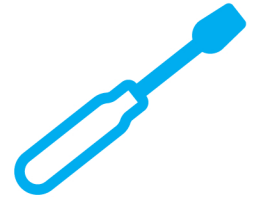


Six embedded projects with open source hardware and software

# Make: Arduino Bots and Gadgets



Learning by  
Discovery

Kimmo Karvinen  
Tero Karvinen

BUILD ROBOTS  
AND OTHER  
ELECTRONIC  
DEVICES

Free Sampler

O'REILLY®

Make:  
makezine.com

## O'Reilly Ebooks—Your bookshelf on your devices!



When you buy an ebook through [oreilly.com](http://oreilly.com), you get lifetime access to the book, and whenever possible we provide it to you in four, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, and Android .apk ebook—that you can use on the devices of your choice. Our ebook files are fully searchable and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at <http://oreilly.com/ebooks/>

You can also purchase O'Reilly ebooks through [iTunes](#), the [Android Marketplace](#), and [Amazon.com](#).

# Make: Arduino Bots and Gadgets

by Kimmo and Tero Karvinen

Copyright © 2011 O'Reilly Media, Inc. All rights reserved.

Printed in Canada.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

**Development Editors:** Brian Jepson and Brian Sawyer

**Production Editor:** Holly Bauer

**Technical Editor:** Joe Saavedra

**Copyeditor:** Rachel Monaghan

**Proofreader:** Jennifer Knight

**Translator:** Marko Tandefelt

**Indexer:** Ellen Troutman Zaig

**Cover Designer:** Mark Paglietti

**Interior Designer:** Ron Bilodeau

**Illustrator/Photographer:** Kimmo Karvinen

**Cover Photographer:** Kimmo Karvinen

**Software Architect:** Tero Karvinen

## Print History:

March 2011: First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. 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 O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

**Important Message to Our Readers:** The technologies discussed in this publication, the limitations on these technologies that technology and content owners seek to impose, and the laws actually limiting the use of these technologies are constantly changing. Thus, some of the projects described in this publication may not work, may cause unintended harm to systems on which they are used, or may not be consistent with current laws or applicable user agreements.

Your safety is your own responsibility, including proper use of equipment and safety gear, and determining whether you have adequate skill and experience. Electricity and other resources used for these projects are dangerous unless used properly and with adequate precautions, including safety gear. These projects are not intended for use by children. While every precaution has been taken in the preparation of this book,

O'Reilly Media, Inc. and the authors assume no responsibility for errors or omissions. Use of the instructions and suggestions in *Make: Arduino: Bots and Gadgets* is at your own risk. O'Reilly Media, Inc. and the authors disclaim all responsibility for any resulting damage, injury, or expense. It is your responsibility to make sure that your activities comply with applicable laws, including copyright.



This book uses Otabind™, a durable and flexible lay-flat binding.

ISBN: 978-1-449-38971-0

[TI]

# Contents

|   |            |
|---|------------|
| <b>Preface</b> .....                                      | <b>vii</b> |
| <b>1. Introduction</b> .....                              | <b>1</b>   |
| Building Philosophy .....                                 | 1          |
| Reusing Parts .....                                       | 4          |
| Buying Components .....                                   | 6          |
| Useful Tools .....  | 7          |
| Electronic Circuit Theory Review .....                    | 14         |
| <b>2. Arduino: The Brains of an Embedded System</b> ..... | <b>17</b>  |
| Why Arduino? .....  | 17         |
| Starting with Arduino .....                               | 18         |
| Hello World with Arduino .....                            | 22         |
| Structure of “Hello World” .....                          | 25         |
| Arduino Uno .....   | 27         |
| Arduino Nano .....  | 28         |
| <b>3. Stalker Guard</b> .....                             | <b>29</b>  |
| What You’ll Learn .....                                   | 30         |
| Tools and Parts .....                                     | 30         |
| Solderless Breadboard .....                               | 31         |
| Jumper Wire .....   | 33         |
| Ping Ultrasonic Sensor .....                              | 33         |
| Vibration Motor .....                                     | 39         |
| Combining Components to Make the Stalker Guard .....      | 41         |
| Making the Motor Vibrate .....                            | 41         |
| Providing Power from a Battery .....                      | 43         |
| What’s Next? .....  | 43         |
| Making an Enclosure .....                                 | 44         |

|  |            |
|--|------------|
| <b>4. Insect Robot</b>                     | <b>53</b>  |
| What You'll Learn                          | 54         |
| Tools and Parts                            | 54         |
| Servo Motors                               | 55         |
| Constructing the Frame                     | 59         |
| Programming the Walk                       | 67         |
| Avoiding Obstacles Using Ultrasound        | 72         |
| What's Next?                               | 77         |
| <br>                                       |            |
| <b>5. Interactive Painting</b>             | <b>79</b>  |
| What You'll Learn                          | 80         |
| Tools and Parts                            | 80         |
| Resistors                                  | 81         |
| LEDs                                       | 83         |
| Detecting Motion Using Ultrasonic Sensors  | 85         |
| Moving Images                              | 97         |
| Installing Python                          | 97         |
| Hello World in Python                      | 101        |
| Communicating over the Serial Port         | 103        |
| Displaying a Picture                       | 107        |
| Scaling an Image to Full Screen            | 107        |
| Changing Images with Button Control        | 111        |
| Gesture-Controlled Painting in Full Screen | 113        |
| Animating the Sliding Image                | 116        |
| Connecting Arduino with Processing         | 122        |
| Processing Code for the Painting           | 124        |
| The Finished Painting                      | 128        |
| Creating an Enclosure                      | 128        |
| Building a Frame                           | 132        |
| <br>                                       |            |
| <b>6. Boxing Clock</b>                     | <b>137</b> |
| What You'll Learn                          | 137        |
| Tools and Parts                            | 138        |
| Android Software Installation              | 138        |
| Creating a Boxing Clock in Android         | 145        |
| What's Next?                               | 176        |
| <br>                                       |            |
| <b>7. Remote for a Smart Home</b>          | <b>177</b> |
| What You'll Learn                          | 177        |
| Tools and Parts                            | 178        |
| The Relay: A Controllable Switch           | 179        |

|   |            |
|---|------------|
| Hacking the Remote Control .....                                | 181        |
| Controlling the Arduino from the Computer .....                 | 184        |
| Creating a Graphical User Interface .....                       | 190        |
| The Finished Remote Control Interface .....                     | 192        |
| Creating an Enclosure .....                                     | 195        |
| <b>8. Soccer Robot.....</b>                                     | <b>199</b> |
| What You Will Learn .....                                       | 200        |
| Tools and Parts .....   | 200        |
| Continuous Rotation Servos .....                                | 203        |
| Modding a Standard Servo into a Continuous Rotation Servo ..... | 207        |
| Connecting the Arduino to the Bluetooth Mate .....              | 211        |
| Testing the Bluetooth Connection .....                          | 215        |
| Building a Frame for the Robot .....                            | 217        |
| Programming the Movements .....                                 | 228        |
| Controlling Movement from a Computer .....                      | 231        |
| Steering with an Android Cell Phone .....                       | 234        |
| The Accelerometer .....   | 238        |
| An Easier Approach to Bluetooth .....                           | 242        |
| Controlling the Robot with Cell Phone Motion .....              | 249        |
| Completing the Soccer Robot .....                               | 253        |
| What's Next? .....  | 262        |
| <b>A. tBlue Library for Android.....</b>                        | <b>263</b> |
| <b>Index.....</b>   | <b>269</b> |

# Introduction

# 1

This chapter will get you started building and designing prototypes for embedded systems. You will learn basic principles that you'll follow in Chapters 3 and 4 as you build the Stalker Guard and Robot Insect. Prototypes in this book are just the beginning. Once you know the techniques, you'll be able to build prototypes for your own inventions.

## IN THIS CHAPTER

- Building Philosophy
- Reusing Parts
- Buying Components
- Useful Tools
- Electronic Circuit Theory Review

## Building Philosophy

When you break a programming problem down into smaller pieces, be sure to test and validate each piece as you go. If you don't do this, you could find yourself wildly off track by the time you've gotten through a few pieces.

## Prototype

This book provides techniques for building *prototypes*, or test versions of a device. A prototype such as the one shown in Figure 1-1 provides a proof of concept—a concrete realization of a device's intended functions.

Try to finish a functional prototype as quickly as possible. Once you've documented a working prototype, you can build in improvements in later versions.

You can make a working end result by stripping out unnecessary functions and taking shortcuts. If it makes testing quicker, use rubber bands and duct tape when you have to. Don't try to optimize your code in the first version.

It's much easier to build an impressive version once the first prototype is finished. Usually, you'll find that many challenging problems you face in the prototype don't even need to be solved for the final version. In the same way, building a prototype can reveal new opportunities for development.



Figure 1-1. Jari Suominen testing a prototype made of Legos

Having a prototype can also help you secure funding for your project. Who would you believe more: someone who talks about a walking robot, or someone who has actually built one?

## Start with Hello World

Starting a project with Hello World is usually a good idea, because it's the simplest possible program. Typically, Hello World will print a row of text to a computer screen or blink an LED. It is used for testing to make sure the development environment works.

If your next, more complicated iteration doesn't work, you can search for the cause of the problem within the added code. Hello World lets you know that the microcontroller, development environment, interpreter, and USB port all function correctly.

## Build in Small Steps

Complex problems (see Figure 1-2, Figure 1-3, and Figure 1-4) are hard to solve, but you can usually make them easier by breaking them down into smaller pieces. You can then solve the problem one manageable piece at a time.

A student of ours once built a burglar alarm after studying embedded systems for a week. The alarm buzzed whenever an infrared sensor detected movement. Users could log into the system wirelessly by presenting an ID in the form of a keychain. Once the system approved the login, the user could then move freely in the space without triggering an alarm.

*How does Arduino say "hello" to the world? By blinking an LED. You'll learn more in Chapter 2.*



Figure 1-2. Juho Jouhtimäki and Elise Liikala building a motion-sensitive soft toy



Figure 1-3. Welding a robot hand



Figure 1-4. Jari Suominen's strobo owl, which uses aesthetics from printed circuit boards and components

A project like this can sound quite complex to a novice, but it really consists of three clearly separate components (motion detector, buzzer, RFID reader). First, the student programmed and tested the motion sensor. That section was finished when the program could detect movement and sound the alarm.

The three components of the system do not affect one another in any way, and the only unifying factor is the code. Program code can check with the motion detector to determine whether movement is present and, if so, it can switch on the buzzer.



## Test in Steps

*"I wrote the code for a singing and dancing robot that can walk up stairs. The code is 30,000 lines long. I just tried compiling it, but it doesn't work. Do you have any advice?"*

Conduct testing as early as possible. If, for example, you build a walking robot, the first thing to test is whether you can make the servo motor move. The next test can make the servo move back and forth.

After you have tested the functionality of a specific version of code, save it separately from the version you are working on.

## Revert to the Last Known Good Version

When you have developed your code into a confusing and nonfunctional state, the solution is easy. Go back to the last working version.

More specifically, go back to a working stage when the situation was already becoming confusing. This method removes the problem areas and lets you start over with a functional clean slate, helping you isolate what went wrong.

## Read the Friendly Manual

RTFM is an old Internet acronym. (Actually, the *F* is not always *friendly*, so we usually stick with just RTM.) The point of the expression is that most answers are out there, written in a manual. When you're surrounded by parts (see Figure 1-5), you're going to need answers.

Friends and students sometimes wonder how we know so much. How do we know the Arduino operating voltage or the way to install SSL encryption to the Apache web server?



Figure 1-5. Mikko Toivonen, surrounded by robots and microcontrollers

The answer is easy. You can find instructions for almost anything if you know where to look.

Instructions don't always come with devices and parts, but you can often find them on manufacturer's web pages (such as <http://www.parallax.com>) or by searching in Google. Good search terms include device names (e.g., "ping ultrasonic sensor") or a sequence of numbers on a circuit board (e.g., "H48C").



Figure 1-6. Jenna Sutela and David Szauder demoing functions of a wearable prototype



Not every device is safe to salvage: for example, a CRT (Cathode Ray Tube) TV retains a hazardous voltage for a long time after you unplug it from the wall.

You could also combine a search sequence with a technology—for example, “H48C arduino.” Some web pages are devoted specifically to Arduino—for example, <http://arduino.cc> and our site, <http://BotBook.com>.

## Document

Most things appear easy once you know them. The details of a project seem obvious on the day you complete them (“of course I remember when I programmed the 16-servo walker”). But a week after building, coding details begin to disappear from your memory. After a year or so, it can be hard for someone who builds many projects to remember anything about a specific one.

For this reason, it is worthwhile to document all projects. Typing notes avoids the potential problem of illegible handwriting, and shooting stages with a digital camera provides an accurate visual snapshot of each stage.

You might also consider publishing your results on the Web. Some projects that would otherwise be collecting dust in your drawer might actually be useful to others. You might even find your own instructions (long since forgotten) when looking to solve a new problem with similar logic. Two sites where you can publish projects are Make: Projects (<http://www.makeprojects.com>) and Instructables (<http://www.instructables.com>).

## Reusing Parts

Prototype mechanics (see Figure 1-6) need all kinds of parts, such as frames, limbs, and joints. Finding appropriate materials can seem daunting. Customizing more complicated parts using homebrew methods isn’t always easy, and even basic materials—such as lightweight and sturdy metal plates—can be significantly expensive at hardware stores.

As a starting point, we recommend using recycled parts. Old devices are filled with usable materials, so remove all salvageable parts before you throw them away.

One additional perk that comes with using recycled parts is a unique aesthetic. Old parts often have interesting shapes, curves, and worn areas (Figure 1-7).

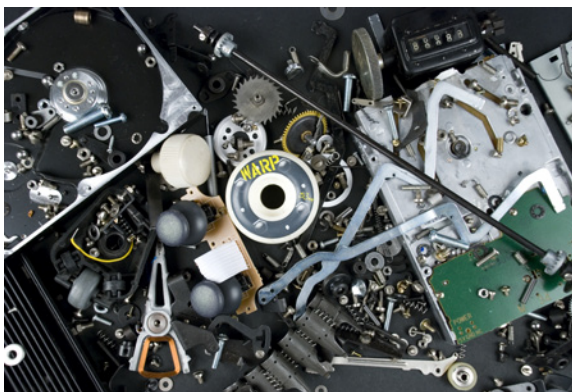


Figure 1-7. An assortment of parts that can be reused

Computer DVD drives and hard drives can make great frames for robots, because their covers are often made of lightweight, easily drillable, and sturdy material. You can also remove DC (direct current) motors and gears from DVD drives. Nowadays, there is more readily available computer junk than you can gather and store in your home. Educational institutions and corporations are particularly good sources, as they're continuously throwing out old devices.

Flea markets can also hold great finds. Mechanical typewriters deserve a special mention here. Though they are relatively hard to disassemble, they house an unbelievable amount of small springs, metal pieces of different shapes, and screws.

Disassemble devices as soon as you find them and then discard or recycle unnecessary parts. This way, you'll avoid turning your home into a graveyard of retired devices, and more importantly, the parts will be immediately usable when you really need them. When you are searching for a suitable attachment piece for a servo, you probably don't want to start a six-hour disassembly operation. Parts usually won't find a new purpose until you've removed them from the original device, at which point inspiration might strike. You might even wonder how a specific "whatchamacallit" fits a new purpose so perfectly.

When you begin working on some difficult new mechanism, think about where you might have seen something similar. You'll often find everyday solutions to many problems. For example, parts purchased from bicycle or automotive shops can sometimes work in other projects. Figure 1-8 shows a hand with fingers that are moved with servo motors; every joint in each finger bends. The fingers were made by attaching sections of a steel pipe to a bicycle chain. They bend when a brake cable is pulled down. Typewriter parts welded to the opposite side of the structure pull the fingers back into a straight position.

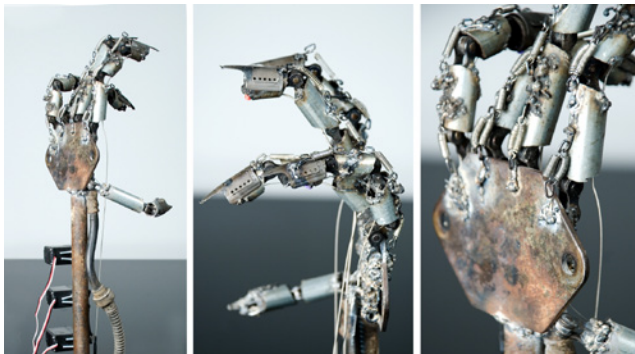


Figure 1-8. Robot hand made of junk

Also keep your eyes open in military surplus stores, where you can find inexpensive, sturdy, and personalized enclosures for prototypes. Various parts and accessories in these shops can also, with a bit of creativity on your part, give devices significantly more street cred. For example, Figure 1-9 shows a porcupine robot cover built from an MG/42 machine gun ammunition belt.

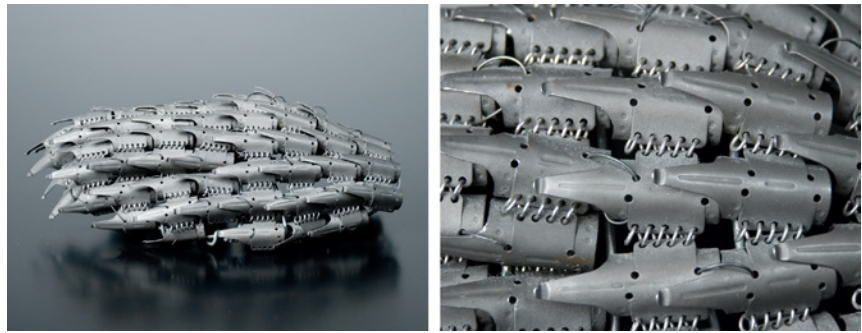


Figure 1-9. Porcupine robot cover made from a machine gun ammunition belt



If you're in the US, you will generally be able to find all the parts you need within the country. However, if you ever need to order large amounts of something (such as hundreds or thousands of LEDs), you may find yourself purchasing from an overseas supplier (for example, many bulk LED sellers on eBay ship from Hong Kong).

When choosing a country to order from, take into consideration customs rules and additional fees incurred by international orders. Shipping costs can also be high in some countries, and some companies won't even ship overseas. Also, consumer protections might not apply to international orders in the event that the package is broken or the product is different from what you ordered.

Regardless of all the scaremongering, ordering internationally usually works out without major problems. We have received everything we have ordered, and the products haven't had any major faults.

## Buying Components

If you can't find exactly what you're looking for in recycled materials, order component parts online. Many unique components can't be found locally at all, or will be overpriced if you do find them. Luckily, comparing prices and ordering online is quite easy.

Because online stores often change, make sure to check the latest links available at <http://BotBook.com/>.

We purchased parts for this book from a variety of sources. We ordered most of the sensors and full-rotation servos from the United States. Arduinos and some of the sensors came from Sweden. We rounded up ordinary components—such as resistors, LEDs, and wiring—from electronics stores in Helsinki. Standard servos came from a Finnish online store specializing in radio-controlled cars and airplanes. Some servos were ordered from Hong Kong. Here are a few sources to consider:

### Maker SHED

MAKE Magazine's store can be found online at <http://www.makershed.com/> and in real life at Maker Faire (<http://makerfaire.com/>). Maker SHED carries Arduinos, project kits, tools, parts bundles, books, and much more. Keep an eye on Maker SHED for special parts bundles or kits dedicated to projects in this book.

### Adafruit Industries

The Adafruit store (<http://www.adafruit.com>) specializes in Arduinos, microcontrollers, electronic and robotic components (including servo motors), tools, and kits. It also has a comprehensive set of Arduino tutorials and produces its own Arduino-compatible boards such as the Boarduino.

### SparkFun Electronics

Among many other things, SparkFun (<http://www.sparkfun.com/>) is a great source for all kinds of sensors—from light and temperature sensors to accelerometers and gas sensors. What's more, it sells the sensors mounted to breakout boards so you can easily connect them to an Arduino without having to do tricky surface-mount soldering. SparkFun has much more, including tools, parts, and Arduinos.

## Useful Tools

When building prototypes, you're going to need some tools (Figure 1-10). The following sections cover the tools that we have found a consistent need for. They are not all mandatory, but depending on your own projects or needs, you may have a use for them in the future.

### Hearing Protectors and Safety Glasses

When using power tools, you must cover your ears with proper hearing protectors and wear safety glasses to protect your eyes from harmful flying debris and material fragments (Figure 1-11). Note that metal can fly forcefully, even when you're cutting or bending with pliers.



Figure 1-10. Wire stripper and side-cutter pliers are sufficient for building prototypes on a prototyping board



Figure 1-11. Hearing protectors and safety glasses

### Needlenose Electronics Pliers

You should immediately purchase good needlenose pliers (Figure 1-12), which can be used to grab small components and parts. The tip for the pliers should be sharp enough to fit into even the smallest of spaces.



Figure 1-12. Needlenose electronics pliers



Figure 1-13. Diagonal-cutter pliers

## Diagonal-Cutter Pliers

Diagonal-cutter (or side-cutter) pliers, shown in Figure 1-13, are used for cutting wires and are also suitable for other small cutting jobs. Always keep at least one set of side cutters in good shape, and use a secondary pair for tasks that cause more wear.

## Metal Saw

A metal saw is a basic, functional tool for shaping and cutting metal (Figure 1-14). Keep a spare blade on hand to keep promising building processes from being interrupted by a broken blade.



Figure 1-14. Metal saw

## Wire Strippers

Wire strippers are used to remove the plastic around a wire to expose a conducting metal within specific areas. Do not use your teeth to strip wires! It is much more expensive to fix dental enamel than to spend just a few dollars on good wire strippers. The adjustable wire strippers on the left side of Figure 1-15 are much more useful than the multigauge model on the right, but they're not as common.



Figure 1-15. Wire strippers

## Screwdrivers

You'll need many different types of screwdrivers, especially when opening devices. Using the wrong screwdriver tip for a particular screw could destroy either the screw or the screwdriver and is just not worth the potential damage. The easiest and most economical thing to do is to buy a kit that comes with a handle and various attachable bits (Figure 1-16). Many electronic devices require a Torx driver and can't be opened with a flat- or Phillips-head screwdriver.

## Alligator Clips

Alligator clips (Figure 1-17) can be useful for quickly connecting components and cables. They can also connect multimeter probes, enabling hands-free measurements.

## Electric Drill

You'll need an electric drill for many projects. A hammer drill, shown in Figure 1-18, is also suitable for drilling into concrete, but a rechargeable cordless drill is easier to handle.

A drill bit can break easily, especially when you're drilling metal with thin bits, so you must wear eye protection when working with a drill. Always position the drill directly into the hole; drilling at an angle will bend the bit and cause it to break under rotation.



Figure 1-16. Screwdriver kit with a variety of bits

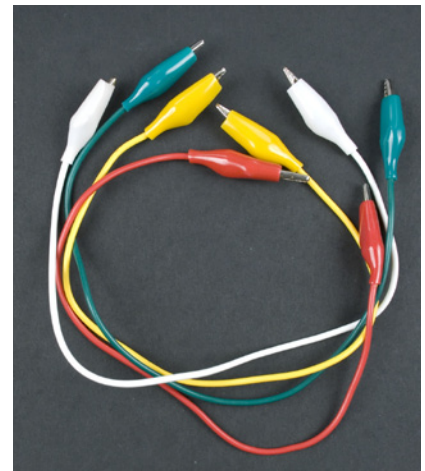


Figure 1-17. Alligator clips



Figure 1-18. Electric drill



Figure 1-19. Leatherman

## Leatherman

A portable handy tool such as a Leatherman (Figure 1-19) is useful during several phases of project building. In this case, it makes sense to invest in the name-brand tool rather than buying cheap imitations. A high-quality multi-purpose tool can withstand heavy use, and its individual parts function in the same way as separate tools.

*Maker SHED sells an assortment of MAKE-branded Leatherman Squirt tools, such as the MAKE: Circuit Breaker Leatherman, a set of electronics tools that can fit on a keychain. See <http://www.makershed.com/SearchResults.asp?Search=leatherman> for more information.*

## Mini Drill

A mini drill (Figure 1-20) is not absolutely necessary, but it makes many tasks easier. Compared to an electric drill, a mini drill is lightweight and relatively precise to work with.

By using an appropriate bit, you can use a mini drill for drilling, sanding, sharpening, shining, cutting, and more. Of course, it doesn't replace a normal drill, because it doesn't have sufficient torque for drilling larger holes.



Figure 1-20. Mini drill



## Headlamp

A headlamp (Figure 1-21) can be handy for focusing light in the direction you're working. Additional light is useful to have, even in well-lit spaces.

## Hot-Glue Gun

A hot-glue gun (Figure 1-22) can adhere items together quickly. The resulting connection is not necessarily very strong, and glued items can bend away from each other, but it works sufficiently well in many prototyping phases. In addition, the fact that hot glue hardens quickly, and items glued with it can be (at least in theory) removed from each other relatively easily, can make the building process less stressful. Still, hot glue is not a replacement for Blu-Tack, and another downside is that if you're unsuccessful in your first attempt to join items together using hot glue, you'll usually need to scrape and shine the surfaces before trying again.

## Nail Punch and Hammer

Drilling metal at home without a drill press can be quite challenging, especially with smooth metal surfaces on which a bit can slide and go through the wrong spot. A nail punch (Figure 1-23, left) can fix this problem. It can create a small dent on the spot where you want to drill a hole, making drilling much easier.

A hammer is a useful tool in its own right, but it's not always the right tool for the job. If you have something to dislodge or to set in place, look for a gentler tool first, so you don't break your project into many little pieces. As Abraham Maslow said, "I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail."



Figure 1-21. Headlamp



Figure 1-22. Hot-glue gun



Figure 1-23. Nail punch and hammer

## Soldering Iron

A soldering iron (Figure 1-24) joins metal sections of components together with molten metal (usually lead, but lead-free solder is available as well). The tip of a soldering iron must be sufficiently thin to enable precise attachment of small parts. Irons with a built-in thermostat are more expensive, but having the capability to adjust the temperature lessens the likelihood of destroying more sensitive components. You will learn the basics of soldering in Chapter 3.



Figure 1-24. Soldering iron

## Multimeter

A multimeter (Figure 1-25) is used for measuring current, voltage, and resistance. You can use it to test a value of a resistor or whether two sections of a circuit are connected. You also can test the condition of a battery by measuring its voltage.

The multimeter shown in Figure 1-25 has two ranges for measuring voltage: DC (direct current) and AC (alternating current). All Arduino circuits in this book use direct current. The correct measurement range for voltage and resistance is the smallest possible range onto which measured readings can fit.

A continuity test works technically in the same way as measuring a value of a resistor. Instead of displaying a resistance value, the continuity test beeps when an unrestricted flow of electricity is detected between two measurement probes.



Figure 1-25. Multimeter

Figures 1-26 and 1-27 illustrate some common uses for a multimeter. The Interactive Painting project in Chapter 5 covers measuring resistance in more detail.



Figure 1-26. The most common functions of a multimeter



Figure 1-27. Studying a remote controller by measuring a voltage difference between two terminals of a button

## Electronic Circuit Theory Review

We'll end this chapter with just enough theory to get you started with the practice.

### Voltage Creates an Electrical Current

*Voltage* refers to a difference in electrical potential between two parts of a circuit. For example, the terminals of a battery can have a 9-volt voltage between them.

If two parts of a circuit with different electrical charges are connected, voltage potential creates a current flow. For example, current will start flowing through a lamp that is connected between the two terminals of a battery, causing the lamp to light up.

A unit of voltage is a *volt* (V). The Arduino microcontroller used in this book functions with a minimum 7V and maximum 12V power adapter (or it can be powered from a 5V USB connection). Voltages inside computers are within a similar range. US AC sockets provide 110 volts and European AC sockets provide 230 volts.

A lamp will be brighter with a 9V battery than with a smaller 4.5V battery. Larger voltage creates a larger current. If a component is used with a voltage higher than what it is rated for, it will usually burn out. If you supply 5 volts to an LED that is rated for 2.4 volts, it will probably make a popping sound, release a little smoke, and cease to function. A running joke among electrical engineers and technicians is that once you've released the "magic smoke" inside an electronic component, you can't put it back in.\*

### A Resistor Resists the Flow of Current

If a resistor is added between a lamp and a battery, the lamp will be dimmer. A resistor resists the flow of current.

All components create at least a bit of resistance. A filament of an incandescent light bulb is sufficient by itself to resist the current flow.

*A resistor may be all that's needed to avoid releasing the magic smoke inside an LED. For example, a 1 kOhm resistor is generally more than sufficient to protect a red LED. If you have the specifications for your LED, you can calculate the value of the resistor. Evil Mad Scientist Laboratories has a handy papercraft pocket LED calculator that you can print out and carry with you: <http://www.evilmadscientist.com/article.php/ledcalc>.*

---

\* [http://en.wikipedia.org/wiki/Magic\\_smoke](http://en.wikipedia.org/wiki/Magic_smoke)

## Short Circuits Are Dangerous

If you bridge a battery's positive and negative terminals with a wire, it forms a *short circuit*. The current flows rapidly through the wire, and the wire and battery will both become warm and may possibly leak or explode. Why do we mention this? Because it's possible to create a short circuit in your own projects if you don't use the correct resistor values. When you follow the instructions to build a project, you must be sure to use the resistor values specified to avoid creating the hazardous condition that comes with a short circuit.

## Closed Circuits Allow Electricity to Flow

When a device is powered, its circuit is closed and electricity will flow through the device. An *open circuit* means that electricity cannot flow through a device. For example, a device that is shut down by its power switch is an open circuit. Electricity can't flow when the circuit is opened by the switch.

Figure 1-28 shows a closed circuit: two batteries powering an LED. The magic smoke didn't come out because the batteries and LED are well matched: the LED has a voltage of 2.6V, which is more than the voltage delivered by two AA rechargeable batteries. Standard AA batteries (1.5V each) might overpower the LED. Still, if you intended to run this circuit for hours on end, it would be advisable to include a low-rated resistor, even a 10 or 100 Ohm.



Figure 1-28. Simple closed circuit powering an LED

## Ground = Zero Voltage Level

To make it easier to discuss topics related to voltage, a single point in a circuit is usually compared to the negative terminal of a power supply. The voltage level of a negative terminal is 0V, against which all other points of the circuit are measured. For example, the positive terminal of a 9V battery can be said to have 9V of voltage.

*Ground* has many names, all of which mean the same thing: *0V*, *minus terminal*, *earth*, and *GND*. Black wire is often used to connect to the ground (red is used for positive voltage). In a circuit, ground is marked with its own symbol (shown in Figure 1-29) to avoid having to always draw a line to the minus terminal.

In this chapter, we've covered prototyping principles, techniques, and tools, and reviewed some basics of electrical theory. Now we'll move on to Chapter 2, where we introduce Arduino, the open source prototyping platform that will be the brain of your projects.

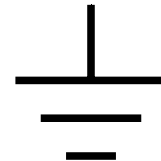


Figure 1-29. Symbol for ground

# Want to read more?

You can find this [book](#) at [oreilly.com](http://oreilly.com)  
in print or ebook format.

It's also available at your favorite book retailer,  
including [iTunes](#), [the Android Market](#), [Amazon](#),  
and [Barnes & Noble](#).



**O'REILLY**<sup>®</sup>

Spreading the knowledge of innovators

[oreilly.com](http://oreilly.com)