

Extracted from:

# Arduino: A Quick-Start Guide, Second Edition

This PDF file contains pages extracted from *Arduino: A Quick-Start Guide, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

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, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

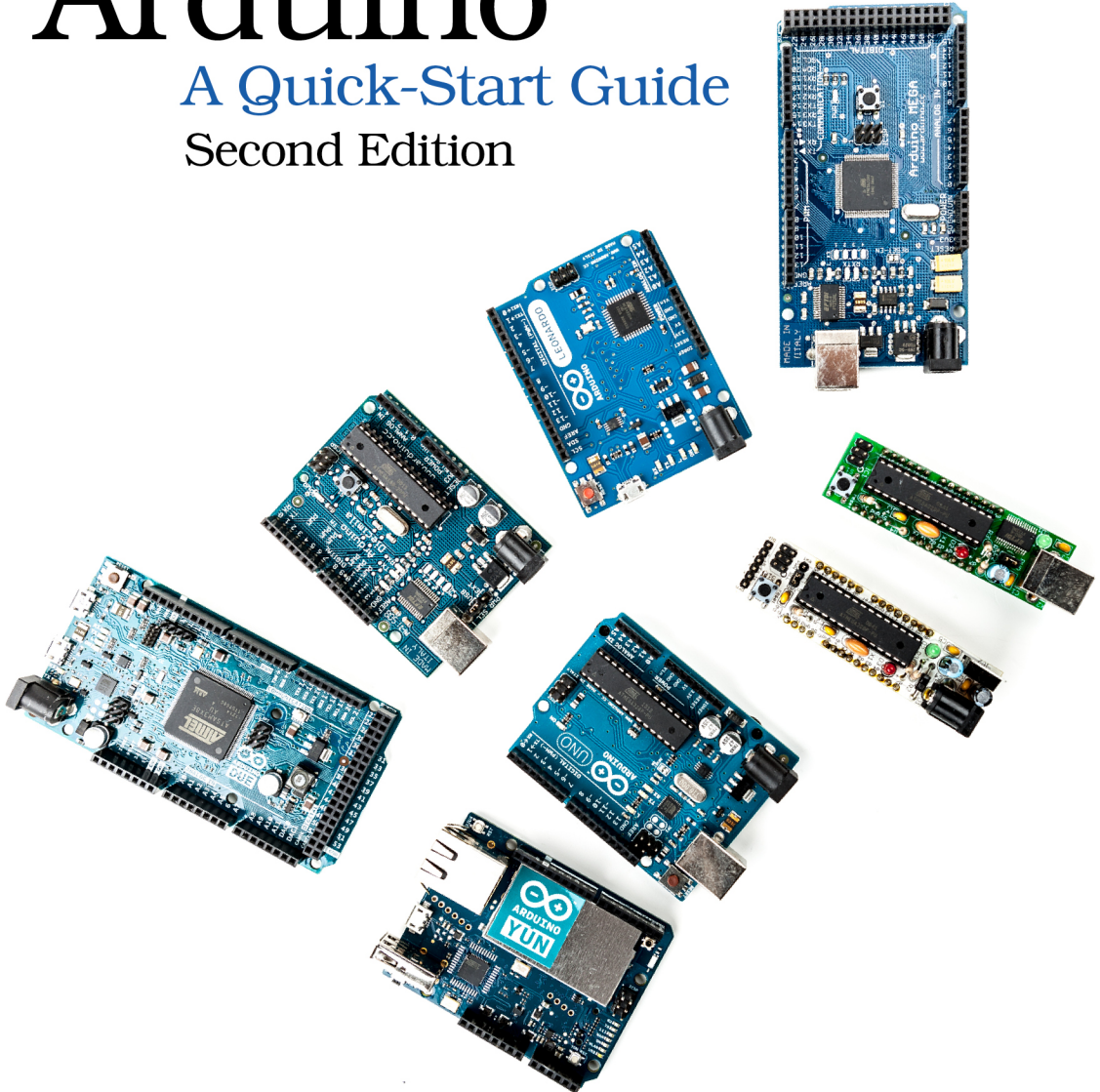
Dallas, Texas • Raleigh, North Carolina

The  
Pragmatic  
Programmers

# Arduino

## A Quick-Start Guide

Second Edition



Maik Schmidt

*Edited by Susannah Davidson Pfalzer*

# Arduino: A Quick-Start Guide, Second Edition

Maik Schmidt

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

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

All circuit diagrams were created with Fritzing (<http://fritzing.org>).

The team that produced this book includes:

Susannah Davidson Pfalzer (editor)

Potomac Indexing, LLC (indexer)

Cathleen Small (copyeditor)

Dave Thomas (typesetter)

Janet Furlow (producer)

Ellie Callahan (support)

For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

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, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-94122-224-9

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—January 2015

# Preface

---

Welcome to Arduino, and welcome to the exciting world of physical computing! Arduino<sup>1</sup> is an open-source project consisting of both hardware and software. It was originally created to give designers and artists a prototyping platform for interaction design courses. Today, hobbyists and experts all over the world use it to create physical computing projects, and you can, too.

Arduino lets you get hands-on again with computers in a way you haven't been able to since the 1980s, when you could build your own computer. And Arduino makes it easier than ever to develop handcrafted electronics projects ranging from prototypes to sophisticated gadgets. Gone are the days when you had to learn lots of theory about electronics and arcane programming languages before you could even get an LED blinking. You can create your first Arduino project in a few minutes without needing advanced electrical engineering coursework.

In fact, you don't need to know anything about electronics projects to read this book, and you'll get your hands dirty right from the beginning. You'll not only learn how to use some of the most important electronic parts in the first pages, you'll also learn how to write the software needed to bring your projects to life.

This book dispenses with theory and stays hands-on throughout. I'll explain all the basics you need to build the book's projects, and every chapter has a troubleshooting section to help when things go wrong. This book is a quick-start guide that gets you up to speed quickly and enables you to immediately create your own projects.

## Who Should Read This Book

If you are interested in electronics—and especially in building your own toys, games, and gadgets—then this book is for you. Although Arduino is a nice

---

1. <http://arduino.cc>

tool for designers and artists, only software developers are able to unleash its full power. So, if you’ve already developed some software—preferably with C/C++ or Java—then you’ll get a lot out of this book.

But there’s one more thing: you have to build, try, and modify the projects in this book. Have fun. Don’t worry about making mistakes. The troubleshooting sections—and the hands-on experience you’ll gain as you become more confident project by project—will make it all worthwhile. Reading about electronics without doing the projects yourself isn’t even half the battle. (You know the old saying: we remember 5 percent of what we hear, 10 percent of what we write, and 95 percent of what we personally suffer.) And don’t be afraid: you really don’t need any previous electronics project experience!

If you’ve never written a piece of software before, start with a programming course or read a beginner’s book about programming first. ([Learn to Program, Second Edition \[Pin09\]](#) is a good starting point.) Then, learn to program in C with [The C Programming Language \[KR98\]](#) or in C++ with [The C++ Programming Language \[Str00\]](#).

## What’s in This Book

This book consists of three parts (“Getting Started with Arduino,” “Eleven Arduino Projects,” and the appendixes). In the first part, you’ll learn all the basics you need to build the projects in the second part, so read the chapters in order and do all the exercises. The chapters in the second part also build on each other, reusing techniques and code from earlier chapters.

There’s one exception, though: in this book you’ll create several Google Chrome apps that connect your web browser to Arduino. [Appendix 4, Controlling the Arduino with a Browser, on page ?](#), explains in detail how Chrome apps work, so you should read it after you’ve read [Chapter 4, Building a Morse Code Generator Library, on page ?](#).

Here’s a short walkthrough:

- The book starts with the basics of Arduino development. You’ll learn how to use the integrated development environment (IDE) and how to compile and upload programs. You’ll quickly build your first project—electronic dice—that shows you how to work with basic parts such as LEDs, buttons, and resistors. By implementing a Morse code generator, you’ll see how easy it is to create your own Arduino libraries.
- Then you’ll learn how to work with analog and digital sensors. You’ll use a temperature sensor and an ultrasonic sensor to build a very accurate

digital metering ruler. Then you'll use a three-axis accelerometer to build your own motion-sensing game controller and a cool breakout game clone.

- At this point you've output data mostly using some LEDs and the Arduino's serial port. Now you'll connect the Arduino to an actual TV set and generate your own video signals. You'll create a graphical thermometer that you can display on the TV set in your living room.
- In electronics, you don't necessarily have to build gadgets yourself. You can also tinker with existing hardware, and you'll see how easy it is to take full control of Nintendo's Wii Nunchuk so you can use it in your own applications. Soon, you'll have everything you need to build your own video game console.
- The Arduino does not have to work in isolation, and it works great with different networking technologies. You'll connect the Arduino to the Internet in various ways, and you'll learn how to send Twitter messages and emails. You'll build a burglar alarm that sends you an email whenever someone is moving in your living room during your absence.
- Using a Nunchuk to control applications or devices is handy, but often it's more convenient to have a wireless remote control. So, you'll learn how to build your own universal remote control that you can even control using a web browser.
- Finally, you'll work with motors by creating a fun device for your next software project. You can connect it to your continuous integration system, so whenever the build fails, it will move an arrow to point to the name of the developer who is responsible.
- In the appendixes, you'll learn about the basics of electricity and soldering. You'll also find advanced information about programming a serial port and programming the Arduino in general. And you'll find an appendix that explains how to control the Arduino using a web browser.

Every chapter starts with a detailed list of all the parts and tools you need to build the chapter's projects. All chapters contain lots of photos and diagrams showing how everything fits together. You'll get inspired by descriptions of real-world Arduino projects in sidebars throughout the book.

Things won't always work out as expected, and debugging circuits can be a challenging task. So in every chapter, you'll find a "What If It Doesn't Work?" section that explains the most common problems and their solutions.

Before you read the solutions in the “What If It Doesn’t Work?” sections, though, try to solve the problems yourself, because that’s the most effective way to learn. In the unlikely case that you *don’t* run into any problems, you’ll find a list of exercises at the end of every chapter to build your skills.

## Arduino Uno and the Arduino Platform

After releasing several Arduino boards and Arduino IDE versions, the Arduino team decided to specify a version 1.0 of the platform. Arduino’s version numbering was counterintuitive before. At the beginning of the project the developers increased the version number by 1 with each new release. They did that up to number 23, and then they decided to use 1.0 as the version number for the next release. That means Arduino 1.0 is more recent than Arduino 23.

Arduino 1.0, released at the end of 2011, has since been the reference point for all developments. The Arduino developers have also released the Arduino Uno board, and they’ve continued to improve the IDE and its supporting libraries.

In parallel, the Arduino team has created more Arduino boards, such as the Arduino Due,<sup>2</sup> the Arduino Leonardo,<sup>3</sup> and the Arduino Yún.<sup>4</sup> These boards either have more powerful microcontrollers or come with additional hardware, such as a Wi-Fi module.

Most of the new boards use a different processor architecture designed by ARM. This architecture isn’t compatible with the architecture of the older board’s AVR processors. To overcome this gap, the Arduino team developed version 1.5.x of the Arduino IDE in parallel to 1.0.x. Version 1.5 will eventually support the different processor architectures transparently. Currently, you need version 1.5 only if you want to program the Arduino Due or the Arduino Yún. For all other boards, use version 1.0.x.

In addition to all that, there’s a separate IDE for the Arduino Galileo.<sup>5</sup> This board was created by Intel and is compatible with the Arduino.

This book is current for version 1.0.6 of the Arduino platform and up to date for the Arduino Uno board. Most of the projects will also work on other recent boards, such as the Leonardo or the Due. They will also work on older Arduino

2. <http://arduino.cc/en/Main/ArduinoBoardDue>
3. <http://arduino.cc/en/Main/ArduinoBoardLeonardo>
4. <http://arduino.cc/en/Main/ArduinoBoardYun>
5. <http://arduino.cc/en/ArduinoCertified/IntelGalileo>



boards, such as the Duemilanove or Diecimila. All code in this book has been tested with Arduino 1.0.6.

## Code Examples and Conventions

Although this is a book about open-source hardware and electronics, you will find a lot of code examples. We need them to bring the hardware to life and make it do what we want it to do.

We'll use C/C++ for all programs that will eventually run on the Arduino. For applications running on our PC, we'll mainly use JavaScript and Processing.<sup>6</sup> In *Serial Communication Using Various Languages, on page ?*, you'll also learn how to use several other programming languages to communicate with an Arduino.

## Online Resources

This book has its own web page at <http://pragprog.com/book/msard2> where you can download the code for all examples. (If you have the ebook version of this book, clicking the little gray box above each code example downloads that source file directly.) You can also participate in a discussion forum and meet other readers and me. If you find bugs, typos, or other annoyances, please let me and the world know about them on the book's errata page.<sup>7</sup>

On the web page you will also find a link to a Flickr<sup>8</sup> photo set. It contains all the book's photos in high resolution. There you can also see photos of reader projects, and we'd really like to see photos of your projects, too!

Let's get started!

---

6. <http://processing.org>

7. <http://www.pragprog.com/book/msard2/errata>

8. <http://flickr.com>