

# ARDUINO SKETCHES

Tools and Techniques for Programming Wizardry

James A. Langbridge



WILEY

---

# Table of Contents

[Title Page](#)

[Copyright](#)

[Dedication](#)

[About the Author](#)

[About the Technical Editor](#)

[Credits](#)

[Acknowledgments](#)

[Introduction](#)

[Overview of the Book and Technology](#)

[How This Book Is Organized](#)

[Who Should Read This Book](#)

[Tools You Need](#)

[What's on the Website](#)

[Summary](#)

[Part I: Introduction to Arduino](#)

[Chapter 1: Introduction to Arduino](#)

[Atmel AVR](#)

[The Arduino Project](#)

[The ATmega Series](#)

[The Different Arduinos](#)

[Shields](#)

[What Can You Do with an Arduino?](#)

[What You Will Need for This Book](#)

[Summary](#)

[Chapter 2: Programming for the Arduino](#)

[Installing Your Environment](#)

[Your First Program](#)

[Understanding Your First Sketch](#)

[Programming Basics](#)

[Summary](#)

[Chapter 3: Electronics Basics](#)

[Electronics 101](#)

[Voltage, Amperage, and Resistance](#)

[The Basic Components](#)

[Breadboards](#)

## [Inputs and Outputs](#)

---

[Connecting a Light-Emitting Diode](#)

[Summary](#)

## [Part II: Standard Libraries](#)

### [Chapter 4: The Arduino Language](#)

[I/O Functions](#)

[Time Functions](#)

[Mathematical Functions](#)

[Trigonometry](#)

[Interrupts](#)

[Summary](#)

### [Chapter 5: Serial Communication](#)

[Introducing Serial Communication](#)

[UART Communications](#)

[Debugging and Output](#)

[Starting a Serial Connection](#)

[Writing Data](#)

[Reading Data](#)

[Example Program](#)

[SoftwareSerial](#)

[Summary](#)

### [Chapter 6: EEPROM](#)

[Introducing EEPROM](#)

[The Different Memories on Arduino](#)

[The EEPROM Library](#)

[Preparing EEPROM Storage](#)

[Adding Nonvolatile Memory](#)

[Summary](#)

### [Chapter 7: SPI](#)

[Introducing SPI](#)

[SPI Bus](#)

[Arduino SPI](#)

[SPI Library](#)

[SPI on the Arduino Due](#)

[Example Program](#)

[Summary](#)

### [Chapter 8: Wire](#)

[Introducing Wire](#)

---

[Connecting I<sup>2</sup>C](#)

[I<sup>2</sup>C Protocol](#)

[Communicating](#)

[Traps and Pitfalls](#)

[Summary](#)

[Chapter 9: Ethernet](#)

[Introduction](#)

[Ethernet](#)

[TCP/IP](#)

[Ethernet on Arduino](#)

[Arduino as a Client](#)

[Summary](#)

[Chapter 10: WiFi](#)

[Introduction](#)

[The WiFi Protocol](#)

[Arduino WiFi](#)

[Example Application](#)

[Summary](#)

[Chapter 11: LiquidCrystal](#)

[Introduction](#)

[LiquidCrystal Library](#)

[Example Program](#)

[Summary](#)

[Chapter 12: SD](#)

[Introduction](#)

[SD Cards](#)

[Using SD Cards with Arduino](#)

[The SD Library](#)

[Example Program and Sketch](#)

[Summary](#)

[Chapter 13: TFT](#)

[Introduction](#)

[Technologies](#)

[TFT Library](#)

[Example Application](#)

[Summary](#)

## [Chapter 14: Servo](#)

---

[Introduction to Servo Motors](#)

[Controlling Servo Motors](#)

[Example Application](#)

[Summary](#)

## [Chapter 15: Stepper](#)

[Introducing Motors](#)

[Controlling a Stepper Motor](#)

[The Stepper Library](#)

[Example Project](#)

[Summary](#)

## [Chapter 16: Firmata](#)

[Introducing Firmata](#)

[Firmata Library](#)

[Example Program](#)

[Summary](#)

## [Chapter 17: GSM](#)

[Introducing GSM](#)

[Mobile Data Network](#)

[Arduino and GSM](#)

[Arduino GSM Library](#)

[Example Application](#)

[Summary](#)

## [Part III: Device-Specific Libraries](#)

### [Chapter 18: Audio](#)

[Introducing Audio](#)

[Digital Sound Files](#)

[Music on the Arduino](#)

[Arduino Due](#)

[Example Program](#)

[Summary](#)

### [Chapter 19: Scheduler](#)

[Introducing Scheduling](#)

[Arduino Multitasking](#)

[Scheduler](#)

[Example Program](#)

[Summary](#)

## [Chapter 20: USBHost](#)

---

[Introducing USBHost](#)

[USB Protocol](#)

[USB Devices](#)

[Arduino Due](#)

[USBHost Library](#)

[Example Program](#)

[Summary](#)

## [Chapter 21: Esplora](#)

[Introducing Esplora](#)

[The Arduino Esplora Library](#)

[Example Program and Exercises](#)

[Summary](#)

## [Chapter 22: Robot](#)

[Introducing Robot Library](#)

[Arduino Robot](#)

[Robot Library](#)

[Example Program and Exercises](#)

[Summary](#)

## [Chapter 23: Bridge](#)

[Introducing Bridge Library](#)

[Bridge](#)

[Example Application](#)

[Summary](#)

## [Part IV: User Libraries and Shields](#)

### [Chapter 24: Importing Third-Party Libraries](#)

[Libraries](#)

[Example Application](#)

[Exercises](#)

[Summary](#)

### [Chapter 25: Creating Your Own Shield](#)

[Creating a Shield](#)

[Your First Shield](#)

[Summary](#)

### [Chapter 26: Creating Your Own Library](#)

[Libraries](#)

[Example Library](#)

# List of Illustrations

[Figure 1.1](#)[Figure 1.2](#)[Figure 1.3](#)[Figure 1.4](#)[Figure 1.5](#)[Figure 1.6](#)[Figure 1.7](#)[Figure 1.8](#)[Figure 1.9](#)[Figure 1.10](#)[Figure 1.11](#)[Figure 1.12](#)[Figure 1.13](#)[Figure 1.14](#)[Figure 2.1](#)[Figure 2.2](#)[Figure 2.3](#)[Figure 2.4](#)[Figure 2.5](#)[Figure 3.1](#)[Figure 3.2](#)[Figure 3.3](#)[Figure 3.4](#)[Figure 3.5](#)[Figure 3.6](#)[Figure 3.7](#)[Figure 3.8](#)[Figure 4.1](#)[Figure 5.1](#)

[Figure 5.2](#)

---

[Figure 5.3](#)

[Figure 5.4](#)

[Figure 6.1](#)

[Figure 6.2](#)

[Figure 6.3](#)

[Figure 6.4](#)

[Figure 7.1](#)

[Figure 7.2](#)

[Figure 7.3](#)

[Figure 8.1](#)

[Figure 8.2](#)

[Figure 8.3](#)

[Figure 8.4](#)

[Figure 8.5](#)

[Figure 9.1](#)

[Figure 10.1](#)

[Figure 10.2](#)

[Figure 10.3](#)

[Figure 11.1](#)

[Figure 11.2](#)

[Figure 11.3](#)

[Figure 11.4](#)

[Figure 12.1](#)

[Figure 12.2](#)

[Figure 12.3](#)

[Figure 12.4](#)

[Figure 12.5](#)

[Figure 12.6](#)

[Figure 12.7](#)

[Figure 13.1](#)

[Figure 13.2](#)

[Figure 14.1](#)

[Figure 14.2](#)

---

[Figure 14.3](#)

[Figure 15.1](#)

[Figure 15.2](#)

[Figure 17.1](#)

[Figure 18.1](#)

[Figure 18.2](#)

[Figure 18.3](#)

[Figure 19.1](#)

[Figure 19.2](#)

[Figure 19.3](#)

[Figure 19.4](#)

[Figure 20.1](#)

[Figure 20.2](#)

[Figure 20.3](#)

[Figure 22.1](#)

[Figure 23.1](#)

[Figure 24.1](#)

[Figure 24.2](#)

[Figure 24.3](#)

[Figure 24.4](#)

[Figure 24.5](#)

[Figure 24.6](#)

[Figure 25.1](#)

[Figure 25.2](#)

[Figure 25.3](#)

[Figure 25.4](#)

[Figure 25.5](#)

[Figure 25.6](#)

[Figure 25.7](#)

[Figure 26.1](#)

[Figure 26.2](#)

[Figure 26.3](#)

# List of Tables

---

[Table 2.1](#)

[Table 3.1](#)

[Table 4.1](#)

[Table 7.1](#)

[Table 7.2](#)

[Table 7.3](#)

[Table 8.1](#)

[Table 8.2](#)

[Table 9.1](#)

[Table 10.1](#)

[Table 10.2](#)

[Table 16.1](#)

[Table 16.2](#)

[Table 17.1](#)

[Table 20.1](#)

[Table 22.1](#)

[Table 22.2](#)

[Table 22.3](#)

---

# **Arduino™ Sketches**

## **Tools and Techniques for Programming Wizardry**

**James A. Langbridge**

**WILEY**

**Published by**

**John Wiley & Sons, Inc.**

**10475 Crosspoint Boulevard**

Indianapolis, IN 46256

[www.wiley.com](http://www.wiley.com)

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-91960-6

ISBN: 978-1-118-91962-0 (ebk)

ISBN: 978-1-118-91969-9 (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 are 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:** 2014948616

**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. Arduino is a trademark of Arduino, LLC. 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.

*To my loving girlfriend, Anne-Laure, who once again put up with entire evenings and weekends spent on my PC. This is the second time I've done that to her, but she put up with me anyway and kept on smiling (most of the time). I still don't know how.*

*To my wonderful daughter, Eléna: I have to admit, I'm addicted to your laugh and smile, something you did every time I showed you the projects I was working on. Again you found a way of telling me when I needed to stop and spend more time playing with you (by unplugging and randomly rewiring my breadboard projects), but coming back home at the end of a long and difficult day to see you smiling and jumping into my arms gave me more energy than you can imagine.*

---

## About the Author

**James A. Langbridge** does not like talking about himself in the third person, but he will try anyway. James was born in Singapore and followed his parents to several countries before settling down in Nantes, France, where he lives with his partner and their daughter.

James is an embedded systems consultant and has worked for more than 15 years on industrial, military, mobile telephony, and aviation security systems. He works primarily on low-level development, creating bootloaders or optimizing routines in assembly, making the most of small processors. When not on contract, James trains engineers on embedded systems, or he makes new gizmos, much to the dismay of his partner.

James wrote his first computer program at age 6 and has never stopped tinkering since. He began using Apple IIs, ZX80s and ZX81s, and then moved to BBC Micros and the Amiga before finally having no other option but to use PCs.

---

## About the Technical Editor

**Scott Fitzgerald** is an artist and educator working with technology and its relationship to people, approaching digital tools from a human-centric perspective. His work has been featured in numerous books and publications such as The New York Times and IDN Magazine. He has edited several books on Arduino and communication technologies, is the author of the book that accompanies the Arduino Starter Kit, and is responsible for documentation of the Arduino platform at <http://arduino.cc>. Scott is currently an assistant arts professor and head of the interactive media program at New York University Abu Dhabi. He enjoys tormenting his cat and partner with early morning work sessions.

---

# Credits

**Project Editor**

Christina Haviland

**Technical Proofreader**

Ying Chin

**Production Editor**

Rebecca Anderson

**Copy Editor**

San Dee Phillips

**Manager of Content Development and Assembly**

Mary Beth Wakefield

**Marketing Director**

David Mayhew

**Marketing Manager**

Carrie Sherrill

**Professional Technology and Strategy Director**

Barry Pruett

**Business Manager**

Amy Knies

**Associate Publisher**

Jim Minatel

**Project Coordinator, Cover**

Patrick Redmond

**Proofreader**

Sarah Kaikini, Word One New York

**Indexer**

Johnna VanHoose Dinse

**Cover Designer**

Michael E. Trent/Wiley

**Cover Image**

© [istock.com/johnbloor](http://istock.com/johnbloor)



---

# Acknowledgments

Writing a book is a huge project. When I was at school, I used to shudder at the thought of writing 1,000 words for an essay, and I was alone to do it. This book is, of course, much longer, and I enjoyed every minute of it, thanks to the team of professionals who helped me every step of the way. Take a quick look at the people involved in this project, and you will soon see what I'm talking about.

I can't thank everyone involved personally; there are just too many people, but there are a few names that I will never forget. My thanks go out to Christina Haviland, my project editor. When I knew that would be working with her again, I was thrilled. She actually managed to put up with me for the entire duration and didn't even shout at me when I was late, despite the fact that some of the chapters were very, very late. I was also thrilled to know that I'd be working with San Dee Phillips, my copy editor. The job they did transforming raw data coming out of my brain into something readable is outstanding. Then there is Scott Fitzgerald, my technical editor, who made sure that I didn't make any mistakes. Believe me, nothing slipped by, and despite all the grumbling I did when I received the corrections, thank you! This wouldn't have been possible without you.

I would also like to thank Atmel for their time and effort, for the engineers I was in contact with to get more information, and to Tom Vu who kept on encouraging me along the way and sending me new evaluation boards to play with. My thanks also go out to Silicon Labs for its excellent UV sensor that is presented in this book and for the time it spent helping me. Thanks to [Materiel .net](#) who managed to get me a new computer, camera, and components in record time when mine broke, allowing me to get the job done. Your coffee mug is still on my desk!

Of course, this book would not have been possible without the amazing people at Arduino. I don't know if they know just how much they have changed the world of makers. Your boards have brought back the joy I had in creating gizmos and contraptions.

This has been a huge adventure, and I've met a lot of amazing people along the way. Thank you to every one of you—for your time, your suggestions, and your encouraging messages.

---

# Introduction

Arduinos have opened up a new world to us. Both makers and professionals use Arduino-based systems to create wonderful and complex devices to help to create fascinating gizmos. From the simplest device that turns on a light when you press a button to advanced 3-D printers, you can use Arduinos in just about every application.

To power all this, Arduinos use *sketches*—software programs that you design to complete your device. They communicate with the outside world and are logic behind your projects. To assist you, the Arduino environment has *libraries*—software that you can add as required, depending on your application or the hardware that you add. Each library is explained in this book with examples for each library.

This book introduces you to Arduino sketches, the software routines that you can use and the different libraries available for the different Arduinos that you will encounter.

The Arduino can be your canvas, and your sketch can be your digital masterpiece.

# **Overview of the Book and Technology**

---

This book covers everything you need to start using Arduinos. It presents the most common Arduinos on the market today, explains how to get your software up and running, and how to program the Arduino, but most important, it explains the Arduino programming languages and the different libraries that you can add to your designs to provide extra functionality. It also gives a primer in electronics to help you in the numerous examples throughout the book.

# How This Book Is Organized

---

This book is designed to give as much information as possible to someone who is starting Arduino programming. It is separated into four parts.

**Part I, “Introduction to Arduino,”** (Chapters 1–3) gives an overview of Arduinos—where they came from and why they are here to stay. It gives a primer on electronics and C programming, and also goes into the Arduino Language, the common elements that you will use for every project.

**Part II, “Standard Libraries,”** (Chapters 4–17) is dedicated to the libraries available for every Arduino, that is, the different software components you can include to add functionality and hardware support. Each library is presented in its own chapter, and an example is provided for each library to help you understand its use.

**Part III, “Device-Specific Libraries,”** (Chapters 18–23) is dedicated to libraries that are specific to different Arduinos; software you can add to a particular Arduino to access hardware or perform specific tasks. Again, each library is presented in its own chapter, and examples are provided.

**Part IV, “User Libraries and Shields,”** (Chapters 24–26) is all about going even further with your Arduino; it explains how to import user libraries and how to design and distribute your own libraries. It also shows how to create your own shield, an electronic board that you can add to your Arduino to provide even more functionality.

## Who Should Read This Book

This book is primarily for makers—people with ideas on how to create amazing applications or automate everyday tasks—and also for developers who want to get into the amazing world of Arduino programming.

## Tools You Need

Each chapter has an example, and the exact components needed for that chapter are listed at the beginning of the chapter. To follow every example in this book, you need the following hardware:

- Computer
- USB cable and micro-USB cable
- power supply
- Breadboard with connector cables
- Several Arduinos:
  - 2 x Arduino Uno
  - Arduino Due
  - Arduino Mega 2560
  - Arduino Esplora
  - Arduino Robot
  - Arduino

- SainSmart LCD Shield
- SainSmart Ethernet Shield
- LM35 Temperature Sensor
- SD card
- Arduino GSM Shield
- Adafruit ST7735 TFT breakout board
- Adafruit MAX31855 breakout board
- Type-K thermocouple wire
- Adafruit SI1145 UV Sensor board
- SainSmart Wi-Fi shield
- DHT11 Humidity sensor
- HC-SR04 ultrasonic distance sensor
- HYX-S0009 or equivalent servo motor
- L293D
- bipolar stepper motor
- Red, green, and blue LEDs
- 10-kilohm resistors
- 4.7-kilohm resistors

## What's on the Website

The source code for the samples is available for download from the Wiley website at [www.wiley.com/go/arduinosketches](http://www.wiley.com/go/arduinosketches).

## Summary

Arduino development is a fascinating subject, one that opens up a whole new world of possibilities. Arduinos are perfectly suited for learning about embedded development, but also for automating everyday tasks or even making amazing gizmos and contraptions. Throughout this book, you'll find numerous examples about how to create simple devices, providing a hardware schematic to get you started, as well as the sketch to get you up and running.

To get the most out of your sketches, each library is introduced and the different functions are explained. Examples are provided for every library, going through the code line by line so you understand what the sketch does. My hope is that this book will serve as a reference for your new projects. Have fun!

# **Part I**

---

## **Introduction to Arduino**

### **In This Part**

[\*\*Chapter 1:\*\*](#) Introduction to Arduino

[\*\*Chapter 2:\*\*](#) Programming for the Arduino

[\*\*Chapter 3:\*\*](#) Electronics Basics

---

# Chapter 1

## Introduction to Arduino

Electronics enthusiasts have always been present. For decades, they have been creating interesting devices and products. Amateur radio enthusiasts typically made their own radio sets using schematics found in magazines or simply from their own design. How many of us built a radio system to discover electronics, only to be hooked? With a few dollars' worth of components, you could create your own radio and listen to glorious long-wave transmissions on a small low-quality speaker, and yet it was better than what could be bought in the shops because it was homemade. If you wanted better sound, you could buy a better speaker. More volume? There were kits for that, too. Audiophiles built their own amplifiers and accessories depending on their needs. Electronics shops proposed books for all levels, from beginner to expert. Kits were also available using the simplest of components all the way to entire computer systems. It was a time in which you could build just about anything, and everything. You could, quite literally, walk into an electronics shop, buy a DIY computer, and spend few hours soldering memory chips onto a printed circuit board. That's how I started.

In the 1990s, things changed slightly. Most hobbyists had a PC on their desk and could use them to create schematics, simulate parts of a system, and even print circuit board with transparent layouts, making the entire process much easier. However, something was missing. Almost all the devices that could be made were not programmable. Microprocessors were available but were either too expensive or too complicated. At the time, the 68000 microprocessor was one of the most reliable components available and was relatively cheap but complex. The microprocessor by itself was useless; it had to be hooked up to external memory. To run a program on every boot, it had to also have read-only memory. If you wanted interrupts, again, you had to add a chip into the design. The end result was complicated and out of the reach of some enthusiasts. To do without this complexity, enthusiasts that wanted programmable devices tended to use what was already on their desk: a personal computer.

Most PCs at the time used the ISA bus, as shown in [Figure 1.1](#). ISA was a simple bus that allowed components to be added to the processor and general computer system. It was a simple system that allowed users to insert add-on cards into their computer, and it was extremely easy to use. It wasn't hard to create a circuit board that could be plugged into an ISA slot, and complete prototyping boards existed, enabling enthusiasts and engineers to test a solution before making their own board. Some of these boards even included breadboards, a simple system allowing users to place their components and wires without the need to solder. This sparked a small revolution, and many enthusiasts turned to this type of board to do what previously could not be done: create programmable systems. An ISA board could have digital inputs and outputs, analog inputs and outputs, radios, communication devices—just about anything was possible. All this would be controlled by the computer's CPU, using simple programming languages such as C or Pascal. My ISA card kept my student apartment nice and warm by reading data from a thermometer and turning on electric heaters, acting like a thermostat. It also served as an alarm clock, programmed depending on my classes the next day. Although I did manage to miss a few morning classes, in all fairness it was usually my fault; the ISA card worked perfectly on a tight budget.

- [Make a Mind-Controlled Arduino Robot: Use Your Brain as a Remote pdf, azw \(kindle\)](#)
- [download Fundamentals of Database Systems \(6th Edition\)](#)
- [click Ugly Love: A Novel](#)
- [L'Ã©veil d'Endymion 1 \(Les Cantos d'HypÃ©rion, Livre 4\) pdf, azw \(kindle\)](#)
- [A Companion to Luis BuÃ±uel \(Wiley-Blackwell Companions to Film Directors\) pdf](#)
- <http://reseauplatoparis.com/library/Make-a-Mind-Controlled-Arduino-Robot--Use-Your-Brain-as-a-Remote.pdf>
- <http://unpluggedtv.com/lib/Fundamentals-of-Database-Systems--6th-Edition-.pdf>
- <http://www.rap-wallpapers.com/?library/Ugly-Love--A-Novel.pdf>
- <http://honareavalmusic.com/?books/L---veil-d-Endymion-1--Les-Cantos-d-Hyp--rion--Livre-4-.pdf>
- <http://aneventshop.com/ebooks/Introducing-Semiotics.pdf>