1

3

3

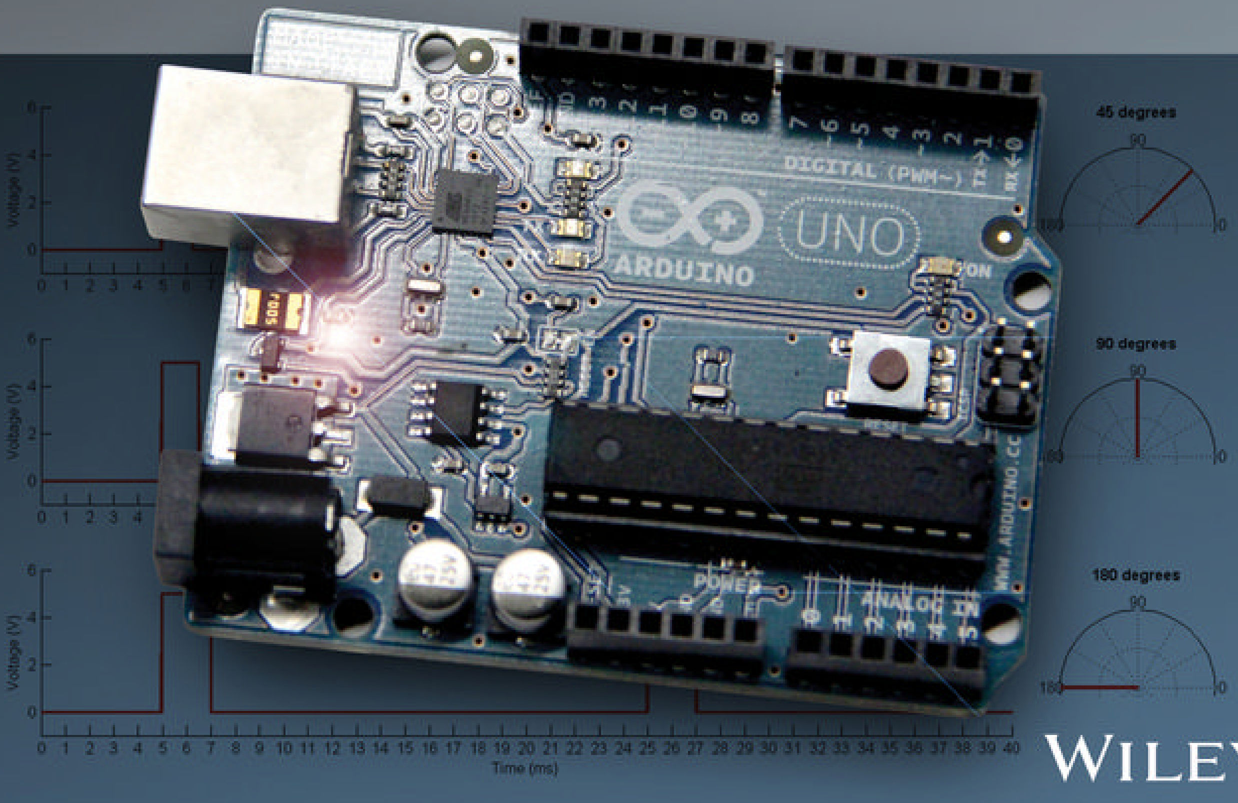
7

"Jeremy uses an Arduino in electrical engineering to make electronic things in the same way that a carpenter uses a hammer to make a house."
— ***Bre Pettis***, *Founder, MakerBot Industries*

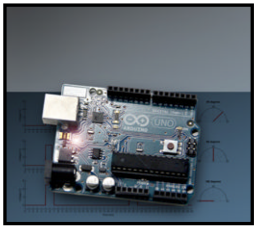
JEREMY BLUM

EXPLORING ARDUINO®

TOOLS AND TECHNIQUES FOR ENGINEERING WIZARDRY



WILEY



Exploring Arduino®

Tools and Techniques for Engineering Wizardry

Jeremy Blum

WILEY

Exploring Arduino®: Tools and Techniques for Engineering Wizardry

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-1-118-54936-0
ISBN: 978-1-118-54948-3 (ebk)
ISBN: 978-1-118-78616-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

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.

Library of Congress Control Number: 2013937652

To my grandmother, whose lifelong curiosity and encouragement inspires me to be a better person every day.

Credits

Acquisitions Editor
Mary James

Project Editor
Jennifer Lynn

Technical Editor
Scott Fitzgerald

Production Editor
Daniel Scribner

Copy Editor
Keith Cline

Editorial Manager
Mary Beth Wakefield

Freelancer Editorial Manager
Rosemarie Graham

Associate Director of Marketing
David Mayhew

Marketing Manager
Ashley Zurcher

Business Manager
Amy Knies

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Neil Edde

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Katie Crocker

Compositor
Cody Gates,
Happenstance Type-O-Rama

Proofreader
James Saturnio, Word One

Indexer
John Sleeva

Cover Designer
Ryan Sneed

Cover Image
Courtesy of Jeremy Blum

About the Author

Jeremy Blum recently received his Master's degree in Electrical and Computer Engineering from Cornell University, where he previously received his Bachelor's degree in the same field. At Cornell, he oversaw the design and creation of several sustainable buildings around the world and domestically through his founding and leadership of Cornell University Sustainable Design, a nationally recognized sustainable design organization that has been specifically lauded by the CEO of the U.S. and World Green Building Councils. In that vein, Jeremy has applied his passion for electrical engineering to design solar home energy monitoring systems, revolutionary fiber-optic LED lighting systems, and sun-tracking smart solar panels. He is also responsible for helping to start a first-of-its-kind entrepreneurial co-working space that contributes to the development of dozens of student start-ups (including some of his own creation) every year.

Jeremy has designed award-winning prosthetic control methods, gesture-recognition systems, and building-automation systems, among many other things. He designed the electronics for the MakerBot Replicator 3D printers (which are used by people around the world, and by notable organizations such as NASA), and the prototype electronics and firmware for the MakerBot Digitizer 3D Scanner. As a researcher in the renowned Creative Machines Lab, he has contributed to the creation of robots that can assemble themselves, self-learning quadrupedal robots, and 3D printers that redefine personal manufacturing. He has presented this research in peer-reviewed journals and at conferences as far away as India.

Jeremy produces YouTube videos that have introduced millions of people to engineering and are among the most popular Arduino tutorials on the Internet. He is well known within the international open source and "maker" communities for his development of open source hardware projects and tutorials that have been featured on the Discovery Channel, and have won several awards and hack-a-thons. Jeremy was selected by the American Institute of Electrical and Electronics Engineers as the 2012 New Face of Engineering.

He offers engineering consulting services through his firm, Blum Idea Labs LLC, and he teaches engineering and sustainability to young students in New York City. Jeremy's passion is improving people's lives and our planet through creative engineering solutions. You can learn more about Jeremy and his work at his website: `www.jeremyblum.com`.

About the Technical Editor

Scott Fitzgerald is an artist and educator who has been using the Arduino platform as a teaching tool and in his practice since 2006. He has taught physical computing in the Interactive Telecommunications Program (ITP) of New York University since 2005, introducing artists and designers to microcontrollers. Scott works for the Arduino team, documenting new products and creating tutorials to introduce people to the platform. He was technical editor of the second edition of *Making Things Talk* in 2011, and he authored the book that accompanies the official Arduino Starter Kit in 2012.

Acknowledgments

First, I must thank my friends at Wiley publishing for helping to make this possible: Mary James, for encouraging me to write this book in the first place; and Jennifer Lynn, for keeping me on track as I worked through writing all the chapters. I also owe a big thanks to Scott Fitzgerald for his critical eye in the technical editing of this book.

Had it not been for the great folks at element14, I may never have gotten into producing my Arduino Tutorial Series, a prelude to the book you are about to read. Sabrina Deitch and Sagar Jethani, in particular, have been wonderful partners with whom I've had the privilege to work.

I wrote the majority of this book while simultaneously completing my Master's degree and running two companies, so I owe a tremendous amount of gratitude to my professors and peers who put up with me while I tried to balance all of my responsibilities.

Finally, I want to thank my family, particularly my parents and my brother, David, whose constant encouragement reminds me why I do the things I do.

Contents at a Glance

Contents

Introduction

You have excellent timing. As I often like to say, "We're living in the future." 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 have it controlling his or her world within minutes. As you may have guessed, a *microcontroller* is a programmable platform 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 great place to start. Are you looking for a general reference for Arduino development? This book is perfect for you, too. This book walks you through a number of particular 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. To facilitate readers of various experience levels, the book features a number of optional sections and sidebars, or short excerpts, that explain a particular concept in greater detail. Although these sidebars are not obligatory 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 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 that it will 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. When particular steps could cause damage to your electronics if performed incorrectly, you'll see one of these asides.

TIP These asides contain quick hints about how to perform simple tasks that might prove useful for the task at hand.

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 following 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

Lucky for you, you can easily obtain the components you need to execute the projects in this book. This book's partner, Newark element14, has created kits specially designed for the contents of this book. You can even use the coupon code at the back of this book to get a discount!

You should order the basic kit first. You can then purchase add-on kits as you progress through the book. Don't want to buy a kit? Don't worry. At the beginning of each chapter, you'll find a detailed list of parts that you need to complete that chapter. The companion website for this book, `www.exploringarduino.com`, also provides links to where you can find the parts for each chapter.

NOTE Did you already buy this book as a bundle from Newark? If so, you're good to go.

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) (Max OSX 10.4+, Windows XP+, or a Linux Distro). I will provide instructions for all operating systems when warranted.

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

- A soldering iron and solder (Note: You will not need to solder to complete the projects in this book, but you may wish to assemble your own circuits on a protoboard, or you may wish to purchase shields that require soldering assembly.)
- A multimeter (This will be useful for debugging concepts within this book, but is not explicitly required.)
- A set of small screwdrivers
- A hot glue gun

Source Code and Digital Content

The primary companion site for this book is `www.exploringarduino.com`, and it is maintained by the author. You will find code downloads for each chapter on this site (along with videos, links, and other useful materials). Wiley also maintains a repository of digital content that accompanies this book at `www.wiley.com`. Specifically for this book, the code download is on the Download Code tab at `www.wiley.com/go/exploringarduino`.

You can also search for the book at `www.wiley.com` by ISBN (the ISBN for this book is 978-1-118-54936-0) to find the code.

At the beginning of each chapter, you can find the location of the major code files for the chapter. Throughout each chapter, you can also find references to the names of code files as needed in listing titles and text.

The code available at `www.exploringarduino.com` and `www.wiley.com` is provided in compressed ZIP archives. After you download the code, just decompress it with an appropriate compression tool.

NOTE Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-1-118-54936-0.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in this book, such as a spelling mistake or faulty piece of code, we would be grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and at the same time, you can help us provide even higher quality information.

To find the errata page for this book, go to `www.wiley.com/go/exploringarduino` and click the Errata link. On this page you can view all errata that has been submitted for this book and posted by Wiley editors.

Supplementary Material and Support

During your adventures with your Arduino, you'll inevitably have questions and perhaps run into problems. One of the best parts about using the Arduino is the excellent support community that you can find on the Web. This extremely active base of Arduino users will readily help you along your way. The following are just a few resources that you'll find helpful on your journey:

- Official Arduino Reference

  `www.arduino.cc/en/Reference/HomePage`
- My Arduino Tutorial Series

  `www.jeremyblum.com/category/arduino-tutorials`
- adafruit Industries' Arduino Tutorial Series

  `learn.adafruit.com/category/learn-arduino`
- SparkFun's Electronics Tutorials

  `learn.sparkfun.com/tutorials`
- The Official Arduino Forum

  `www.arduino.cc/forum`
- The element14 Arduino Community

  `www.element14.com/community/groups/arduino`

If you've exhausted all of those resources and still cannot solve your problem, reach out to me on Twitter (`@sciguy14`); maybe I can help. You can also get in touch with me directly via the contact page on my website (`www.jeremyblum.com/contact`), but I generally don't guarantee fast response times.

What Is an Arduino?

The best part about the Arduino prototyping platform is that it's whatever you want it to be. The Arduino could be an automatic plant-watering control system. It can be a web server. It could even be a quadcopter autopilot.

The Arduino is a microcontroller development platform paired with an intuitive programming language that you develop using the Arduino integrated development environment (IDE). By equipping the Arduino with sensors, actuators, lights, speakers, add-on modules (called *shields*), and other integrated circuits, you can turn the Arduino into a programmable "brain" for just about any control system.

It's impossible to cover everything that the Arduino is capable of, because the possibilities are limited only by your imagination. Hence, this book serves as a guide to get you acquainted with the Arduino's functionality by executing a number of projects that will give you the skills you need to develop your own projects.

You'll learn more about the Arduino and the available variations of the board in Chapter 1, "Getting Up and Blinking with the Arduino." If you're eager to know all the inner workings of the Arduino, you're in luck: It is completely open source, and all the schematics and documentation are freely available on the Arduino website. Appendix A, "Deciphering the ATMega Datasheet and Arduino Schematics," covers some of the Arduino's technical specifications.

An Open Source Platform

If you're new to the world of open source, you are in for a treat. This book does not go into detail about the open source hardware movement, but it is worth knowing a bit about the ideologies that make working with the Arduino so wonderful. If you want a full rundown of what open source hardware is, check out the official definitions on the Open Source Hardware Association website (`www.oshwa.org/definition`).

NOTE Learn all about the open source movement from my TEDx Talk: `www.jeremyblum.com/portfolio/tedx-cornell-university-2011/`. You can also find this video on the Wiley website shown at the beginning of this Introduction.

Because the Arduino is open source hardware, all the design files, schematics, and source code are freely available to everybody. Not only does this mean that you can more easily hack the Arduino to serve a very particular function, but you can also even integrate the Arduino platform into your designs, make and sell Arduino clones, and use the Arduino software libraries in other projects. Although this book focuses mostly on using official Arduino hardware, you could also use hundreds of Arduino derivative boards (often with particular functions added on to them) to create the projects in this book.

The Arduino open source license also permits commercial reuse of their designs (so long as you don't utilize the Arduino trademark on your designs). So, if you use an Arduino to prototype an exciting project and you want to turn it into a commercial product, you can do that. For example, you'll read about products like the MakerBot Replicator 3D printer, which uses electronics based on the Arduino Mega platform (`www.thingiverse.com/thing:16058`). (Full disclosure: I designed that motherboard.)

Be sure to respect the licenses of the source code and hardware that you use throughout this book. Some licenses require that you provide attribution to the original author when you publish a design based on their previous work. Others require that you always share improvements that you make under an equivalent license. This sharing helps the community grow, and leads to all the amazing online documentation and support that you'll undoubtedly refer to often during your Arduino adventures. All code examples that I've written for this book (unless otherwise specified) are licensed under the GNU General Public License (GPL), enabling you to use them for anything you want.

Beyond This Book

Some of you might already be familiar with my popular series of YouTube Arduino and electronics tutorials (`www.youtube.com/sciguy14`). I refer to them throughout this book as a way to see more-detailed walkthroughs of the topics covered here. If you're curious about some of the remarkable things that you can do with clever combinations of electronics, microcontrollers, computer science, and creativity, check out my portfolio (`www.jeremyblum.com/portfolio`) for a sampling of projects. Like Arduino, most of what I do is released via open source licenses that allow you to easily duplicate my work for your own needs.

I'm anxious to hear about what you do with the skills you acquire from this book. I encourage you to share them with me and with the rest of the world. Good luck on your Arduino adventures!

Part I

Arduino Engineering Basics

In This Part

Chapter 1: Getting Up and Blinking with the Arduino
Chapter 2: Digital Inputs, Outputs, and Pulse-Width Modulation
Chapter 3: Reading Analog Sensors

CHAPTER

1

Getting Up and Blinking with the Arduino

Parts You'll Need for This Chapter:

Arduino Uno

USB cable

CODE AND DIGITAL CONTENT FOR THIS CHAPTER

Code downloads, videos, and other digital content for this chapter can be found at `www.exploringarduino.com/content/ch1`.

In addition, all code can be found at `www.wiley.com/go/exploringarduino` on the Download Code tab. The code is in the chapter 01 download and individually named according to the names throughout the chapter.

Now that you have some perspective on the Arduino platform and its capabilities, it's time to explore your options in the world of Arduino. In this chapter, you examine the available hardware, learn about the programming environment and language, and get your first program up and running. Once you have a grip on the functionality that the Arduino can provide, you'll write your first program and get the Arduino to blink!

NOTE To follow along with a video that introduces the Arduino platform, visit `www.jeremyblum.com/2011/01/02/arduino-tutorial-series-it-begins/`. You can also find this video on the Wiley website shown at the beginning of this chapter.

Exploring the Arduino Ecosystem

In your adventures with the Arduino, you'll depend on three main components for your projects:

- The Arduino board itself
- External hardware (including both shields and hand-made circuits, which you'll explore throughout this book)
- The Arduino integrated development environment, or Arduino IDE

All these system components work in tandem to enable you do just about anything with your Arduino.

You have a lot of options when it comes to Arduino development boards, but this book focuses on using official Arduino boards. Because the boards are all designed to be programmable via the same IDE, you can generally use any of the modern Arduino boards to complete the projects in this book with zero or minor changes. However, when necessary, you'll see caveats about using different boards for various projects. The majority of the projects use the Arduino Uno.

You start by exploring the basic functionality baked in to every Arduino board. Then you examine the differences between each modern board so that you can make an informed decision when choosing a board to use for your next project.

Arduino Functionality

All Arduino boards have a few key capabilities and functions. Take a moment to examine the Arduino Uno (see Figure 1-1); it will be your base configuration. These are some key components that you'll be concerning yourself with:

- Atmel microcontroller
- USB programming/communication interface(s)
- Voltage regulator and power connections
- Breakout I/O pins
- Debug, Power, and RX/TX LEDs
- Reset button
- In-circuit serial programmer (ICSP) connector(s)

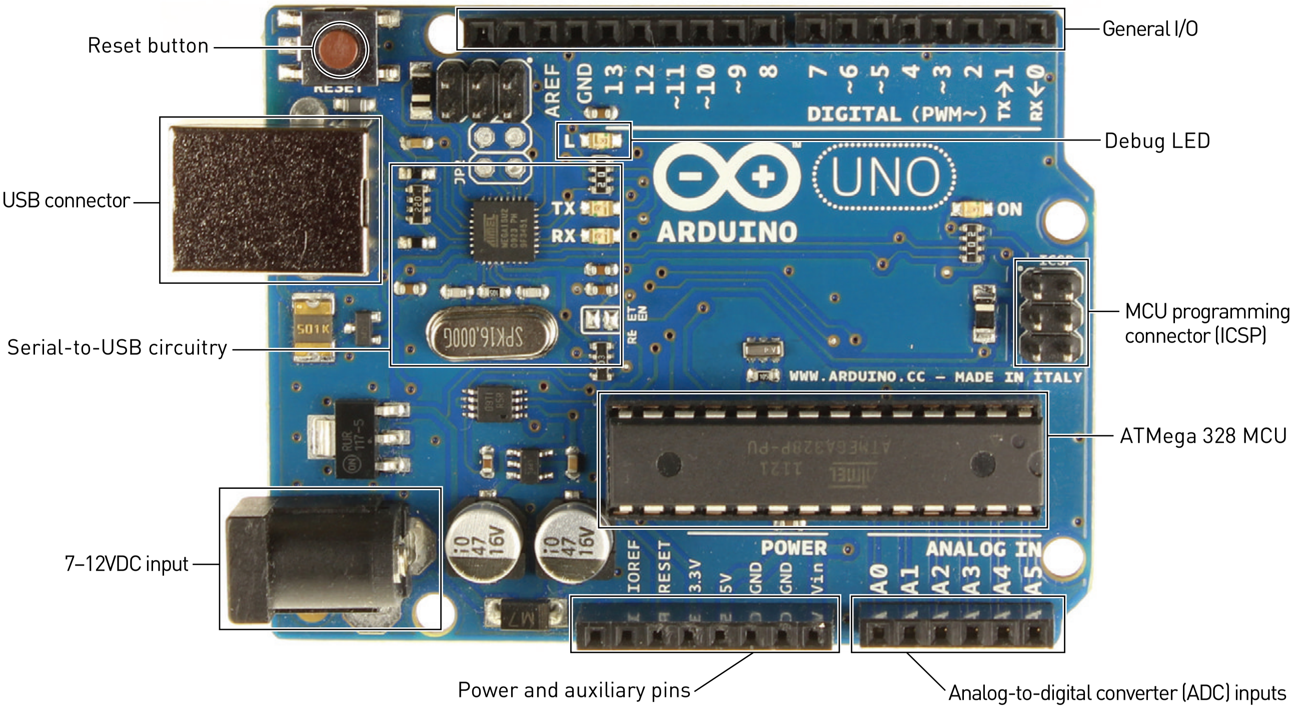


Figure 1-1: Arduino Uno components

Credit: Arduino, `www.arduino.cc`

Atmel Microcontroller

At the heart of every Arduino is an Atmel microcontroller unit (MCU). Most Arduino boards, including the Arduino Uno, use an AVR ATMega microcontroller. The Arduino Uno in Figure 1-1 uses an ATMega 328p. The Due is an exception; it uses an ARM Cortex microcontroller. This microcontroller is responsible for holding all of your compiled code and executing the commands you specify. The Arduino programming language gives you access to microcontroller peripherals, including analog-to-digital converters (ADCs), general-purpose input/output (I/O) pins, communication buses (including I^{2}C and SPI), and serial interfaces. All of this useful functionality is broken out from the tiny pins on the microcontroller to accessible female headers on the Arduino that you can plug wires or shields into. A 16 MHz ceramic resonator is wired to the ATMega's clock pins, which serves as the reference by which all program commands execute. You can use the Reset button to restart the execution of your program. Most Arduino boards come with a debug LED already connected to pin 13, which enables you to run your first program (blinking an LED) without connecting any additional circuitry.

Programming Interfaces

Ordinarily, ATMega microcontroller programs are written in C or Assembly and programmed via the ICSP interface using a dedicated programmer (see Figure 1-2). Perhaps the most important characteristic of an Arduino is that you can program it easily via USB, without using a separate programmer. This functionality is made possible by the Arduino bootloader. The bootloader is loaded onto the ATMega at the factory (using the ICSP header), which allows a serial USART (Universal Synchronous/Asynchronous Receiver/Transmitter) to load your program on the Arduino without using a separate programmer. (You can learn more about how the bootloader functions in "The Arduino Bootloader and Firmware Setup" sidebar.)

In the case of the Arduino Uno and Mega 2560, a secondary microcontroller (an ATMega 16U2 or 8U2 depending on your revision) serves as an interface between a USB cable and the serial USART pins on the main microcontroller. The Arduino Leonardo, which uses an ATMega 32U4 as the main microcontroller, has USB baked right in, so a secondary microcontroller is not needed. In older Arduino boards, an FTDI brand USB-to-serial chip was used as the interface between the ATMega's serial USART port and a USB connection.