

1

Getting started

7

Introducing Raspberry Pi	8
Gathering the components	10
Setting up the SD card	12
Configuring the system	14
Logging into the console	16
Starting the desktop	18
Automating the login	20
Summary	22

2

Exploring the desktop

23

Understanding the taskbar	24
Choosing your preferences	26
Examining the filesystem	28
Launching a terminal	30
Creating a text file	32
Browsing the internet	34
Extending the system	36
Summary	38

3

Commanding the system

39

Listing contents	40
Getting applications	42
Saving web pages	44
Reading & writing text	46
Finding files	48
Adding users	50
Changing permissions	52
Employing the shell	54
Creating shell scripts	56
Summary	58

4

Animating with Scratch

59

Walking a sprite	60
Changing directions	62
Adding another sprite	64
Editing costumes	66
Turning around	68
Reacting to touch	70
Playing sounds	72
Summary	74

5

Programming with Python

75

Introducing the interpreter	76
Writing your first program	78
Writing lists	80
Manipulating lists	82
Fixing in tuples	84
Collecting in sets	86
Associating in dictionaries	88
Testing expressions	90
Looping while true	92
Looping over items	94
Looping for a number	96
Breaking out of loops	98
Defining functions	100
Supplying arguments	102
Summary	104

6

Importing modules

105

Storing functions	106
Owning function names	108
Formatting strings	110
Reading & writing files	112
Pickling objects	114
Handling exceptions	116
Defining classes	118
Inheriting features	120
Interrogating the system	122
Performing mathematics	124
Calculating decimals	126
Telling the time	128
Summary	130

7

Producing games

131

Creating a game window	132
Painting shapes	134
Blitting objects	136
Playing sounds	138
Moving images	140
Animating sprites	142
Detecting collisions	144
Summary	146

8

Developing windowed apps

147

Introducing Tkinter	148
Adding widgets	150
Responding to buttons	152
Producing messages	154
Gathering entries	156
Listing options	158
Polling radio buttons	160
Checking boxes	162
Displaying images	164
Adjusting attributes	166
Managing layouts	168
Summary	170

9

Driving header pins

171

Understanding pin numbering	172
Lighting a lamp	174
Directing output	176
Adding more lamps	178
Recognizing input	180
Adding more buttons	182
Controlling projects	184
Summary	186

Index

187

Foreword

For such a small inexpensive device the Raspberry Pi has become a huge global sensation. Creation of this book has provided me, Mike McGrath, a welcome opportunity to demonstrate features of the Raspberry Pi that encourage an interest in computing and programming. Example code listed in this book describes how to develop programs in easy steps – and the screenshots illustrate the actual results. I sincerely hope you enjoy discovering the exciting possibilities of Raspberry Pi and have as much fun with it as I did in writing this book.

In order to clarify the code listed in the steps given in each example I have adopted certain colorization conventions. Components of the Python programming language are colored blue, programmer-specified names are red, numeric and string data values are black, and comments are green, like this:

```
# Write the traditional greeting.  
greeting = "Hello World!"  
print( greeting )
```

Additionally, in order to identify each source code file described in the steps a colored icon and file name appears in the margin alongside the steps:



script.py



data.txt



image.gif



sound.wav



font.ttf

For convenience I have placed source code files from the examples featured in this book into a single ZIP archive. You can obtain the complete archive by following these easy steps:

- 1 Browse to <http://www.ineasysteps.com> then navigate to the “Resource Center” and choose the “Downloads” section
- 2 Find “Raspberry Pi in easy steps” in the “Source Code” list, then click on the hyperlink entitled “All Code Examples” to download the archive
- 3 Next extract the archive contents to your home directory folder, such as `/home/pi`
- 4 Now call upon the Python interpreter from a Terminal to execute any code example, for instance issue the command `python hello.py` to see the output

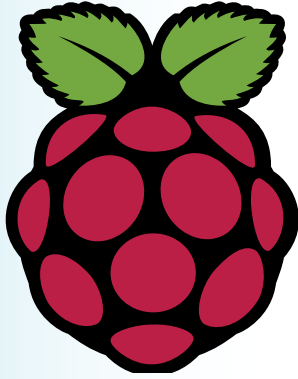
1

Getting started

Welcome to the exciting world of the Raspberry Pi.

This chapter demonstrates how to establish a fully functional computer system.

- 8** Introducing Raspberry Pi
- 10** Gathering the components
- 12** Setting up the SD card
- 14** Configuring the system
- 16** Logging into the console
- 18** Starting the desktop
- 20** Automating the login
- 22** Summary



Discover more about Raspberry Pi online at www.raspberrypi.org

Hot tip



The choice of Linux distributions optimized for the Raspberry Pi are described on page 12.

Introducing Raspberry Pi

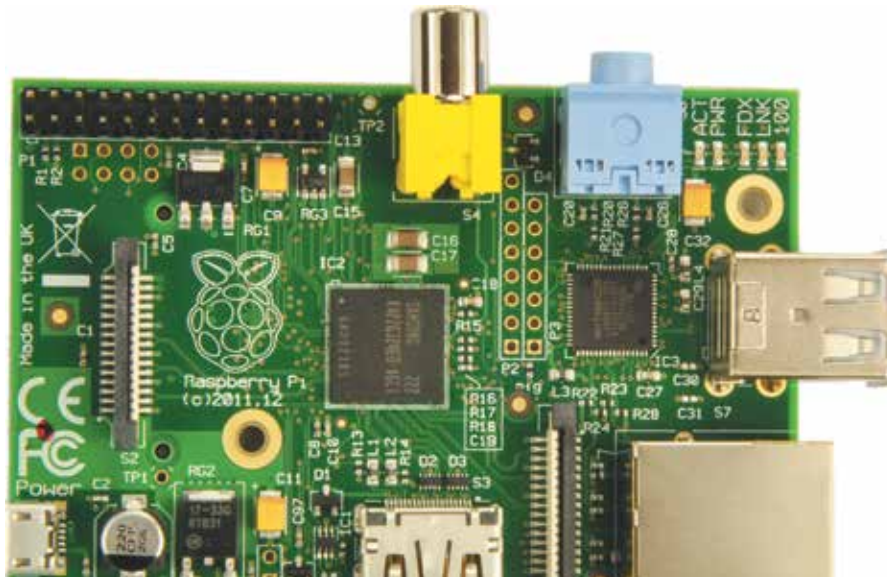
The Raspberry Pi is an inexpensive computer built on a single printed-circuit board. It was developed in the UK by the Raspberry Pi Foundation to encourage the teaching of basic computer science in schools and to put the fun back into learning about computing. The foundation recognized that the school ICT curriculum had changed, placing emphasis on the use of applications, such as Word and Excel, or to writing webpages. Additionally they noticed that the home PC and games console had replaced the Amigas, BBC Micros, Spectrum ZX and Commodore 64 machines that people of an earlier generation learned to program on. Nowadays young people had become merely passive users of computers who the foundation considers could benefit from knowing how computers work and how to program them – so they created the cheap, accessible Raspberry Pi computer.

In order to keep the price low the Raspberry Pi has some innovative design features:

- At its heart is an ARM processor that has System-on-Chip (SoC) architecture to integrate several traditionally separate components onto a single chip. The processor runs at 700mhz – but can be tweaked to run at a faster speed. Typically ARM processors have previously been used mainly in cellphones
- Unlike traditional computer design the Raspberry Pi does not have a hard drive but instead employs an SD card to contain the operating system and to store the files you create. The operating system can be one of several specially optimized variants of the Linux operating system
- The Raspberry Pi model B has a total memory of just 512MB – which is small compared to that of today's traditional computers. Even with this limitation surprisingly good performance is achieved as neither the processor nor the operating system are “memory hungry”. This in turn allows programs running on the Raspberry Pi to use very low amounts of memory
- Most noticeably, the Raspberry Pi is supplied without a case so it can be easily built into another device, such as a monitor, and its components can be easily identified

...cont'd

The Raspberry Pi is shown below together with a table describing the specifications of the Raspberry Pi model B components.



Component:	Specification:
CPU	700MHz ARM1176JZFS with FPU
SoC	Broadcom BCM2835
GPU	Videocore 4
SDRAM	512MB shared with GPU
USB2.0	2 via integrated USB hub
Video out	1 HDMI socket plus 1 RCA socket
Audio out	3.5mm jack socket (plus HDMI)
Storage	SD card slot
Network	10/100 Ethernet socket (RJ45)
Low-level	8 GPIO pins plus UART
Power rating	700mA (3.5W)
Power source	5 volt via MicroUSB
Size	85.6 x 53.98 x 17 mm
Weight	1.6 oz (45g)

Hot tip



Raspberry Pi model B also has a 15-pin MIPI camera interface that is unsupported at the time of writing – but support should be added soon.

Don't forget

You can find the Raspberry Pi suppliers online at pi.rsdelivers.com and www.farnell.com.

Beware

Don't confuse the larger MiniUSB standard socket, found on many digital cameras, with the MicroUSB standard power socket found on the Raspberry Pi.

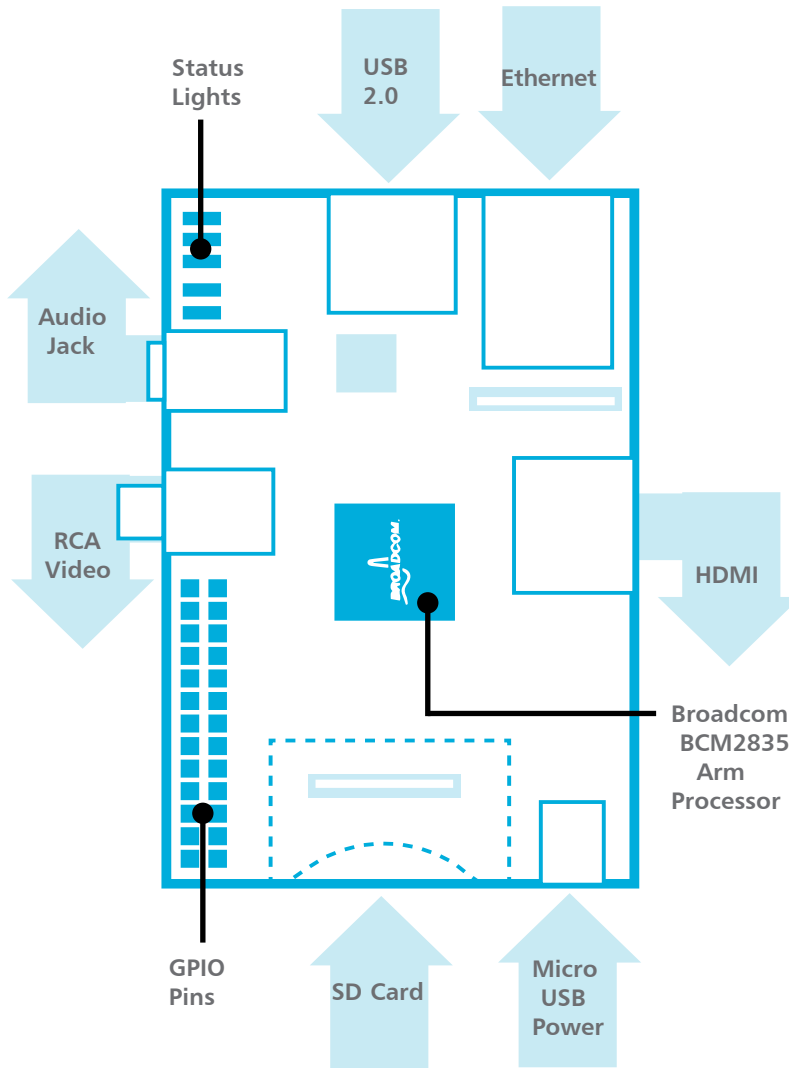
Gathering the components

The first step in establishing a computer system with Raspberry Pi is to gather together all the necessary components listed below:

- **Raspberry Pi** – first of course you will need to get your hands on a Raspberry Pi board itself, available worldwide through Premier Farnell/Element 14 and RS Components
- **SD card** – to contain the operating system so must be at least 2GB capacity and Class 4 or higher speed
- **SD card reader** – to write the operating system onto the card
- **MicroUSB 5 volt power source** – typically an Android cellphone charger or Kindle e-book reader charger
- **USB mouse**
- **USB keyboard**
- **USB hub** – optional but useful to add peripherals and to save wear and tear of the onboard USB sockets
- **TV or monitor**
- **HDMI cable** – to connect to a TV/monitor, or an RCA cable to connect to an older TV set
- **Ethernet cable** – for wired internet connection, or a wireless adapter for wireless internet connection

This seemingly long list of requirements is broadly similar to the components of a traditional PC so you may well already have some items on this list, but others you may need to purchase.

...cont'd



Hot tip



The status lights signify
ACT - SD card access
PWR - 3.3 volt power
FDX - full duplex (LAN)
LNK - link activity (LAN)
100 - 10/100Mb (LAN).

When you have gathered together all the necessary components they can be connected as shown in the illustration above – but the Raspberry Pi is not yet functional as the SD card does not contain an operating system.

Setting up the SD card

The operating system used by the Raspberry Pi can be one of several specially optimized versions of the Linux operating system. These are known as “distributions” or “distros” for short. Each available distro is offered as a disk image which must then be written onto the SD card to be inserted into the Raspberry Pi. The choice of currently available distros are listed for free download at www.raspberrypi.org/downloads. At the time of writing the list comprises these distros:

- **Raspbian** – recommended for beginners, this distro provides a comprehensive system that includes the LXDE graphical desktop, Midori web browser, and development tools to get you started programming
- **Arch Linux ARM** – not recommended for beginners, this distro aims for simplicity and allows the experienced Linux user to install packages from the online Arch repository to create a system customized to their needs
- **QtonPi** – not recommended for beginners, this is a specialized distro containing an SDK optimized for developing and running Qt 5 applications on Raspberry Pi

Hot tip



Linux and MacOSX users can use the **dd** tool on their systems to write the extracted disk image onto an SD card.

This book demonstrates features of the Raspbian distro. Disk images of the distros cannot simply be copied onto the SD card but must be written (“burnt”) onto the card by a special tool. For this on Windows you can use the free Win32 Disk Imager tool available at <https://launchpad.net/win32-image-writer/+download>. All disk images and the Win32 Disk Imager tool are contained in ZIP archives whose contents must be extracted after download.

- 1 Download the Win32 Disk Imager and extract its contents to a new folder on your desktop named “DiskImager + Raspbian OS”
- 2 Now download the Raspbian distro and extract the disk image into the same folder on your desktop – the disk image will be a file named like **XXXXXX-raspbian.img**

...cont'd

- 3 Next insert an SD card into your card reader – notice that it gets assigned a drive letter on your system



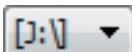
Don't forget



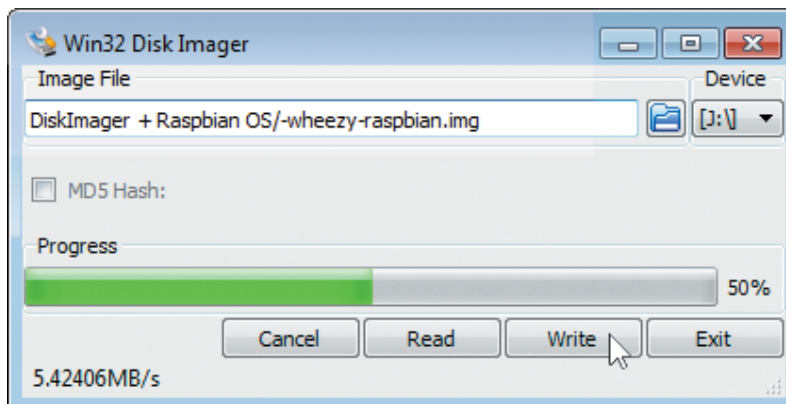
The SD card should be at least 2GB capacity and at least Class 4 speed.

- 4 Double-click on the executable (.exe) file in the folder to launch the Win32 Disk Imager tool

- 5 Next click on the  folder button in the Disk Imager and browse to select the image file

- 6 Now click on the  dropdown button in the Disk Imager and select the drive letter of your SD card

- 7 Finally click the Write button to burn the Raspbian disk image onto your SD card



Beware



The drive letter assigned to the SD card on your system may be different to the one shown here.

Don't forget

The root partition on your system will get expanded when you next boot Raspberry Pi.

Hot tip

If the Raspberry Pi screen does not fill your monitor choose the “overscan” configuration menu item then select the “disable” option.

Configuring the system

The first time you boot your Raspberry Pi a configuration tool where you can change options will appear after the boot messages:

- 1 With all components connected to the Raspberry Pi insert the SD card containing the Raspbian image then turn on the power to start the boot process

```

Raspi-config

info          Information about this tool
expand_rootfs Expand root partition to fill SD card
overscan      Change overscan
configure_keyboard Set keyboard layout
change_pass   Change password for 'pi' user
change_locale Set locale
change_timezone Set timezone
memory_split  Change memory split
overclock     Configure overclocking
ssh           Enable or disable ssh server
boot_behaviour Start desktop on boot?
update        Try to upgrade raspi-config

Select                               <Finish>

```

- 2 When the configuration menu appears use the up-down arrow keys on your keyboard to choose the “expand_rootfs” item and use the left-right arrow keys to choose “Select”, then hit Return – so your system uses all available space

- 3 Next use the arrow keys to choose “configure_keyboard” and select your keyboard type from the list that appears – or choose the default item “Generic 105-key (Intl) PC”

- 4 Now choose your keyboard layout – such as “Other” then “English (US)”

```

Configuring keyboard configuration
The layout of keyboards varies per country, with some countries having multiple common
layouts. Please select the country of origin for the keyboard of this computer.

Country of origin for the keyboard:

Dhivehi
Dutch
Dzongkha
English (Cameroon)
English (Ghana)
English (Nigeria)
English (South Africa)
English (UK)
English (US)

Cancel

```

...cont'd

- 5 Use the arrow keys to now choose “change_locale” and deselect the current character set from the list then select a character set for your locale – such as “en_US.UTF-8”

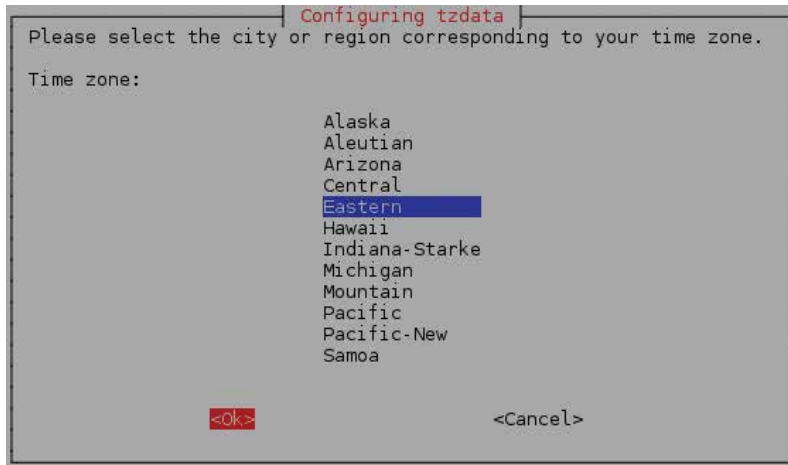


Beware



Unlike the other menus you must use the spacebar to choose a character set then hit Return to select it.

- 6 Finally use the arrow keys to choose “change_timezone” and select your region and timezone – such as “America” then “Eastern”



Hot tip



You can re-visit the Configuration Menu at any time from a command prompt by issuing the command `sudo raspi-config`.

- 7 The other menu options can be safely ignored so now use the arrow keys to select “Finish” then hit Return – configuration is complete and Raspberry Pi will offer to reboot your system then return you to a login prompt

```
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login:
```

Don't forget

If you make a mistake entering the user name or password you will simply be returned to the login prompt so you can try again.

Beware

Linux is case-sensitive so you must always observe correct capitalization. For example **pi**, **Pi**, and **PI** are distinctly different.

Don't forget

The password will not be displayed as you type it at the password prompt – so it cannot be stolen from over your shoulder.

Logging into the console

After configuring your system preferences, as described on the previous page, each time you start your Raspberry Pi you will see the boot messages then be placed at a login prompt where you must enter a user name and password to proceed.

The Raspbian distro is supplied configured with a single default user named “pi” whose login password is “raspberrypi”. These can therefore be used at the login prompt to access the system at its Command-Line Interface (CLI). This is known as the “console” – historically describing the keyboard and monitor combination from the days before the mouse and graphical desktop. When a login attempt succeeds a colorful user command prompt appears where can enter commands to interact with the system.

- 1 Start up your Raspberry Pi and watch the boot process until a login prompt appears – type **pi** then hit Return
- 2 When the password prompt appears type **raspberrypi** then hit Return – see the user prompt appear after system info

```
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Thu Nov  1 08:28:32 UTC 2012 on tty1
Linux raspberrypi 3.2.27+ #160 PREEMPT Mon Sep 17 23:18:42 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi ~ $
```

The Raspberry Pi user prompt contains several components describing your user name **pi**, machine name **raspberrypi** and current file location **~** (tilde – an alias for the **/home/pi** directory). You can confirm your current location on the filesystem at any time by entering the command **pwd** (print working directory).

- 3 At the user prompt type **pwd** then hit Return – see the current directory appear then another user prompt

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

...cont'd

You can discover the contents of the current directory by entering the command **ls** (list sorted). This displays a list of folders and files within the current directory, but not hidden system files. Like most Linux commands “switches” can be added after the command to change how they respond. So with the **ls** command adding a **-a** switch will display all content, including system files.

- 4 At the user prompt type **ls** then hit Return – see the current directory contents appear

```
pi@raspberrypi ~ $ ls
Desktop python_games
pi@raspberrypi ~ $
```

- 5 Now at the user prompt type **ls -a** then hit Return – see the current directory contents including system files

```
pi@raspberrypi ~ $ ls -a
.          .bashrc   Desktop   python_games
..         .cache    .gvfs     .thumbnails
.bash_history .config   .local    .xauthority
.bash_logout .dbus     .profile  .xsession-errors
pi@raspberrypi ~ $
```

You can clear past command results using the **clear** command and discover the switches available for any command, plus what they achieve, using the **man** command followed by the command name. For example, **man ls** to display the switches for the **ls** command.

It is important to recognize that Linux is designed as a multi-user system so regular users cannot perform all tasks – some can only be performed by the system administrator (“root” or “superuser”). But for convenience you can temporarily assume superuser status to do something with the **sudo** command. For example to close Raspberry Pi with the **shutdown** command. This accepts a **-h** (halt) switch or a **-r** (reboot) switch, followed by a number indicating a delay in minutes or zero for no delay.

- 6 Finally at the user prompt type **sudo shutdown -h 0** then hit Return – see the system close down immediately

```
pi@raspberrypi ~ $ sudo shutdown -h 0
```

Hot tip



Switches can be combined, so that **ls -al** is equivalent to **ls -a -l** as both will display all content in long format.

Hot tip



Notice that system content names are prefixed by a period and directories are blue color.

Hot tip



The system remembers your past commands – use the up and down arrow keyboard keys to scroll through them at the user prompt then hit Return to re-run your chosen past command.

Hot tip



There are other window managers available for Linux, such as Gnome and KDE, but Raspbian favors LXDE because it is lightweight – so it runs well on small devices.

Starting the desktop

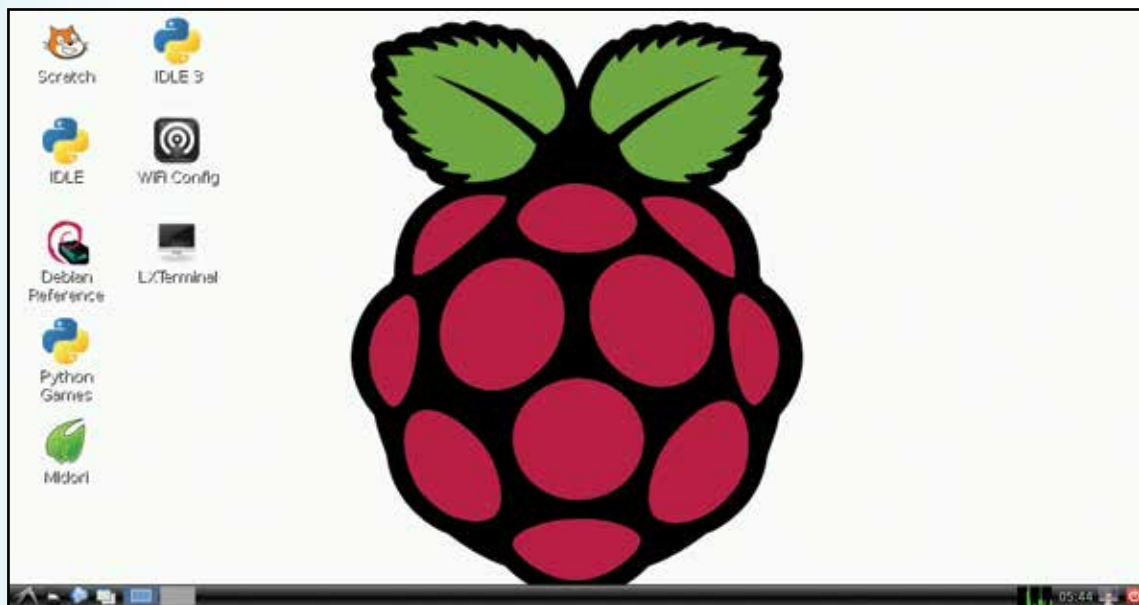
Interacting with your Raspbian system at the Command-Line Interface (CLI) gives an insight into the power of commands. Many Linux “boxes” are used at just that level, especially web servers, but you will probably want to see a more familiar graphical desktop that lets you interact using your mouse.

With the Raspbian distro an “X” server provides a Graphical User Interface (GUI) that automatically employs a window manager named “LXDE” (Lightweight X Desktop Environment).


When you boot Raspbian the X server does not automatically get started as you may wish to run your system as a basic console. Simply issuing a **startx** command at the command prompt will start the X server and load the graphical desktop environment.

- 1 Start up your Raspberry Pi then login to the system as the default user **pi** with the **raspberrypi** password
- 2 When the user prompt appears type **startx** then hit Return – see the graphical desktop load in the screen

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



...cont'd

- 3 Next click the  Logout button at the extreme right of the taskbar then click Logout on the dialog that appears – see both LXDE and the X server close, returning you to a command prompt in the console




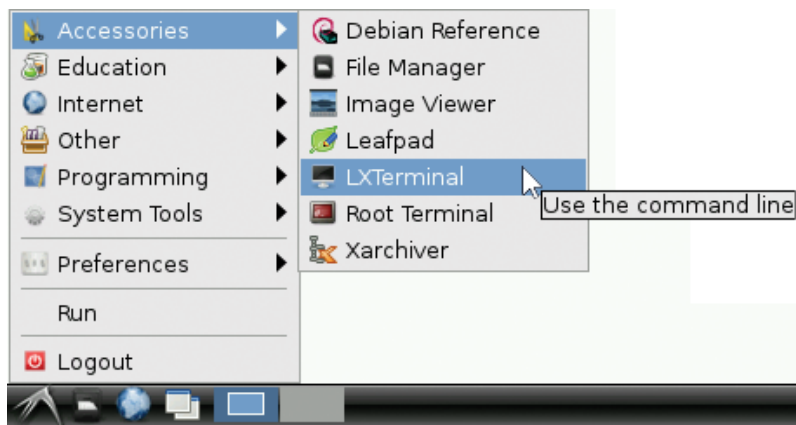
Don't forget



Notice here that the Logout dialog offers no buttons to Shutdown or Reboot the system because this LXDE "session" was started from the console.

If you wish to both logout and shutdown the system you can do so from a "Terminal" window in the GUI, which provides a command prompt similar to that in the console.

- 4 Issue a **startx** command at the prompt to restart the X server and the LXDE graphical desktop
- 5 Now click the  Start button at the extreme left of the taskbar and choose Accessories, LXTerminal – see a Terminal window appear containing a command prompt

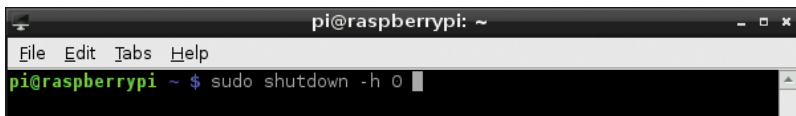


Hot tip



Features of the LXDE desktop are described in the next chapter – but feel free to explore the menus and taskbar now.

- 6 Finally at the command prompt type **sudo shutdown -h 0** then hit Return – see the system close down immediately



Don't forget

You need to temporarily assume superuser status with **sudo** to make configuration changes.

Automating the login

Unless you particularly wish to use your Raspberry Pi as a console system it becomes tedious to login and start the X server manually so the process can be automated using the configuration tool.

- 1 Start up your Raspberry Pi then login to the system as the default user **pi** with the **raspberrypi** password
- 2 When the user prompt appears type **sudo raspi-config** then hit Return – see the configuration tool appear

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

- 3 When the configuration menu appears use the up-down keyboard arrow keys to choose the “boot_behaviour” item – then hit Return to select it

```
Raspi-config

info          Information about this tool
expand_rootfs Expand root partition to fill SD card
overscan     Change overscan
configure_keyboard Set keyboard layout
change_pass  Change password for 'pi' user
change_locale Set locale
change_timezone Set timezone
memory_split Change memory split
overclock    Configure overclocking
ssh          Enable or disable ssh server
boot_behaviour Start desktop on boot?
update       Try to upgrade raspi-config

                <Select>                <Finish>
```

Beware

The ARM processor runs at 700MHz by default but you can choose the “overclock” menu item to increase its speed and performance – but this may reduce the life of your Raspberry Pi.

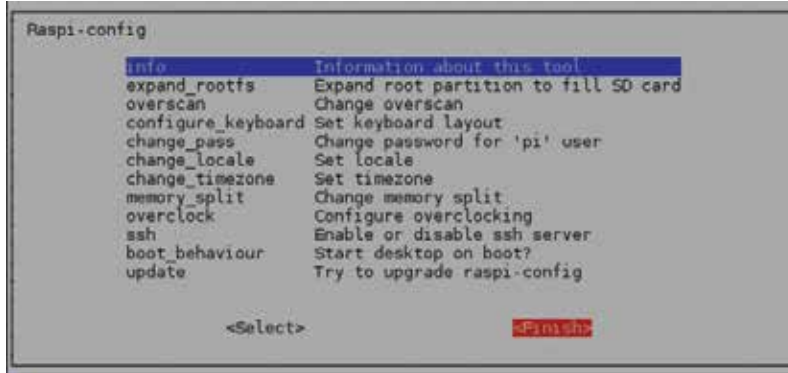
- 4 In the dialog that appears next use the left-right keyboard arrow keys to choose “Yes” to boot straight to the desktop

```
Should we boot straight to desktop?

                Yes                No                <Back>
```

...cont'd

- 5 Next use the left-right keyboard arrow keys to choose “Finish” then hit Return to apply the configuration change




Hot tip



If the Raspbian screen does not fit your monitor choose the “overscan” menu item and select “Disable” then reboot to see the difference.

- 6 In the dialog that now appears use the left-right keyboard arrow keys to choose “Yes” – then hit Return to reboot the system straight to the GUI desktop



- 7 When the desktop has loaded click the  Logout button at the extreme right of the taskbar then click Shutdown to close down the system, or click Reboot to restart the system, or click Logout to return to a command prompt in the console



Don't forget



Notice here that the Logout dialog now offers to Shutdown or Reboot the system because this LXDE “session” was started automatically.

Summary

- Raspberry Pi is an inexpensive credit-card sized computer developed to put fun back into learning about computing
- Raspberry Pi has an ARM processor using System-on-Chip architecture and uses an SD card as its hard drive for economy
- Peripheral components must be connected to the Raspberry Pi board to establish a complete computer system
- The Raspbian Linux distro is recommended for beginners and must be written onto an SD card for use with Raspberry Pi
- When you boot your Raspbian system for the first time a configuration tool appears where you can change options
- The Raspbian distro is supplied configured with a default user name of “pi” whose login password is “raspberry”
- After the boot messages have completed a console login prompt appears in the Command-Line-Interface (CLI)
- After login you can confirm your location on the filesystem by entering the command **pwd** (print working directory)
- You can discover the contents of the current directory folder by entering the command **ls** (list sorted)
- The **clear** command removes past command results and the **man** command can be used to discover command switches
- Linux systems, such as Raspbian, are designed as multi-user systems so regular users cannot perform administrative tasks
- You can temporarily assume superuser status to perform administrative tasks with the **sudo** command
- With Raspbian the X server provides a Graphical User Interface (GUI) desktop with the LXDE window manager
- The desktop GUI can be started from a console prompt by entering the command **startx** or it can be automated by modifying the “boot_behaviour” with the configuration tool