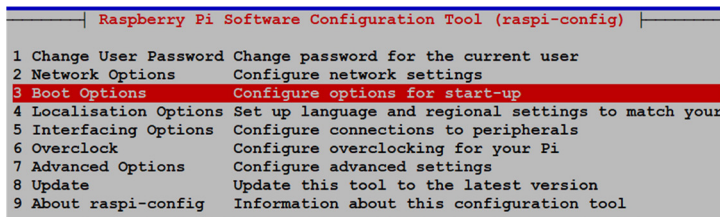


Figure 1.4 Select Boot Options

- 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



	09AA01AC491808W2.home	192.168.1.200	Nest Labs Inc.	64:16:66:93:79:43
	raspberrypi.home	192.168.1.202		DC:A6:32:00:E4:29

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 doesn't 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.

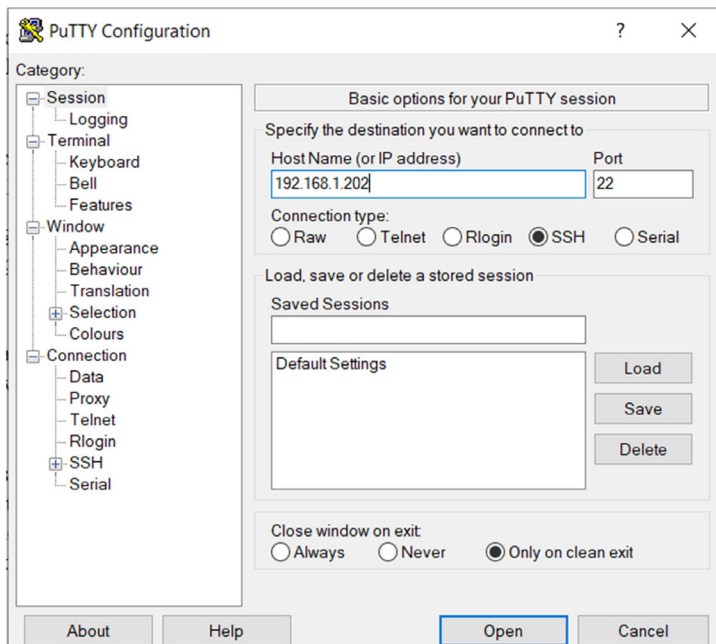


Figure 1.7 Putty startup screen

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

username: **pi**  
password: **raspberrypi**

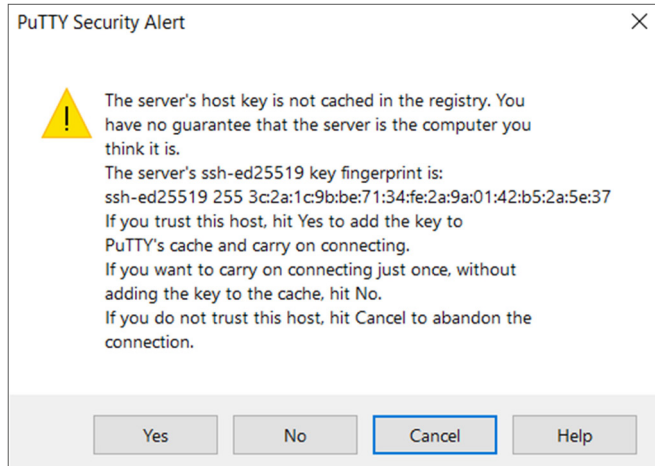


Figure 1.8 Click Yes to accept

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.

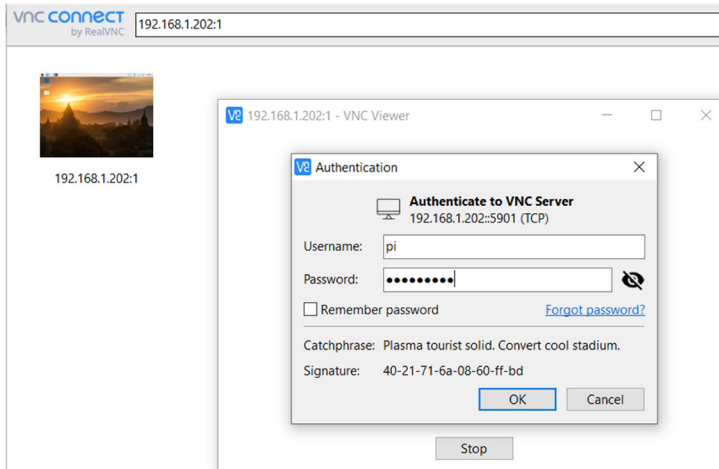


Figure 1.9 Start the TightVNC and enter the IP address

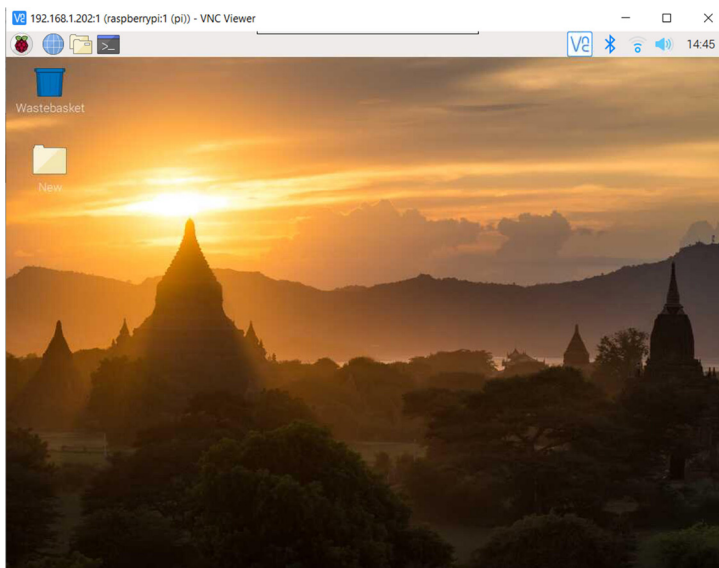


Figure 1.10 Raspberry Pi Desktop on the PC screen

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@raspberrypi:~ $ 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 pre:
   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, statu
   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@raspberrypi:~ $ sudo service dhcpcd start
pi@raspberrypi:~ $ 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.

## View your basic network information and set up connections

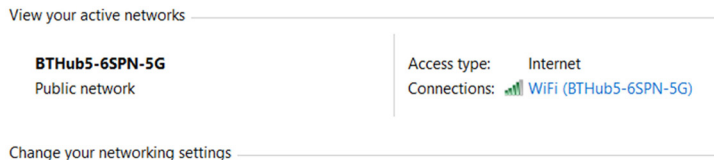


Figure 1.12 Click Internet

- 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/dhcpd.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@raspberrypi:~ $ sudo nano /etc/dhcpd.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/dhcpd.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.



---

**• Index****A**

Adafruit\_DHT 73, 211, 213

Adafruit Neopixel 109

Address field 83, 84

AMOLED 125

**B**

BarGraphIn 62, 63, 200, 201, 202, 203

BarGraphLatch 62, 63, 200, 201, 202, 203

**C**

command mode 14, 28, 31, 43, 73, 123, 147

common anode 66, 67

common cathode 42, 66

ConvBCD 87, 221, 222, 224, 225

crondeamon 44

crontab 44, 45, 46, 47, 48, 49, 51

Crontab Generator 49

current source mode 34

**D**

Data field 83

DC56-11EWA 67, 68, 74

Decode mode 84, 85

Desktop GUI mode 28, 31

dhcpcd 22, 23, 24

DHT11 71, 72, 73, 117, 122, 153, 154, 277, 297

direct connection 14

DisplayInt 87, 88, 222, 223, 225, 226

dot matrix library 99

Dot Matrix R 88, 89, 92, 93, 230, 236, 242

Dot Register 89, 91, 93

Duty Factor 99

**E**

Etcher 13

**F**

Flow Charts 41

**G**

GPIO connector *32, 37, 40, 139*

GPIO library *36, 39*

GPIO.RPi *59, 62, 63, 87*

## H

HC595IN *59, 197, 199*

HD44780 *112, 113, 114*

## I

ILOGIC *99*

Input Configuration *37*

## L

Latch Clock *57, 58*

LEDClear *59, 196, 197*

LED strips *11, 106, 107*

LoadControlWord0 *97, 98, 228, 229, 233, 240, 241*

## M

matplotlib *146, 147, 148, 149, 151, 154, 279*

MatrixAscii *100, 240, 242*

MOSFET *35, 36, 61*

mycron.sh *45, 46, 47*

myfile *25, 26, 47, 48, 50*

myscript.sh *47, 48, 49, 50*

## N

nano *23, 25, 26, 27, 31, 166, 168*

networked connection *15*

numpy *141, 147, 148, 150, 271, 279*

## O

Output configuration *39*

## P

PIL library functions *129*

Pin numbering *37*

PMOLED *125*

PORT\_Output *52, 53, 54*

Power Dissipation *86*

Program Description Language *41*

Putty *15, 16, 17, 18, 19, 20*

Python 2 *147*

---

## R

Raspbian Buster *12, 13, 25*

RegisterLatch *80, 217, 218*

relays *36, 60*

Remote access *17*

## S

Sclock *59, 60, 62, 63, 79, 87, 197, 201,*

Serial Data Input *57*

Shift Clock *57, 58*

shift registers *56, 57, 59, 60*

Shutdown mode *84, 85, 87*

SQH *57*

State Machines *41*

Static IP address *21*

sudoraspi-config *103, 121, 127, 141*

## T

Test mode *85*

TFT display *155, 157, 275, 277*

Thonny *28, 29, 30, 31, 43*

Tkinter *163, 169, 170, 181, 182, 286*

try-except *59*

## V

VLED *99*

VLOGIC *99*

VNC Viewer *20, 147*

## W

Waveshare *139, 141*

## X

Xming *20*

Learn to program displays and GUIs with Python

# Using Displays in Raspberry Pi Projects

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 the GPIO ports.

The core of the book explains the following topics in simple terms with fully tested and working example projects:

- › Simple LED projects
- › Bar graph LED projects
- › Matrix LED projects
- › Bitmap LED projects
- › LED strips
- › LCDs
- › OLED displays
- › E-paper displays
- › TFT displays
- › 7-inch touch screen
- › GUI Programming with Tkinter

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.

Python program listings of all Raspberry Pi projects developed in this book are available for download at [Elektor.com](http://Elektor.com). Readers can use these programs in their projects. Alternatively, they can modify the programs to suit their applications.



Prof. Dr. Dogan Ibrahim is a Fellow of the Institution of Electrical Engineers. He is the author of over 60 technical books, published by publishers including Wiley, Butterworth, and Newnes. He is the author of over 250 technical papers, published in journals, and presented in seminars and conferences.

**Elektor International Media BV**  
[www.elektor.com](http://www.elektor.com)

