



## Praise for *Designing Circuit Boards with EAGLE*

“Matt Scarpino has succeeded where scores of others have failed—he’s managed to make the formidable EAGLE software understandable and, more importantly, useable. His presentation is not only approachable and logical, but it’s complete. When you’ve finished his book, you’ll be able to do something meaningful with EAGLE. This book belongs on every engineer’s bookshelf or tablet.”

—**Bryan Bergeron**, Editor, *Nuts & Volts Magazine*

“Matt Scarpino’s *Designing Circuit Boards with EAGLE* is a great resource for electronics enthusiasts who are ready to get serious and produce their own circuit boards. Matt’s sensible instructions take readers through the steps to design simple and not-so-simple circuit boards, and you can really tell that he’s been using EAGLE for 10 years and loves it. I’m recommending this book to all my maker friends.”

—**John Baichtal**, Author of *Arduino for Beginners: Essential Skills Every Maker Needs*

“With the rising popularity of open source hardware projects, the EAGLE circuit board software has become a vital tool for both hobbyists and professional engineers alike. *Designing Circuit Boards with EAGLE* provides all the information you’ll need to get up to speed with the EAGLE software, and to start creating your own circuit board designs. Matt Scarpino has provided a great tool for the hobbyist starting out in the circuit board design world, demonstrating all of the features you’ll need to know to create your own circuit board projects. However, the experienced engineer will also benefit from the book, as it also serves as a complete reference guide to all the EAGLE software configuration settings and features. His insightful guidance helps simplify difficult tasks in the EAGLE software, and his handy tips will help save you hours of trial-and-error experimenting in your circuit board designs.”

—**Rich Blum**, Author of *Sam’s Teach Yourself Arduino Programming in 24 Hours* and *Sam’s Teach Yourself Python Programming for Raspberry Pi in 24 Hours*

*This page intentionally left blank*

# **Designing Circuit Boards with EAGLE**

*This page intentionally left blank*

# Designing Circuit Boards with EAGLE

*Make High-Quality PCBs at Low Cost*

**Matthew Scarpino**



Pearson

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [international@pearsoned.com](mailto:international@pearsoned.com).

Visit us on the Web: [informit.com](http://informit.com)

Library of Congress Control Number: 2013957510

Copyright 2014 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

ISBN-13: 978-0-13-381999-1

ISBN-10: 0-13-381999-X

Text printed in the United States on recycled paper at Edward Brothers Malloy in Ann Arbor, Michigan.

First printing: April 2014

**Executive Editor**

Bernard Goodwin

**Managing Editor**

Kristy Hart

**Senior Project Editor**

Betsy Gratner

**Copy Editor**

Apostrophe Editing Services

**Indexer**

Tim Wright

**Proofreader**

Kathy Ruiz

**Technical Reviewers**

John Baichtal

Bryan Bergeron

Rich Blum

**Editorial Assistant**

Michelle Housley

**Interior Designer**

Matthew Scarpino

**Cover Designer**

Chuti Prasertsith

**Composer**

Matthew Scarpino

**Senior Composer**

Gloria Schurick

# Contents

Preface .....	xi
Acknowledgments .....	xiv
About the Author .....	xv

## Part I: Preliminary Introduction

<b>Chapter 1: Introducing EAGLE.</b> .....	<b>1</b>
1.1 A Whirlwind Tour of EAGLE .....	2
1.2 Obtaining EAGLE .....	6
1.3 Licensing .....	7
1.4 Organization of This Book .....	8
1.5 More Information .....	10
1.6 Conclusion .....	11
<b>Chapter 2: An Overview of Circuit Boards and EAGLE Design</b> .....	<b>13</b>
2.1 Anatomy of a Printed Circuit Board .....	14
2.2 Overview of Circuit Design with EAGLE .....	22
2.3 Conclusion .....	28
<b>Chapter 3: Designing a Simple Circuit.</b> .....	<b>31</b>
3.1 An Inverting Amplifier .....	31
3.2 Initial Steps .....	33
3.3 The Inverting Amplifier Schematic .....	35
3.4 Board Layout .....	44
3.5 Routing .....	48
3.6 CAM Processor .....	52
3.7 Conclusion .....	55

## Part II: Designing the Arduino Femtoarduino

<b>Chapter 4: Designing the Femtoarduino Schematic</b> .....	<b>57</b>
4.1 Initial Steps .....	59
4.2 The Reset Switch .....	61
4.3 Voltage Regulation .....	65
4.4 The ATmega328P Microcontroller .....	67
4.5 Header Connections .....	71
4.6 Net Classes .....	73
4.7 Electrical Rule Check .....	74
4.8 Generating the Board Design .....	76

4.9 Framing the Schematic . . . . .	76
4.10 Attributes and Assembly Variants . . . . .	77
4.11 Conclusion . . . . .	79
<b>Chapter 5: Layout and Design Rules . . . . .</b>	<b>.81</b>
5.1 Layers . . . . .	81
5.2 Board Layout . . . . .	84
5.3 Design Rule Check . . . . .	90
5.4 Conclusion . . . . .	100
<b>Chapter 6: Routing . . . . .</b>	<b>101</b>
6.1 An Overview of Routing . . . . .	102
6.2 Manual Routing . . . . .	103
6.3 Follow-Me Routing . . . . .	108
6.4 The Autorouter . . . . .	109
6.5 Home PCB Fabrication . . . . .	113
6.6 Conclusion . . . . .	118
<b>Chapter 7: Generating and Submitting Output Files . . . . .</b>	<b>119</b>
7.1 Jobs and the CAM Processor . . . . .	119
7.2 Viewing Gerber Files . . . . .	125
7.3 Drill Files . . . . .	126
7.4 Submitting Design Files . . . . .	130
7.5 Conclusion . . . . .	140
<b>Part III: Advanced Capabilities</b>	
<b>Chapter 8: Creating Libraries and Components . . . . .</b>	<b>141</b>
8.1 Creating the Library . . . . .	142
8.2 Creating the SIMPLE-TQFP16 . . . . .	143
8.3 Creating the VACUUM-TH . . . . .	149
8.4 Creating the TW9920 . . . . .	153
8.5 Conclusion . . . . .	160
<b>Chapter 9: Simulating Circuits with LTspice . . . . .</b>	<b>161</b>
9.1 Introducing LTspice . . . . .	161
9.2 Designing a Schematic . . . . .	163
9.3 Simulating the Circuit . . . . .	170
9.4 Exchanging Designs with EAGLE . . . . .	174
9.5 Conclusion . . . . .	176

## Part IV: Automating EAGLE

<b>Chapter 10: Editor Commands</b> .....	177
10.1 Introducing Editor Commands .....	177
10.2 Schematic Editor Commands .....	180
10.3 Commands for Board Designs .....	187
10.4 Commands for Library Interface .....	192
10.5 The Assign, Change, and Set Commands .....	202
10.6 Configuration Scripts .....	207
10.7 Conclusion .....	208
<b>Chapter 11: Introduction to the User Language (UL)</b> .....	209
11.1 Overview of UL .....	209
11.2 Simple Data Types and Functions .....	211
11.3 Builtins .....	214
11.4 Control Structures .....	219
11.5 The exit Statement .....	221
11.6 Conclusion .....	222
<b>Chapter 12: Examining Designs with the User Language</b> .....	225
12.1 UL-Specific Data Types .....	226
12.2 Schematic Designs (UL_SCHEMATIC) .....	227
12.3 Board Designs (UL_BOARD) .....	243
12.4 Conclusion .....	256
<b>Chapter 13: Creating Dialogs and Menu Items</b> .....	257
13.1 Predefined Dialogs .....	257
13.2 Custom Dialogs and Widgets .....	264
13.3 Dialog Layouts .....	275
13.4 The Menu Command .....	278
13.5 Conclusion .....	279

## Part V: The BeagleBone Black

<b>Chapter 14: Schematic Design for the BeagleBone Black</b> .....	281
14.1 Overview of the BeagleBone Black .....	282
14.2 Advanced EAGLE Schematics .....	284
14.3 AM3359 Memory/JTAG Connections .....	286
14.4 AM3359 I/O Connections .....	290
14.5 System Memory .....	293
14.6 Power Management .....	295

14.7 Ethernet and the Universal Serial Bus . . . . .	297
14.8 Graphics Display . . . . .	300
14.9 Conclusion . . . . .	301
<b>Chapter 15: Board Design for the BeagleBone Black. . . . .</b>	<b>303</b>
15.1 Configuring the Stackup . . . . .	304
15.2 Creating and Routing Ball Grid Arrays . . . . .	307
15.3 Trace Length and Meander . . . . .	313
15.4 The BBB Board Design. . . . .	316
15.5 Conclusion . . . . .	321
<b>Appendix A: EAGLE Library Files . . . . .</b>	<b>323</b>
A.1 Anatomy of a Library File . . . . .	323
A.2 Settings, Grid, and Layers . . . . .	325
A.3 Overview of Library Elements . . . . .	327
A.4 Symbols . . . . .	329
A.5 Packages . . . . .	336
A.6 Devicesets . . . . .	340
A.7 Conclusion . . . . .	346
<b>Appendix B: The Gerber File Format. . . . .</b>	<b>347</b>
B.1 Introducing the Gerber Format . . . . .	347
B.2 Setting Global Properties. . . . .	349
B.3 Aperture Definitions . . . . .	353
B.4 Drawing Shapes . . . . .	356
B.5 Custom Apertures . . . . .	361
B.6 Conclusion. . . . .	368
<b>Index . . . . .</b>	<b>369</b>

# Preface

As I write this in late 2013, the Maker Movement has flourished from a tiny group of tinkerers into a passionate community of millions. Hobbyists have become entrepreneurs and entrepreneurs have become large-scale manufacturers. 3-D printers have fallen into the price range of the average consumer, and the printers' capabilities have risen to such an extent that they're being used to fabricate high-precision aircraft parts and medical equipment. With good reason, many economists and journalists have likened the rise of the Maker Movement to a second Industrial Revolution.

Nothing better illustrates the movement's success than the popularity of the Arduino platform. The first Arduino board design, the Arduino USB, was released in 2005, giving students and hobbyists a low-cost means of programming Atmel microcontrollers. Since then, hundreds of thousands of Arduino boards have been sold, and the Arduino family has expanded to include a vast array of boards, shields, kits, and accessories. Arduino boards have found their way into robots, musical instruments, game platforms, and even unmanned aerial vehicles. The boards have become so popular that many hobbyists-turned-entrepreneurs use them to build prototypes of new inventions.

But Makers still demand more: more capability, more affordability, and more customization. This means designing new circuit boards, a task that requires specialized knowledge and software. Most professional design tools are beyond the price range of the average Maker, but not EAGLE. Since its release in 1988, EAGLE has grown steadily in features and stability while maintaining a price that even cash-strapped students can afford. EAGLE wins legions of admirers with every new version, and the analogy couldn't be clearer: What Arduino is to hardware, EAGLE is to software. It's no wonder that all open-source Arduino designs are released in EAGLE's format.

In writing this book, my mission is to show Makers how to take full advantage of EAGLE's capabilities. This requires a basic understanding of circuit theory, including Ohm's law and Kirchoff's laws, but nothing beyond that. You won't find any transistor analysis or differential equations here. Instead, my goal is to provide a practical, hands-on exploration of EAGLE so that readers can design practical circuit boards, thereby bringing exciting new gadgets to the marketplace and continuing the extraordinary momentum of the Maker Movement.

Matthew Scarpino

## Structure of This Book

This book presents EAGLE by walking through a series of circuit design projects, starting with a simple inverting amplifier and proceeding to a six-layer, single-board computer. As the circuits grow in complexity, I'll explain more advanced features of EAGLE and show how to automate repetitive tasks. This book also includes a great deal of material to help readers understand the fundamentals of circuit boards and the theory behind the example circuits.

Chapters 1, 2, and 3 introduce the topics of EAGLE and circuit board design. Their primary purpose is to familiarize you with EAGLE's capabilities and present the terminology used throughout the book. Chapter 3 breezes through the complete design of a trivially simple circuit.

Chapters 4 through 7 present the design of a practical circuit board: the Arduino Femtoarduino. These chapters take a hands-on approach to explaining the four fundamental steps of circuit board design: drawing a schematic, laying out components, routing connections, and generating Gerber/Excellon files.

Chapters 8 through 13 discuss an assortment of topics related to EAGLE circuit design. These include circuit simulation, the process of creating custom components, and the all-important subject of design automation. Design automation is one of the most powerful aspects of EAGLE, but it's also one of the most overlooked. For this reason, I highly recommend becoming familiar with editor commands and User Language programs.

Chapters 14 and 15 present the book's advanced example design: the BeagleBone Black. The name may sound silly but there's nothing silly about the circuit. It has six board layers, hundreds of components, and thousands upon thousands of routed connections. As I present the design, I'll discuss EAGLE's advanced capabilities and ways to take advantage of design automation.

## Example File Archive

To supplement the text, all the circuit designs, programs, and support files in this book are provided in a zip file called eagle-book.zip. This can be freely downloaded from <http://eagle-book.com>. As you follow the discussion, I recommend that you compare the theoretical discussion to the real-world EAGLE designs. In addition, the color figures for this book can be accessed at [www.informit.com/title/9780133819991](http://www.informit.com/title/9780133819991).

# Acknowledgments

First and foremost, I'd like to thank Bernard Goodwin of Pearson North America for his support and sage wisdom during the creation of this book. Thanks to his deft handling, the development process was as frictionless as could be asked.

I'd like to express my deep appreciation to San Dee Phillips of Apostrophe Editing Services, who caught so many of my formatting, spelling, and grammar errors. I'd also like to thank Betsy Gratner for her cheerful yet firm masterminding of this book's production, Gloria Schurick for her painstaking efforts in compositing, Kathy Ruiz for her eagle-eyed proofreading, and Laura Robbins for managing this book's images.

Last but not least, I'd like to extend my gratitude to Bryan Bergeron of *Nuts & Volts Magazine*, Richard Blum, author of *Sams Teach Yourself Arduino Programming in 24 Hours*, and John Baichtal, author of *Arduino for Beginners*. These reviewers bravely made their way through the book in its rough state and provided many useful suggestions. Their comments have improved the book's accessibility to newcomers and expanded the number of topics.

# About the Author

**Matthew Scarpino** is an engineer with more than 12 years of experience designing hardware and software. He has a Master's degree in electrical engineering and is an Advanced Certified Interconnect Designer (CID+). He currently resides in Massachusetts where he develops software for embedded systems. In his spare time, he uses EAGLE to design accessories for his Android smartphone and the Google Glass.

*This page intentionally left blank*

# *Chapter 1*

## **Introducing EAGLE**

Circuit design applications can be divided into two categories: those intended for large design firms and those intended for everyone else. Applications in the first category provide high reliability, a wide range of features, and responsive technical support. But these advantages come with a hefty price tag. A perpetual license for Altium Designer costs more than \$7,000 and Cadence's OrCad suite costs nearly \$10,000.

Applications in the second category are less expensive, and this makes them accessible to students, individuals, and small-to-medium businesses. Unfortunately, they tend to be unreliable and plagued with bugs. Without technical support, there may be no way to work around these difficulties. What's worse, the companies that release these tools tend to be as flaky as their software and may disappear before their support contracts expire.

But not CadSoft's EAGLE. The Easily Applicable Graphical Layout Editor provides the best of both worlds: the quality of a first-tier design application for the price of a second-tier application. EAGLE has been around since 1988, and with each year, it has improved in capability and reliability. It provides a complete set of features for designing circuit boards, and despite thousands of hours of use, it has never crashed on me. If problems arise, users can visit multiple online forums or read through the many online articles.

EAGLE has one major drawback: its user interface. If you're a frequent Windows user, you're accustomed to applications behaving in a certain manner. You're used to a common set of toolbar items and mouse gestures. But EAGLE has its own unique behavior, and it's impossible to simply start the application and figure out how everything works. It takes time to understand the many editors, dialogs, menus, and commands. And because circuit design is such a complex task to begin with, many newcomers to EAGLE give up.

The goal of this book is to ease the process of learning EAGLE. In these chapters, I'm going to walk through the process of designing circuits, starting with a simple circuit (a noninverting amplifier), proceeding to an intermediate circuit (the Arduino Femtoduino), and finally reaching an advanced circuit (the BeagleBone Black). During the course of this presentation, I'll describe both the EAGLE interface and the general process of designing circuit boards.

In addition to point-and-click design, a significant portion of this book is devoted to automation. EAGLE has a rich command language that can be accessed through scripts and User Language programs, or ULPs. When you have a solid grasp of how to create circuit designs in code, you can perform long, repetitive tasks with a single command. With this automation, your errors will decrease and your productivity will skyrocket.

## 1.1 A Whirlwind Tour of EAGLE

EAGLE is a software application that makes it possible to design circuit boards. Boiled down to its essentials, EAGLE consists of six features:

- **Component library**—The set of devices that can be inserted into a design
- **Schematic editor**—An editor that makes it possible to draw the circuit's preliminary design
- **Board editor**—An editor that defines the circuit board's physical layout and routing
- **Device editors**—Editors used to design new components
- **Autorouter**—A tool that automatically determines how circuit elements can be connected
- **CAM (Computer Aided Manufacturing) processor**—A tool that reads in a board design and produces files for the board's fabrication

This section briefly describes each of these features and how they relate to the overall process of circuit design.

### 1.1.1 The Component Library

One of the most important features of any circuit design tool is the set of available parts. This set of components is called a *library*, and the larger the library, the less time the designer needs to spend defining new devices.

Thanks to its longevity, EAGLE's set of libraries has expanded to thousands and thousands of components, from vacuum tubes to field programmable gate arrays. No matter how complex the design, the odds are that EAGLE will have most of the

required parts. If it doesn't, the site <http://www.cadsoftusa.com/downloads/libraries> provides more libraries for free download. If a part still can't be found, Chapter 8, "Creating Libraries and Components," explains how to design custom parts.

One new feature of EAGLE 6 is the format used by the library files. Each library is defined within a \*.lbr file, and the format for this file is the eXtensible Markup Language (XML), which is popular throughout the world of computing. Appendix A, "EAGLE Library Files," describes the XML schema that defines the structure of EAGLE's library files.

## 1.1.2 The Schematic Editor

After you verified that your circuit's components are available, you can select and connect them inside a schematic design, as shown in Figure 1.1.

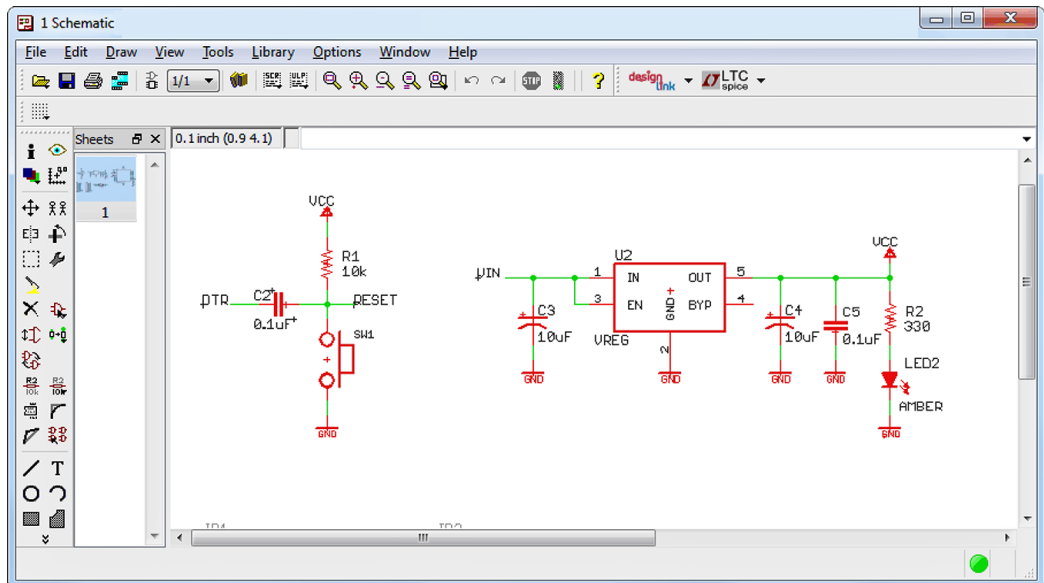


Figure 1.1: The EAGLE Schematic Editor

As with most schematic editors, this keeps track of four important pieces of information:

- Which components are present in the design
- Connections between the components' pins
- Names and values associated with the components
- Properties of the components' connections

EAGLE's schematic editor makes it easy to design a preliminary circuit. Just select a part from the library, move it to a position, and draw connections between it and other components. Afterward, you may assign names and values to the component, such as a resistor's resistance in ohms. Chapter 3, "Designing a Simple Circuit," and Chapter 4, "Designing the Femtoarduino Schematic," discuss the schematic editor in detail.

### 1.1.3 The Board Editor

After a schematic design is complete, EAGLE can generate a board file (\*.brd) that defines the layout of the actual circuit board. Board files are modified in EAGLE's board editor, as shown in Figure 1.2.

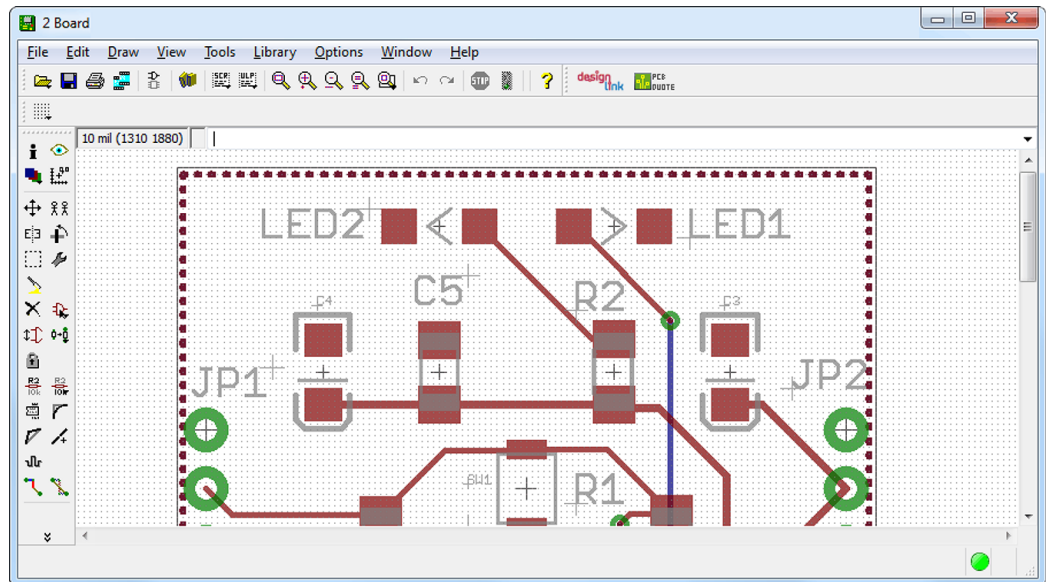


Figure 1.2: The EAGLE Board Editor

In this editor, the designer positions the real-world devices corresponding to the components in the schematic. This position includes not only x and y coordinates, but also whether the components are on the top or bottom layer.

### 1.1.4 The Device Editors

If the EAGLE library doesn't contain a crucial part, the device editors make it possible to design a new one. This process has three steps:

1. Create a design for the schematic editor. This is called a *symbol*.
2. Create a design for the board editor. This is called a *package*.
3. Create an association between the symbol and its package. This is called a *device*.

EAGLE provides editors for laying out a component's symbol and package. These are collectively called the device editors, as shown in Figure 1.3.

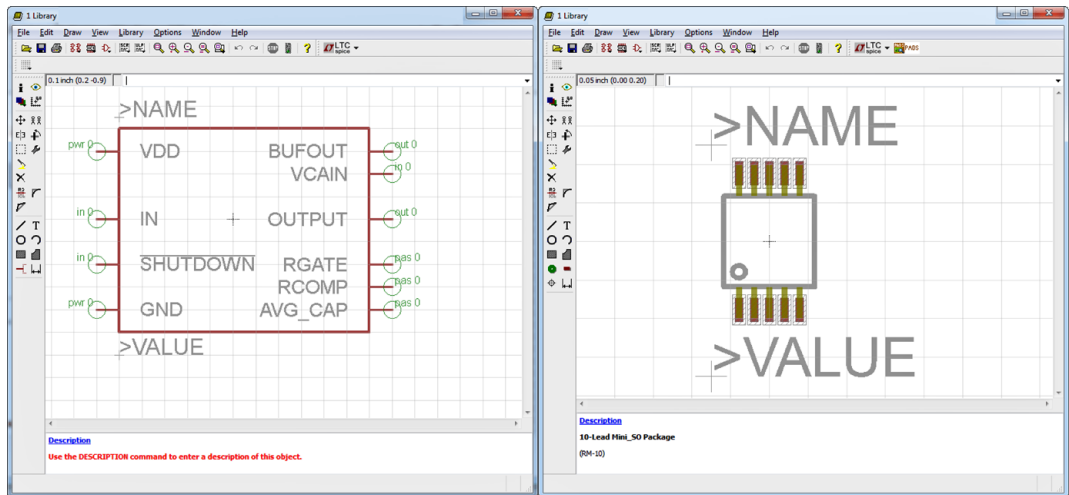


Figure 1.3: The EAGLE Device Editors

The left side of the figure displays the symbol for Analog Devices' SSM2167 component. The right side displays the component's package, which can be used in the board design.

Don't be concerned about terms like symbol, package, and device just yet. These topics will be explored throughout this book, and Chapter 8 presents the entire process of designing new components. Appendix A explains the file format used by EAGLE to store these designs.

### 1.1.5 The Autorouter

After the boards' devices are in place, the next step is to create the connections between them. This is called *routing*, and even with high-end design tools, this process can be complex and time-consuming.

EAGLE's autorouter simplifies the routing process and provides insight into how circuit components can be connected. But for large-scale circuits, it generally isn't capable of completely routing a board on its own. However, if a designer manually creates initial routes, it will help the autorouter do its job. Chapter 6, "Routing," explains all the different routing methods supported by EAGLE.

### 1.1.6 The CAM Processor

Most fabrication facilities don't accept EAGLE design files, so EAGLE's CAM (Computer Aided Manufacturing) Processor converts EAGLE designs into different formats. Figure 1.4 shows the processor's dialog.

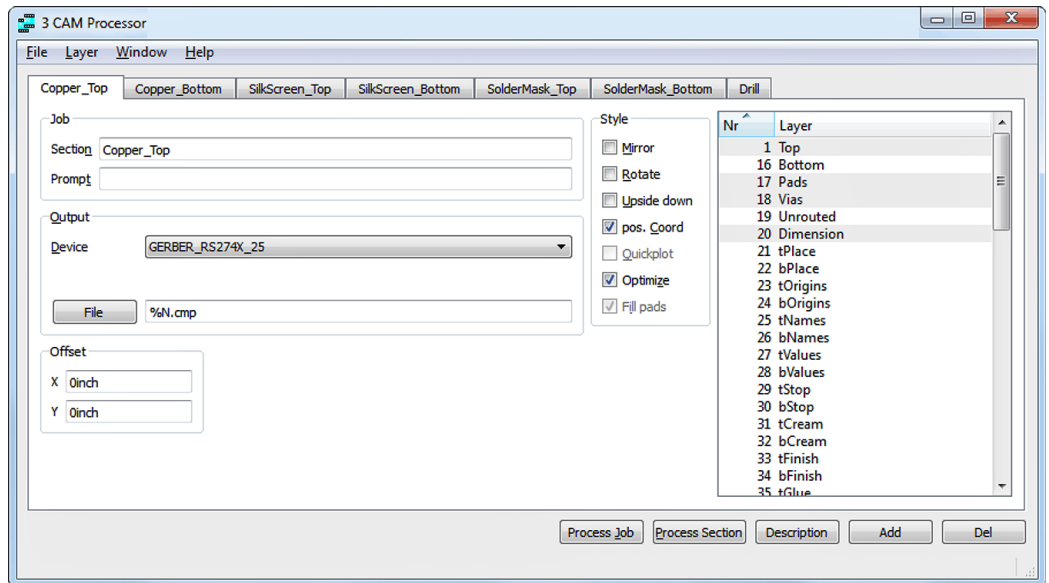


Figure 1.4: The CAM Processor

When the Process Job button is pressed, the processor executes a sequence of tasks called a *job*. A designer can load a job from a file (\*.cam) or create a new job from scratch. As a job executes, each of its tasks reads a portion of the board design and creates a file of the selected type.

To fabricate a circuit board, most manufacturers require two types of files. To define a circuit's geometry and connections, the accepted file format is RS-274X, also called the Gerber format. To specify drill diameters and drill locations, the accepted format is the Excellon format. The CAM Processor generates files of both types.

## 1.2 Obtaining EAGLE

CadSoft makes it easy to start with EAGLE. After you download the executable, you can try it out without registering or paying anything. If you're interested in more features, you can make decisions regarding purchasing, licensing, and registration.

If you run a supported operating system and connect to the Internet, you need to download only a single file. Currently, EAGLE can run on any of the following operating systems:

- Windows 8, Windows 7, Windows Vista, or Windows XP
- Mac OS 10.6, 10.7 on Intel-based processors
- Linux (kernel 2.6, Intel processors, 32-bit runtime environment)

CadSoft's primary web site is <http://www.cadsoft.de> but the company can also be accessed at locale-specific sites such as <http://www.cadsoftusa.com>. At these sites, you can download EAGLE by finding the Downloads link in the upper menu and selecting Download EAGLE. This takes you to a page with download links, and you can choose between the Windows, Linux, and Mac OS offerings.

## 1.3 Licensing

When you first launch EAGLE, a dialog appears and gives you the option of providing a license key or running the tool as freeware. The Freeware option enables you to run EAGLE in a special configuration called the Freeware version of EAGLE Light. This enables you to access EAGLE's editing and routing for free, but with a limited set of features. In addition, this version can be used only for evaluation or nonprofit purposes. If you intend to make money through your PCB design, CadSoft asks that you purchase a license.

EAGLE provides four types of licenses that appeal to different segments of the PCB design community. Each has a different price and set of features (the higher the price, the more features). Specifically, the license type determines the maximum number of schematic sheets, the maximum number of board layers, and the maximum routing area. Table 1.1 lists the characteristics of the different licenses.

**Table 1.1**

EAGLE License Features

License	Number of Sheets	Number of Layers	Routing Area in mm <sup>2</sup>
Light	1	2	100 × 80
Hobbyist	99	6	160 × 100
Standard	99	6	160 × 100
Professional	999	16	4000 × 4000

Looking at this table, you may wonder what the difference is between the Hobbyist and Standard licenses. The Hobbyist license is much, much less expensive but carries the requirement that EAGLE can be used only for noncommercial purposes. CadSoft requires a signed statement to this effect.

Table 1.1 doesn't list the prices for these licenses for three reasons:

1. EAGLE's prices change over time and any listed price will prove inaccurate in the near future.
2. For the Standard and Professional licenses, CadSoft doesn't sell EAGLE as an integrated application. Instead, it splits EAGLE into three parts (schematic editor, autorouter, and board editor), and sells them separately.
3. For the Standard and Professional licenses, pricing depends on how many users can use the tool at once.

For a full presentation of the EAGLE pricing structure, visit the CadSoft web site. For prices in American dollars, the link is <http://www.cadsoftusa.com/shop/pricing/?language=en>.

In writing this book, I have made no assumptions regarding which license readers have purchased or if any license has been purchased at all. But this book covers every aspect of EAGLE, so if one or more features are unavailable in your installation, skip over the corresponding material. The first and second example circuits can be designed with any of the licenses, but the final design (discussed in Chapters 14, “Schematic Design for the BeagleBone Black,” and 15, “Board Design for the BeagleBone Black”) requires more advanced capabilities.

## 1.4 Organization of This Book

This book is structured so that the material proceeds from the simple to the complex and from the fundamentally important to the esoteric. More specifically, the chapters in this book can be divided into five parts, each of which focuses on a different task or aspect of EAGLE.

### Part I: Preliminary Introduction

The first part of this book provides essential information for readers new to circuit board design and EAGLE. Chapter 2, “An Overview of Circuit Boards and EAGLE Design,” explains what circuit boards are and how they’re manufactured, thereby establishing the vocabulary that will be used throughout this book. It also explains the overall circuit board design process with EAGLE.

Chapter 3, “Designing a Simple Circuit,” expands on this introduction and walks through the schematic design and board design for a simple amplifier circuit. This circuit isn’t intended to be manufactured, but the design process will be helpful to inexperienced readers.

### Part II: Designing the Arduino Femtoduino

The second and largest part of this book centers on designing an Arduino Femtoduino. The Arduino family of circuit boards enjoys a great deal of popularity among amateurs and professionals, and Chapters 4 through 7 explain how to design one for yourself. Chapter 4, “Designing the Femtoduino Schematic,” explains how to create the schematic and Chapter 5, “Layout and Design Rules,” explains how to position the packages in the board editor.

Chapter 6, “Routing,” discusses the process of design rule checking and shows how to route the connections on the Arduino Femtoduino. Lastly, Chapter 7, “Generating and Submitting Output Files,” presents the Computer Aided Manufacturing (CAM) processor and explains how to generate the final artwork files for the Femtoduino.

It also presents five different fabrication services that accept these files and deliver finished circuit boards.

### **Part III: Advanced Capabilities**

The next part of the book covers two topics that go beyond regular schematic/board design. Chapter 8, “Creating Libraries and Components,” explains how to create custom components for EAGLE and walks through two designs. The first creates a symbol and package for a through-hole component and the second creates a symbol and package for a surface-mount component.

Chapter 9, “Simulating Circuits with LTspice,” delves into one of EAGLE’s newest and most interesting features: circuit simulation with LTspice. LTspice is a freely downloadable simulation tool that makes it possible to draw circuits, assign inputs, and simulate the circuit’s operation. By combining EAGLE and LTspice, designers can test a design before sending it out for fabrication.

### **Part IV: Automating EAGLE**

The fourth part of this book focuses on automating EAGLE using scripts and program files. Chapter 10, “Editor Commands,” presents the EAGLE command language, which executes design operations in text. For example, the `add` command adds a new component to a schematic or board design, and the `auto` command launches the autorouter.

Chapters 11 through 13 explain how to write User Language programs (ULPs), which make it possible to examine circuit designs automatically. These chapters provide many useful examples that can simplify EAGLE usage and reduce time associated with the design process.

### **Part V: The BeagleBone Black**

The last two chapters of this book focus on designing the BeagleBone Black. This advanced circuit board has six layers and hundreds of components, some of which have high-density ball grid array (BGA) pins. Though difficult to design, the BeagleBone Black has gained a significant following among programmers because of its extraordinary amount of computing power.

### **Example File Archive**

All the designs, programs, and support files discussed in this book are freely available online. The archive is called `eagle-book.zip` and it can be downloaded from <http://eagle-book.com>.

## 1.5 More Information

One of EAGLE's greatest advantages is the staggering amount of information available. No matter what problem you face, it's likely that someone has already encountered it and found a solution. In addition to this book, here are four sources of information that I highly recommend.

### 1.5.1 Element14—[www.element14.com](http://www.element14.com)

EAGLE is maintained and released by CadSoft, but in 2009, CadSoft was acquired by Premier Farnell PLC, a distributor of electronic components. That same year, Premier Farnell created element14, an online community to provide support for circuit designers. This community has grown significantly over the years, and each day its forum receives hundreds of designers asking and answering questions. In addition, it provides a library of documentation and videos related to electronic design.

EAGLE isn't the only topic discussed at element14, but the subforum devoted to EAGLE support is one of the busiest places on the site. Here, users ask questions ranging from routing issues to library entries to converting file formats to those used by other tools. Richard Hammerl, a chief technician at CadSoft, frequently answers questions, which means the subforum is nearly as good as full professional support.

### 1.5.2 SparkFun—[www.sparkfun.com](http://www.sparkfun.com)

In 2003, Nathan Seidle founded SparkFun Electronics to “make electronics accessible to the average person.” This site sells development tools and kits, such as Arduino boards, and it also provides articles related to electrical design. The list of tutorials includes SMT soldering, programming, robotics, and of course, EAGLE. Nathan Seidle has written a series of articles that discuss EAGLE, and SparkFun provides its own EAGLE scripts, programs, and CAM Processor jobs.

The SparkFun forum is very active and its subforums discuss topics as diverse as wireless/RF design, GPS projects, and shipping times for fabrication facilities. In the PCB Design Questions subforum, many EAGLE users submit questions and receive answers.

### 1.5.3 YouTube—[www.youtube.com](http://www.youtube.com)

If you search for **EAGLE** and **PCB** or **CadSoft** on YouTube, you'll find many YouTube videos devoted to explaining EAGLE usage. Some may be out of date, but taken as a whole, they provide a friendly introduction to this complicated topic.

### 1.5.4 CadSoft—[www.cadsoftusa.com](http://www.cadsoftusa.com)/[www.cadsoft.de](http://www.cadsoft.de)

Last but not least, I recommend CadSoft's main site. CadSoft provides a great deal of documentation on EAGLE, but in general, you can find the same documentation inside EAGLE's top-level doc directory. One major point of interest is the Downloads link, which makes it possible to download additional libraries, ULPs, and actual EAGLE projects.

## 1.6 Conclusion

I first used EAGLE around 2003, and though it had many of the same capabilities as today, it tended to crash at least three times an hour. On message forums, users railed against EAGLE's instability and exchanged workarounds for dealing with its many bugs. But CadSoft persevered in its work on EAGLE, and nearly 10 years later, the bugs and instability are gone. Instead of complaining, today's users defend the application fiercely.

I'm a devoted EAGLE user, and my goal in writing this chapter is to explain why I think the tool is so wonderful. EAGLE provides a full suite of design features, including a schematic editor, a board editor, device editors, and a CAM processor. Its libraries contain thousands and thousands of electronic components. It's stable, runs at high speed, and if I encounter issues, there are many online resources I can turn to.

One of the reasons I'm so impressed with EAGLE is its generous licensing. Users can try out the tool for free and continue using it indefinitely. If they'd like to take advantage of its advanced features, they can purchase a license without spending great sums of money.

*This page intentionally left blank*

# *Chapter 2*

## **An Overview of Circuit Boards and EAGLE Design**

Before you start using EAGLE, you should have a basic understanding of printed circuit boards (PCBs)—what they are and how they're constructed. This is a complex topic with many specialized concepts and terms, but I'll assume you know nothing about them. Therefore, my goal in writing this chapter is to bring you up to speed.

The first part of this chapter provides a general overview of circuit board technology. It starts by explaining what PCBs are meant to accomplish and how this is reflected in their design. It explains the basics of PCB construction for single-sided boards, double-sided boards, and multilayer boards.

The second part of this chapter explains how PCBs can be designed using EAGLE's capabilities. This process consists of five steps:

1. Create a new project.
2. Design a schematic that defines the circuit's components and their connections.
3. Create a circuit board design from the schematic and position the packages corresponding to the schematic's components.
4. Route connections between the packages.
5. Convert the board design into files that can be sent to a fabrication facility.

If you already have a solid grasp of topics like silk-screening and solder mask, feel free to skim this chapter or skip it altogether. If not, I recommend that you take the time to understand how PCBs are constructed and how EAGLE makes it possible to design them. Then you'll have no trouble with the initial circuit design presented in the next chapter.

## 2.1 Anatomy of a Printed Circuit Board

We all know what circuit boards look like: They're thin, rigid, and usually rectangular, with components attached to one or both surfaces. The top and bottom are generally colored in dark blue or green. Lines running between the components have a slightly different color.

In addition to the top and bottom sides, modern circuit boards have internal planes called *layers*. Internal layers don't have components but may contain metal lines that carry electricity to and from the components on the top and bottom. For example, the circuit board in the iPhone 4 handset has 10 layers.

Layers are critically important in PCB design, so circuit boards are commonly divided into three categories: single-sided, double-sided, or multilayer. This section discusses each of these categories and the manner in which the circuit boards are constructed.

But first, let's answer the question of why circuit boards exist. At the very least, a circuit board serves two purposes:

1. Provides mechanical support for a set of components
2. Provides electrical connections between the components

Given this focus on components, it's a good idea to take a brief look at the types of electric components that board designers have to deal with.

### 2.1.1 Electrical Components

From the dawn of circuit design to the present day, engineers have created thousands of different types of components. They can be categorized in many ways, and one common distinction involves whether they need power to operate. Components that require power, such as transistors and integrated circuits (ICs), are called *active components*. Components that don't need power, such as resistors and capacitors, are called *passive components*.

For circuit board designers, components are categorized according to their *leads*, also referred to as *terminals*. A lead is a metal extrusion that serves as the component's connection point to a circuit board. Figure 2.1 depicts three popular types of leads:

- **Through-hole**—Leads are wires that enter holes in the board.
- **Surface mount technology (SMT)**—Leads are metal tabs on the perimeter of the device.
- **Ball grid array (BGA)**—Leads are metal balls on the bottom of the device.

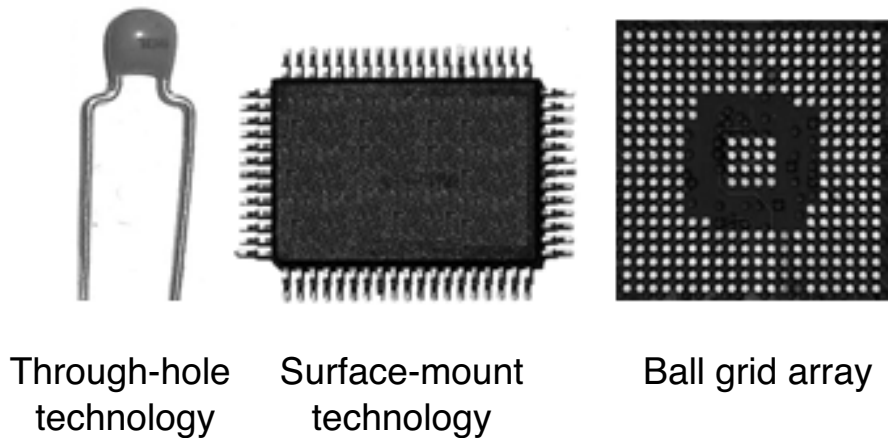


Figure 2.1: Component Leads

Through-hole components dominated the twentieth century but SMT components have become much more popular. This is because SMT components are smaller, don't require drilling, and can be automatically positioned using devices called pick-and-place machines.

The drawback of SMT components is that the number of leads is limited by the component's perimeter. In contrast, the number of leads in a BGA component is limited by the component's area. Therefore, BGA components allow for many more leads.

The location on a board that comes in contact with a component's lead is called a *pad*. Leads are attached to pads using solder, and because solder connects metal to metal, pads must be metallic. The second reason that pads must be metallic is that they need to conduct electricity to or away from the lead.

In general, through-hole pads and BGA pads are round, and SMT pads are rectangular. Figure 2.2 presents typical pads for through-hole and SMT components.

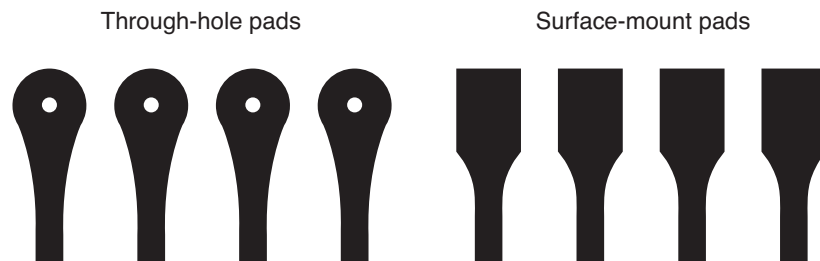


Figure 2.2: Through-Hole and SMT Pads

In all cases, the pads should be large enough so the leads can be reliably soldered to the board. But if the pads are too large, they may come in contact with other metallic surfaces.

The second function of a circuit board is to provide connections between a circuit's components. This means connecting pads to one another using a conductive material. To establish these connections, circuit boards use conductive lines called *traces*. The process of laying out a board's traces is called *routing*, and it's a crucial part of the circuit design process.

In theory, circuit board design is fairly simple: Just figure out where the component pads should be and create connections between them. But in practice, this process has a vast number of details that need to be considered. How wide should the traces be? How thick should the metal be? Should Component A be on the top or the bottom side? What's the best way to connect Pads 1, 3, 5, and 7 on Component X to Pads 2, 4, 6, and 8 on Component Y?

I won't pretend that this book will provide answers to these questions or even provide reliable methods for finding them. Instead, my goal is to explain how EAGLE can assist you as you search for your own answers.

### 2.1.2 Single-Sided Boards

The simplest circuit board to understand and fabricate is the single-sided board. Figure 2.3 presents a cross-section.

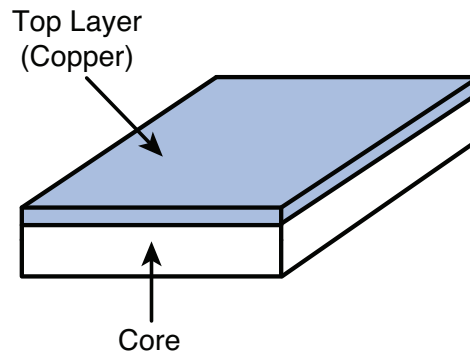


Figure 2.3: Cross-Section of a Single-Sided Board

To understand how single-sided boards are fabricated, it's important to be familiar with three topics: board materials, photolithography, and post-processing.

#### Circuit Board Materials

The body of a single-sided board is made of a hard, nonconductive material, typically fiberglass epoxy-resin, or FR4. Some literature refers to a board's material as substrate, but in this book, we'll refer to this material as *core*. Core thickness is given in thousandths of inches called *mils* (1 mil = 0.001 in). Standard board thicknesses are 31 mils, 39 mils, and 62 mils.

A thin layer of metal is attached (usually glued or electrodeposited) to one side of the board, and we'll call this the top side. This metal is almost always copper, which is inexpensive and provides a low-resistance path for electricity. As Table 2.1 shows, the thicker the copper layer, the lower the resistance.

**Table 2.1**

Copper Thickness and Sheet Resistance

Thickness in oz	Sheet Resistance in $\mu\Omega$
0.5	971
1.0	486
2.0	243
3.0	162

It may seem odd that the copper thicknesses in Table 2.1 are given in ounces, but this is the common measurement used in the industry. For circuit boards, an ounce corresponds to the weight of copper per square foot. Table 2.2 relates copper ounces to the thickness of the metal in mils.

**Table 2.2**

Copper Thicknesses

Copper Thickness in oz	Copper Thickness in mils
0.5	0.68
1.0	1.35
2.0	2.70
3.0	3.05

Thick copper provides less resistance than thin copper and can carry greater current. But thin copper makes it easier to form small features. This is why many fabrication houses allow a smaller minimum dimension for thin copper than for thick copper.

## Photolithography

To remove unwanted copper from a board, PCB fabrication houses employ photolithographic (*photo* - light, *lithos* - stone, *graphein* - write) methods. Figure 2.4 shows how this works.

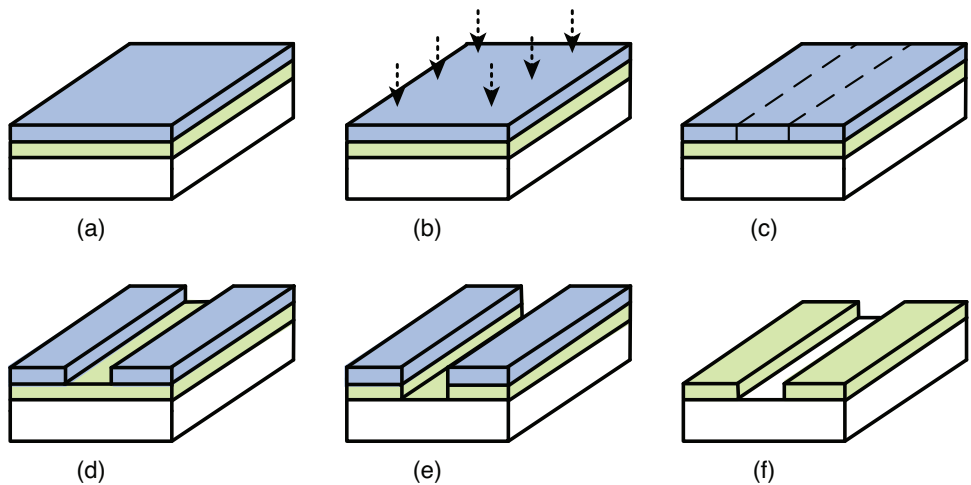


Figure 2.4: Circuit Board Photolithography

The six steps depicted in the figure are given as follows:

- a. The copper layer is coated with light-sensitive material called photoresist.
- b. A photoplotter uses a light source to selectively expose portions of the photoresist.
- c. The photoresist's chemical properties change when exposed to light. If the photoresist is positive, the exposed photoresist softens. If the photoresist is negative, the exposed photoresist hardens. The photoresist in the figure is negative.
- d. The softened photoresist is removed using a chemical called developer. The hardened photoresist remains and covers parts of the copper.
- e. The uncovered copper is removed using a strong acid, such as cupric chloride. This process is called etching and the chemical is called an etchant.
- f. The remaining photoresist is removed, leaving only the patterned copper on the circuit board.

## Post-Processing

After photolithography, electrical components can be soldered to the board. But most board fabrication processes perform at least three steps beforehand:

1. Cover circuit board (except pads) with solder mask.
2. Apply solder paste to pads.
3. Use silk-screening to print letters and draw symbols.

To protect the copper traces, board fabrication facilities frequently cover the circuit board with a tough, nonconductive material called *solder mask*. Solder mask is usually dark green or blue, which explains why so many circuit boards have these colors. The solder mask shouldn't cover the pads because that would make it difficult to solder components to the board.

To assist in the soldering process, it's common to apply a small amount of conductive glue to SMT pads. This glue is called *solder paste*, and it is an important part of modern circuit board assembly.

On many circuit boards, you'll notice lettering and symbols printed in white. These markings may identify where components should be placed, which leads are available for testing, and which company designed the board. This printing process is referred to as *silk-screening* because silk was originally used to form the stencils that mask the printed symbols.

### 2.1.3 Double-Sided Boards

As the number of components on a board increases, so do the number of traces. In many instances, the full set of traces can't be connected on a single plane without intersecting one another. To solve this problem, traces need a way to jump over other traces. This jumping is made possible by adding copper to the board's bottom side. This bottom side may also support additional components. A board with copper on both sides is called a double-sided board.

The process of fabricating double-sided boards is similar to that of single-sided boards. Copper is glued to both sides of the core material and both sides are processed using similar photolithographic methods.

The important difference between double-sided and single-sided boards is the need for electrical paths between the top and bottom sides. These paths are called *vias*, and they're formed by creating tunnels through the core material and filling them with metal. Figure 2.5 depicts the cross-section of a via connecting the sides of a double-sided board.

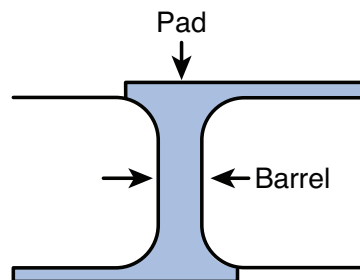


Figure 2.5: A Via Through a Double-Sided Board

The body of the via between the two layers is called the barrel. A pad is formed where the via touches either surface. A via's aspect ratio is the ratio of its height to its diameter.

**NOTE** A via's pad isn't necessarily a regular pad. That is, a via's pad is usually not connected to a component's lead. If a via pad is directly connected to a component's lead, the process is called *via-in-pad*.

Vias are categorized according to how their holes are created. The most popular method of creating a via is to drill through the core material and fill the hole with metal. In this case, the via is called a *plated-through hole*, or *PTH*. PTH diameters are commonly given in mils, and common via diameters are 12, 13, and 25 mils.

The second method of creating vias is to use lasers, photolithography, or plasma etching to create holes smaller than can be created by drills. These are called *microvias* and their diameters are commonly given in millimeters. Typical diameters are 0.1 and 0.3 millimeters. The holes are usually filled with solder paste.

### 2.1.4 Multilayer Boards

Double-sided boards enable more complex routing than single-sided boards, but sometimes two layers still aren't enough. This occurs frequently when designing circuits whose components have hundreds of closely spaced leads. It's also important for designs that require entire layers for ground or power supply.

To resolve these issues, board designers create multilayer board designs. In essence, a multilayer board is a group of double-sided boards sandwiched together using a material called *pre-preg*. Similar to glue, pre-preg is soft to begin with but hardens when heat and pressure are applied. A board's arrangement of core and pre-preg layers in a multilayer board is called its *stackup*, and Figure 2.6 shows a stackup for an eight-layer board.

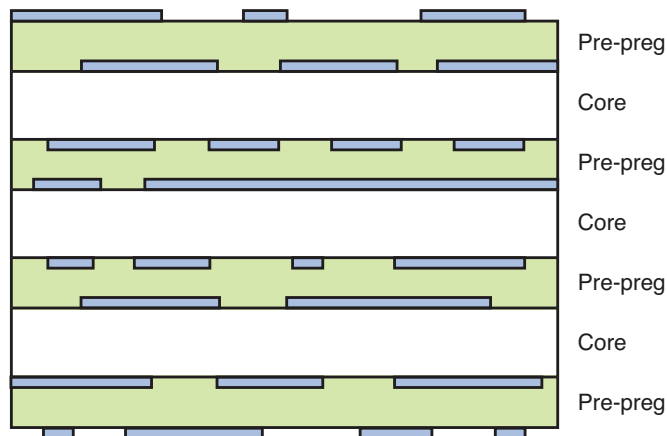


Figure 2.6: Stackup of an Eight-Layer Board

This stackup contains three layers of core material and four layers of pre-preg. But the board has eight layers because there are eight surfaces containing copper.

As with double-sided boards, vias pass electricity from layer to layer. For a multisided board, vias come in four types:

- *Through vias* connect the board's top and bottom sides. They don't come in contact with any of the internal layers.
- *Stub vias* run all the way through the board. They can connect one or both external layers to any of the internal layers.
- *Blind vias* connect an external layer to an internal layer but don't run through the entire board. They can be seen from only one side.
- *Buried vias* connect internal layers but don't touch any external layer. They can't be seen from either side of the board.

Figure 2.7 presents an example of each of these vias in a multilayer board. It's worth noting that, while many fabrication houses support different via types, the prices differ according to difficulty.

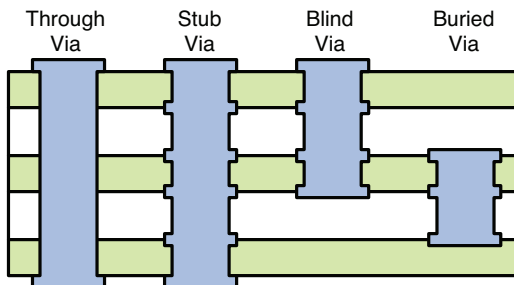


Figure 2.7: Vias in a Multilayer Board

### 2.1.5 Gerber and Excellon Files

Circuit board designers usually don't get involved in the fabrication process. Instead, our job is to give a fabrication facility the information it needs to construct our designs. This information consists of three parts:

- **Locations**—For each feature (such as a pad or a via), we need to identify its layer and (x, y) coordinates.
- **Dimensions**—We need to provide geometric data including the width of the traces, the spacing of the pads, and the thickness of the copper.
- **Drill holes**—For each hole to be drilled, we need to identify the hole's position and the drill diameter.

Designers provide these properties to the fabrication facility using computer files. Multiple formats exist to describe circuit boards, but at the time of this writing, the most commonly accepted format is RS-274X, which is commonly referred to as the extended Gerber format or just the Gerber format. Appendix B, “The Gerber File Format,” discusses the Gerber format in detail.

Gerber files control how the photolithographic tools should be used to pattern copper on the board (see Figure 2.4). To tell the fabrication facility about the board’s holes, another type of file is needed. Files containing drill information are called Excellon files.

The ultimate goal of PCB design is to create the files that accurately describe the circuit board. The next section will explain how EAGLE generates these important files automatically.

## 2.2 Overview of Circuit Design with EAGLE

Now that you have a basic understanding of how circuit boards are constructed, you’re ready to see how EAGLE makes it possible to design them. This section provides a brief overview of the design process, from creating a project to generating the design files. The next chapter walks through the process of designing a simple circuit.

### 2.2.1 Creating a Project

EAGLE stores information about a circuit in computer files, and all the files for a specific circuit are stored in a directory called a project. In essence, a project is a special type of folder that stores all the files related to a single circuit design.

By default, EAGLE places each new project directory inside a folder called eagle, which is located in your Documents folder. On my Windows 7 system, the eagle directory can be found at the following path:

```
C:\Users\Matt\Documents\eagle
```

When you launch EAGLE, the Control Panel dialog box appears. The last entry in the dialog’s vertical list is called Projects. Figure 2.8 presents the initial projects that ship with EAGLE 6.5.

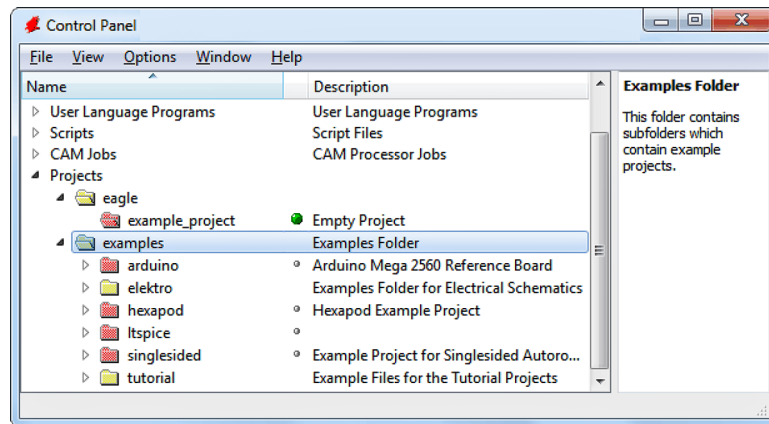


Figure 2.8: Initial Contents of EAGLE's Projects Directory

The red folders represent projects. Inside each project folder, you'll find files related to a circuit design. In the figure, the folders named *arduino*, *hexapod*, and *singlesided* are project folders.

The yellow folder icons correspond to regular folders. These contain project folders and other regular folders. In the figure, the folders named *eagle* and *examples* are regular folders.

If you look at the list of projects, you see that each has a circle between the project's name and description. A small circle means the project is closed. If you right-click a closed project and select *Open Project*, EAGLE reads the project's design files into memory and launches editors for its schematic (\*.sch) and board (\*.brd) files. When a project is open, its circle becomes large. Only one project can be open at a time.

Creating a new project is easy. Go to the main menu in EAGLE's Control Panel and select *File > New > Project*. Chapter 3, "Designing a Simple Circuit," will explain this further in its presentation of an initial circuit design.

## 2.2.2 Creating a Schematic Design

After you create a project, the first design file you need is a schematic. A schematic is a high-level description of a circuit's structure. It provides information about the circuit's components, the components' values (resistance, capacitance, and so on) and the manner in which they're connected.

Schematics do not provide information about the circuit's physical characteristics. That is, the schematic doesn't say anything about the board's dimensions or where the components are actually placed on the board.

**NOTE** Different versions of EAGLE have different limits for circuit board size, but there are never any limitations for schematic sizes.

Schematics are stored inside projects as \*.sch files. Its format is based on the eXtensible Markup Language (XML), which means it can be read and modified with any XML editing tool. But it's easier to access a schematic's content by executing scripts and user language programs (ULPs). These topics are explored in Chapters 10 through 13.

When you open a project containing \*.sch files or double-click a \*.sch file in the main window, the schematic editor appears. This lets you choose circuit components and connect them together to form the design. Figure 2.9 shows what an empty schematic editor looks like.

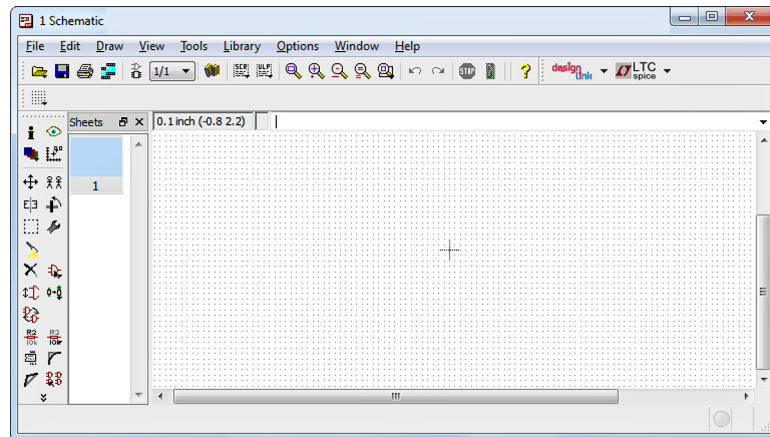


Figure 2.9: The Schematic Editor

Later chapters discuss the schematic design process in detail and the many items in the editor's toolbars and menus. For now, it's important to understand the four steps of using the schematic editor:

1. Insert components from EAGLE's library into the schematic.
2. Arrange components and set their names/values.
3. Draw connections between components.
4. Create a board file (\*.brd) from the schematic.

To add a component to the schematic, you need to look through EAGLE's set of component libraries. Thankfully, this set of libraries is *huge*, so no matter what integrated circuit you're looking for, you'll probably find it or something similar. Figure 2.10 presents a small portion of the list of available components. In this case, the component to be added is the MC9S12XF512 integrated circuit from Freescale Semiconductor.

The vastness of the EAGLE libraries can be daunting, particularly if you're searching for a component without knowing the manufacturer. Thankfully, pressing Ctrl-F opens a search bar at the bottom of the window. This makes it possible to find a component using some or all the characters in its name.