

Extracted from:

Arduino

A Quick-Start Guide

This PDF file contains pages extracted from *Arduino*, 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 printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

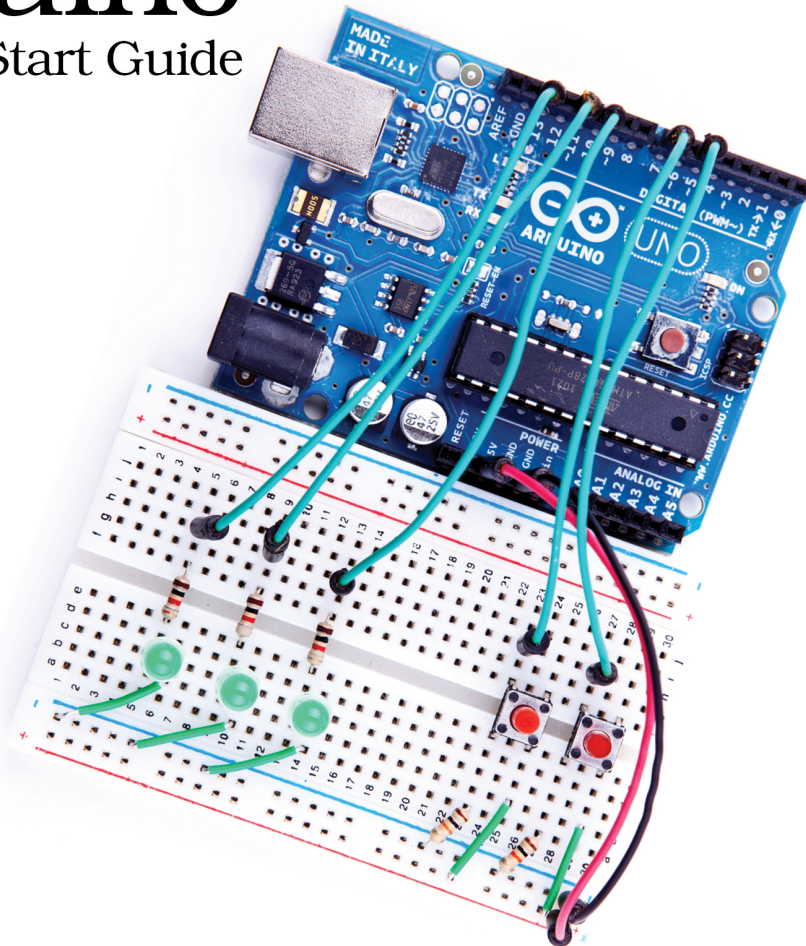
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

Arduino

A Quick-Start Guide



Maik Schmidt

Edited by Susannah Davidson Pflazer

Preface

Welcome to the Arduino, and welcome to the exciting world of physical computing! Arduino¹ 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.

The Arduino lets us get hands-on again with computers in a way we 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 course work.

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 the Arduino is a

1. <http://arduino.cc>

nice 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* [Pin06] is a nice 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,” “Eight 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.

Here’s a short walk-through:

- The book starts with the basics of Arduino development. You’ll learn how to use the 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, together with a cool breakout game clone.
- 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.

- Using a Nunchuk to control applications or devices is nice, 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.
- Speaking of web browsers: connecting the Arduino to the Internet is easy, so you'll build a burglar alarm that sends you an email whenever someone is moving in your living room during your absence.
- Finally, you'll work with motors by creating a fun device for your next software project. It connects to your continuous integration system, and whenever the build fails, it moves 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.

Every chapter starts with a detailed list of all parts and tools you need to build the chapter's projects. Every chapter contains lots of photos and diagrams showing how everything fits together. You'll get inspired with 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 difficult and 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 of learning. In the unlikely case that you *don't* run into any problems, you'll find a list of exercises to build your skills at the end of every chapter.

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 is a bit counterintuitive. 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, will be the reference point for all future developments. The Arduino developers have also released the Arduino Uno board, and they've continued to improve the IDE and its supporting libraries.

The Arduino team has tried to keep Arduino 1.0 as backward compatible as possible.² However, they still needed to add some changes that break old code.

This book is current for version 1.0 of the Arduino platform and it is up-to-date for the new Arduino Uno boards. All the projects will also work with older Arduino boards such as the Duemilanove or Diecimila. All code in this book has been tested with Arduino 1.0 and most of the code also runs on Arduino IDE versions 19 to 23. Only projects using the Ethernet or Wire libraries need some changes to run on older Arduino versions. The code on the book's web site contains versions not only for Arduino 1.0 but also for older versions. In addition, I'll point out the most important differences in the text.

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 use C/C++ for all programs that will eventually run on the Arduino. For applications running on our PC, we use Processing,³ but in [Section A3.2, *Serial Communication Using Various Programming Languages*, on page ?](#), you'll also learn how to use several other programming languages to communicate with an Arduino.

Whenever you find a slippery road icon beside a paragraph, slow down and read carefully. They announce difficult or dangerous techniques.

Online Resources

This book has its own web page at <http://pragprog.com/titles/msard> 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

2. At <http://code.google.com/p/arduino/wiki/Arduino1> you can find a list of all changes

3. <http://processing.org>

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.⁴

On the web page you will also find a link to a Flickr⁵ 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!

4. <http://www.pragprog.com/titles/msard/errata>

5. <http://flickr.com>