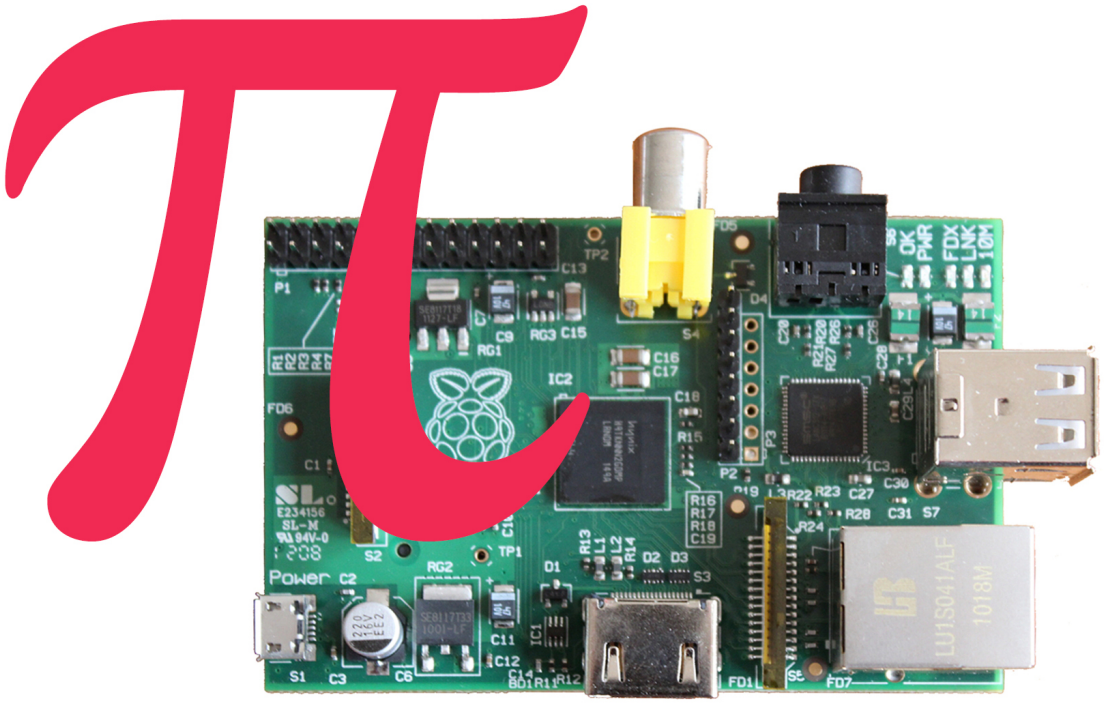
The Pragmatic Programmers

Pragmatic Press

Raspberry Pi

A Quick-Start Guide



Maik Schmidt

Edited by Jacquelyn Carter

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Jacquelyn Carter (editor)
Kim Wimpsett (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Printed in the United States of America.
ISBN-13: 978-1-937785-04-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 8, 2012

Preface

Over the past decades computers have gotten cheaper and cheaper, so today you can find them not only beneath your desk but in nearly every consumer electronics device such as smartphones or DVD players. Still, computers aren't so cheap that you spontaneously buy one when shopping for your groceries. Usually, you carefully plan your next PC, because you have to use it for a couple of years.

Computers like the Raspberry Pi will change the situation completely in the near future. The Raspberry Pi, or Pi for short, is a full-blown desktop PC that costs only $35. You can directly connect it to the Internet, and it is able to display high-definition videos. Also, it runs Linux, so you do not have to pay for an operating system. This makes the Pi probably the first throwaway computer in history.

Originally, the Raspberry Foundation[1] built the Pi to teach children how to program, so it comes as no surprise that the Pi is an excellent device for exactly this purpose. On top of that, you can use the Pi for many other exciting things. For example, you can turn it into a multimedia center, use it as a cheap but powerful web server, or play some classic games.

The Pi is also a great machine for experimenting with electronics. In contrast to many popular microcontroller boards like the Arduino, the Pi runs a full-blown operating system, and you can choose from a wide range of programming languages to implement your projects.

With cheap and small devices like the Raspberry Pi, a new era of ubiquitous computing has begun, and you can be part of it. This book helps you get up to speed quickly.

1. http://www.raspberrypi.org/

Who Should Read This Book?

This book is for everyone who wants to get started with the Raspberry Pi. Even if you have some experience with other computers, you'll quickly see that the Pi is different in many regards, and this book helps you avoid the most common pitfalls.

You can choose from a variety of operating systems for the Pi, but this book's focus is on Debian Linux (Raspbian), because it is the most convenient choice for beginners. If you've never worked with Linux before, you should start with Appendix 1, *A Linux Primer*, on page ?. Even if you've worked with Linux before, you still might learn a few things, because running Linux on the Pi is different in some ways.

Of course, you'll get the most out of this book if you have a Raspberry Pi and follow all the book's examples closely.

What's in This Book?

The Raspberry Pi does not come with a user guide, but in this book you'll learn step-by-step how to get the most out of your mini-computer quickly. You'll learn not only how the Pi's hardware works in principle but also how to run different operating systems and use the Pi for special purposes such as turning it into a multimedia center.

Here's a list of all the things you're going to learn:

- The book starts with an introduction to the Raspberry Pi's hardware. You'll learn what the Pi's connectors are for and which additional hardware you need to start the Pi for the first time.
- After you've connected all necessary devices to your Pi, you need an operating system. Although the Pi is a fairly young project, you can already choose from several, and you'll learn what their pros and cons are.
- Installing an operating system on the Pi is quite different from installing an operating system on a regular PC. So, you'll learn how to get Debian Linux up and running on the Pi.
- Debian Linux runs fine out of the box on the Pi, but to get the most out of it, you have to tweak a few configuration parameters. For example, it's beneficial to set the right layout for your keyboard. In addition, you'll learn how to install, update, and remove new software.

- The Pi's hardware, especially its graphics hardware, is special in many regards. Depending on the display you're using, you have to adjust some low-level settings for the Pi's firmware. You'll learn what settings are available and how to solve the most common firmware problems.
- To see what can be achieved with the Pi with a minimum of effort, you'll turn it into a kiosk system. It will be able to display a set of static slides as well as live information from the Internet.
- Until this point, you've used the Pi more or less in isolation, but now you'll learn how to integrate it with networks. You'll use the Pi for everyday tasks such as browsing the Web, you'll make it accessible via Secure Shell, and you'll even turn it into a full-blown web server. Also, you'll learn how to share your Pi's desktop with a PC, and vice versa.
- With the XBMC project, you can turn your Raspberry Pi into a multimedia center with ease. Not only can you show your photos collections to your friends in your living room, but you can also play music in all popular formats, and you can watch your favorite movies and TV shows in high definition.
- The Raspberry team originally built the Pi for educational purposes, but you can easily use it to play some entertaining games. Even though it's possible to run some first-person shooters, you might prefer some classic genres such as interactive fiction and point-and-click adventures.
- One of the greatest advantages the Pi has over regular PCs is its GPIO pins. In the book's final chapter, you'll learn how to easily use them to attach your own electronics projects to the Pi.
- The appendix contains a short introduction to Linux. If you've never worked with Linux before, you should read the appendix before you start with Chapter 3, *Configure Raspbian*, on page ?.

Where Can I Get a Raspberry Pi and Additional Hardware?

At the time of this writing, only two distributors produce and sell the Raspberry Pi. To buy a Pi, visit the web shops of Farnell[2] or RS Components.[3] These shops also sell many accessories such as power supplies, keyboards, mice, and so on, for the Pi. Adafruit[4] sells useful accessories, too.

2. http://www.farnell.com/
3. http://www.rs-online.com/
4. http://adafruit.com/category/105

You can find a growing list of compatible hardware on the project's wiki,[5] but when in doubt, it's better to buy hardware from one of the shops mentioned here.

Debian Linux

The most popular operating system for the Pi is Linux. Several Linux distributions are available for the Pi, and we chose Debian. Recently the Debian team has frozen the latest version named *wheezy*, and because of the great efforts of the Raspbian team,[6] it is available for the Pi already. Raspbian supersedes Debian squeeze, which has been the reference operating system for the Pi for a long time.

The Raspbian distribution has many advantages over all its predecessors. It is much faster, it has more recent software, and it will soon be more stable. Also, it is the preferred solution of the Raspberry team, so this book's focus is on Raspbian.

Code Examples and Conventions

In this book you'll find a few code examples written in PHP, in HTML, and in the programming language of the Bash shell. They are all very short, and if you've done some programming before, you'll have no problems understanding them. If you haven't developed software before, you'll still be able to copy the code to the Pi and make it run.

Online Resources

This book has its own web page at http://pragprog.com/titles/msraspi where you can download the code for all examples, or you can click the file name above each code example to download the source file directly. On the web page, you can also participate in a discussion forum and meet other readers and me. If you find bugs, typos, or other annoyances, please let me and the world know about them on the book's errata page.

Now it's time to unbox your Raspberry Pi and have some real fun!

5. http://elinux.org/RPi_VerifiedPeripherals
6. http://www.raspbian.org/

6.5 Add WiFi to the Pi

Wireless networks are everywhere. Coffee shops, airports, and hotels all offer free WiFi to their customers these days. You probably run a wireless network at home, too, so you can conveniently access your most important devices from your smartphone while you're having a barbeque with the family in the garden. On Windows or Mac OS X you usually do not have to think much about joining wireless networks because the process is nearly seamless.

On the Pi it's different. In this section you'll learn how to connect the Pi to a wireless network.

Joining wired networks via Ethernet is usually a piece of cake: plug in the Ethernet cable and you're done. To access a wireless network you need to do a bit more. First, you need a WiFi stick for the USB port, because the Pi's hardware does not support WiFi by default. Plug your WiFi stick into one of the Pi's USB ports and run the `lsusb` command to see if the Pi recognizes it properly:

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 050d:0237 Belkin Components F5U237 USB 2.0 7-Port Hub
Bus 001 Device 005: ID 04e8:2018 Samsung Electronics Co., Ltd WIS09ABGN
                                 LinkStick Wireless LAN Adapter
Bus 001 Device 006: ID 046d:c312 Logitech, Inc. DeLuxe 250 Keyboard
Bus 001 Device 007: ID 046d:c05a Logitech, Inc. Optical Mouse M90
```

In this case, device 005 is a WiFi stick manufactured by Samsung. You can take a closer look at the Pi's boot message with the `dmesg` command and see whether the WLAN stick has been initialized properly:

```
pi@raspberrypi ~ $ dmesg | less
...
usb 1-1.3.6: new high speed USB device number 5 using dwc_otg
usb 1-1.3.6: New USB device found, idVendor=04e8, idProduct=2018
usb 1-1.3.6: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.3.6: Product: 802.11 n WLAN
usb 1-1.3.6: Manufacturer: Ralink
usb 1-1.3.6: SerialNumber: 1.0
...
```

Press the space bar to go down one page and press 'b' to go up one page. Press 'q' to go back to the shell prompt. As you can see in the current case the Samsung stick uses the WiFi chipset from a company (manufacturer) named Ralink. This chipset is pretty popular, so Debian recognized it out of

the box. If the output of dmesg contains any errors right after the initialization of your WiFi stick, check the Pi's Wiki.[13] Often you have to download the firmware for your WiFi stick manually and reconfigure the Linux kernel.

If no errors occured, Debian Linux has recognized your WiFi stick successfully. You can use the following command to get the current status of your Pi's wireless network interfaces:

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

eth0      no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
```

At the moment the Pi is not connected to a wireless network, but the wlan0 interface is up and running. The following command searches for a wireless network:

```
pi@raspberrypi ~ $ sudo iwlist scan | grep ESSID
        ESSID:"darknet"
        ESSID:"valhalla"
```

In this case, two wireless networks named darknet and valhalla are within reach. To connect to one of them you have to edit the configuration file /etc/network/interfaces using a text editor, such as the nano text editor, for example. To connect to darknet add the following lines to the file:

```
auto wlan0
iface wlan0 inet dhcp
wpa-ssid darknet
wpa-psk t0p$ecret
```

These lines will activate the wlan0 interface automatically the next time you boot the Pi. Also they'll make the Pi obtain an IP address using DHCP. The Pi will try to join the network named darknet and it'll use the password 't0p$ecret;. Of course, you have to adjust the network name and the password accordingly.

If you're impatient, you do not have to reboot the Pi. Run the following command to make the Pi join your wireless network:

```
pi@raspberrypi ~ $ sudo ifup wlan0
Internet Systems Consortium DHCP Client 4.2.2
```

13. http://elinux.org/RPi_VerifiedPeripherals#USB_WiFi_Adapters

```
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/00:12:fb:28:a9:51
Sending on   LPF/wlan0/00:12:fb:28:a9:51
Sending on   Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 14
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 14
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.1.1
DHCPACK from 192.168.1.1
bound to 192.168.1.101 -- renewal in 2983 seconds.
```

The Pi has the IP address 192.168.1.101 now and is connected to your network wirelessly (your IP address probably will differ). Use the `ping` command to check whether you can access a web site like Google, for example:

```
pi@raspberrypi ~ $ ping -c 3 google.com
PING google.com (173.194.69.100) 56(84) bytes of data.
64 bytes from google.com (173.194.69.100): icmp_req=1 ttl=45 time=26.7 ms
64 bytes from google.com (173.194.69.100): icmp_req=2 ttl=45 time=32.3 ms
64 bytes from google.com (173.194.69.100): icmp_req=3 ttl=45 time=34.8 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 26.752/31.338/34.863/3.395 ms
```

As you can see, the Pi is connected to the Internet via WiFi now. Run the following command to get some statistics about the signal strength etc.:

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

eth0      no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:"darknet"
          Mode:Managed  Frequency:2.442 GHz  Access Point: 54:E6:FC:CF:77:8A
          Bit Rate=135 Mb/s   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=40/70  Signal level=-70 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:1  Invalid misc:6   Missed beacon:0
```

Keep in mind that a computer like the Pi can have more than one IP address. If you connect it via Ethernet and WiFi, for example, your boot message will display something like the following:

```
My IP address is 192.168.2.109 192.168.1.101
```

That means that you've connected your Pi using two network interfaces and for each interface it has a different IP address.

9.2 Meet the Pi's GPIO Pins

To connect your own electronics projects to the Pi, you can use the expansion header in the top-left corner of the Pi (see Figure 1, *The front side of a Model B*, on page ?). It consists of 26 pins arranged in two rows containing 13 pins each. The top row contains the even-numbered pins, and the other row contains the odd-numbered pins. That is, the first pin in the lower row is pin 1, and you can find the label "P1" on the Pi below the pin.

In Figure 27, *The Pi's GPIO pins*, on page 5, you can see the meaning and the numbering of the pins. With pin 6, the Pi can share a common ground with your electronics projects. Using pins 1 and 2, you can power external devices connected to the Pi with 3.3 volts or 5 volts. The Pi limits the output of pin 1 to 50mA, while pin 2 allows for a current draw that depends on the USB input current. If you power the Pi with a 1A power supply, for example, you can draw up to 300mA from pin 2, because the Pi Model B needs 700mA for itself.

	5V	-	Ground	GPIO14	GPIO15	GPIO18	-	GPIO23	GPIO24	-	GPIO25	GPIO8	GPIO7
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Pin	2	4	6	8	10	12	14	16	18	20	22	24	26
Pin	1	3	5	7	9	11	13	15	17	19	21	23	25
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	3v3	GPIO0	GPIO1	GPIO4	-	GPIO17	GPIO21	GPIO22	-	GPIO10	GPIO9	GPIO11	-

Figure 27—The Pi's GPIO pins

Pins 4, 9, 14, 17, 20, and 25 are reserved for future enhancements, so you cannot use them in your own projects. The remaining pins are general-purpose input/output (GPIO) pins that you can use as digital input or output pins. Note that the GPIO pin names do not correspond to the pin numbers of the expansion header.

You can use the GPIO pins, for example, to read the state of a push button or to turn an LED on and off. For this chapter's examples, you can assume that all GPIO pins work the same, but you should know that some of the Pi's pins are special. Pin 12, for example, supports Pulse Width Modulation (PWM),[6] which can be handy for controlling motors. If you're going to build more complex projects, you should take a look at a more detailed description of the Pi's pins.[7]

6. http://en.wikipedia.org/wiki/Pulse_width_modulation
7. http://elinux.org/RPi_Low-level_peripherals

9.3 Build a Basic Circuit

To warm up, you'll build one of the most basic circuits possible. You'll connect an LED to the Pi and make it shine as long as the Pi is running. For this you need an LED, a resistor, a breadboard, and two female/male jumper wires. Using these parts, you'll build the circuit in Figure 28, *A basic circuit*, on page 6.

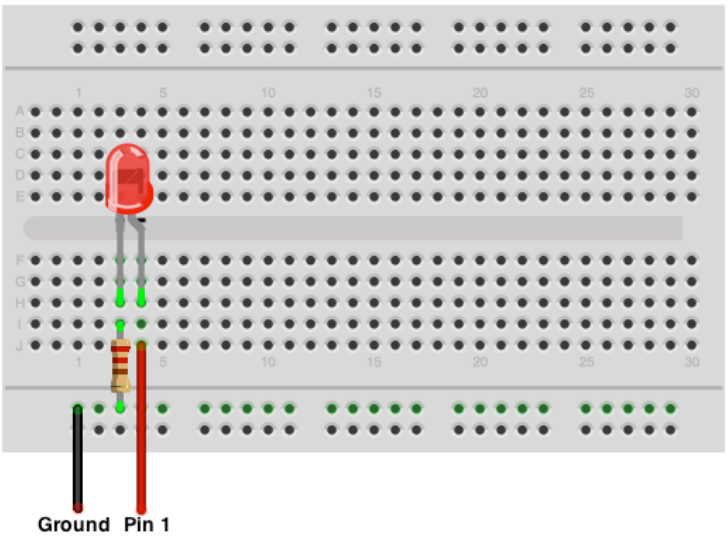


Figure 28—A basic circuit

Before you actually build the circuit, you should know what all the parts are for and how they work. Breadboards are useful tools for prototyping circuits. You can simply plug in parts like LEDs and resistors, so you do not have to solder them. Breadboards come in various sizes, but they all look very similar. On all of them, you can find many sockets arranged in columns. Most breadboards also have two rows of sockets at the top and at the bottom.

The main trick of a breadboard is that it automatically connects the sockets belonging to a certain column and to a certain row. In the basic circuit in Figure 28, *A basic circuit*, on page 6, you connect the Pi's Ground pin to the second-to-last row of the breadboard. This automatically connects all sockets in this row to the Pi's ground (that's why all the sockets in this row have a light green color). The same happens in the two columns connected to the LED. The resistor indirectly connects the Pi's Ground pin to one of the LED's connectors. In addition, you connect the Pi's pin 1 directly to the other connector of the LED by plugging it into a socket in the same column.

By the way, LED stands for light-emitting diode, so an LED is basically a diode. Diodes are very useful, because they let electricity pass in only one direction. That's true for LEDs, too, and in addition, LEDs emit light as a side effect.

Working with LEDs isn't very difficult, but you have to take care of a few things. First, you have to connect them the right way. LEDs have two wires, and one of them is a bit shorter than the other. The shorter wire is named *cathode* (negative), and you have to connect it to the Pi's ground pin. The longer wire is named *anode* (positive), and you have to connect it to one of the Pi's power supply or GPIO pins. You can also identify the anode and cathode by taking a close look at the LED's case. The flat side belongs to the cathode and the round side to the anode. In Figure 28, *A basic circuit*, on page 6, the anode is slightly bent.

Also, you always have to put a resistor in front of an LED. If you don't, the LED consumes too much power and will be destroyed. Simply put, a resistor limits the amount of current that flows through an electric connection and protects the LED. Calculating the resistor value for a certain type of LED is not difficult, but it's beyond the scope of this book. Simply keep in mind that the lower the resistor value, the brighter the light will shine. When in doubt, use a 330Ω or 470Ω resistor.

Now it's time to actually build the circuit. First, connect the LED to the breadboard. Make sure that the direction of the LED is right, and plug it in. You have to press firmly but not too hard—otherwise, you'll bend the connectors, and they won't fit. It's usually easier to plug parts in after you've shortened the connectors. When cutting the connectors, wear safety glasses to protect your eyes!

The resistor is next, and this time the direction doesn't matter. Before plugging the resistor in, you have to bend its connectors. Also, it usually helps to shorten them.

Finally, connect the two jumper wires to the Pi and to the breadboard. Connect the female side to the Pi and the male side to the breadboard. Make sure you're using the right pins on the Pi, and then turn on the Pi. If you've connected everything correctly, the LED will turn on, too. Otherwise, take a look at Section 9.7, *What If It Doesn't Work?*, on page ?.