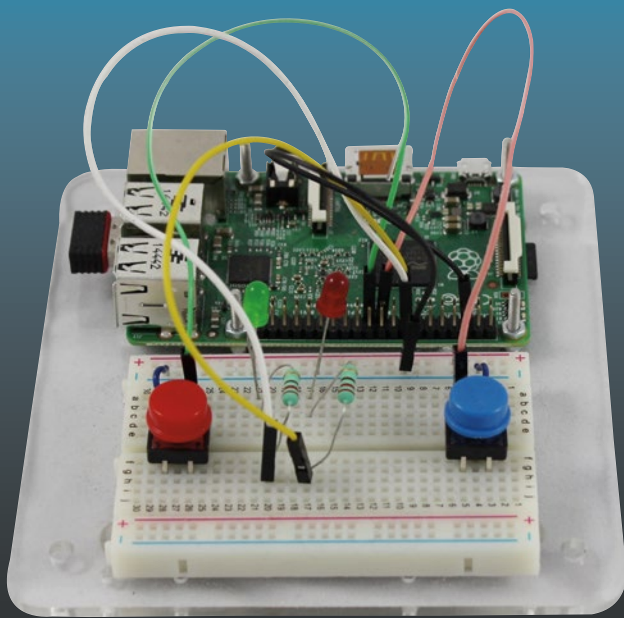
TECHNOLOGY IN ACTION™



Learn Electronics with Raspberry Pi

Physical Computing with Circuits, Sensors, Outputs, and Projects

—

Stewart Watkiss

Apress®

Learn Electronics with Raspberry Pi

Physical Computing with Circuits, Sensors, Outputs, and Projects

Stewart Watkiss

Apress®

Learn Electronics with Raspberry Pi: Physical Computing with Circuits, Sensors, Outputs, and Projects

Stewart Watkiss
Redditch
United Kingdom

ISBN-13 (pbk): 978-1-4842-1897-6 ISBN-13 (electronic): 978-1-4842-1898-3
DOI 10.1007/978-1-4842-1898-3

Library of Congress Control Number: 2016943841

Copyright © 2016 by Stewart Watkiss

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Lead Editor: Steve Anglin
Technical Reviewer: Chaim Krause
Editorial Board: Steve Anglin, Pramila Balan, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing
Coordinating Editor: Mark Powers
Copy Editor: Kezia Endsley
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail `rights@apress.com`, or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/bulk-sales`.

Any source code or other supplementary materials referenced by the author in this text is available to readers at `www.apress.com/9781484218976`. For detailed information about how to locate your book's source code, go to `www.apress.com/source-code/`. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

For Amelia and Oliver

Contents at a Glance

Contents

About the Author

Stewart Watkiss has been a keen electronics hobbyist since the early 1990s. He first studied electronic engineering at Huddersfield Technical College and subsequently at the University of Hull, where he earned a master's degree in electronic engineering.

He then became more involved in the software side, studying the Linux operating system and programming. During this time he launched a web site (`www.penguintutor.com`) to teach Linux and to help those working toward Linux certification.

His interest in electronics was revitalized thanks in part to the Raspberry Pi. Stewart created a number of projects, some of which have been featured on the Raspberry Pi blog and *The MagPi* magazine. Stewart also volunteers as a STEM Ambassador, going into local schools to help support teachers and teach programming and physical computing to teachers and children.

About the Technical Reviewer

Chaim Krause presently lives in Leavenworth, Kansas, where the U.S. Army employs him as a simulation specialist. In his spare time, he likes to play PC games and occasionally develops his own. He has recently taken up the sport of golf to spend more time with his significant other, Ivana. Although he holds a bachelor's in political science from the University of Chicago, Chaim is an autodidact when it comes to computers, programming, and electronics. He wrote his first computer game in BASIC on a Tandy Model I Level I and stored the program on a cassette tape. Amateur radio introduced him to electronics, while the Arduino and the Raspberry Pi provided a medium to combine computing, programming, and electronics into one hobby.

Acknowledgments

My family has been very supportive during the writing of this book. Thank you to my wife Sarah for her support and especially to my children, Amelia and Oliver, who have been both a source of inspiration and enthusiastic testers of the games and activities.

I'd also like to thank the team behind the Raspberry Pi, including the Raspberry Pi foundation and the community that has grown around it. The Raspberry Pi has reinvigorated my love of electronics, making it possible to interact with hobby electronics projects in a way that I'd only dreamed about before. Raspberry Jams and community events have been a great way to meet the team behind the Raspberry Pi as well as other members of the community. All this has encouraged me to pursue it further.

There have been many other people have helped in the making of this book. Special thanks to the technical reviewer, Chaim Krause, who has tested all the projects. Michelle Lowman encouraged me to write this book and Mark Powers ensured that the book moved forward. I'd also like to thank Corbin Collins for his comments and suggestions and to the rest of the team at Apress.

Introduction

Learning computer programming is fun in itself, but when the computer is connected to external sensors and outputs, your programs can interact with the real world. This is known as *physical computing* and it opens up the opportunity to create some fun projects.

I am a big fan of learning by doing. It's much easier to learn when you get to make the projects rather than just read about what other people have been done. It's even better when those projects are fun. This book covers simple projects that you can do at home to make games, control toys, create your own films, or just have fun.

The Raspberry Pi computer is great for learning physical computing, thanks to special pins that provide access to ports on the processor. These 40 pins (26 on earlier versions) provide a simple way to extend computing into the physical world. The circuits in this book are adding sensors, outputs, and electronic circuits to a Raspberry Pi. With a little bit of programming, these can do some pretty amazing things.

We will start with some simple circuits, which can be controlled from Scratch, and then move up to Python and some more complicated circuits. By then end of the book, we will have covered enough so that you can start designing your own circuits.

Most of the circuits can be created by plugging wires into a solderless breadboard, but there are tips on how to solder, which opens the possibilities further. You'll learn how to design custom circuit boards and look at how you can use some of the common Raspberry Pi add-on boards and HATs.

Who Is This Book For?

This book is for anyone who wants to learn about electronics and have fun in the process. This book focuses on fun projects so is great for older children and young adults. My eight-year-old son helped with some of the easier projects so young children could have a go with adult help. While the fun aspect appeals to younger adults, there's no maximum age for having fun, so this is just as useful to adults of any age who want to learn about electronic circuits and connecting to the Raspberry Pi.

You don't need to know anything about electronic circuits before you start. Having a basic knowledge about computers and computer programming will be useful, but is not required and will be explained as we go. We'll be using Scratch and Python, as they are good programming languages for those learning programming, but the electronic circuits can be controlled using any programming language that can communicate with the GPIO ports.

How to Use This Book

As with many books, you can read this book from cover to cover, or you can jump straight to the project that you find most interesting. The book introduces new concepts and components in each chapter. The first few chapters start with simple circuits and work through to more complicated circuits. There is an explanation of each circuit as you go, so for the first few chapters you will find it most useful to follow in order, making the circuits as you go. Most of the circuits in these first few chapters are based on low-cost components that should be within the reach of most readers.

Most of the projects are within a single chapter, but a few need some of the concepts explained in later chapters and so are split between the chapters. The notes explain which chapter you will need to refer to for the rest of the project.

Some of the later projects do use more expensive components or add-on boards for the Raspberry Pi or are designed to interact with more expensive toys. Where possible, boards have been chosen to keep costs as low as possible. You may want to just read about how the circuit works or look for the suggestions on how these can be adapted for use with cheaper components or items you already own.

Creating the examples in this book should not be considered the end. I hope that the information in this book will provide you with the inspiration and knowledge to go on to learn more about electronics and design your own circuits.

After working through the projects, it's useful to have a summary of the components so that you can refer to the book when designing your own circuits. To make it easier to refer back, I've added a summary of some of the components in the appendix along with some extra technical details that are useful when designing your own circuits.

Is Soldering Required?

When I've talked to teachers and students about electronic projects, I often get asked the question of whether soldering is required. Unfortunately, I think that this is something that puts some people off from learning electronics. I don't think it should.

First, a lot can be learned through creating circuits that don't need any soldering. Many of the projects in the first few chapters, and in some of the subsequent chapters, are designed to be made without any soldering. These are typically using solderless breadboards, but some can also be made using crocodile clips or with an inexpensive crimp tool. There are, however, some components that are not suitable for use on a breadboard, or that need a small amount of soldering so that they can be used with a breadboard. In fact many "breadboard friendly" components may need headers to be soldered on to them first.

The second point I'd like to make is that soldering is not as hard, expensive, or dangerous as some people have been led to believe. Chapter 10 explains soldering and will hopefully dispel some of the myths surrounding it. If you are still uncertain, see if you have a local maker club or Hackspace where you can speak to someone experienced in soldering.

Buying a Raspberry Pi

If you are reading this book then there is a good chance that you already have a Raspberry Pi. Since the Raspberry Pi was first released in 2012 there have been several versions. Some have had only minor changes, but one of the bigger changes was increasing the size of the GPIO connectors from 26 to 40. This was introduced in the Raspberry Pi B+ and the larger connector has been used on all new models since, including the Pi Zero and Raspberry Pi 2 and 3 models. While most of the projects in this book use only the first 26 pins, some of them do need the additional pins. If you don't have a Raspberry Pi or only have an original version with 26 pins on the connector, I recommend buying a Raspberry Pi 2, which includes a quad-core processor, or a Raspberry Pi 3 with the 64-bit processor and built-in Wi-Fi and Bluetooth. You don't need the extra processor power for the projects in this book, but it does make it possible to use the computer for other things. After all, nobody ever says, "This computer is too fast!"

The official Raspberry Pi suppliers are listed on the Raspberry Pi web site, but they are also widely stocked by various electronics and hobbyist suppliers, so you shouldn't have a problem finding a supplier.

Buying the Components

In order to follow the instructions, you will need some electronic components. There is no single kit that will provide all the items required; it will depend on which circuits you decide to make as well as the suggested variations. Details of the main components required for each project are listed in Appendix A. One thing that is worth stocking up on is a variety of different resistors. You may want to buy either an E6 or E12 series resistor multi-pack (see Appendix C for an explanation of the resistor series).

Most of the components are fairly common and can be bought from any good electronics retailer. There are several retailers that are specifically geared toward makers. In the United States, there are companies such as Adafruit and Sparkfun, and in the UK, two popular suppliers are Maplin and CPC Farnell. You may also want to look at Raspberry Pi retailers, many of whom have an increasing range of electronic sensors and other components. In particular, Pimoroni has created several add-on boards and HATs specifically for the Raspberry Pi. There are also international electronic retailers such as RS and Farnell or many smaller independent suppliers located around the world.

One thing about electronic components in particular is that a device with an almost identical name may work differently. It may be possible to substitute a similar product, but when a specific component is required, I've tried to list the specific part number to help find the correct one. Watch out for codes that are almost the same but have different electrical properties.

Installing Raspbian

The official operating system for the Raspberry Pi is *Raspbian*. It is based on Debian Linux, but has been customized for the Raspberry Pi and comes with some additional software. The operating system needs to be loaded onto an SD card before it can run. The easiest way to install Raspbian is through the NOOBs installer. More details on installing NOOBs are available on the Raspberry Pi web site at `https://www.raspberrypi.org/help/noobs-setup/`.

The Raspbian image has been updated on a regular basis since the Raspberry Pi was launched. If you've had your Raspberry Pi for some time and haven't updated it for a while, now is a good time. For many of the projects in this book, you need a version that was released from at least November 2015. If you downloaded the image after November 2015, that is sufficient, but if you installed prior to that or used a pre-installed image that may not have been updated to that version then you will need to perform an update.

In the event of minor updates to the operating system, it is normally sufficient to run a `dist-upgrade` to update to the latest version of the installed software. To do this, launch the terminal under the Accessories category of the Start menu. Once you are in the terminal shell, enter the following commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

If the version is much older, or it has been a long time since your last update, then it is recommended that you download a new version of NOOBs from the Raspberry Pi web site. The new image can then be installed onto the SD card (this will involve formatting the SD card, so wipe off any data that is stored on it).

If you are unsure of your version, you can run the following command from the terminal:

```
uname -a
```

This provides a date that indicates the date of the kernel. If no date is shown, assume it is an old version and needs to be upgraded. To upgrade from an old version, you will first need to wipe the SD card using the SD formatter tool `https://www.sdcard.org/downloads/formatter_4/` and then copy the latest NOOBs files to the SD card.

Software Required

All the software required in this book is available for free and mostly available as open source software. This does include some libraries that have been created by others, and in those cases, links have been provided to the original location.

The source code used in the book can be typed in manually or downloaded from the Internet. For most short examples, I believe you can learn more through typing and experimentation, although downloads can be useful when trying to get a circuit working or when the amount of typing is enough to make your fingers ache. You can download the source code and media files as a ZIP file at:

https://github.com/penguintutor/learnelectronics/archive/master.zip

Unzip the `master.zip` file using the following commands:

```
unzip master.zip
mv learnelectronics-master learnelectronics
```

The first command unzips the file and the second command renames the directory to `learnelectronics`. The files are then contained in the relevant subdirectories.

You are free to use the software source code and circuit designs that I have provided in your own projects, but some of the accompanying files or suggested downloads may come under a different license. In particular, some of the files are provided under a Creative Commons license. Any additional copyright information is provided in the `LICENSE` file in the appropriate directory.

Safety Information

All the circuits in this book are designed to run at a low voltage and, as long as an appropriate power supply is used, they are safe to touch. To make these projects permanent may involve the use of power tools where the safety information related to the power tools needs to be followed.

Some of the projects use bright LED lights. Some people may be sensitive to high-frequency flashing lights, which in some circumstances may cause seizures. This is more likely if the flash rate is increased beyond that used in the supplied code. Please take this into consideration when modifying the source code, especially if you're using the lights in a public space. You should also avoid looking directly into any bright light, including the LED lights.

More Electronics

The projects in this book should be considered a starting point toward achieving more using electronics and the Raspberry Pi. At the end of each chapter there is a section that provides a summary of the key points in the chapter and suggestions on how the projects could be improved or ideas for related projects. These have been left as an exercise for the reader. I hope that this will result in future projects inspired from this book and I look forward to seeing some on the Internet in the future.

CHAPTER 1

Getting Started with Electronic Circuits

Most of this book involves connecting circuits to a Raspberry Pi, but before you actually plug anything into the Raspberry Pi, you will need a basic understanding of electronic circuits. This is going to be a gentle introduction, so if you already know how to build your own simple circuits and would like to jump straight in to connecting into the Raspberry Pi, feel free to jump to Chapter 2.

An *electronic circuit* combines individual electronic components to perform a specific function. This could be as simple as a light circuit in a torch/flashlight that turns on when the on switch is pressed or incredibly complex such as the circuitry inside a laptop computer. Electronic circuits are all built around the same principles.

The most basic principle is the concept that an electronic circuit must make a complete physical circuit. So for a circuit that includes a battery, there must be a complete path starting from the positive (+) side of the battery, through any components (such as a switch and buzzer), and then back to the negative (-) side of the battery. This is shown in the circuit in Figure 1-1.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-1898-3_1) contains supplementary material, which is available to authorized users.

© Stewart Watkiss 2016
S. Watkiss, *Learn Electronics with Raspberry Pi*, DOI 10.1007/978-1-4842-1898-3_1

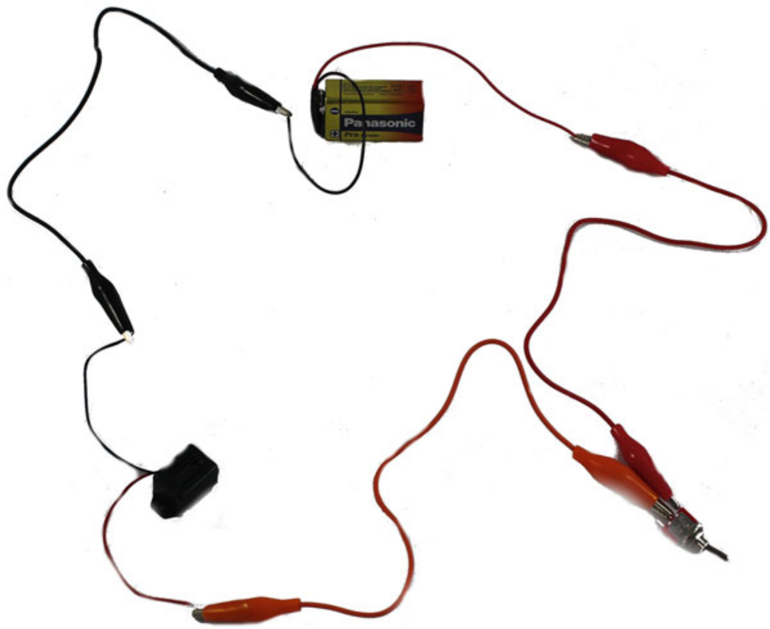


Figure 1-1. *Switch and buzzer circuit*

This is a simple circuit connected using crocodile clip leads. The circuit has a buzzer and a switch, which can turn the buzzer on and off. When the switch is closed (turned on), the circuit is complete, allowing current to flow around the circuit, and the buzzer will sound. When the switch is open (off), there is a gap between the connections inside the switch preventing the current flow and causing the buzzer to stop.

Obviously this is a very basic circuit, but it's also the basis of almost all the circuits you'll make. You will replace the mechanical switch with an electronic component and use different sensors to turn the switch on and off. You will also use different outputs, including LEDs and motors.

Voltage, Current, and Resistance

I'm going to keep the theory as simple as possible, but voltage, current, and resistance are some terms that I use throughout the book. Understanding how the circuit works and the math involved is going to be important by the time you get to the stage where you are designing your own circuits. I have avoided putting too much math into the projects, but there are some worked examples that you may find useful in future.

The *voltage* is the difference in energy between two terminals or points in a circuit. It is measured in volts indicated by a letter V. For example, you could have a 9V PP3 battery such as the one used in the buzzer circuit in Figure 1-1. The battery has a difference of 9 volts between its positive and negative terminals. We would consider the negative terminal to be at 0 volts and the positive terminal at 9 volts. The 0V connection is considered the ground terminal with voltages relative to that point. Although the battery is designed for 9V, the actual voltage may vary depending on how much charge is in the battery and what load is connected to it.

The *current* is the flow of electric charge around a circuit. Current is measured in *amperes*. This is normally abbreviated to amps and its value is indicated by a letter A. The current is related to the voltage: the higher the voltage of the power supply, the more current is able to flow through the circuit, although it also depends on what other components are in the circuit. Using conventional electric current, you say that the current flows from the positive to the negative terminal. In the electronic circuits you'll create, most currents will be small and so will normally be measured in *milliamps* (mA), where 1mA = 0.001A.

The electrical *resistance* is a measure of how difficult it is for the current to flow around a circuit. It is measured in *ohms*. The value ohms is represented by the Greek omega character (Ω). There is resistance in all components of a circuit, but you can normally disregard the small amount of resistance in a good conductor such as a wire. The resistors you will be using normally range from around two hundred ohms to several thousand ohms (kΩ).

Ohm's Law

When creating advanced circuits, some of the math can get quite complicated; fortunately, you don't need to do many calculations for most of the circuits in this book. However, there are still some basic calculations that you will need to do to. In particular for some of the circuits, you will need to work out a suitable resistor size to ensure that the current cannot damage any components, but is sufficient to allow the circuit to work.

To do this, you use a single formula that's almost certainly the most important formula used in electronics. It's also one of the simplest. This relationship was discovered by German scientist Georg Ohm and is known as Ohm's Law. The basic formula is this:

$$I = V/R$$

As you may expect, V represents voltage and R represents resistance (measured in ohms), but *I* is not so obvious. I is used to indicate current, based on the French phrase, "intensité de courant".[1]

So this formula says that to find the current through a circuit, divide the voltage by the resistance. This can be rearranged to find the voltage using this formula:

$$V = I \text{ x } R$$

To calculate the required resistor size, use:

$$R = V/I$$

An easy way to remember this is using the Ohm's Law triangle, shown in Figure 1-2.

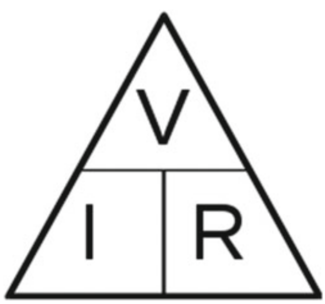


Figure 1-2. *Ohm's Law triangle*

[1]Much of the early research into electric current and magnetism was pioneered by the French scientist André-Marie Ampère. His surname is also used for the measure of current known as an ampere, which is usually abbreviated to amps.

To use the triangle, hide the value you want to calculate and read the remaining entries. So to find the voltage, you hide the letter V, leaving I and R. So multiply the current and resistance to find the voltage. To find the required resistor size, hide the letter R which leaves V above I. So you divide the voltage by the current to get the required resistor size.

Electrical Safety

Electricity can be dangerous. All the projects in this book are designed to work at low voltages up to 12V and, as long as an appropriate safe power supply (such as a wall wart or plug-in power supply) is used, there is little risk of electric shock. The same does not apply to the high voltage present in the main electricity supply.

In fact it's not the voltage that's dangerous but the amount of current that can flow through the body. Electric fences used for farm animals give a shock of several thousand volts, but although they give a nasty shock, they are considered safe for use near people as they are limited to short bursts of very low current (although should still be avoided particularly by children or those with heart conditions). The main electricity to your house is between about 100V and 250V (depending on which country you live in) and is very dangerous—it can supply enough current to be fatal. As a general rule to avoid any risk of electric shock, I recommend only working with circuits designed for 24V or less, unless you are 100% sure you know what you are doing.

It's not only electric shock that poses a risk. Even at much lower voltages, too much current can create a lot of heat and potentially start a fire. This is particularly important when using low voltage (12V) electrical lighting or car batteries, which can provide very high currents in the event of a short-circuit. I recommend only using power supplies with short-circuit and over-current protection and consider adding a fuse (this is explained later during the disco lights project).

■ **Caution** Do not try to connect any of these circuits to the main electricity in your home, except using the appropriate power supply adapter.

ANALOG VS. DIGITAL

The world we live in is varied. If we take sound as an example, we may use various words to describe the amount of sound something is making, from saying that someone is very quiet, or that the MP3 player is very loud, or even that a pneumatic drill is deafening. We don't normally know or care about the actual values of the sound (measured in decibels), but we do know if we want it to be louder or quieter. A computer however does not understand these terms. It only deals in actual values. In fact, at a most basic level, it only thinks of things being on and off, but it can compare against different levels to interpret this as a more accurate value.

An *analog circuit* can interpret any number of variations of the input. Perhaps one of the few remaining purely analog circuits you will find at home today is a simple amplifier built into a set of speakers. Here, as you turn the volume control clockwise, the volume smoothly increases compared to the input signal. Compare this to a modern TV, where you press the volume button on the remote control and the volume moves up a fixed amount, say between 1 and 40.

Most electronic circuits are now digital and in fact most include some kind of micro-processor, either a full computer such as a Raspberry Pi or a more basic micro-controller such as the ATMega micro-controllers used in the Arduino. The real world continues to be analog, so there is often an analog sensor or output and an element of conversion between analog and digital and vice versa.

Breadboard

Many of the circuits in this book are built on a solderless breadboard, sometimes called a *plugboard*. A breadboard is a good way of creating temporary circuits to allow testing prior to committing with solder. They consist of a plastic board with a matrix of holes. The holes are then connected in small rows so that components plugged into the same section are connected.

Breadboards are very easy to use and don't damage the components, so it's easy to change the circuit and experiment with different components. If you don't want the circuit any more, then the components can be removed and both the breadboard and the components can be used again for another circuit. Integrated circuits (ICs) can be also be inserted and wired to other components. To connect wires to a breadboard, you should use solid core wire or special jumper wires that have a solid end that can be plugged into the breadboard. The alternative type of wire is known as *multi-stranded wire* and it's more flexible and so more popular with soldered circuits, but it doesn't plug into the breadboard properly.

Breadboards are available in a variety of sizes, from small ones with 170 holes to large boards with multiple breadboards mounted onto a single panel. You can also connect multiple boards together, which slot together. Unfortunately, there is no standard for how the boards slot together, so this may work only if you're using the same manufacturer. A selection of different breadboards is shown in Figure 1-3.

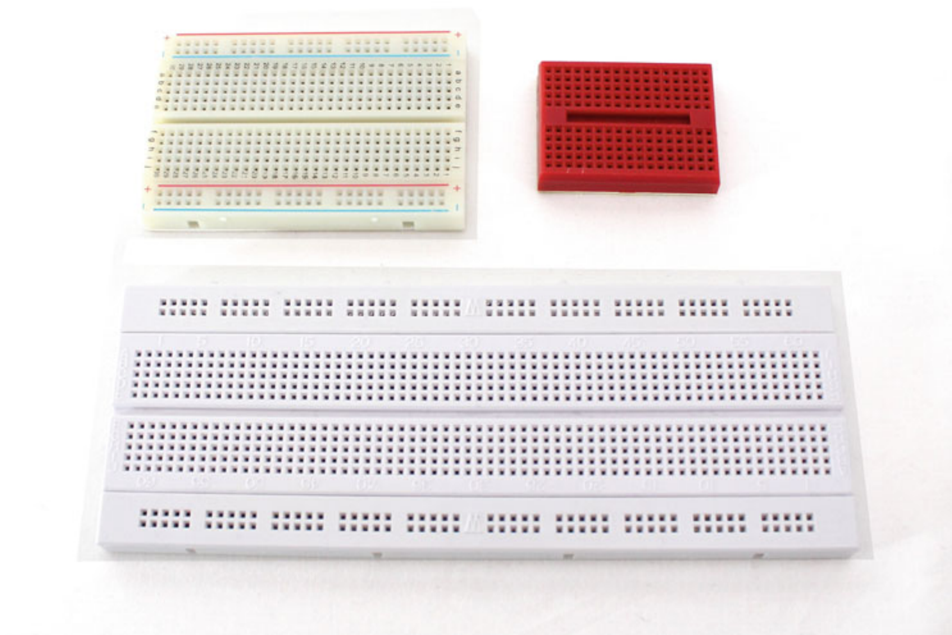


Figure 1-3. *A selection of different breadboards*

Each size of breadboard has a set of circumstances in which it works best. The smallest can be included in a small box, whereas the large one is great for larger circuits and includes connectors that are useful for plugging banana plugs from an external power supply.

For most of the circuits in this book, the half-size breadboard is an ideal size. It's about the same size as the Raspberry Pi and is a good compromise between the space taken up and the amount of space needed for connecting circuits. An example of the half-size breadboard layout is shown in Figure 1-4.