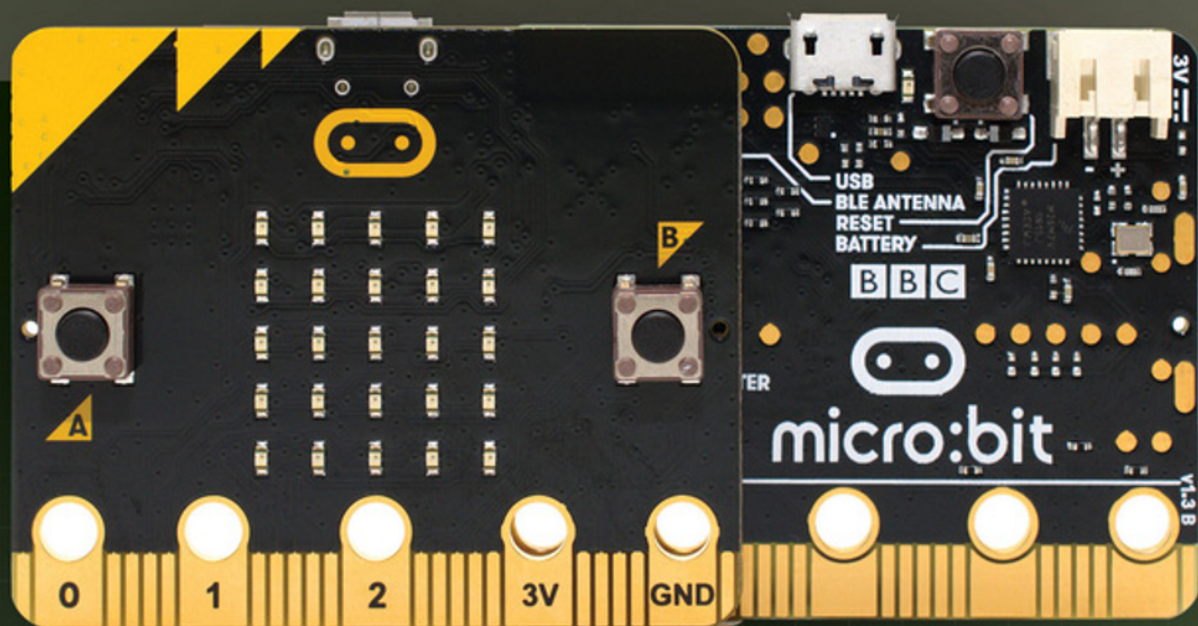


GARETH HALFACREE

# THE OFFICIAL BBC MICRO:BIT<sup>®</sup> USER GUIDE



WILEY



# **The Official BBC micro:bit<sup>®</sup> User Guide**

Gareth Halfacree

## **The Official BBC micro:bit® User Guide**

Published by  
John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2018 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-38673-5

ISBN: 978-1-119-41376-9 (ebk)

ISBN: 978-1-119-41384-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2017950180

**Trademarks:** Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Micro:bit is a registered trademark of the British Broadcasting Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

*For my father, the enthusiastic past, and my daughters,  
the exciting future.*

—Gareth

## About the Author

**GARETH HALFACREE** is a freelance technology journalist and the co-author of the *Raspberry Pi User Guide* alongside project co-founder Eben Upton. Formerly a system administrator working in the education sector, Gareth's passion for open source projects has followed him from one career to another, and he can often be seen reviewing, documenting or even contributing to projects such as GNU/Linux, LibreOffice, Fritzing and Arduino. He is also the creator of the Sleepduino and Burnduino open hardware projects, which extend the capabilities of the Arduino electronics prototyping system. A summary of his current work can be found at <http://freelance.halfacree.co.uk>.

## About the Technical Editor

**DAVID WHALE** is an embedded software engineer whose career of over 30 years has involved him designing and building embedded software for a diverse collection of high tech products. David is a STEM Ambassador and volunteer for the Institution of Engineering and Technology (IET), where he regularly helps schools and teachers introduce and run computing clubs, as well as trains teachers nationally. It was through his association with the IET that David became involved with the micro:bit project, where he has helped to develop a large base of teaching resources, as well as train and support the delivery of the IET Faraday national STEM challenge days using the BBC micro:bit. David now works with the Micro:bit Educational Foundation, where he continues to develop projects and resources with many partner organisations, which includes the Doctor Who team at the BBC. David is the co-author of the successful Wiley title *Adventures in Minecraft*, a book that teaches Python coding to children via their interest in the Minecraft game, and he has been technical editor of a wide range of technology and computing books.

## Credits

### Project Editor

John Sleeva

### Technical Editor

David Whale

### Production Editor

Barath Kumar Rajasekaran

### Copy Editor

Karen A. Gill

### Production Manager

Katie Wisor

### Manager of Content Development and Assembly

Mary Beth Wakefield

### Marketing Manager

Christie Hilbrich

### Professional Technology & Strategy Director

Barry Pruett

### Business Manager

Amy Knies

### Executive Editor

Jody Lefever

### Associate Acquisitions Editor

Riley Harding

### Project Coordinator, Cover

Brent Savage

### Proofreader

Debbye Butler

### Indexer

Estalita M. Slivoskey

### Cover Designer

Wiley

### Cover Image

Courtesy of Gareth Halfacree

# Contents

Foreword . . . . . xi

Introduction . . . . . xiii

## Part I

### CHAPTER 1

Meet the BBC micro:bit . . . . . 3

A Tour of the Board . . . . . 3

Breaking It Down . . . . . 5

Display . . . . . 6

Buttons . . . . . 7

Processor . . . . . 8

Radio . . . . . 9

Accelerometer . . . . . 10

Compass . . . . . 11

Input-Output Pins . . . . . 12

Micro-USB Port . . . . . 13

Battery Connector . . . . . 14

### CHAPTER 2

Getting Started with the BBC micro:bit . . . . . 17

Handling the BBC micro:bit . . . . . 17

Powering the BBC micro:bit . . . . . 18

USB Power . . . . . 18

Battery Power . . . . . 20

Greetings from the BBC micro:bit . . . . . 23

Signs of Life . . . . . 24

Testing the Buttons . . . . . 24

Motion Gaming . . . . . 24

Get Coding . . . . . 25

Resetting the BBC micro:bit . . . . . 25

### CHAPTER 3

Programming the BBC micro:bit . . . . . 27

USB Connectivity . . . . . 27

Drag-and-Drop . . . . . 29

Automatic Flashing . . . . . 31

The Code Editor . . . . .	32
Downloading Your Program . . . . .	33
About Flash Memory . . . . .	38

## Part II

### CHAPTER 4

<b>Programming Languages . . . . .</b>	<b>41</b>
About Programming Languages . . . . .	41
The Three Main BBC micro:bit Languages . . . . .	42
JavaScript Blocks . . . . .	43
JavaScript . . . . .	44
Python . . . . .	45
Comparing Programming Languages . . . . .	46
Choosing a Programming Language . . . . .	48
Other Programming Languages . . . . .	49

### CHAPTER 5

<b>JavaScript Blocks . . . . .</b>	<b>51</b>
Introducing the JavaScript Blocks Editor . . . . .	51
Program 1: ‘Hello, World!’ . . . . .	54
Loops . . . . .	57
Program 2: Button Inputs . . . . .	58
Multiple Buttons . . . . .	60
Program 3: Touch Inputs . . . . .	61
Variables . . . . .	62
Program 4: The Temperature Sensor . . . . .	65
Formatting the Output . . . . .	67
Program 5: The Compass Sensor . . . . .	67
Program 6: The Accelerometer Sensor . . . . .	70
Delays . . . . .	73
Reading Raw Accelerometer Data . . . . .	74
Program 7: The Fruit Catcher Game . . . . .	76
The Setup . . . . .	77
The Main Program Loop . . . . .	78
Conditional Loops . . . . .	80
Conditional Statements . . . . .	82
The Control Events . . . . .	84
Further Steps . . . . .	86



**CHAPTER 6**

**JavaScript . . . . . 87**

- Introducing the JavaScript Editor . . . . . 88
- Program 1: ‘Hello, World!’ . . . . . 90
  - Loops . . . . . 93
- Program 2: Button Inputs . . . . . 94
  - Multiple Buttons . . . . . 97
- Program 3: Touch Inputs . . . . . 98
  - Variables . . . . . 99
- Program 4: The Temperature Sensor . . . . . 102
  - Formatting the Output . . . . . 104
- Program 5: The Compass Sensor . . . . . 104
- Program 6: The Accelerometer Sensor . . . . . 107
  - Delays . . . . . 109
  - Reading Raw Accelerometer Data . . . . . 110
- Program 7: The Fruit Catcher Game . . . . . 112
  - The Setup . . . . . 113
  - The Main Program Loop . . . . . 115
  - The Conditional Loops . . . . . 116
  - The Conditional Statements . . . . . 117
  - The Control Events . . . . . 120
- Further Steps . . . . . 123

**CHAPTER 7**

**Python. . . . . 125**

- Introducing the Python Editor . . . . . 126
- Program 1: ‘Hello, World!’ . . . . . 128
  - Loops . . . . . 132
- Program 2: Button Inputs . . . . . 133
  - Multiple Buttons . . . . . 136
- Program 3: Touch Inputs . . . . . 137
  - Variables . . . . . 138
- Program 4: The Temperature Sensor . . . . . 141
  - Formatting the Output . . . . . 142
- Program 5: The Compass Sensor . . . . . 143
- Program 6: The Accelerometer Sensor . . . . . 145
  - Delays . . . . . 147
  - Reading Raw Accelerometer Data . . . . . 148

Program 7: The Fruit Catcher Game . . . . .	150
The Setup . . . . .	150
The Main Program Loop . . . . .	153
Conditional Loops . . . . .	154
Conditional Statements . . . . .	155
Drawing the Sprites . . . . .	156
Finishing the Program . . . . .	157
Further Steps . . . . .	160

## **Part III**

### **CHAPTER 8**

#### **The Wireless BBC micro:bit . . . . . 163**

The BBC micro:bit Radio . . . . .	163
Program 1: One-to-One Communication . . . . .	164
Program 2: One-to-Many Communication . . . . .	167
Program 3: Radio Groups . . . . .	169
Testing the Group Feature . . . . .	171
Using the BBC micro:bit with a Smartphone or Tablet . . . . .	173

### **CHAPTER 9**

#### **The BBC micro:bit and the Raspberry Pi . . . . . 175**

About the Raspberry Pi . . . . .	176
Connecting the Raspberry Pi to the BBC micro:bit . . . . .	177
Reading from the BBC micro:bit . . . . .	180
Using the BBC micro:bit Display . . . . .	186
Practical Example: A CPU Monitor . . . . .	189

### **CHAPTER 10**

#### **Building Circuits . . . . . 193**

Electronic Equipment . . . . .	194
The Input-Output Pins . . . . .	196
The Large Pins . . . . .	197
The Small Pins . . . . .	199
Serial Peripheral Interface (SPI) . . . . .	201
Inter-Integrated Circuit (I <sup>2</sup> C) . . . . .	201
Universal Asynchronous Receiver/Transmitter (UART) . . . . .	201

Your First Circuits . . . . .	202
Reading from a Button Input . . . . .	202
Reading Resistor Colour Codes . . . . .	205
Writing to an LED Output . . . . .	207
Fading an LED via PWM . . . . .	211
Reading an Analogue Input . . . . .	213
<b>CHAPTER 11</b>	
<b>Extending the BBC micro:bit . . . . .</b>	<b>217</b>
Extending via Breakout Boards . . . . .	217
Kitronik Edge Connector Breakout Board . . . . .	218
ScienceScope Micro:bit Breakout Board . . . . .	219
Proto-Pic Bread:Bit . . . . .	220
Proto-Pic Exhi:Bit . . . . .	220
Robotics and the BBC micro:bit . . . . .	222
Kitronik Line-Following Buggy . . . . .	222
Kitronik Motor Driver Board . . . . .	223
Technology Will Save Us Micro:Bot . . . . .	224
4tronix Bit:Bot . . . . .	225
BinaryBots . . . . .	226
Other BBC micro:bit Add-Ons . . . . .	227
Kitronik Mi:Power . . . . .	227
Proto-Pic Micro:Pixel Board . . . . .	228
Proto-Pic Simon:Says Board . . . . .	229
4tronix Bit:2:Pi Board . . . . .	230
Kitronik Mi:Pro Protector and Mi:Power Cases . . . . .	231
<b>CHAPTER 12</b>	
<b>The Wearable BBC micro:bit . . . . .</b>	<b>233</b>
Advantages of the Wearable BBC micro:bit . . . . .	234
Conductive Thread . . . . .	235
Using Conductive Thread . . . . .	237
The Rain-Sensing Hat . . . . .	241
Building the Hat . . . . .	242
Mounting the BBC micro:bit . . . . .	244
The Rain-Sensing Program . . . . .	245
Battery Power . . . . .	246

**CHAPTER 13**

**Additional Resources . . . . . 249**

- The Micro:bit Educational Foundation . . . . . 249
- Official Teaching Resources . . . . . 251
- Third-Party Teaching Resources . . . . . 252
  - The Institution of Engineering and Technology . . . . . 252
  - Computing At School . . . . . 253
  - Micro:bit for Primary Schools . . . . . 253
  - TES Magazine . . . . . 255
  - Code Clubs . . . . . 256

**Part IV**

**APPENDIX A**

**JavaScript Blocks Recipes . . . . . 259**

**APPENDIX B**

**JavaScript Recipes . . . . . 267**

**APPENDIX C**

**Python Recipes . . . . . 275**

**APPENDIX D**

**Pin-Out Listing . . . . . 285**

**Index . . . . . 289**

# Foreword

**IN APRIL 2015**, I spent a lunchbreak searching my local supermarket for the cheapest thing I could cannibalise to finish a demo for the BBC. What they saw that afternoon was the battery clip from a £1 alarm clock, glued onto an early BBC micro:bit prototype. I was building a demo as one of the 31 partner organisations that the BBC had recruited to help them deliver a bold and audacious project: to give a coding device to every year 7 in the UK, for free. However, the BBC micro:bit wasn't designed to be just another programmable 'development board', but a seamless plug-and-play tool that puts creativity, learning, and ease-of-use for teachers and young people first.

As an engineer on the project, the most compelling thing about working with the BBC micro:bit is seeing the exciting (and sometimes ridiculous) things these new audiences choose to do with this technology: build a team game based on a teleporting duck, measure a rocket car's acceleration, tell the interactive story of pizza, build a portable heart-rate-monitor, or invent the fantastic 'rain detecting hat' you can find in this book. This guide brilliantly captures the exhilaration, simplicity, and creative potential of the BBC micro:bit, and I'm sure it will help many more people become coders and inventors.

Instantly interactive, the BBC micro:bit's sensors and slick design make it accessible and exciting to the widest possible audience, even (perhaps especially) those who didn't realise that coding was for them. It takes no time at all to build your first program, and the simplicity of the tools means that what follows is an upward spiral of success and satisfaction that encourages your imagination to run wild.

Part of the magic of the BBC micro:bit is how effortlessly it brings the virtual and physical together. Within minutes, things you've made with the computer start controlling the real world. It's this physicality and immediacy that create the 'micro:bit moment'. It's amazing to see people of all ages have that empowering experience when they realise the potential of the device in their hands, and want to make something new with it—the moment people start to feel excited about playing with technology instead of nervously holding back from it.

But micro:bit isn't really about learning to code; it's about learning to do things that you care about with technology. Learning to code happens along the way: you learn about loops almost by accident because you're making your BBC micro:bit dance, and the song has a verse and chorus that repeat. This approach helps engage new coders of all ages. Independent research tracked the progress of the BBC micro:bit in the UK during its first year and found that 85% of teachers who used the device agreed that it made Computer Science more enjoyable for their students, and 90% of the children who used it said it helped show them that anyone can code.

The cool design, broad appeal, and technological simplicity of the micro:bit tools come as a direct result of the diverse backgrounds, outlooks, communities, and interests within the micro:bit partnership. This broad, interdisciplinary team has shaped the device, the editors, the learning materials, and the concept of the micro:bit itself into a truly unique ecosystem.

Founded in September 2016, the Micro:bit Educational Foundation has been entrusted with supporting and developing that ecosystem, and taking the micro:bit project around the world. At the time of writing, the device is available in more than 50 countries, with the website available in 12 languages. We have a growing library of lesson plans, projects and ideas, new, more advanced editors developed with our partners (which are explained in this book), and a growing community of enthusiasts, volunteers, great partners, and educators.

So, getting started with the BBC micro:bit isn't just about building your own creations. It's also about joining a worldwide community of people who are using technology creatively to express themselves, solve problems they care about, make life better for others, and help change the way it feels to learn to code.

This book will introduce you to the world of micro:bit, but it's only the start of your journey. Welcome to the micro:bit community. We're looking forward to seeing what you create!

—Jonny Austin, CTO, Micro:bit Educational Foundation

# Introduction

**EDUCATIONAL COMPUTING HAS** a long and storied history, beginning with the adoption of mechanical calculators to aid with mathematics classes through to the early days of microcomputing with initiatives like the BBC Computer Literacy Project in the 1980s. As the cost of computers came down and their capabilities increased, schools around the world rapidly went from a single shared computer to entire rooms filled with computers, integrating them into lessons from languages and history to engineering and art.

Today many homes have a computer of their own, or in some cases more than one. While access to computers has increased, actually operating them brings with it a sense of being disconnected from their inner workings. The BBC Micro, the 1980s microcomputer designed by Acorn Computers and at the heart of the BBC Computer Literacy Project, loaded straight into a text-based programming language known as the Beginner's All-Purpose Symbolic Instruction Code (BASIC) and invited experimentation; today, the majority of computers load into a graphical user interface (GUI) which emphasises the use of pre-written programs over creating your own.

The BBC micro:bit is designed to bring back the days of learning to write your own code on a low-cost, easily-understood platform. Designed to sit at the heart of an international computer literacy programme directly inspired by the BBC's original Computer Literacy Project, the BBC micro:bit is an affordable microcontroller on which you can run your own programs to do everything from spell out your name or play a game to turn lights on and off or communicate via radio.

Schools around the world have begun to adopt the BBC micro:bit in their curricula, but it's by no means exclusively for structured educational use. The device's readily accessible nature means it can be used to teach programming and computing concepts to children of any age, its robustness and small size mean it's extremely portable and well-suited to wearable projects, and its surprising power and flexibility mean that you're unlikely to find yourself restricted by its capabilities even when using it at the heart of more complex projects.

## Who This Book Is For

This book is written for anyone interested in working with the BBC micro:bit. You don't need pre-existing knowledge of computers, electronics, or programming to be able to pick up a BBC micro:bit and get started.

All you need to get the most from this book is the ability to read and a willingness to learn. If you've used computers before, you'll find that you are able to skim through some of the early chapters on general concepts. If you've used other microcontroller-based development boards, you can skip still more. If you've written your own computer programs, then you'll find programming for the BBC micro:bit immediately familiar. And if you know your way around an electrical circuit, the later chapters should hold few surprises.

Whether you're an existing user of the BBC micro:bit or a complete newcomer, this book aims to get you started on your journey with a minimum of fuss and maximum enjoyment.

## What This Book Covers

The march of technology is constant, and the BBC micro:bit is no exception. This book has been written based on the most recent version of the BBC micro:bit hardware, revision 1.3b, but it is entirely applicable to all versions going back to the first prototype versions given to schools for testing purposes. Equally, it will remain applicable to all future revisions thanks to the sterling work of the Micro:bit Educational Foundation, which has been instrumental in the development of this book.

The software for the BBC micro:bit is, as with the hardware, under constant development. References to the software in this book are accurate at the time of writing, and all screenshots and related materials have been captured on the very latest software versions. Over time, small changes may be made to the way the software looks, but the way it works will remain the same.

This is especially important for the programs contained within the book. Although the languages available for the BBC micro:bit will expand over time and gain additional features, the existing features will always be available. A program taken from this book today will still be usable with the BBC micro:bit years down the line.

## How This Book Is Structured

Part I, 'An Introduction to the BBC micro:bit', offers a guide to the hardware and how it works, step-by-step instructions on unpacking your first BBC micro:bit and exploring its sample program, and connecting the BBC micro:bit to your computer so you can load a program of your own. This section also contains a number of tips on working with the BBC micro:bit, including how to handle it to protect it against possible damage. Even if you've already worked with a BBC micro:bit, reading this section is well advised.



Part II, ‘Coding for the BBC micro:bit’, gets you started writing your own programs. You learn the languages available for the BBC micro:bit and how they differ, and you learn to write your own programs from a simple message scrolling across the BBC micro:bit’s display and reading from its various built-in sensors to writing a simple game.

There’s a chapter dedicated to each of the three main programming languages used with the BBC micro:bit: JavaScript Blocks, JavaScript, and Python. Each chapter is designed to be as close to identical to the others as possible, allowing you to quickly see how the process of writing each program differs between languages. You can use the comparison table at the start of the section to pick a favoured language and read only that language’s chapter, or you can work through all three chapters in turn to get a real feel for how each operates.

Part III, ‘Advanced BBC micro:bit Projects’, goes a step further, introducing the radio module with examples on communicating between individual BBC micro:bits and groups of BBC micro:bits without the need for wires. There’s also a chapter dedicated to using the BBC micro:bit with the popular Raspberry Pi educational single-board computer, extending the capabilities of both devices.

In addition, you learn how to add external components like switches and LEDs to the BBC micro:bit, building electronic circuits from basic components to further extend its functionality. There’s no soldering involved, and the circuits described are designed to be safe and accessible for even the youngest reader; they require only a small number of affordable electronic components.

You will now have a sound understanding of how the BBC micro:bit works, how to program it, and how to use it with other devices. You still won’t have reached the end of its capabilities, though, so the final chapter offers information on additional resources, including add-on hardware which can further increase the BBC micro:bit’s flexibility, and websites offering more project ideas and formal lesson plans for use in structured education.

Finally, the appendices have full program listings for every program mentioned in the book in all three languages, making it easy to type them in without getting distracted by comments and explanations of what each part of the program is doing. If you’d rather save your fingers, you can download the same program files from the book’s website at [www.wiley.com/go/bbcmicrobituserguide](http://www.wiley.com/go/bbcmicrobituserguide). You’ll also find a *pin-out* diagram of the BBC micro:bit with a full list of its capabilities.

## What You Need to Use This Book

Technically speaking, you can begin using this book even without a BBC micro:bit of your own; simulators allow you to write programs designed for the BBC micro:bit and see how

they run even without loading them onto a physical BBC micro:bit. You'll get the most from the book if you have at least one BBC micro:bit with which to experiment, however, along with a few extras detailed here.

To run the main program samples listed in this book, you need the following:

- A BBC micro:bit
- A fully-wired micro-USB cable
- A computer running Microsoft Windows, Apple macOS, or Linux, with a free USB port
- An up-to-date web browser and working Internet connection

For the radio programs included in Chapter 8, 'The Wireless micro:bit', you also need:

- A total of three BBC micro:bits

To run the programs listed in Chapter 9, 'The BBC micro:bit and the Raspberry Pi', you need:

- A Raspberry Pi Model B+, Raspberry Pi Model 2, Raspberry Pi Model 3, or Raspberry Pi Zero W
- A micro-USB On-The-Go (OTG) adapter cable, if using the Raspberry Pi Zero W

To build the circuits detailed in Chapter 10, 'Building Circuits', you also need the following:

- Wires with crocodile clip or 4mm banana plug connectors
- A button or switch
- An LED
- A current-limiting resistor (see Chapter 10 for an explanation)
- A potentiometer

These parts are readily available via the Internet or in high-street electronics component shops, and they can frequently be supplied by BBC micro:bit resellers alongside the BBC micro:bit itself.

## Conventions

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

Technical terms are presented *in italic* when they're first used. The same applies to acronyms and initialisms, which are presented in full when first used and then in their abbreviated form.

Metric measurements are used throughout this book, with imperial measurements provided in brackets where appropriate.

When a line of code would extend past the border of the page, a ↔ symbol is printed. When you see this symbol, continue to type the code without pressing the Enter or Return keys. If you're not sure how a line of code should be entered, visit the website at [www.wiley.com/go/bbcmicrobituserguide](http://www.wiley.com/go/bbcmicrobituserguide) to download plain-text versions of each program; these can then be used for reference or even simply copied and pasted directly into the editors.

## Contact Me

Comments, corrections, and questions from readers are heartily welcomed via email at [microbit@halfacree.co.uk](mailto:microbit@halfacree.co.uk), while other publications of mine can be found at [freelance.halfacree.co.uk](http://freelance.halfacree.co.uk).

You can also get in touch with me via Twitter at [twitter.com/ghalfacree](https://twitter.com/ghalfacree) and via encrypted message at [keybase.io/ghalfacree](https://keybase.io/ghalfacree).

Enjoy the book, and happy travels on your BBC micro:bit journey!

—Gareth Halfacree



Part I

# An Introduction to the BBC micro:bit

**Chapter 1** Meet the BBC micro:bit

**Chapter 2** Getting Started the BBC micro:bit

**Chapter 3** Programming the BBC micro:bit



# Chapter 1

## Meet the BBC micro:bit

### In this chapter

- A look at what the BBC micro:bit is and how it came about
- A tour of the BBC micro:bit and an explanation of its major components

**THE BBC MICRO:BIT** is an incredible device, capable of educating and entertaining in equal measure. It can form the heart of a complex robotic or home automation system, or it can simply show a smiley face when you press a button. It can help you come to grips with programming, learn about how electronics work, and even communicate wirelessly with more BBC micro:bits or other devices such as a smartphone or tablet.

You can write programs for the BBC micro:bit in a variety of *programming languages*, or you can use programs others have written. You can use the BBC micro:bit in the classroom, the club, the playground, or at home. You can play games, solve problems, and invent new devices, all with your BBC micro:bit.

Before all this, though, you'll need to meet the BBC micro:bit.

### A Tour of the Board

The BBC micro:bit is what is technically known as a *microcontroller development board*. That is, it's a *printed circuit board* (PCB) which contains a *microcontroller* on which you can run your own programs and connect your own hardware.

The first microcontroller development boards were expensive and complicated to use. In the decades since their first introduction, they have become steadily cheaper and more accessible, until the BBC micro:bit became possible: a minimal-cost, highly-functional board designed to help teach programming, or ‘coding’, to anyone regardless of experience.

Your programming journey begins simply: learning about the BBC micro:bit itself.

**WARNING**

The BBC micro:bit is designed to be robust, but it’s still a complex electronic device. It’s designed as a bare circuit board so that you can see what all its components are and what they do, but this does mean you need to take a little more care with handling it than if it were in an enclosure. Always make sure to handle it by its edges to avoid damage through *electrostatic discharge*. For more information on preventing electrostatic discharge damage, see Chapter 2, ‘Getting Started with the BBC micro:bit’.

The BBC micro:bit itself is a small printed circuit board with a range of components fitted. It has two sides: the front side of the BBC micro:bit includes the *display* and *buttons*, while the back side has components like the *micro-USB connector* and *radio*. Figure 1-1 shows the front side of the BBC micro:bit, and Figure 1-2 shows the back.

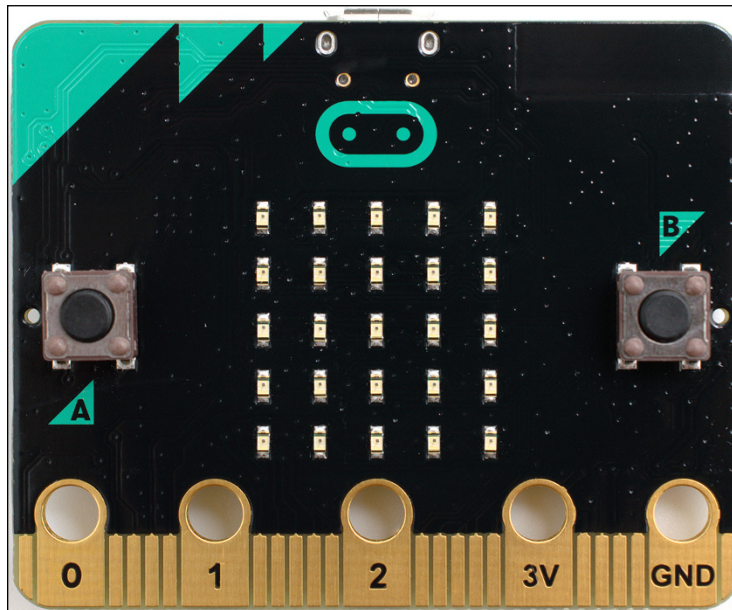


FIGURE 1-1: The front of the BBC micro:bit





FIGURE 1-2: The back of the BBC micro:bit

The BBC micro:bit includes a layer of print known as a *silkscreen layer*, which labels some of the more important components. The buttons on the front side are labelled A and B so you know which is which, while the back side picks out key components like the *processor* and the *accelerometer*. These components, and others, are described in more detail in the next section.

## Breaking It Down

Like any complex device, the BBC micro:bit is made up of numerous relatively simple parts. From the more obvious, like the display which dominates the front of the board, to the more subtle, like the radio which allows two or more BBC micro:bits to communicate with each other or connect to a smartphone or tablet, each component works with the others to give the BBC micro:bit its impressive flexibility.

If you're impatient to get started with your BBC micro:bit, you can skip ahead to Chapter 2. Reading the rest of this chapter, though, will give you a good understanding of just what the BBC micro:bit can do, putting you in good stead for not only learning how it works but coming up with practical projects of your own as your skills progress.

## Display

The most obvious feature of the BBC micro:bit is its display, which sits in the centre of the board's front side (see Figure 1-3). This is the BBC micro:bit's primary *output device*, a means for a program running on the BBC micro:bit to communicate with the outside world—whether that's reading off the position of one of the BBC micro:bit's sensors or simply displaying a smiling face.

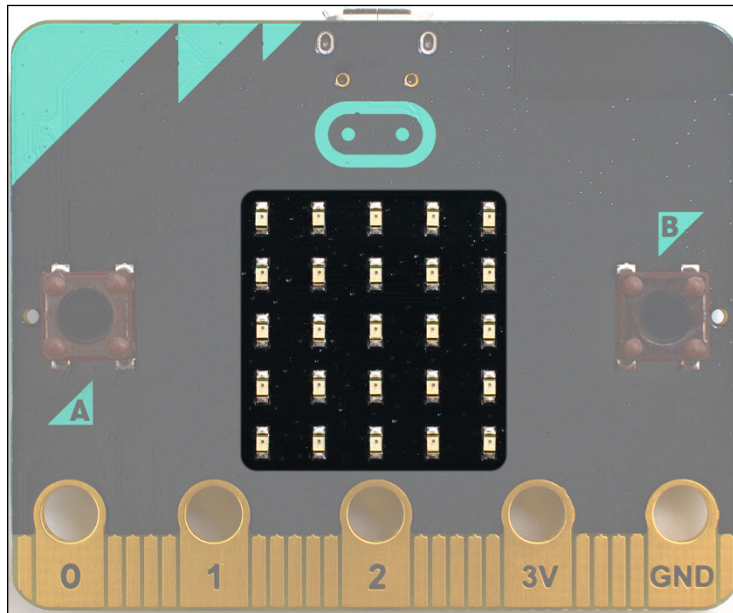


FIGURE 1-3: The BBC micro:bit's display

The BBC micro:bit's display serves the same purpose as the monitor on a desktop computer but is of a considerably lower *resolution*. Where a desktop monitor may be made up of 1,920 columns of 1,080 picture elements or *pixels*, the BBC micro:bit has just five columns of five picture elements for a total of 25 pixels.

Although 25 pixels may not sound like much, it's more than enough to view useful information such as a bar graph or the elements of a simple game or to scroll messages of almost any length. You'll learn more about using the display for these tasks and more later in the book.

From a technical perspective, the BBC micro:bit's display is made up of a 5×5 *light emitting diode (LED) matrix*. Each LED in the matrix makes up one pixel and can display a single colour in varying brightnesses. By altering the brightness and rapidly changing the image being shown, the BBC micro:bit's display can show animation as well as still images.

## Buttons

Next to the display, the buttons are the BBC micro:bit's second most obvious feature. The two main buttons, Button A and Button B, are positioned either side of the display at the front of the board and are labelled with their letter to avoid any confusion (see Figure 1-4). These two buttons form the BBC micro:bit's primary *input devices*. Where the display allows information to be output from a program running on the BBC micro:bit, the buttons allow you to send simple inputs into the program to change the image being displayed, for example, or control a character in a game.

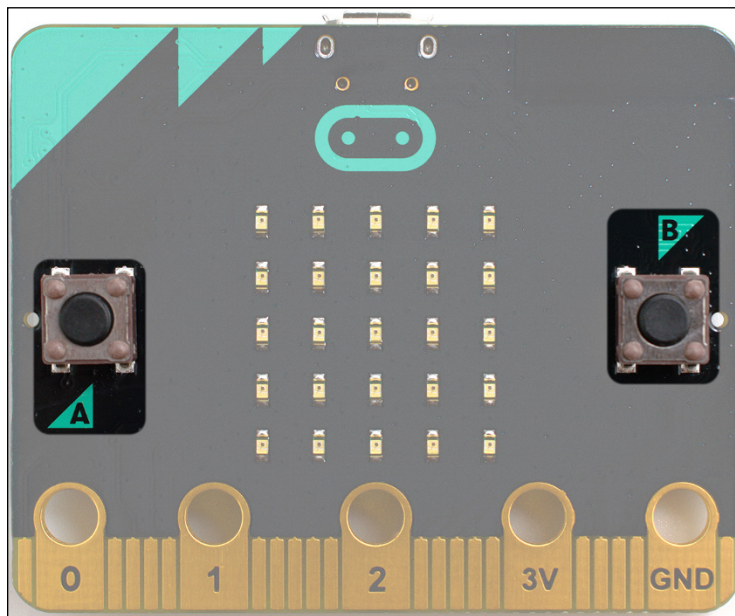


FIGURE 1-4: The BBC micro:bit's Button A and Button B

These buttons are technically known as a *momentary switch*. These are different to the switches you might use to turn on a light, which are known as *latching switches* or *toggle switches*. A momentary switch stays 'on' only as long as you continue to push it down, while a latching switch goes from 'off' to 'on' and stays there until you push it again.

Although the buttons are permanently connected to the BBC micro:bit, they won't do anything unless the program running on the BBC micro:bit is listening for a button input. Depending on what you're using the BBC micro:bit for, you may want to use only one or even neither of the two buttons; alternatively, you may find you need more buttons, in which case the *input-output pins* will help you expand the BBC micro:bit.

In addition to Button A and Button B, there's a third button on the back of the BBC micro:bit: the *Reset button*. Like the Reset button of a desktop computer, the BBC micro:bit's Reset button acts like cutting the power. Whatever the BBC micro:bit is doing at the time will be interrupted, and the BBC micro:bit will restart and begin its stored program again from the start. The Reset button is useful when experimenting with programs that may cause the BBC micro:bit to stop responding, but be careful not to press it accidentally when you're in the middle of something. There are other uses for the Reset button, too, which will be explained in later chapters.

## Processor

The *processor* is often called the 'brain' of a computer, and the BBC micro:bit's processor—technically known as a *microcontroller*—is no exception. Found on the upper-left of the rear of the BBC micro:bit and labelled on the board's *silkscreen layer*, the processor is a tiny black square no bigger than your little fingernail called an *integrated circuit* (see Figure 1-5).



FIGURE 1-5: The BBC micro:bit's processor

Despite its small size, this integrated circuit is surprisingly complex. It's here that any program the BBC micro:bit is using is both stored and run. While in a desktop computer, memory, storage, and the central processing unit are all separate, the BBC micro:bit's processor is all-in-one; this is known as a *system-on-chip* (SoC).

The BBC micro:bit's processor uses a special set of instructions, known as an *instruction set architecture*, called the *ARM Architecture*. Named for the company which invented it, ARM processors are designed to offer high performance at the lowest possible power usage. The BBC micro:bit uses this to great effect: It's possible to run the BBC micro:bit for months when using the recommended batteries.

For those interested in the technical side, the BBC micro:bit's processor is a Nordic nRF51822, which contains a single-core ARM Cortex-M0 running at 16Mhz, with 16 KB of random access memory (RAM) and 256 KB of non-volatile memory (NVM) for program storage.

The BBC micro:bit also has a second processor, not labelled on the silkscreen and found at the upper-right of the board. When you connect the BBC micro:bit to your computer using a micro-USB cable, as described in Chapter 3, 'Programming the BBC micro:bit', this second processor takes over and handles communication with your computer, accepting new programs and transferring them to the main processor to run.

## Radio

A major feature of the BBC micro:bit is its *radio*, which allows it to communicate with other BBC micro:bits or with other devices, such as a smartphone or tablet. The radio itself is a part of the BBC micro:bit's main processor, forming a segment of the system that makes up the system-on-chip design. As a result you won't find it labelled on the silkscreen as with other components, but instead covered under the 'PROCESSOR' label.

The BBC micro:bit's radio has two main functions. The first function is to communicate with other BBC micro:bits wirelessly, allowing you to group multiple BBC micro:bits without having to string cables between them. The second function is to communicate with other devices, which it does using *Bluetooth Low Energy (BLE)*, a low-power version of the popular Bluetooth wireless standard built into most modern smartphones and tablets.

Unlike the type of radio you might use to listen to music, there's no external antenna for the BBC micro:bit's radio chip. Instead, it uses a cleverly-shaped copper track built into the circuit board itself. You can find this at the upper-left on the rear of the BBC micro:bit, labelled 'BLE ANTENNA' (see Figure 1-6). It doesn't look like much, but if you hold it to the light, you should be able to see a raised line tracing a series of rectangular shapes—providing, that is, the antenna isn't covered by regulatory compliance stickers required of all radio transmitters in selected countries.



FIGURE 1-6: The BBC micro:bit’s radio antenna

## Accelerometer

One of two built-in sensors, the BBC micro:bit’s *accelerometer* is an integrated circuit even smaller than the processor chips (see Figure 1-7). It’s so small that if it weren’t labelled on the board’s rear silkscreen, it’d be easy to overlook it. Despite its small size, this chip is extremely clever: it knows exactly how the BBC micro:bit is positioned in space at any given time.

When you rotate your smartphone from portrait to landscape and vice versa, it’s an accelerometer that tells the device what you’ve done and allows it to automatically rotate the screen. The BBC micro:bit’s accelerometer works in the same way: it can track the angle of the device in all three *axes*—sideways, backward and forward, and up and down, or X, Y, and Z—by tracking what is known as *proper acceleration*.

The accelerometer can be seen in action during the BBC micro:bit’s demonstration program, which is explored in Chapter 2.