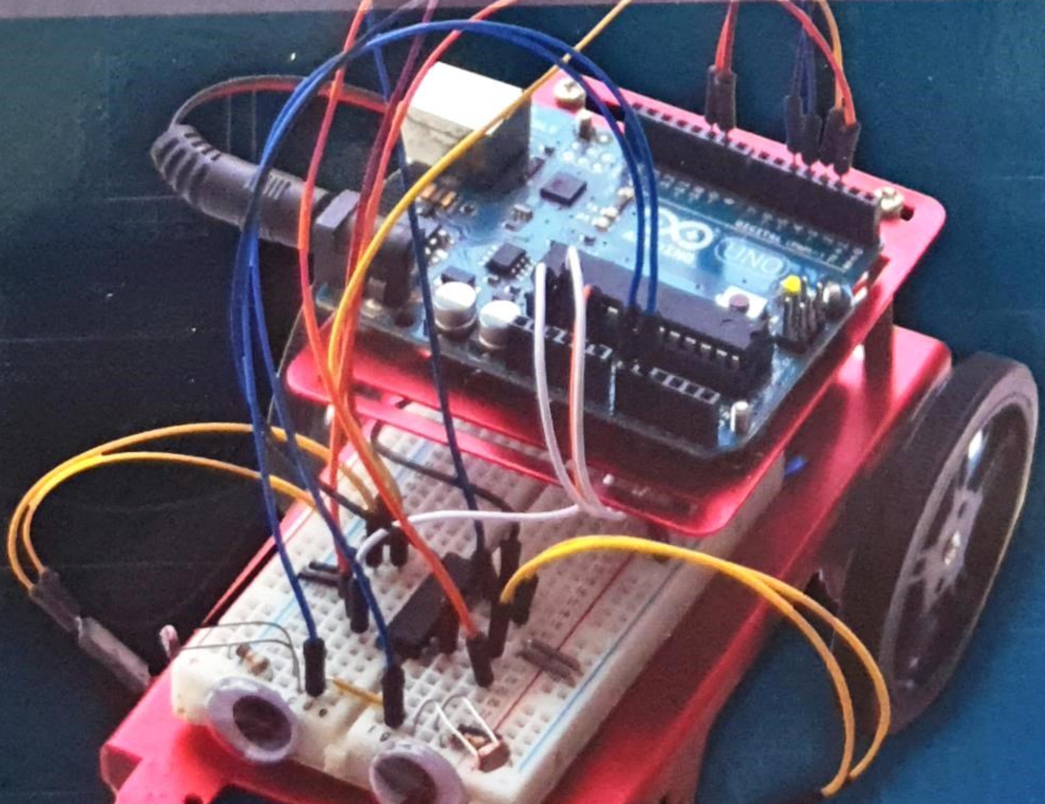
SECOND EDITION

New color photographs and diagrams

JEREMY BLUM

EXPLORING ARDUINO®

TOOLS AND TECHNIQUES FOR ENGINEERING WIZARDRY



WILEY

EXPLORING ARDUINO®

Tools and Techniques for Engineering Wizardry

Second Edition

Jeremy Blum

WILEY

Contents at a Glance

Introduction xxv

PART I Arduino Engineering Basics 1

1 Getting Started and Understanding the Arduino Landscape 3

2 Digital Inputs, Outputs, and Pulse-Width Modulation 23

3 Interfacing with Analog Sensors 47

PART II Interfacing with Your Environment 67

4 Using Transistors and Driving DC Motors 69

5 Driving Stepper and Servo Motors 99

6 Making Sounds and Music 125

7 USB Serial Communication 141

8 Emulating USB Devices 171

9 Shift Registers. 183

PART III Communication Interfaces 199

10 The I²C Bus 201

11 The SPI Bus and Third-Party Libraries 223

12 Interfacing with Liquid Crystal Displays 247

PART IV Digging Deeper and Combining Functions 273

13 Interrupts and Other Special Functions 275

14 Data Logging with SD Cards. 295

3 Interfacing with Analog Sensors 47

Understanding Analog and Digital Signals 48

Comparing Analog and Digital Signals 48

Converting an Analog Signal to Digital 49

Reading Analog Sensors with the Arduino: *analogRead()* 51

Reading a Potentiometer 51

Using Analog Sensors 56

Using Variable Resistors to Make Your Own Analog Sensors 60

Using Resistive Voltage Dividers 61

Using Analog Inputs to Control Analog Outputs 64

Summary 66

PART II Interfacing with Your Environment 67

4 Using Transistors and Driving DC Motors 69

Driving DC Motors 70

Handling High-Current Inductive Loads 71

Using Transistors as Switches 72

Using Protection Diodes 73

Using a Secondary Power Source 74

Wiring the Motor 74

Controlling Motor Speed with PWM 76

Using an H-Bridge to Control DC Motor Direction 78

Building an H-Bridge Circuit 80

Operating an H-Bridge Circuit 82

Building a Roving Robot 86

Choosing the Robot Parts 87

Selecting a Motor and Gearbox 87

Powering Your Robot 87

Constructing the Robot 89

Writing the Robot Software 92

Bringing It Together 96

Summary 97

5 Driving Stepper and Servo Motors 99
Driving Servo Motors 100
Understanding the Difference between Continuous Rotation and Standard Servos 100
Understanding Servo Control 101
Controlling a Servo 104
Building a Sweeping Distance Sensor 105
Understanding and Driving Stepper Motors 109
How Bipolar Stepper Motors Work 111
Making Your Stepper Move 113
Building a "One-Minute Chronograph" 117
Wiring and Building the Chronograph 117
Programming the Chronograph 119
Summary 124

6 Making Sounds and Music 125
Understanding How Speakers Work 126
The Properties of Sound 126
How a Speaker Produces Sound 128
Using *tone()* to Make Sounds 129
Including a Definition File 129
Wiring the Speaker 130
Making Sound Sequences 133
Using Arrays 133
Making Note and Duration Arrays 134
Completing the Program 134
Understanding the Limitations of the *tone()* Function 136
Building a Micro Piano 136
Summary 139

7 USB Serial Communication 141
Understanding the Arduino's Serial Communication Capabilities 142
Arduino Boards with an Internal or External FTDI or Silicon Labs USB-to-Serial Converter 143

Arduino Boards with a Secondary USB-Capable ATmega MCU Emulating a Serial Converter . . . 146

Arduino Boards with a Single USB-Capable MCU . . . 147

Arduino Boards with USB-Host Capabilities . . . 147

Listening to the Arduino . . . 148

Using *print* Statements . . . 148

Using Special Characters . . . 150

Changing Data Type Representations . . . 152

Talking to the Arduino . . . 152

Configuring the Arduino IDE's Serial Monitor to Send Command Strings . . . 152

Reading Incoming Data from a Computer or Other Serial Device . . . 153

Telling the Arduino to Echo Incoming Data . . . 153

Understanding the Differences between Chars and Ints . . . 154

Sending Single Characters to Control an LED . . . 156

Sending Lists of Values to Control an RGB LED . . . 158

Talking to a Desktop App . . . 161

Installing Processing . . . 162

Controlling a Processing Sketch from Your Arduino . . . 163

Sending Data from Processing to Your Arduino . . . 166

Summary . . . 169

8 Emulating USB Devices . . . 171

Emulating a Keyboard . . . 173

Typing Data into the Computer . . . 173

Commanding Your Computer to Do Your Bidding . . . 177

Emulating a Mouse . . . 178

Summary . . . 182

9 Shift Registers . . . 183

Understanding Shift Registers . . . 184

Sending Parallel and Serial Data . . . 185

Working with the 74HC595 Shift Register . . . 186

Understanding the Shift Register pin Functions . . . 186

Understanding How the Shift Register Works . . . 187

Shifting Serial Data from the Arduino 189
Converting Between Binary and Decimal Formats 192
Controlling Light Animations with a Shift Register 192
Building a "Light Rider" 192
Responding to Inputs with an LED Bar Graph 194
Summary 197

PART III Communication Interfaces 199

10 The I²C Bus 201
History of the I²C Bus 202
I²C Hardware Design 203
Communication Scheme and ID Numbers 203
Hardware Requirements and Pull-Up Resistors 206
Communicating with an I²C Temperature Probe 208
Setting Up the Hardware 208
Referencing the Datasheet 210
Writing the Software 212
Combining Shift Registers, Serial Communication, and I²C Communications 214
Building the Hardware for a Temperature Monitoring System 214
Modifying the Embedded Program 215
Writing the Processing Sketch 218
Summary 221

11 The SPI Bus and Third-Party Libraries 223
Overview of the SPI Bus 224
SPI Hardware and Communication Design 225
Hardware Configuration 225
Communication Scheme 227
Comparing SPI to I²C and UART 227
Communicating with an SPI Accelerometer 228
What Is an Accelerometer? 229
Gathering Information from the Datasheet 231
Setting Up the Hardware 233

Writing the Software 235
Installing the Adafruit Sensor Libraries 236
Leveraging the Library 237
Creating an Audiovisual Instrument Using a 3-Axis Accelerometer 241
Setting Up the Hardware 242
Modifying the Software 242
Summary 246

12 Interfacing with Liquid Crystal Displays 247
Setting Up the LCD 248
Using the LiquidCrystal Library to Write to the LCD 251
Adding Text to the Display 252
Creating Special Characters and Animations 254
Building a Personal Thermostat 258
Setting Up the Hardware 258
Displaying Data on the LCD 261
Adjusting the Set Point with a Button 264
Adding an Audible Warning and a Fan 265
Bringing It All Together: The Complete Program 266
Taking This Project to the Next Level 270
Summary 271

PART IV Digging Deeper and Combining Functions 273
13 Interrupts and Other Special Functions 275
Using Hardware Interrupts 276
Knowing the Tradeoffs Between Polling and Interrupting 277
Ease of Implementation (Software) 277
Ease of Implementation (Hardware) 277
Multitasking 278
Acquisition Accuracy 278
Understanding the Arduino Hardware Interrupt Capabilities 278

Building and Testing a Hardware-Debounced Button Interrupt Circuit 279

Creating a Hardware-Debouncing Circuit 280

Assembling the Complete Test Circuit 284

Writing the Software 285

Using Timer Interrupts 288

Understanding Timer Interrupts 288

Getting the Library 289

Executing Two Tasks Simultaneously(ish) 289

Building an Interrupt-Driven Sound Machine 290

Sound Machine Hardware 291

Sound Machine Software 291

Summary 294

14 Data Logging with SD Cards 295

Getting Ready for Data Logging 296

Formatting Data with CSV Files 297

Preparing an SD Card for Data Logging 297

Formatting Your SD Card Using a Windows PC 298

Formatting Your SD Card Using Mac OS 300

Formatting Your SD Card Using Linux 302

Interfacing the Arduino with an SD Card 304

SD Card Shields 304

SD Card SPI Interface 307

Writing to an SD Card 307

Reading from an SD Card 312

Real-Time Clocks 317

Understanding Real-Time Clocks 317

Communicating with a Real-Time Clock 317

Using the RTC Arduino Third-Party Library 318

Using a Real-Time Clock 319

Installing the RTC and SD Card Modules 319

Updating the Software 320

Building an Entrance Logger .327
Logger Hardware .328
Logger Software .329
Data Analysis .334
Summary .335

PART V Going Wireless .337

15 Wireless RF Communications 339
The Electromagnetic Spectrum .340
The Spectrum .342
How Your RF Link Will Send and Receive Data .343
Receiving Key Presses with the RF Link .346
Connecting Your Receiver .346
Programming Your Receiver .347
Making a Wireless Doorbell .351
Wiring the Receiver .351
Programming the Receiver .351
The Start of Your Smart Home—Controlling a Lamp .354
Your Home's AC Power .356
How a Relay Works .356
Programming the Relay Control .358
Hooking up Your Lamp and Relay to the Arduino .360
Summary .361

16 Bluetooth Connectivity 363
Demystifying Bluetooth .364
Bluetooth Standards and Versions .364
Bluetooth Profiles and BTLE GATT Services .365
Communication between Your Arduino and Your Phone .366
Reading a Sensor over BTLE .366
Adding Support for Third-Party Boards to the Arduino IDE .367
Installing the BTLE Module Library .369
Programming the Feather Board .369

Connecting Your Smartphone to Your BTLE Transmitter 377
Sending Commands from Your Phone over BTLE 379
Parsing Command Strings 380
Commanding Your BTLE Device with Natural Language.................384
Controlling an AC Lamp with Bluetooth 389
How Your Phone "Pairs" to BTLE Devices 389
Writing the Proximity Control Software 390
Pairing Your Phone 394
Pairing an Android Phone..394
Pairing an iPhone 395
Make Your Lamp React to Your Presence 396
Summary 397

17 Wi-Fi and the Cloud 399
The Web, the Arduino, and You 400
Networking Lingo 401
The Internet vs. the World Wide Web vs. the Cloud....................401
IP Address 401
Network Address Translation..402
MAC Address..402
HTML..402
HTTP and HTTPS...402
GET/POST..403
DHCP...403
DNS 403
Clients and Servers 403
Your Wi-Fi–Enabled Arduino 404
Controlling Your Arduino from the Web.................................404
Setting Up the I/O Control Hardware 404
Preparing the Arduino IDE for Use with the Feather Board..............406
Ensuring the Wi-Fi Library Is Matched to the Wi-Fi Module's Firmware 407
Checking the WINC1500's Firmware Version.............................408
Updating the WINC1500's Firmware.......................................408

Writing an Arduino Server Sketch. 408
Connecting to the Network and Retrieving an IP Address via DHCP 409
Writing the Code for a Bare-Minimum Web Server 412
Controlling Your Arduino from Inside and Outside Your Local Network. 423
Controlling Your Arduino over the Local Network 423
Using Port Forwarding to Control Your Arduino from Anywhere 425
Interfacing with Web APIs. 427
Using a Weather API 428
Creating an Account with the API Service Provider 429
Understanding How APIs Are Structured 430
JSON-Formatted Data and Your Arduino 430
Fetching and Parsing Weather Data 431
Getting the Local Temperature from the Web on Your Arduino 433
Completing the Live Temperature Display 440
Wiring up the LED Readout Display. 440
Driving the Display with Temperature Data 443
Summary 449

Appendix A: Deciphering Datasheets and Schematics. 451

Index. 461

Figure Credits

All images, icons, and marks as displayed in Figure 3-7 and Figure 10-3 are owned by Analog Devices, Inc. (ADI), copyright © 2019. All Rights Reserved. These images, icons, and marks are reproduced with permission by ADI. No unauthorized reproduction, distribution, or usage is permitted without ADI's written consent.

This book contains copyrighted material of Microchip Technology Incorporated replicated with permission. All rights reserved. No further replications may be made without Microchip Technology Inc.'s prior written consent.

Atmel, AVR, ICSP, and In-Circuit Serial Programming are trademarks or registered trademarks of Microchip Technology Inc.

Arm and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the United States and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs, and trade secrets.

Introduction

When the first edition of this book came out in 2013, I opened it with the following greeting:

> *You have excellent timing. As I often like to say, "We're living in the future."*

I think I backed myself into a corner with that introduction, because if 2013 was "the future," then I'm not quite sure what to call the present! The *far future*? The *future-future*? My point is, the march of progress has been swift, and the possibilities for what you can do with even a cursory knowledge of embedded electronics and software continue to expand every day.

Since the first edition of this book was released, electronics and software have continued to become increasingly accessible with every passing day. In 2013, I was hesitant to include a chapter about connecting your hardware projects to the internet because the process for doing so was still quite fussy. The "Internet of Things" (IoT) was just an emerging nerdy buzzword in 2013. Now, it's a key part of the global vernacular. It seems like every product for sale nowadays contains a microcontroller. Everything is "smart" and most of those things also feature phone or web connectivity. I bet you didn't think you'd be buying a Bluetooth-enabled toothbrush back when "Bluetooth" just referred to people talking to themselves through their wireless cellphone headsets.

Considering all this, I felt it was time to release a new edition of *Exploring Arduino*. This second edition expands upon everything that was covered in the first edition. It updates all the projects with new challenges and details, clarifies questions that people had from the first edition, and adds a plethora of new content, including a lot more details on wireless connectivity, new Arduino hardware, changes to the Arduino ecosystem and software, and more.

Why Arduino?

With the tools available to you today, many of which you'll learn about in this book, you have the opportunity and the ability to bend the physical world to your whim. Until very recently, it has not been possible for someone to pick up a microcontroller and use it to control their world within minutes. A *microcontroller* is a programmable integrated circuit (IC) that gives you the power to define the operation of complex mechanical, electrical, and software systems using relatively simple commands. The possibilities are endless, and the Arduino microcontroller platform will become your new favorite tool as you explore the world of electronics, programming, human-computer interaction, art, control systems, and more. Throughout the course of this book, you'll use the Arduino to do everything from detecting motion to creating wireless control systems to communicating over the internet.

Whether you are completely new to any kind of engineering or are a seasoned veteran looking to get started with embedded systems design, the Arduino is a great place to start. Are you looking for a general reference for Arduino development? This book is perfect for you, too. It walks you through a number of separate projects, but you'll also find it easy to return to the book for code snippets, best practices, system schematics, and more. The electrical engineering, systems design, and programming practices that you'll learn while reading this book are widely applicable beyond the Arduino platform and will prepare you to take on an array of engineering projects, whether they use the Arduino or some other platform.

Who This Book Is For

This book is for Arduino enthusiasts of all experience levels. Chapters build upon each other, utilizing concepts and project components from previous chapters to develop more complex ideas. But don't worry. Whenever you face new, complex ideas, a cross-reference reminds you where you first encountered any relevant building-block concepts so that you can easily refresh your memory.

This book assumes that you have little or no previous experience working with programming or electrical engineering. Using feedback from readers of the first edition of this book, I've taken special care to be very detailed in my explanation of the more confusing topics you may encounter. To effectively support readers of various experience levels, the book features several optional sections and *sidebars*, or short excerpts, that explain a particular concept in greater detail. Although these sidebars are not necessary for you to gain a good understanding of how to use the Arduino, they do provide a closer look at technical topics for the more curious reader.

What You'll Learn in This Book

This book is not a recipe book. If you want to follow step-by-step instructions that tell you exactly how to build a particular project without actually explaining why you are doing what you are doing, this book is not for you. You can think of this book as an introduction to electrical engineering, computer science, product design, and high-level thinking using the Arduino as a vehicle to help you experience these concepts in a hands-on manner.

When building hardware components of the Arduino projects demonstrated in this book, you'll learn not just how to wire things together, but also how to read schematics, why particular parts are used for particular functions, and how to read datasheets that will allow you to choose appropriate parts to build your own projects. When writing software, I provide complete program code, but you will first be stepped through several iterative processes to create the final program. This will help to reinforce specific program functions, good code-formatting practices, and algorithmic understanding.

This book will teach physics concepts, algorithms, digital design principles, and Arduino-specific programming concepts. It is my hope that working through the projects in this book will not just make you a well-versed Arduino developer, but also give you the skills you need to develop more-complex electrical systems, and to pursue engineering endeavors in other fields, and with different platforms.

Features Used in This Book

The following features and icons are used in this book to help draw your attention to some of the most important or useful information in the book:

WARNING Be sure to take heed when you see one of these asides. They appear when particular steps could cause damage to your electronics if performed incorrectly.

TIP These asides contain quick hints about how to perform the task at hand more easily and effectively.

NOTE These asides contain additional information that may be of importance to you, including links to videos and online material that will make it easier to follow along with the development of a particular project.

SAMPLE HEADING

These asides go into additional depth about the current topic or a related topic.

Getting the Parts

In preparing the projects outlined in this book, I've taken special care to use components that are readily available through a variety of retailers, both in the United States and internationally. I've also partnered with Adafruit (`adafruit.com`), a popular retailer of hobbyist electrical components. You can purchase all the components required for completing the projects in this book from Adafruit. A convenient listing of Adafruit parts for each chapter is available at `exploringarduino.com/kits`.

At the beginning of each chapter, you'll find a detailed list of parts that you need to complete that chapter—all of these parts are available from many sources. The companion website for this book, `www.wiley.com/go/exploringarduino2e`, also provides links to multiple sources where you can find the parts for each chapter.

What You'll Need

In addition to the actual parts that you'll use to build your Arduino projects, there are a few other tools and materials that you'll need on your Arduino adventures. Most importantly, you'll need a computer that is compatible with the Arduino integrated development environment (IDE) (Mac OS X 10.7 Lion or newer, Windows XP or later, or a Linux distro). I will provide instructions for all operating systems when warranted.

Arduino now also has an entirely web-based editor, but this book will generally focus on the desktop IDE. All the instructions for the desktop software generally apply to the online IDE as well. The first version of this book was read by people all over the world, representing a wide range of internet speeds and reliability. To ensure that Arduino remains easily accessible to all, I'll mostly provide instructions that use the offline IDE, as constant internet access isn't always an option for everybody.

You may also want some additional tools that will be used throughout the book to debug and assemble hardware. These tools are not only necessary to complete the projects in this book. As you develop your electrical engineering skillset, they will come in handy for other projects, too. I recommend the following:

- A soldering iron and solder (Note: A few shields and microcontroller boards used in the final chapters of this book may be sold with some soldering required—this usually involves easy soldering of thru-hole pins to a circuit board.)
- A multimeter (This will be useful for debugging concepts within this book, but is not required.)
- A set of small screwdrivers
- Tweezers
- Wire cutters and wire strippers
- A hot glue gun
- A magnifying glass (Electronics are small, and sometimes it's necessary to read the tiny, laser-etched markings on integrated circuits in order to look up their datasheets online.)