
Application Note: PCB Design with EAGLE



Kenji Aono ¹
Department of Electrical & Computer Engineering,
Michigan State University,
East Lansing,
MI 48824-1226,
United States of America
aonokenj@egr.msu.edu

April 4, 2011

¹The author is an undergraduate student at Michigan State University

The EAGLE Layout Editor is copyright protected software,
©1988—2011 CadSoft Computer GmbH, all rights reserved worldwide

The acronyms

EAGLE - *Easily Applicable Graphical Layout Editor*

EAGLE - *Einfach Anzuwendender Grafischer Layout Editor*

are trademarks of CadSoft Computer GmbH

The EAGLE logo® is a registered trademark of CadSoft Computer[†]

Windows® is a registered trademark of Microsoft Corporation

Linux® is a registered trademark of Linus Torvalds

Copyright © 2011 Kenji Aono, created using L^AT_EX

Attribution-NonCommercial 3.0 Unported (CC BY-NC 3.0)

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

With the understanding that:

Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

Your fair dealing or fair use rights, or other applicable copyright expectations and limitations;

The author's moral rights;

Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to [this web page](#).

[†]CadSoft Computer in the USA is a trading division of Newark Corporation, a Premier Farnell company

Abstract

Background

For professional, students, and hobbyists alike, a **Printed Circuit Board (PCB)** is an important component of creating useful electrical projects. The quality and intricacy of these boards can vary from cheap, single layer boards used as a replacement to a protoboard to boards on-board spacecraft that must guarantee proper operation of mission critical components in the harshest of environments. In the electronics industry, as parts continue to get smaller, surface mount packaging has become the preferred method of delivering components to customers. The proliferation of surface mount devices has increased the need for creating quality **PCBs** for prototyping and validating designs as well as final packaging of a product.

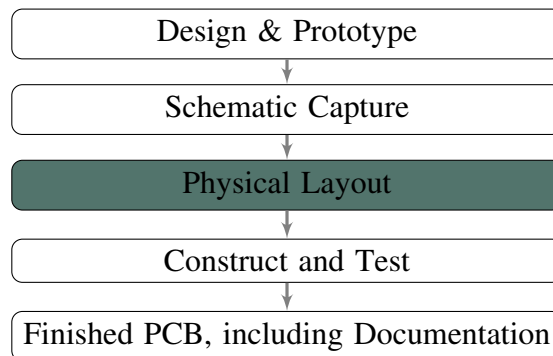
Topics Covered

The goal of this application note is to introduce users to one method of completing the "Physical Layout" step of the **PCB** design flow.

† ‡ There are several software packages that students have access to for **PCB** layout editing, such as; Cadence© Allegro© PCB Design, EAGLE©, and Sonnet© Suites™. For simple **PCB** designs, the full versions provide more than the required tools to complete the job,

which can lead to obfuscating the process for beginners. Based on the low cost of obtaining, and the relatively short learning curve, this application note will focus on the offerings of **Easily Applicable Graphical Layout Editor (EAGLE)**. Some of the topics covered with regards to **EAGLE** are; Installation , Libraries, Schematic Design, Layout, Outputting, and Other Notes of Interest.

*Please refer to the **Contents** for a more detailed list of topics...*



Keywords

Autorouter, EAGLE, Gerber, LaTeX, Layout, Library, Package, Pads, PCB, Pins, Ratsnest, ECE, 480, Design Team 5, Kenji, Aono.

*Please refer to the **Glossary** and **Index** for more keywords...*

† Adapted from ECE Shop document: How to Create a Printed Circuit Board (PCB)[[PDF Document](#)]

‡ For completing the "Construct and Test" step, see [Joshua Myer's Application Note](#).

Contents

Abstract	i
Background	i
Topics Covered	i
Keywords	i
1 Introduction: Setting Up EAGLE	1
1.1 Download	1
1.2 Install	2
1.2.1 Windows	2
1.2.2 Linux	3
1.2.3 Mac	3
1.3 Initial Startup	3
2 Library Creation	6
2.1 The Symbol	7
2.1.1 The Symbol Editing Mode Toolbar	8
2.1.2 Creating the Symbol	11
2.2 The Package	12
2.2.1 Creating Custom Packages	12
2.3 The Device	14
2.4 Some Tips	15
3 Schematic Design	17
3.1 Starting a Project	17
3.2 Placing Parts	20
3.3 Finishing Up	21
4 Board Layout	22
4.1 The Board Editor	22
4.2 Routing - The Auto Router	24
4.2.1 Manual Routing	24
4.2.2 Setting Up a Simple Auto Route	25
4.2.3 Optimizations	26

4.3	Manual Traces & Shapes	27
5	Output the Design	28
5.1	Chemical Etching	28
5.1.1	Editing in Inkscape	29
5.2	Gerber	30
5.3	Scripts	31
6	Other Notes of Interest	32
6.1	User Languages	32
6.2	Command Line	32
6.3	Getting Help	33
	Glossary	34
	Index	38

Chapter 1

Introduction: Setting Up EAGLE

EAGLE provides users with a relatively easy to learn and use schematic and layout editor, while allowing for advanced users to implement more complex designs as well. But before one can begin creating designs in **EAGLE**, they will have to gain access to a working installation.

If you already have access to a working copy of EAGLE skip ahead to 1.3

EAGLE comes in several license types, from the full-blown professional version that costs over USD 1000¹ to the freeware version that is ...free! Before downloading the software, users should refer to the [pricing page](#) at www.cadsoft.de. If one is attempting a 5cm x 5cm **PCB** design and only requires two **Signal** layers, it is unnecessary to spend money purchasing the professional version of the software. Also of interest is that a non-profit license that unlocks the potentials of the Standard Edition are available for a significantly reduced price, which may be of interest for students. Another edition of **EAGLE** is the Freemium version, which is another free version, which offers more features than the Standard Edition, but is only available for a limited time.² For users in the United States of America and Canada, there is a 30-day money back guarantee from CadSoft on purchases of **EAGLE** licenses. If at any time a user wishes to upgrade or extend a license, they can start **EAGLE** and navigate to *Help* → *EAGLE License* to change between license types and perform updates. This application note will assume that users are installing and using the free version of **EAGLE**. Please note that installation and use in a network situation will require more attention and details not covered in this application note, please refer to the **EAGLE** manual supplied with your particular installation files.

1.1 Download

Now that the user has decided on which license they are going to use, it is time to obtain a copy of the installation files. If you you have the installation files on physical media, skip ahead to 1.2.

For everyone else, you will need to download the latest version of **EAGLE** from <http://www.cadsoft.de/download.htm>. Be sure to download the correct files for the expected oper-

¹The price for a 1-User Professional license is USD 1494.-- as of April 2, 2011

²See: [CadSoft EAGLE Forums](#) for more information.

ating system. Available operating systems include Windows, Linux, and Mac. For users without Internet, it may be possible to purchase physical media containing the installation files, please contact one of the [authorized dealers](#).

1.2 Install

```
Before we begin, let the user be warned that the following
instructions for installation are intended for new
installations. Users who are upgrading an existing
installation should refer to the documentation provided in the
downloads(of the new installation files) for pertinent update
information. It is also advised that users create a backup of
existing libraries, projects, and any customizations. It is
also advised that users keep files required for reverting to an
old installation in the event that earlier version files are
not compatible with newer versions.
```

Install the software for your preferred operating system: **Windows, Linux, Mac**.

1.2.1 Windows

This will most likely be the most popular operating system for students at Michigan State University, most lab computer will be using Windows, as well as the laptops supplied to students enrolled in the Senior Design Capstone. For users who have install programs in a Windows environment before, the installation process should be very easy, no hidden junkware or obscure settings to worry about. Running the downloaded eagle-win5.XX.X.exe file will bring the *WinZip Self-Extractor* which will unpack the installation files to a temporary folder before beginning the setup. Be sure to read over the instructions at each step so you don't make any mistakes in installation, such as the directory to install to. Once the program is installed, the user will be prompted to decide how to license EAGLE. For this application note, we will *Run as Freeware*, which can later be changed using the steps mentioned **earlier**. Execution of EAGLE is done by launching the application, typically installed in Program Files.

1.2.2 Linux

The download for Linux users is a simple self-extracting shell script, which can also be deflated manually if a user feels compelled to. For normal installation, the downloaded shell script (in .run format) should be marked as executable using the following (be sure to change the code to match the version of EAGLE you downloaded.)

```
chmod u+x ./eagle-5.11.0.run
```

Running the script will extract files from the archive, after which installation will follow. Again, this will be a similar process as other installs, please be sure to read the instructions at each step. If EAGLE was not installed as a superuser, it may be possible that the license files for the free version did not get copied, to fix this run the following sequence of commands from "*.../installation directory of EAGLE/bin*"

```
chmod a+w freeware.key
cp freeware.key eagle.key
chmod ug-w *.key
```

Execution of EAGLE is done by executing *bin/eagle* in the installation directory.

1.2.3 Mac

This author does not have access to a computer running a Mac system, and will omit specifics on how to install EAGLE on a Mac. From a quick inspection of the download files, it appears that a Mac user will need to unpack a .zip file which will contain a Mac installer compatible with PowerPC and Intel processors alike. Execution of EAGLE is done through the *Finder's Applications* on a Mac.

1.3 Initial Startup

Once the program is installed, go ahead and execute it as mentioned in the respective section earlier. You will be presented with the EAGLE Control Panel, which should look similar to 1.1. From here, users will have access to predefined libraries, example projects, and several other features that will be explained throughout this document. If at any time a user needs guidance, be sure to check the Help files and documentation included with EAGLE, or go on the Internet

to check forums dedicated to helping users.³ A side note on default languages in case any users

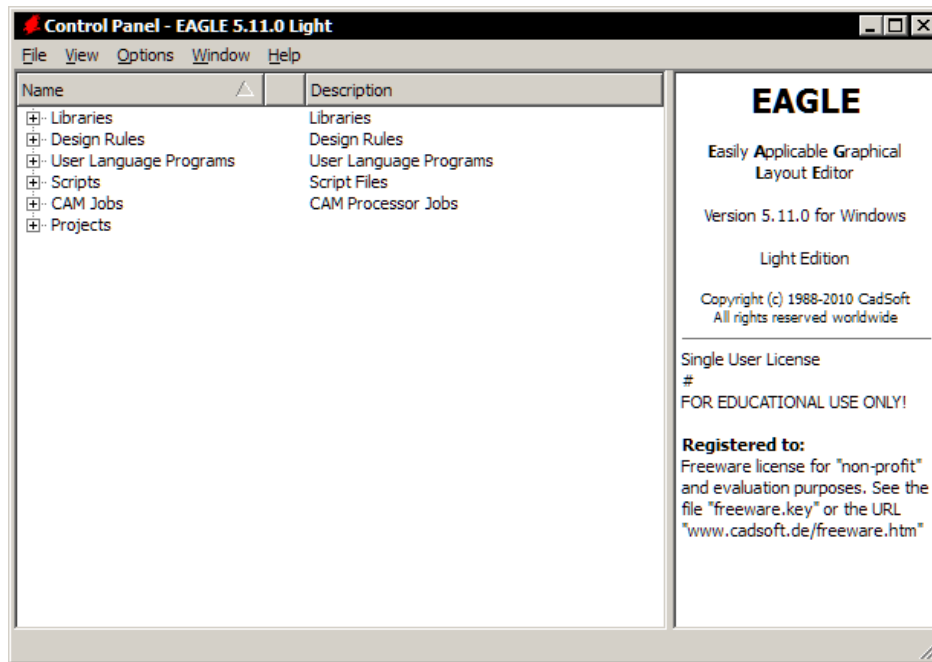


Figure 1.1: The EAGLE Control Panel running in a Windows environment.

experience problems; the default EAGLE language depends on the language setup of a user's operating system. If the systems *LANG* variable is set to an English speaking region, the program will default to English, for other languages such as German, the language may default to German. If you would like to run the program in English run the following command.

Windows (from Command Prompt)

```
SET LANG=en_GB  
cd C:\Program Files\eagle-5.x.x  
start \bin\eagle.exe
```

Linux and Mac

```
LANG=en_US  
/home/user/eagle-5.x.x/bin/eagle
```

Now that we have EAGLE installed and working on our systems, we can begin working on our designs. As mentioned earlier, this application note is focused on the Physical Layout step of the PCB design flow. As such, we are assuming that users will have completed a design and

³Documentation is included in *.../installation directory of EAGLE/doc*", and the CadSoft Forum is available as a [Mailing List](#) or [Web Interface](#).

schematic to implement on a **PCB**. The first step in going from a schematic or PSPICE simulation is to recreate the design in **EAGLE** by adding parts and wiring them in the schematic editor. In order to do this, **EAGLE** must know that the components we wish to add, whether it be a simple through-hole resistor, or a more complicated 100 pin microcontroller, actually exist and what their dimensions are. **EAGLE** does this by looking up devices from it's library, which is discussed in the following chapter.

Chapter 2

Library Creation

EAGLE only knows of components that exist in its libraries, if it isn't contained in one of the many default libraries you will have to create the device yourself. For standard components and many typical components, **EAGLE** already supplies the **Device Set** so we don't have to spend time recreating devices from scratch every time we wish to use it. First things first, look through your design and list all of the components that you expect to use and look through the libraries that come prepackaged with **EAGLE**. If all of your parts exist then you can go ahead and skip to 3.¹ To look through the libraries, double-click on *Libraries* from 1.1 or click on the plus sign to the left of the label, this will open up several libraries that follow the naming convention of *.lbr and also have a short description of what they contain. Go ahead and double-click on one of the libraries, or click on the plus sign to the left of the label, you should have something that looks like 3.2.

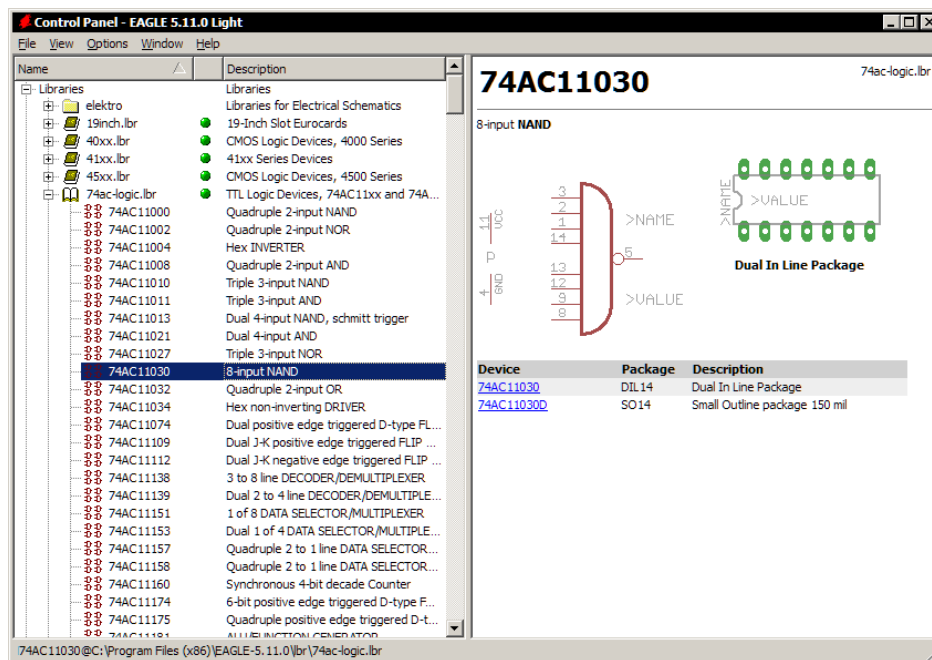


Figure 2.1: The EAGLE Control Panel viewing the 74AC11030 from the 74ac-logic.lbr library.

If the **Device Set** does not exist, try looking [online](#), if it doesn't exist there you'll have to make a custom **Device Set**, which is discussed in the following sections. To begin with, either right click

¹Be sure to double check the package dimensions defined in **EAGLE** to those that are given in the component's datasheet, or the dimensions that are expected based on measurements of the physical device.

and select *Open* on an existing library to simply add a part to it, or navigate to *File* → *New* → *Project* to bring up a new **Library Window**. It is from here that all changes to the library will be made.

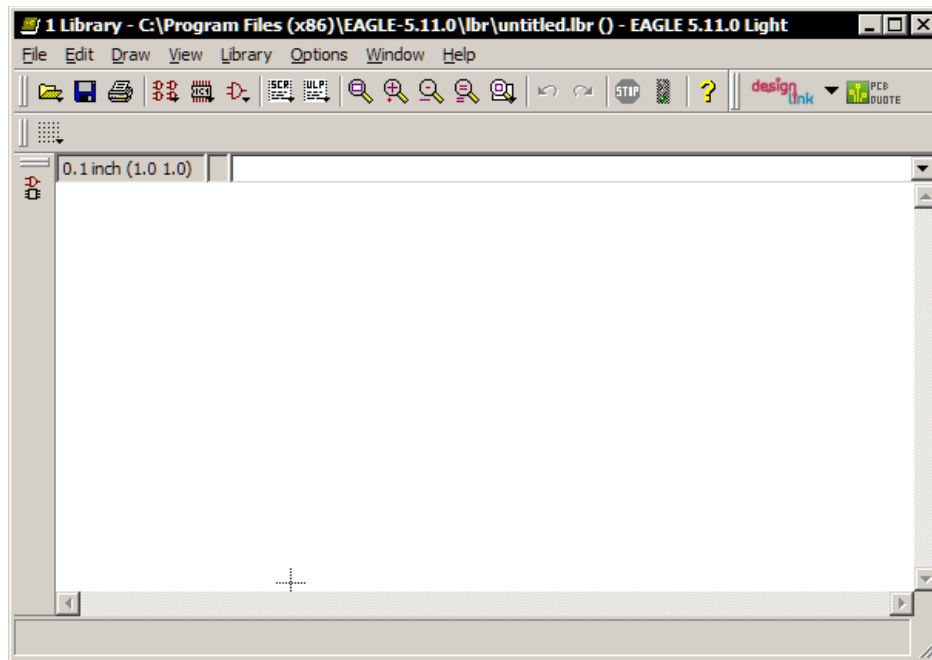



Figure 2.2: The Library Window without any devices loaded.

2.1 The Symbol

To begin a new symbol, either navigate to *Library* → *Symbol*, or click on the  w *Symbol* icon.² Either action will result in a popup window titled *Edit* to appear, at this time the list of available symbols should be empty. We can begin populating this list by typing a name into the *New:* field of the popup window and clicking OK. You will be prompted with a *Warning* about creating a new symbol, go ahead and say *Yes*. We should now return to the Library Window in Symbol editing mode, which will allow us to define a **Symbol** for a **Device** that may be placed into a schematic. You may notice that there is now a new toolbar in the Library Window, this contains the many tools that are required to make a symbol. Before continuing, lets take the time to look at this new toolbar.

²Please note that for a clean install of **EAGLE** that does not have any existing symbols, the little black arrow may not appear on the icon

2.1.1 The Symbol Editing Mode Toolbar

Please take the time to look through what each icon does now, it is a lot to take in at once, but it is important to understand what each icon looks like and what it does. Don't be intimidated by the number of icons presented, many of them are not exclusive to the Symbol editing mode, and will show up in future chapters. Taking the time to learn them now will make things easier on us later.

INFO

Clicking on this icon and selecting an object will display a popup window its properties such as; Name, Position, Angle, and others exclusive to each type of object. Some properties can also be altered through this popup window. This icon is equivalent to typing `INFO OBJECTNAME` in the command line.

SHOW

Using this tool will simply highlight an object, especially useful when trying to see where a **Net** is connected. The user can either activate this mode and click on an object, or can also enter an object's name in the command line and have the part highlighted for them. (An equivalent of searching for an object.) An example would be to highlight all pins that have been named using the default convention of P\$1, P\$2, etc. with the command `P$*`, that is, wildcard use of `*` and `?` are allowed.

DISPLAY

As the name suggests, this icon allows a user to choose what is going to be displayed. This is done by clicking on the icon and selecting which **Layer** to display. The window that appears from clicking on the icon also allows users to create new layers that do not have predefined usages. (This can also be done in the command line using `LAYER #FORNEWLAYER NAMEOFNEWLAYER.`) If a user right-clicks on the icon instead of the standard left-click, they can define an **Alias**.

MARK

Clicking on this icon, and then a location on the grid will create a relative distance marker in rectangular and polar coordinates, displayed between the distance from origin and command window. This is useful for measuring the actual dimensions of designs to ensure that they are as expected.



Figure 2.3: The new toolbar

To remove the relative coordinates, click on this icon followed by the traffic light icon in the *Actions* toolbar just below the menus.

 **MOVE**

Click on this icon, then single click on an object to highlight it for movement, the object should now move as your cursor does. Although you can click and drag an object, it is possible that you end up moving a component you did not intend to, therefore it is suggested that users click once to ensure that the correct part is highlighted before moving their cursor. While the object is highlighted, or while click and drag is active, right-clicking will rotate the component.

 **COPY**

This icon will allow users to copy components, albeit with a newly assigned name.

 **MIRROR**

This icon allows users to mirror components. Users can mirror a pin on a symbol horizontally, or even mirror top side wires to the bottom side of a **PCB**.

 **ROTATE**

This icon allows for more precise rotation of objects than the simple right-click option of the *MOVE* icon does.

 **GROUP**

This icon, when used correctly, can become a very powerful tool for making large scale changes. Users can either click and drag to use a rectangular selection, or create a polygon to select components. (Use left-click to define points on a polygon, and the right-click to close the polygon) After selecting components, users can click on any other command icon that supports group commands, such as *MOVE* or *CHANGE*, then right-click in the editing window and select *COMMAND: ROTATE*.

 **CHANGE**

Using this icon, users can change various properties of an object without having to navigate into the properties through *INFO*. An often useful combination is to use this with the *GROUP* command to make changes to multiple components.

 **CUT**

Please note that this command is not the same as cutting an object in a typical operating system.

The CUT icon will copy a group into the paste buffer, then allow a user to select a reference point and destination for pasting the group.



Also unlike a typical cut and paste situation, this will insert an object from the paste buffer into a design.



Click this icon, then click on a object to delete, also allows deletion of a group. For more advanced options, such as deleting an entire net, refer to the command line usage.



Click on this icon, then an object to give that object a name, although all objects are given a name, they are not always intuitive, and naming object yourself will allow for easier referencing later.



This icon allows users to round of wire joints.



This icon will allow users to insert a bend in a wire, taking a long stretch of wire and creating a new control point where selected.



Click on this icon, then click on the drawing area to draw lines, note that you can also select which layer you are drawing on by changing layers in the *Parameters* toolbar.



Similar usage as *WIRE*, except instead of lines you will be placing text. It is recommend that users turn on the *Always vector font* by navigating to *Options* → *User Interface* and clicking the checkbox under *Misc*. This option will ensure that texts are drawn as vectors instead of bitmaps, which allows for accurate scaling when outputting.



Click this icon to activate the circle drawing, the first click on the drawing area will define the circle's center, and the second click will dictate the circle's radius.



Similar to the *CIRCLE* command, but adds a third click that determines how much of the circle to draw.



This icon allows the drawing of a rectangle by defining two opposite corners. Although one can click and drag, it is recommended to click once to define one corner, then click again to define the opposite corner, this method will show a preview unlike the click and drag method.



Activate this icon, then left click on the drawing to choose points on a polygon. Unlike the *GROUP* polygon, right-clicking will not close the polygon, but instead change the type of bend to use. To close the polygon create a point that overlaps with the initial point.



Using this icon, users can place pins on a symbol that are used as inputs and outputs of a device. The names of pins can be changed using the *NAME* command, the direction or type of the pin can be changed using *INFO* or *CHANGE*.



2.1.2 Creating the Symbol

Now that the icons on the toolbar have been described, there isn't too much guidance to offer for users creating the symbol. Go ahead and draw out the symbol, typically using the rectangle (for ICs) or line (discrete components) drawing tools, be sure to add appropriate pins with the correct direction. The only tricky part to creating a symbol is making sure to name your pins correctly, done by using the *NAME* icon, clicking on a pin, and giving it a useful name, such as GND. The final touch is to add two text labels, ">NAME" and ">VALUE" that will allow **EAGLE** to assign names and values to your components. To do this, click on the *TEXT* icon, which will prompt for the input of the text label, type NAME and OK. Before placing the text at the top of your symbol, be sure to change the layer to layer #95 Names by selecting it from the drop down box in the *Parameters* toolbar. Do this for VALUE as well, but place it at the bottom of your symbol. Save your new symbol by saving the library.



For samples of good design practices, check out symbols from the included libraries of a default




EAGLE installation, these libraries contain nearly every type of electrical component one is likely to typically use.

2.2 The Package

Once you are done creating a symbol, it is time to create a package for the device. If you are using typical components, it is likely that the package you require already exists in another library, if so, it may be useful to copy that package into your own library. (If allowed by the creator of the library you intend to copy from.) To do this, keep the *Library Window* open and switch back to the *Control Panel*. Navigate to the library you want to copy from, like `rc1.lbr` which contains many surface mount packages for resistors, capacitors, and inductors. Note that the components with a  icon are devices, to copy only the package scroll down until you find the  icons. Right-click on the package and *C*opy to library, it is now in your library as well. Return to the *Library Window* and save to see the changes. Congratulations, you now have a package. Now if you weren't lucky enough to find the particular package you need, you'll have to read on to find out how to make your own.

2.2.1 Creating Custom Packages

First things first, look for documentation on the part you need a package for. Several companies will have datasheets with the dimensions of the parts, and many times the suggesting pad size and positions. If not, you'll have to make measurements on the actual component and decide for yourself how large the pads need to be and where to place them. Create a new package from the *Library Window* by clicking on  this icon. Insert a name for it as you did in the last section. The first thing you'll want to do is draw the outline of the physical component using something like the *WIRE* command. When trying to match the dimensions of your component, you may find that **EAGLE** keeps you from precisely defining the size of a wire by snapping to larger dimension. (In multiples of 0.05 inches in a default installation.) To alleviate this issue, one needs to define a custom grid size. Navigate to *View* → *G*rid or click on the  icon in the upper left hand corner of the *Library Window*. Change the *Size:* field to whatever suits your design needs, for small surface mount devices 0.1 mm may be a good choice. You can also change the *Alt:* field to 0.1mm and keep the *Size:* field as 1mm if you don't like fine grids. Then when you need to place objects with 0.1mm precision you simply hold down the `Alt` key before proceeding, releasing the `Alt` key will return your grid to whatever you have the *Size:* field set to.

The packages view will add three new icons to your toolbar, the  **PAD**,  **SMD**, and  **HOLE**. The pad icon will create a typical through-hole (**Hole**) for drilling through all layers with plating to allow for easy soldering. The SMD icon will allow you to place a pad by defining the width and height of said pad instead of manually placing it using the rectangle command. The hole icon is similar to the pad command except it is meant for mounting components that don't require an electrical connection so it doesn't supply the extra plating.

Again, the majority of the work comes from accurately recreating the dimensions given in the component's datasheet in the package drawing, resulting in something like 2.4 when done. There

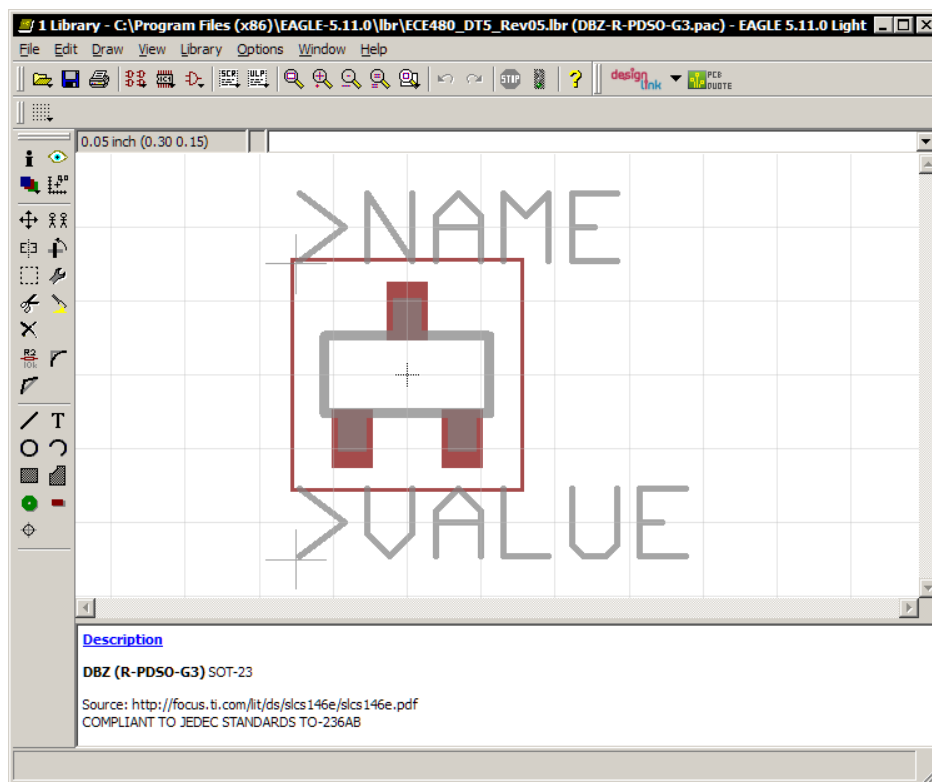



Figure 2.4: A completed package, including restricted areas

are however, a few intricacies to follow when doing this. Any symbols that you wish to place on a silkscreen (such as location of pin one) should be placed on layer 21 `tPlace`. Other information that is useful when designing, but not necessary for the final **PCB**, such as the outline of the IC, use the documentation layer 51 `tDocu`. Note that by replacing the prefix `t` with a `b` you can place this information on the bottom layer. As you did for the symbol, add the ">NAME" and ">VALUE" on layers 25 `tNames` and `tValues`, respectively. If you plan on using the Autorouter for easier

routing later, and a certain components requires that routing remains a set distance away from it, you may have to define restricted areas, please refer to the **EAGLE** manual for more information. (The default layers for defining restrictions are layers 39 through 43.)

The final step is to create a description of the package by running the command `DESCRIPTION` in the command line with the package open, or click on the Description link that should have appeared at the bottom of the *Library Window* when you started the new package. Formatting of the description uses simple HTML syntax, such as `TEXT` to bold TEXT, or `<p>` for starting a new paragraph or `
` for a carriage line. You should have something similar to 2.4. Save the library to reflect the changes you've made.

2.3 The Device


Now that the symbol and package have been created, it is time to create the device. This time you will click on the  icon to begin a new device. First things first, lets get the description out of the way, use the same procedure as described in the previous section to edit the device's description.

74AC11030

74ac-logic.lbr

8-input **NAND**

As a sample, in the case 3.2(Relevant section shown above), the device description was "8-input **NAND**". (Input as 8-input `NAND`)

Once you are done with creating the device, you should have a window that looks like the following screenshot. To get to this point, the first thing to do is add the symbol using , be sure to center the symbol on the origins cross hair at (0,0). Next, click on the large *New* button in the bottom right to open the "Create new package variant for SAMPLE" window from which you can select the appropriate package. The package should appear in the upper right hand window as well as be listed above the *New* button. However, you will notice that it has a yellow circle with an exclamation point warning you that something is not properly configured. Don't fret, you simply need to click the *Connect* button to open a new window that allows you to create a connection between your symbol and package. There should be three columns, from left to right:the Pin, Pad, and Connection. At this time the Connection column is empty, which lead to the warning displayed earlier. You need to select a Pin from the first column (this is why it is useful to name

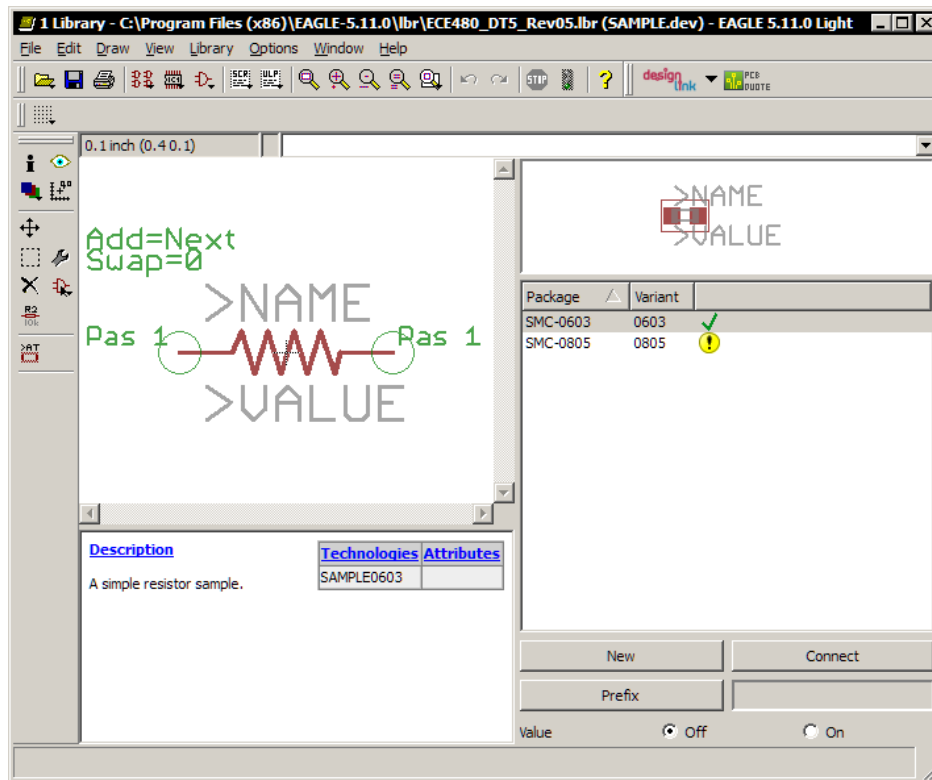


Figure 2.5: The device editor with a complete device.

your pins, otherwise you're left with pin names such as G\$1.1) and also click on the corresponding Pad from the second column. (Refer to your datasheets and double-check the connections first) The click the big Connect button just below the two columns. The two entries should disappear from the Pin and Pad list and be transferred to the connection column. Keep defining connections this way until the Pins and Pads list are empty. It is also possible that you have extra pads left over that don't have corresponding pins, this isn't a problem, continue. Click the OK button. Your yellow warning should now be replaced with a green check mark showing that everything is just peachy. Save your library now. Congratulations, you have created a new library from scratch!

2.4 Some Tips

Again, check online for available packages to copy, it will save a lot of time. Check for alternate names as well. For example, Texas Instruments offers a product that uses a package they refer to as SOT-23. If one checks the datasheet on that product, they will find that SOT-23 is also known as DBZ (R-PDSO-G3), which is a package that conforms to JEDEC standard TO-236AB.

Although that might be a little hard to follow, it simply means that the same package that Texas Instruments refers to as SOT-23 is the same as any other manufacturer's product that complies to JEDEC standard TO-236AB, so if you can find another package that conforms to that standard you don't need to create a package from scratch, you can just copy that one instead.

A tip on useful descriptions, if possible add whether your package conforms to standards such as JEDEC and if you are creating a resistor such as the SMC-0603, go ahead and put in the metric size as well SMC-1608 (METRIC) so others can easily reference your work. If you didn't get your dimensions directly from measurements made on a device, add a reference to the source of your information, something like: "Source: <http://focus.ti.com/lit/ds/slcs146e/slcs146e.pdf>" in the description will suffice.

When creating multiple packages for a single product, such as two different sized resistors, you don't need to create a whole new device, just add a new package as you did the initial one. If you get a warning *Package variant " already defined!* then you need to actually name the currently existing package. Right-click on it and select *Rename*, alternatively, click on the large *Prefix* button, and rename it using something practical like naming it as 0603 if that is the package size. You should notice that the *Variant* column next to the package no longer says "", but has an actual name, you can now add a new package.

Another tip for creating a library is to add a description to the entire library. Navigate to *Library* → *Description*, editing is done the same way as all other descriptions. Users should add an additional `<author>AUTHOR'S NAME</author>` to the bottom of the description, with an e-mail address in case others need to contact the people who made the library to request clarification or report errors.

The final tip is that the library does not always need to contain a real device. That is, you can create a device that has no real working parts, such as a through-hole or **Drill**. There are actually several items like these in the libraries installed by default.

Chapter 3

Schematic Design

This chapter will perhaps be the easiest for readers to relate to, if you're attended Michigan State University, you are probably very familiar with having to create schematics.

3.1 Starting a Project

Before we get into making the schematic, we will have to create a project. Up to this point we have installed **EAGLE 1** and created a custom library **2**, we are now about to start a project that will be used in the creation of the schematic³ and for the layout of the **PCB4**. If you still have the *Library Window* open, go ahead and close it, we should be done with it if you successfully created your device sets. Go to the *Control Panel*. As usual, there are several methods to doing the same thing, one could use a series of right-clicks and selecting from the resulting menus to accomplish the same task that will be described in the following paragraph.

First close your *Libraries* tab to give us more space to see other options in the *Control Panel* window, you should see a *Projects* entry at the bottom of the list, expand it. It should, by default, include a directory name `eagle` and `examples`, the `eagle` will save your projects to your My Documents/`eagle` folder in Windows, or the equivalent of My Documents in Linux and Mac. Navigate to *File* → *New* → *Project* to name and begin a new project. You should now have a new entry under the project tab, and next to its name should be a green dot that indicates the project is active. If it is a gray dot, the project is not active, right-click on the project and select *Open Project* to activate it. Once you have an active project navigate to *File* → *New* → *Schematic*. (Or you could right-click on the project *New* → *schematic*) This will open up a new window that should look familiar^{2.4}, with some new entries in the toolbar as well an extra toolbar that shows the available sheets. Since this application note is designed using the free version, the concept of sheets will not be covered, since it is assumed that simple designs being completed in the free version probably won't require the advance use of sheets.

The new toolbar will show several new icons, each of which is described in the following list.

**NET**

This will create a **Net**, which is useful as an interconnect for pins between devices.

**JUNCTION**

This icon will allow users to manually specify a junction, it is best practice to do this if you have wires that overlap and should be considered connected to each other instead of leaving it implied.

**LABEL**

Click on this icon and then click on a **Net**, you will be able to give it a name. As we saw earlier in the case of naming pins, names can come in handy later.

**ATTRIBUTE**

Using this icon and clicking on a component will allow the changing of any attribute, even overriding global attributes, this is a feature left to advanced users.


**ERC**

This icon will initiate an **Electrical Rule Check (ERC)** and create a list of errors.


**ERRORS**

This icon will show active errors and warnings in the circuit that require attention, if you know something is going to work and it keeps showing up as an error you can add an exception to not consider that particular event.

Once you are familiar with the differences in the toolbar, be sure to activate¹ the library you wish to use for your design, keep in mind that you may activate multiple libraries at once if needed.

¹The **USE** command can indicate which libraries are active, alternatively one can go to the *Control Panel* and right-click on libraries and activate them as needed, or use the  icon near the top of the *Schematic editor window* to define active libraries.

3.2 Placing Parts

You should still have a clear drawing space right now, click the  *ADD* icon or navigate to *Edit* → *Add* to open a new window. This window will contain a list of active libraries on the left hand side, and the right hand side will include two windows on the top and one window on the bottom. The top left window on the right hand side will show the currently selected device's symbol, the window to its right will be the package for the device. The bottom window on the right hand side contains information about the device, based on the description that has been defined for it. If you do not see these windows then be sure to check the *Preview* box near the bottom left corner of the *Add* window. If there is a library listed that you do not want to use you can either ignore it, or select it and press the *Drop* button. To add a part, select the part and package and click okay. The part should now be attached to your cursor, much as if you have clicked on a device after initiating a *MOVE* command. Place the part where you need it. The placement of parts on the schematic will not affect the placement on the final **PCB**, so don't worry too much about making the schematic look neat.

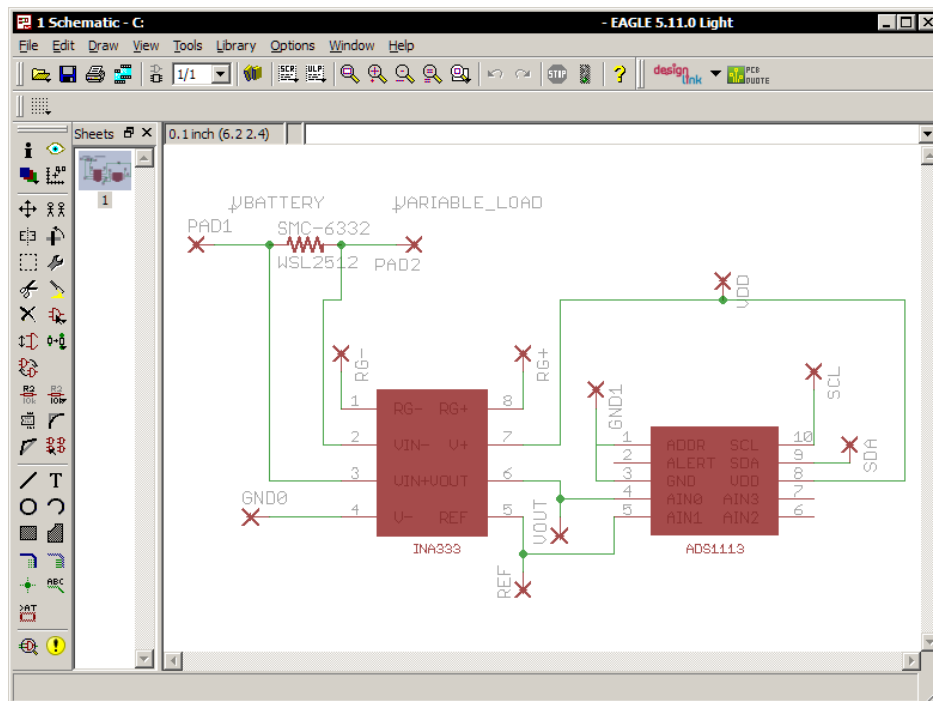







Figure 3.2: A sample schematic, nothing special.

3.3 Finishing Up

After the parts have been placed, you will need to make connections between pins and devices, the suggested tool for this is the  *NET* icon. Try to avoid criss-crossing the **Nets** when possible, if you have a **Net** split from one **Pin** into two other **Pins**, be sure to add a  *Junction* to the spot that they overlap. Go ahead and add a  *Label* to **Nets** if possible, such as GND or PWR. On a related note, some devices will not show their power or ground pins on the schematic, but will give them hidden pins that you cannot see on the schematic, but they will still need to be connected in the final **PCB** layout for the device to work. A note on connecting power and ground, there is a special library in a default **EAGLE** installation called `supply1.lbr` which contains the required devices for automatic wiring of supply **Signals**. Look through the default libraries, there are many useful devices in here that you may miss when skimming through the contents. If you know that a certain **Net** is going to have to be wider than the rest because it needs to carry more current, or if you have some **Nets** that are especially sensitive to interference and you need to ensure a certain amount of separation, it may be a good idea to specify custom **Net** classes. Go to *Edit* → *Net classes* which will open a new window with four columns. The first column is the name of the **Net** that you wish to edit, make sure this matches the label of the **Net** you wish to edit. The second column, the *Width* will determine how wide your traces will be on the layout. (Difference in width not shown in the schematic.) The third column, *Drill* defines how large the **Vias** connecting to this **Net** should be. The final column, *Clearance*, is especially interesting if you intend to use the Autorouter, this will make sure that traces that Autorouter generates stay at least this amount away from traces of this **Net**. Once you think you have completed the schematic, run the  *ERC* and see if the sentient machine agrees with your silly hypothesis. The descriptions in the  *Errors* should give you enough of a hint to go back and fix any mistakes in the schematic. Once all of that is cleared up, we are finally ready to begin the **PCB** layout process.

Chapter 4

Board Layout

Although it is possible to begin a layout without recreating the schematic in **EAGLE**, you will find that having a schematic will allow for easier routing. Creating a schematic first also ensures that the design that **EAGLE** is looking at is electrically equivalent to your designed schematic. Whether you skipped ahead to the chapter on board layout or got to this point after finishing the chapter on schematic, you're here now. Lets get this party started.

If creating a layout from scratch, skip ahead to 4.1, go along, skedaddle. Now, for those of you who were thorough enough to create a schematic, you are at a relative advantage when starting the layout design process. Open your completed design in the *Schematic Editor* and navigate to *File* → *Switch to board*, you will be prompted by **EAGLE** that the board `SCHEMATICNAME.brd` does not exist, and should one be created from the schematic. Click *Yes*. You should be presented with a new window, similar to Figure 4.1 which contains all of the components from your schematic in the lower left hand corner. There are also yellow lines all over the layout, these lines indicate which pins on the layout need to be electrically connected, this is the one of the advantages of creating a schematic first. The use of these lines will allow for Autorouter to know where to create traces, as well offer a reminder to the designer that there are connections that still require traces to complete, when manually creating routes.

4.1 The Board Editor

Now that the *Board Window* is active, lets take a minute to check out the new toolbar, please note that icons with similar functionality as **the symbol editing toolbar** are not covered in this section. For those users who have opted to begin the layout process without first creating the schematic, the devices for your design will need to be added (Use the `ADD` command) using the tools described in **the symbol editing toolbar**, and until the nets for the parts are defined (as a schematic would have done), the Autorouter will not be of any use, as it will not know where to make the connecting traces.¹

¹This point is reiterated, for the Autorouter is one of the most useful tools in creating a **PCB** in **EAGLE**. Please, make the schematic first unless you really know what you are doing because you can easily end up with **Forward&Back Annotation** errors otherwise.

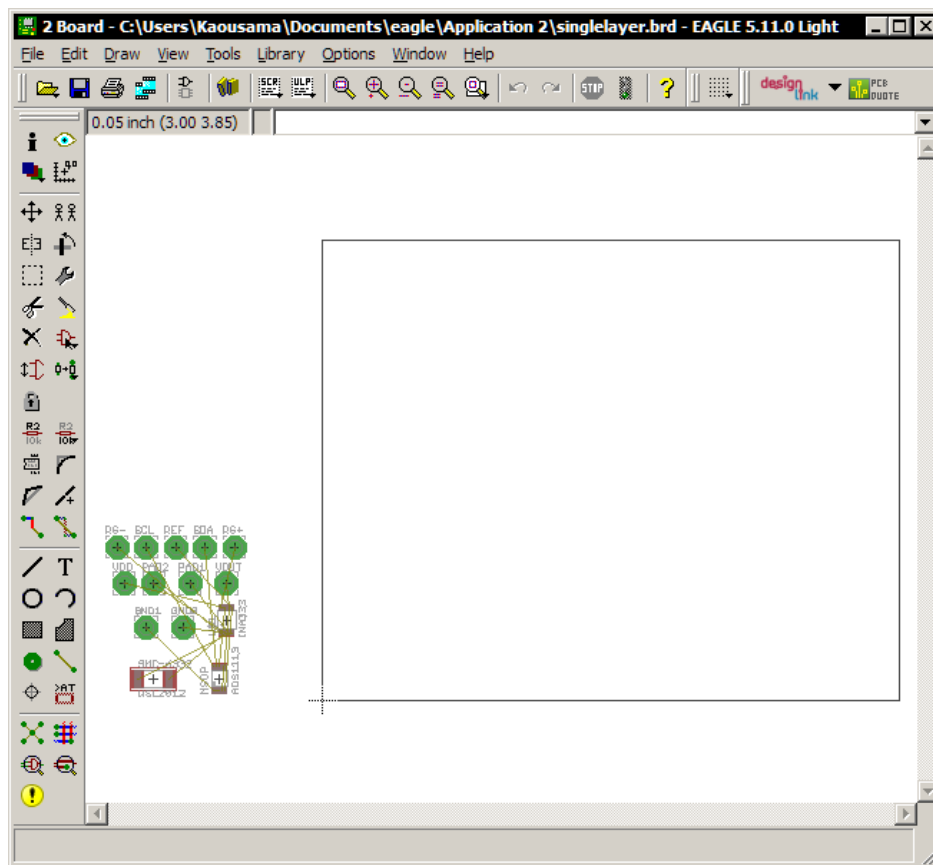


Figure 4.1: The layout editor, with schematic design imported

Here is a list of icons with a short description of their use from the layout toolbar shown in the bottom left of Figure 4.1. For icons that are shared throughout the **EAGLE** editor, refer back to the **symbol editing toolbar** and **schematic toolbar**.

LOCK

This icon will lock a component's position and angle in a drawing to prevent the accidental moving of the part. Using **Shift** before clicking on the part will unlock the component.

ROUTE

Click on this icon, then you can manually change **Airwires** into **Wires**. A right-click will change the type of bend to draw, and left-clicking while holding **Shift** will create a via.

RIPUP

This option is similar to **DELETE**, except it is intended for use with **Wires** and will revert them to an **glsplairwire**.

 **RATSNEST**

This icon will recalculate the shortest path for unrouted connections (also known as **Airwires**), and can also be used to fill a polygon.


 **AUTOROUTER**

The icon that will greatly simplify the task of routing the layout and connecting devices, this will invoke the Autorouter which will connect **Airwires** where possible.

 **DRC**

Using this icon will start the **Design Rule Check (DRC)**, which checks the layout for errors against convention or a manufacturer's specific design rules to ensure that the board can be properly fabricated.

4.2 Routing - The Auto Router

Although Autorouter is very useful in quickly laying out traces, it is not perfect, and should never be the only tool used by a designer. To find the optimal placement of traces, one will find that the Autorouter will need to be re-run after some parts have been added. Before it can run a second time, there need to be some **Airwires** for it to work with, so one will need to "erase" the traces that need to be redone. If the intent is to redo all traces, simply use the command `RIPUP` then click the  *Traffic Light* to revert all **Wires** to their respective **Airwires**, it is also a good idea to re-run the **Ratsnest** when devices have been moved, as the shortest distance **Airwire** may also change. The following sub sections will describe in more detail on how to achieve a good layout for a **single sided pcb**, for multi layered boards, many of the concepts will carry over; users will need to be mindful of changing the layers between edits. Autorouter will also work in multi layered situations without any trouble with a few adjustments in settings.

4.2.1 Manual Routing

The laying out of the devices on the board is done manually, you will need to use the `MOVE` command to place the parts that were initially placed to in the lower left hand corner off of the board.²

²The large empty rectangle to the right of the bunch of parts is the outline in the #20 Dimensions layer for the **PCB**, these dimensions can be reduced if you are not going to utilize the entire board by using the `MOVE` command and clicking on each side of the outline and moving it.

Place each part as you would like it to appear on your final board, if you do not have specific requirements, a tip is to set the part with the most connections (like a microcontroller) in the center of your drawing, and then place the supporting devices around it. When placing the parts, try to reduce the number of unrouted lines (shown as yellow lines) that cross over each other. Chances are, that each time an unrouted line is shown crossing over another, the shortest path to connect the devices is going to require a crossing of traces, which requires either changing layers, or in the single layer case it can lead to paths that cannot be routed. When you are placing the devices to minimize the number of criss-crossing unrouted lines, it may be useful to rotate a device, this has in many cases solve a few of the overlapping **Signals**. Another tip is to provide more connections to power and ground if possible, so a single ground trace does not need to extend across the entire board.³

Before going crazy with the Autorouter, check your design for sections that require special attention. As mentioned earlier, one may rename a **Net** and give it a special class definition. In this way, a **Net** that is designated as a power line can automatically be set to a bigger width, with a larger clearance.⁴ For these cases, Autorouter will know to make the traces wider and to stay the prescribed distance away from those traces. Traces such as 4.2, that are symmetrical and not necessarily shortest distance, a design choice that is intended to minimize offsets in resistance, will not be created automatically by Autorouter. Use the `ROUTE` command to make these manually, to make symmetrical traces use the `MIRROR` command and change the layers to match if needed.

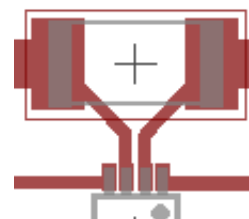



Figure 4.2: An example that requires manual routing.

4.2.2 Setting Up a Simple Auto Route

Now the exciting part, running the  Autorouter. Clicking the icon will bring up the Autorouter settings window. For simple designs, you will not need to worry about the other tabs besides the *General* tab. Here you will want to set the *1 Top*'s preferred direction to `*`, and the *16 Bottom*'s preferred direction to `N/A` for single layered designs. Go ahead and click `OK`. In a few short moments the Autorouter should be complete, in the bottom left of the *Layout Window*, the status bar


³This is easily done if using multi layers, simply dedicate a large portion of one of the layers to ground, however, note that a **Via** will add a resistance to ground, to minimize the resistance, add as many **Vias** as possible. (Resistances in parallel will divide in their total impedance)

⁴Refer back to **Schematic Design: Finishing Up** for details.

should tell you whether the Autorouter was able to connect 100% of the **Signals**. If not, use the RIPUP command and remove all of the traces and start over. If you have manual traces that need to be preserved, click on the RIPUP icon and remove all but the manual traces by clicking on them.

Before removing the traces, visually inspect the drawing and try to dissect where the bottleneck might be. If you see that a two devices are too close to allow a trace to go between them, it may be useful to increase their separation so the Autorouter can use that space, these are decisions that you, as the designer, will have to make. If the Autorouter still cannot complete, try decreasing the *Routing Grid* in the same window that you set the preferred directions in. This will decrease the grid size that Autorouter uses to try and find optimal placement, if possible, do this from a design that does not have any traces to allow the program to optimize all paths. Although this will increase the chances that a path is found, it will also greatly increase the computation time that Autorouter takes to complete. To find the best layouts, you will likely need to run Autotracer and RIPUP many times and change device placement or orientation between each run. If only a slight change needs to be made, you can use the RIPUP and remove a single trace and manually place the route.

4.2.3 Optimizations

To create a copper well, all one has to do is begin by drawing a polygon around the entire design area. Use the polygon icon and click in the upper left hand of the board design, then go clockwise to the top right and click once, then to the bottom right and click once. Finally, go to the bottom left hand corner and **double-click**, this will close the polygon and make a clean trace that outlines the board. Now simply click on the  RATSNEST icon, which will fill the polygon. The fill will be on the same layer as you used to draw the polygon, and it will automatically remain the set clearance distance away from other traces on the same layer. This feature can be used with named **Nets**, to create a ground plane on a single plane in a few simple steps, just be sure to give both the ground pin and polygon the same ">NAME" before using the RATSNEST to fill the polygon.

When dealing with multiple layers, one may wish to choose the *1 Top*'s preferred direction to -, and the *16 Bottom*'s preferred direction to | in a Manhattan routing scheme. The *Autorouter Setup* window also has several tabs besides the *General* that allow users to define how the Autorouter will choose a particular trace over another possible routing choice. These are settings left to advanced users, and those interested should refer to the forums for additional guidance.

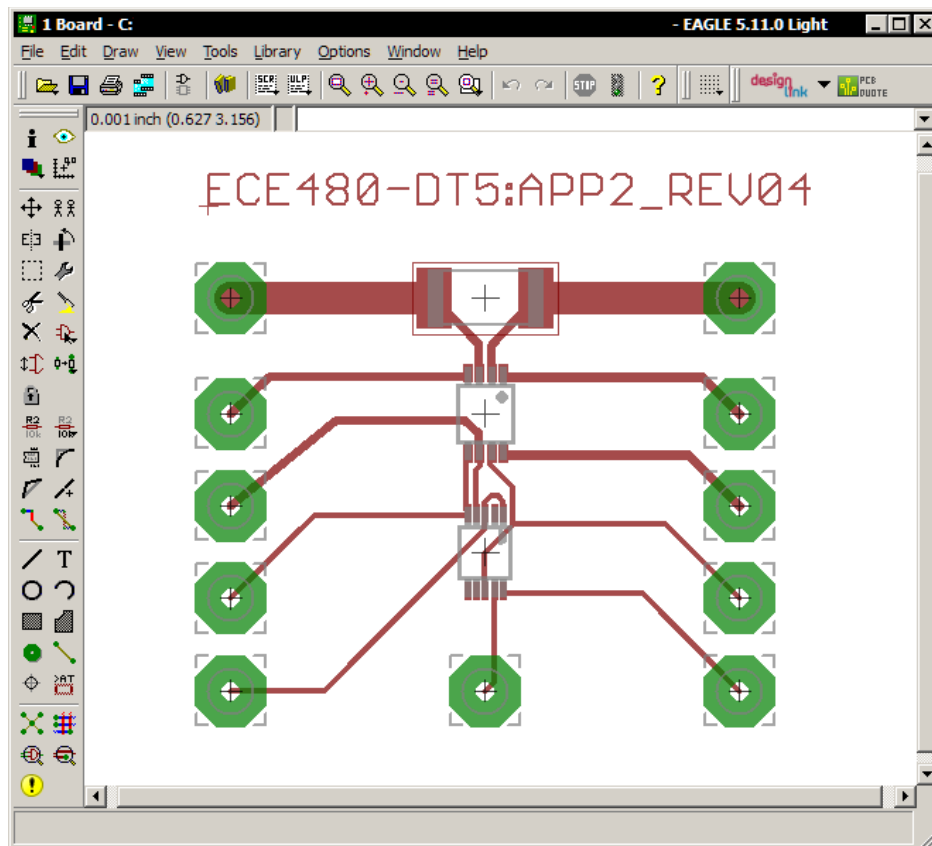


Figure 4.3: A sample layout with completed routing

4.3 Manual Traces & Shapes

As long as the layers and traces do not overlap with traces that will be carry critical **Signals**, a user can use these layers to create drawings on the **PCB**. Although some fabrication companies will require additional payment for such features, it does not require additional techniques beyond regular fabrication methods. That is, you could easily create these logos if you use a copper layer to do it, if you intend to do it on silkscreen it should be just as easy to get these shapes on your board. These layers can contain wires, rectangles, text, and anything else, just avoid the use of electrical connections so you don't make an inadvertent connection that messes up the integrity of the original circuit. A possible use for creating manual traces that don't contribute to the electrical performance of the circuit is to label what the board's intended use is, or perhaps the revision number if creating multiple prototypes.⁵

⁵See **User Languages** section for importing custom logos.

Chapter 5

Output the Design

Once the layout design is complete, it is likely that one will require an output in a format that is not the **EAGLE** default `.BRD` format. This application note is intended for use in conjunction with [Joshua Myer's Application Note](#) that informs users on how to conduct rapid, cost-effective chemical etching. As such, this application note will focus on outputting a proper mask to a transparency. For projects requiring a Gerber output, see [5.2](#).

5.1 Chemical Etching

For fabrication using chemical etching, we will require a mask that will be used to expose the pre-etched **Core** to ultraviolet radiation that transfers our layout design to the top layer. A typical method for accomplishing this is to use a printer and print our design onto a transparency. (Please be sure to use the correct settings on your printer, with the correct type of transparency, otherwise you could seriously damage the printer.) The first step is to make sure your board passes **DRC** and to double check the schematic to your designs to make sure that there are no missing or extra components. For chemical etching of a single layer **Core** you will only need layers 1 Top and 17 Pads, or any other layers that designate what copper should not be etched away. For example, if you have a logo on 200 SomeLayer then you should also activate that layer to be visible, or if you have text that you need to display, be sure that is active. Double-check the board with the required layers visible, check to make sure that there is proper clearance between traces and layers, when this is printed to the mask the layers will effectively be flattened into one layer, so if the layers aren't supposed to touch and they're overlapping, you will need to make adjustments.

With the board ready to be printed, go to *File* → *Print* to open the print setup. Set the *Printer:* to Print to File (PDF) and the *Paper:* size to Letter, now check the *Mirror*, *Black*, and *Solid* check boxes and uncheck the rest. Set the borders to something like 10mm so the board still prints, but you aren't wasting paper. Finally, be sure to check that the *Scale factor:* is set to 1 (one). Make note of where the *Output file:* is pointing to, change it if needed, and rename it to something you can remember, then click OK to print the layout to a `.pdf` file. If you want to save printing cost, you can save multiple layouts to separate `.pdf` files as per the method described above, and refer to [5.1.1](#) for how to merge these files into a single page. This can allow you to print several board

designs on a single page, if you do not have access to Inkscape or other similar software, **EAGLE** allows you to choose an *Alignment*: when printing, so you align each board layout to a different section of the transparency, and just print on the same piece of transparency multiple times.

Once you have the .pdf file with your design ready, it may be quicker to print directly to a local printer without using .pdf, however, it is likely that the printers in the ECE labs cannot supply the high printing quality required for micron level layout prints. If you are using large traces for parts large enough to fit in protoboards it may be quicker to simply output the layout as a .jpeg image, open it in paint and invert the colors and print, the decrease in quality from taking these quick steps isn't going to impact yours designs as much. For smaller layouts, be sure to inspect the .pdf output, try zooming in as far as you can to make sure that there are no jagged edges on the traces. These jagged edges are typical artifacts from converting to a raster image format such as .jpeg. (Try zooming in on **5.1** to see that a vector image does not have this problem.) As I mentioned in the quick and dirty method for printing layouts that use large parts, you need to invert the colors. That's right, invert the colors, think about it. In chemical etching, our mask will be used in the exposure to ultraviolet light step, which will imprint our layout onto the negative (or positive) film that we will have placed on the un-etched **PCB**. The exposed portion of the film will become resilient to the chemical used in the etching process, which means that as the portions of the film that were not exposed to the ultraviolet light deteriorate, and leave the copper exposed to the etching chemicals, the exposed portion will remain intact, as will the copper underneath it. This is exactly what we want, the copper to remain un-etched, so we have electrical traces. Anyway, for inverting colors of the simple .jpeg files you can use paint, but if you have a vector graphic .pdf file how do you go about inverting the colors?

5.1.1 Editing in Inkscape

Inkscape is available for [download](#) and is also available in portable formats for Windows that don't require a installation. Some of the advantages of this program are that it allows us to edit the .pdf files without degrading the vector image quality, it's free, and it is also available for Linux and Mac. Because of the abundance of tutorials on how to use the software, this application note will just give an overview on the steps one needs to take. For inverting the colors first open the .pdf and click OK when asked about the import settings, switching between the rough and fine settings will not impact our layouts. This is shown on the left side of the figure below. Next, draw a black rectangle around our board with sufficient margins around the edges to ensure we cover the entire layout. This would be the center portion of the figure. Then send that layer to the back, and select

the original layout design. Set the layout design's *fill* and *stroke* to white, as shown in the right hand side of the figure.

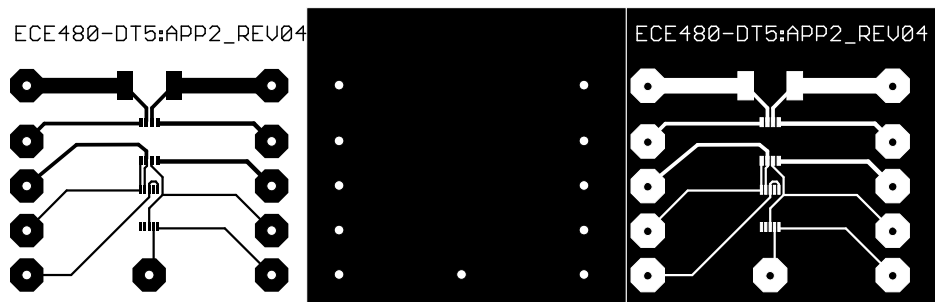



Figure 5.1: The Inkscape program, with the three steps described above.


For the purpose of the mask, you will only need the results as shown in the right hand side of the image. You can do this for all of the .pdf files you intend to print, and then combine them all into a single document in Inkscape. Once you have done that, save the document as a .pdf file, be sure to check that the document size remains as *Letter*, you don't want to have this scaling wrong. You can now take the .pdf generated by Inkscape to a professional printer service, and use a high dpi printer with the darkest black setting you can use to print to your transparency. By combining multiple designs to a single transparency, you can save money since you don't need to print off as many pages. Before continuing to fabrication, if possible, set the actual components on top of the transparency to check that you got the pad size and spacings correct, this quick step can save you several hours that you might spend fabricating a board that is obviously wrong.

5.2 Gerber

The preferred file format for machine milling at the ECE shop is the Gerber RS-274 format. To create a **Gerber Format** file, make sure that the design's .brd file is active in the *Board editor* and click on the  CAM Processor icon that is to the right of the print icon. (Or navigate to *File* → *CAM Processor*.) For users familiar with **Gerber Format** and the specifics required for proper output, this is where it is done. For those uninitiated, it is slightly beyond the scope of this application note to give instruction on how **Gerber Format** is useful, that is left as an exercise for the reader. However, it is likely that you've been told you need a **Gerber Format** file format and you only care to create one after spending all this time creating a design in **EAGLE**, don't fret, there's an easy way to accomplish this task. Download and unpack the [Eagle CAM Job File](#) from MSU's [ECE shop website](#) or simply use the `gerb274x.cam` file in the `cam` directory of your

EAGLE installation.¹ Once you have the .cam file ready, go back to the *CAM Processor* window we opened earlier and navigate to *File* → *Open* → *Job* and open the .cam file. You should notice that several settings have been defined, wasn't that easy? Now all that is left to do is click the *Process Job* button, **EAGLE** should process the files and place the **Gerber Format** files such as the **Wheel** in the same directory as the .brd file is saved in. If you are using Windows, the default location for new **EAGLE** projects is in `My Documents/eagle/YOUR APPLICATION` so go there to get the outputted files. Depending on the requirements of the group that is going to do the **PCB** fabrication, they will have different requirements for which files need to be submitted. For the ECE shop at MSU you will need to submit any files with the following extensions: .CMP², .SOL³, .BOL⁴, and .DRD⁵.

5.3 Scripts

Besides the two methods mentioned above, there are several scripts available on the Internet for creating output that follows an organization's specific requirements. These can be run from *File* → *Script* or the  icon, and offer tremendous amounts of customization. A popular format to output CAD designs is the .dxf format, and users can use the scripts installed with **EAGLE** to output to this file format. For users who wish to create their own scripts, please refer to the **EAGLE** manual that is in the `doc` directory of your installation. Also under the *File* menu, one can *Export* useful information such as the parts list and netlist for the design.

¹Companies that require specific Gerber settings will often have a .CAM file that defines these settings for **EAGLE**, such as [SparkFun Electronics](#).

²Top layer (component side)

³Bottom layer (solder side)


⁴Board outline

⁵Drill file

Chapter 6

Other Notes of Interest

6.1 User Languages

Similar to how scripts make outputting easier, another interesting feature of **EAGLE** is the use of **User Language**, which allows users to complete tasks that would otherwise take hours in mere seconds. Start the process of using a **User Language** by going to *File* → *Run* or click on the  icon near the *USE* icon at the top of the window. As an example, try opening `/ulp/import-bmp.ulp` from your **EAGLE** installation directory while in an active board design. A window with an outline of the capabilities of the program should appear, click OK. Now find a bitmap image that is suitable.¹ Open it, and select the number of colors you want to evaluate, note that the program will attempt to place each color on its own layer, the more colors you evaluate, the more layers it will use. Read through the settings that you can change before clicking OK, for simple cases the default settings are appropriate. You will be prompted by **EAGLE** to accept the script, look through it if you like, it shows the amount of work one would have to do by hand if not for the **User Language**. Run the script. Your bitmap image should now appear at the origin of the board drawing 6.1. This can be useful for adding logos to your **PCB** design without having to draw out the paths by hand.

6.2 Command Line

With the knowledge of operating **EAGLE** using command line, users can create scripts and adapt **User Language** to do their bidding. This is a topic left to users who truly wish to harness the power of **EAGLE** and the manual in the `/doc` directory contains some hints on how to use such commands, for more help please refer to the forums. With the use of command line, users can also customize how **EAGLE** works, down to the finest settings, however, this is a topic beyond the scope of this application note.

¹This application note will use the Spartan logo of Michigan State University, trademark and copyright of MSU.

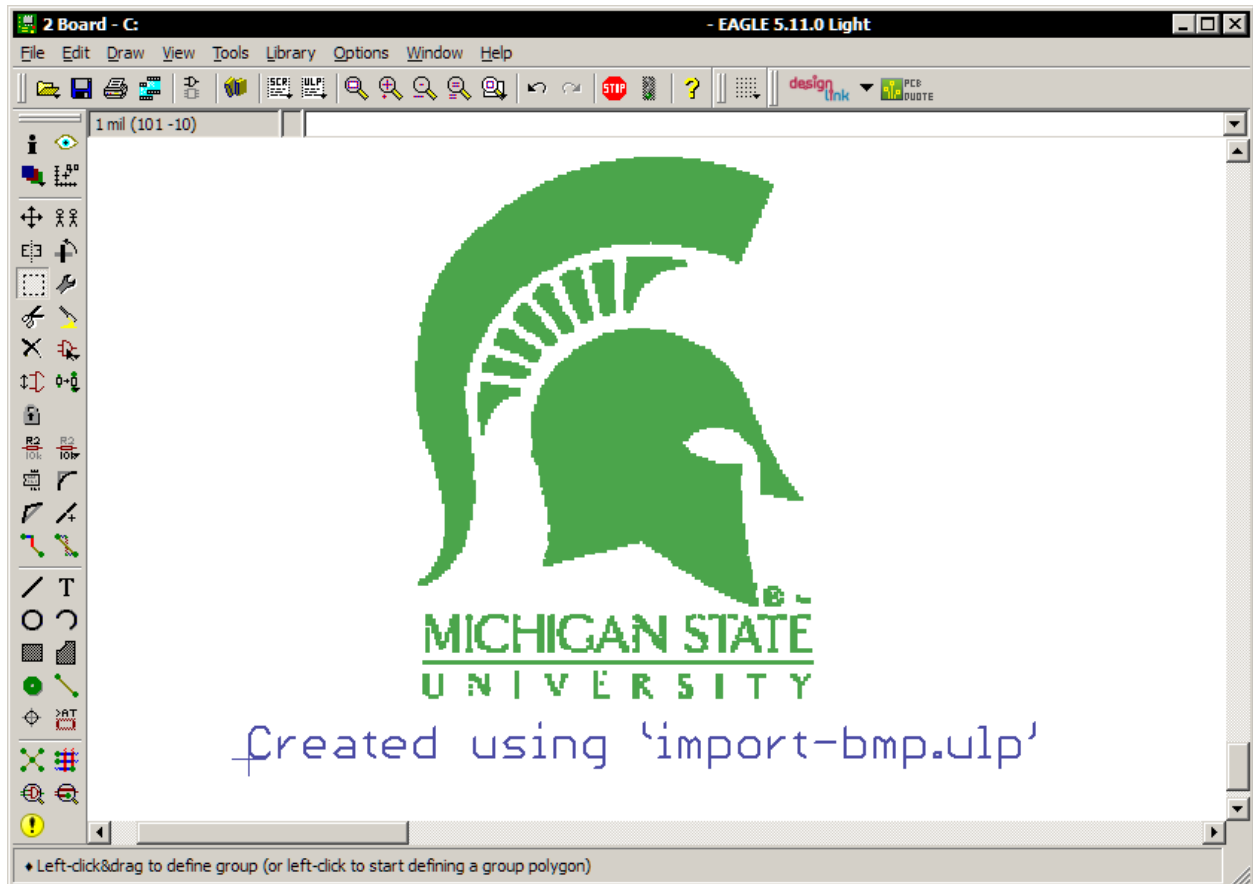


Figure 6.1: The Spartan logo successfully imported into the board design.

6.3 Getting Help

For additional help, check out the following sources.

[CadSoft Homepage](#)

[EAGLE Forums](#)

[User Files](#)

[ECE Shop²](#)

²This document was prepared using [LaTeX](#), an extremely useful tool for generating documentation.

Glossary

Airwire

Unrouted connection on a board, displayed in the unrouted layer.. 23, 24, 37

Alias

An alias is a predefined or user-defined configuration, which allows for rapid switching between frequently used settings as well as keeping track of settings that may be too cumbersome to remember and change each time they need to be used.. 8

Blind Via

A plated-through hole for changing the layer of a track which has not been drilled through all layers in the production process of a multilayer board.. 36, 37

Buried Via

A plated-through hole, which has been drilled through the current layer stack in the production process like a normal (through) via, but does not connect all layers of the whole board.. 37

Core

Two copper layers applied to a solid substrate.. 28

Device

A fully defined element in a **Library**. Consists of a **Package** and a **Symbol**.. 7, 34

Device Set

Consists of **Devices** that use the same **Symbols** for the schematic but have different package variants or technologies.. 6

DRC

Design Rules are a series of parameters provided by semiconductor manufacturers that enable the designer to verify the correctness of a mask set. Design rules are specific to a particular semiconductor manufacturing process. A design rule set specifies certain geometric and connectivity restrictions to ensure sufficient margins to account for variability in semiconductor manufacturing processes, so as to ensure that most of the parts work correctly.. 24, 28

Drill

Plated-through drilling in the layout (in **Pads** and **Vias**). [16](#)

EAGLE

Also known as "Einfach Anzuwendender Grafischer Layout Editor," is a schematic capture and **PCB** layout tool for hobbyists and DIY enthusiasts. Copyrights owned by CadSoft Computer GmbH.. [i](#), [1](#), [5--7](#), [11](#), [12](#), [14](#), [17](#), [21--23](#), [28--32](#), [35](#), [36](#)

ERC

Similar to **ERC**, **EAGLE** can identify the violation of certain electrical rules (e.g. if two outputs are connected) with the ERC. It also checks the consistency of the schematic and the layout.. [19](#), [35](#), [37](#)

Forward&Back Annotation

Transforms all the actions one makes in a schematic online into the layout (and with limitations from layout into schematic). Both files are consistent all the time.. [22](#)

Gate

The term *gate* is used in this manual for a part of a component which can be individually placed on a schematic. This can be one gate of a TTL component, one contact pair in a relay, or an individual resistor from a resistor array.. [18](#)

Gerber Format

The Gerber format is a family of file formats used by **PCB** industry software to describe the images of a **PCB** (copper layers, solder mask, legend, etc.) as well as the drilling and milling data. The RS-274X format is considered the de facto industry standard for photoplotters.. [30](#), [31](#)

Hole

Non plated-through drilling in the layout (e.g. a mounting hole).. [13](#)

LaTeX

\LaTeX is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. \LaTeX is not a word processor! Instead, \LaTeX encourages authors not to worry

too much about the appearance of their documents but to concentrate on getting the right content.. [33](#)

Layer

Layers have a unique number associated with a name, as well as a short description of what they are to contain. The purpose of layers is to create modules within designs, to make designing large projects easier. For example, the layer #94 Pins will contain any pins that are defined in a device, by turning the layer off in the display does not delete the actual pins, they simply won't be shown. A list of predefined layers can be found in the manual of a default **EAGLE** installation directory.. [8](#)

Library

A collection of devices that can be user-created, or download from sites such as [CadSoft Online](#).. [34](#), [36](#)

Micro Via

A plated-through hole (like **Blind Via**) with a relatively small drill diameter which connects an outer layer with the next reachable inner layer.. [37](#)

Net

Electrical connection in a schematic.. [8](#), [19](#), [21](#), [25](#), [26](#)

Package

Component footprint stored in a **Library**.. [34](#), [36](#)

Pad

Through-hole pad associated with a **Package**.. [35](#)

PCB

A printed circuit board, or PCB, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or signal traces etched from copper sheets laminated onto a non-conductive substrate. It is also referred to as printed wiring board (PWB) or etched wiring board. A PCB populated with electronic components is a printed circuit assembly (PCA), also known as a printed circuit board assembly (PCBA). Printed circuit boards are used in virtually all but the simplest commercially-produced electronic devices.. [i](#), [1](#), [5](#), [9](#), [13](#), [17](#), [20--22](#), [24](#), [27](#), [29](#), [31](#), [32](#), [35](#)

Pin

Connection point on a schematic **Symbol**.. 21

Ratsnest

Command for calculating the shortest **Airwires**.. 24

Signal

Electrical connection in a board.. 1, 21, 25--27

Symbol

Schematic representation of a component, stored in a library.. 7, 34, 37

User Language

Freely programmable, C-like language for data import and export.. 32

Via

Plated-through hole for changing the layer of a track. See also **Micro Via**, **Blind Via**, and **Buried Via**.. 21, 25, 35

Wheel

Aperture configuration file. Generated with Gerber data for board manufacturing.. 31

Wire

Electrical connection in a board, or a line (since lines are drawn with the WIRE command).. 23, 24

Index

- Add, 18
- Attribute, 19
- Author, 16
- Autorouter, 13, 21, 22, 24, 25

- Bus, 18

- Chemical Etching, 28
- Command Line, 32
- Control Panel, 6

- Decive, 12, 14
- Default Language, 4
- Description, 14, 16, 20
- Device, 14
- Device Editor, 15
- Download, 1
- Download Libraries, 6
- DRC, 24

- ERC, 19
- Errors, 19

- Freemium, 1

- Gate Swap, 18
- Grid, 12

- Hidden Pins, 21

- Import BMP, 32
- Inkscape, 29

- Installation
 - Linux, 3
 - Mac, 3
 - Windows, 2
- Installation Files, 1
- Invoke, 18

- Junction, 19

- Label, 19
- Library, 6
- Library Window, 7
- Lock, 23

- Mask, 30
- Multiple Layers, 24

- Name, 11
- Net, 19

- Package, 12
- Package Editor, 13
- Part Placement, 20
- Physical Media, 2
- Pin Swap, 18
- Pricing, 1

- Ratsnest, 24
- Replace, 18
- Ripup, 23, 24
- Route, 23

- Schematic, 17

- Schematic Toolbar, 18
- Silkscreen, 13
- Smash, 18
- Symbol Editing Toolbar, 8
 - Arc, 11
 - Change, 9
 - Circle, 10
 - Copy, 9
 - Cut, 9
 - Delete, 10
 - Display, 8
 - Group, 9
 - Info, 8
 - Mark, 8
 - Militer, 10
 - Mirror, 9
 - Move, 9
 - Name, 10
 - Paste, 10
 - Pin, 11
 - Polygon, 11
 - Rectangle, 11
 - Rotate, 9
 - Show, 8
 - Split, 10
 - Text, 10
 - Wire, 10

- User Languages, 32

- Value, 11, 18