**EE4: Circuit Board Design**

Idea to schematic to board in 2 hours flat

Richard “Ducky” Lin

**Application:** You've designed and prototyped circuits. But you never see a finished product on a breadboard - rather, they are almost always on a custom circuit board. Now, you will learn how to design these boards.

Table of Contents

[Introduction 3](#_Toc337742671)

[Technical Manual 4](#_Toc337742672)

[What is a printed circuit board? 4](#_Toc337742673)

[Anatomy of a printed circuit board 4](#_Toc337742674)

[Electronic design automation (EDA) 5](#_Toc337742675)

[Workflow overview 6](#_Toc337742676)

[Schematic capture 6](#_Toc337742677)

[Components 6](#_Toc337742678)

[Basic concepts 8](#_Toc337742679)

[Good practices 8](#_Toc337742680)

[Board layout 9](#_Toc337742681)

[Basic concepts 9](#_Toc337742682)

[Design rules 10](#_Toc337742683)

[Practical considerations 10](#_Toc337742684)

[Good practices 10](#_Toc337742685)

[Fabrication 11](#_Toc337742686)

[Verification & design rule checking (DRC) 11](#_Toc337742687)

[Gerbers 11](#_Toc337742688)

[Assembly 12](#_Toc337742689)

[Advanced: Component creation 12](#_Toc337742690)

[Schematic Symbols 12](#_Toc337742691)

[Component footprints / patterns 13](#_Toc337742692)

[Parts List 14](#_Toc337742693)

[Lab Manual 14](#_Toc337742694)

[Lab 3.0: Pre-lab: setup and circuit overview 14](#_Toc337742695)

[Lab 3.1: Circuit design 15](#_Toc337742696)

[Lab 3.2: Schematic entry 15](#_Toc337742697)

[Lab 3.3: Layout 18](#_Toc337742698)

[Lab 3.4: Forward annotation 20](#_Toc337742699)

[Lab 3.5: Design verification 22](#_Toc337742700)

[Lab 3.6: Fabrication data 22](#_Toc337742701)

[Lab 3.7: Extra for Experts: Component creation & surface mount 24](#_Toc337742702)

[Pattern creation 24](#_Toc337742703)

[Component creation 28](#_Toc337742704)

[Revising the schematic 31](#_Toc337742705)

[Revising the board 32](#_Toc337742706)

[Design Software 33](#_Toc337742707)

[Free & Open Source 33](#_Toc337742708)

[Commercial 33](#_Toc337742709)

[Software you can afford 33](#_Toc337742710)

[Software for the 1% 33](#_Toc337742711)

[PCBs and You 34](#_Toc337742712)

[Advanced Circuits 34](#_Toc337742713)

[BatchPCB 34](#_Toc337742714)

[Dorkbot PDX / OSH Park 34](#_Toc337742715)

[SeeedStudio 34](#_Toc337742716)

[Sources & Further Reading 35](#_Toc337742717)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changelist** |
| 1.0 | Fall 2011 | Initial release |
| 1.1 | 11 Oct 2012 | Various typo fixes and clarifications  Expanded section on good practices |

# Licensing

Creative Commons License

This document (including source elements, like diagrams), is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License (<http://creativecommons.org/licenses/by-sa/3.0/deed.en_US>).

While commercial use is not prohibited under the terms of the license (the drawbacks outweigh the benefits, nor is a solar car team going to start suing people for obscene amounts over copyright infringement), you shouldn’t be shamelessly trying to turn a profit off someone else’s work. What exactly this means is completely up to you – but please behave like responsible adults. Keep in mind that you must license any derivative works under a similar license – which gives recipients redistribution rights.

# Introduction

In the previous labs, you've been introduced to basic circuit design, and have even prototyped them on a solderless breadboard.

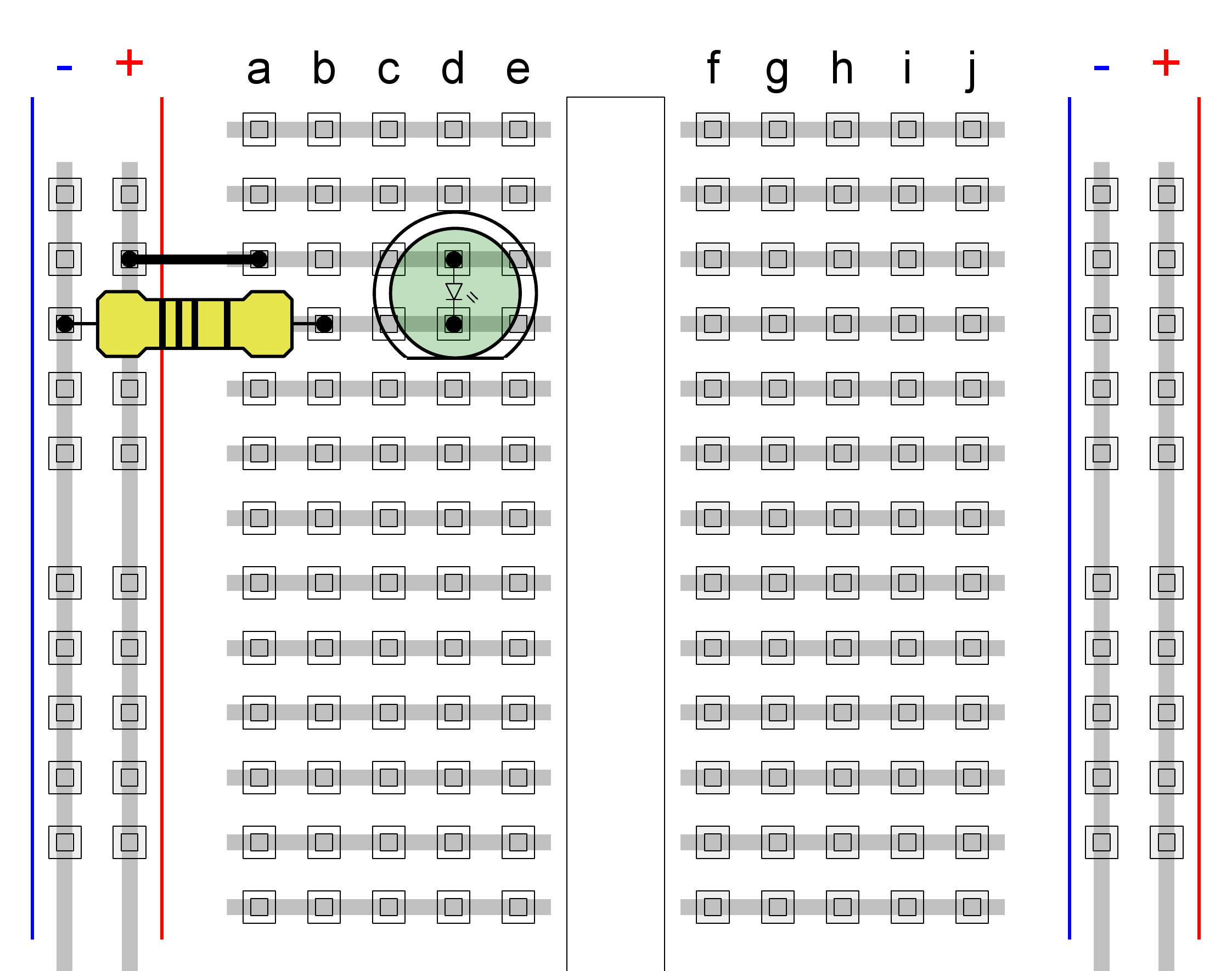


Figure 1: Simple LED circuit on breadboard

However, a breadboarded circuit certainly is not going to go onto our final solar car (or any production device, for that matter). Breadboards are handy when you are prototyping since you can easily swap out components and make changes, but what if you want something more permanent? Something which won't fall apart given the vibrations in a car?

Enter the printed circuit board.

# Technical Manual

## What is a printed circuit board?

A **printed circuit board (PCB)** is a board on which electronic components like chips and resistors are placed and connected.

### Anatomy of a printed circuit board

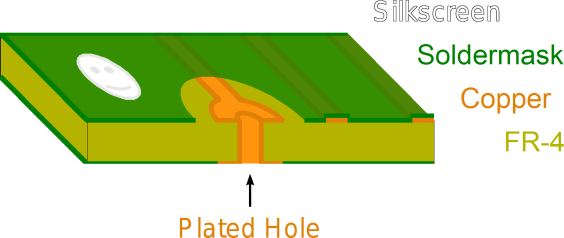


Figure 2 Anatomy of a two-layer circuit board

The major parts of the board are as follows (from center upwards):

* The **substrate** makes up the bulk of the board. This is an insulating layer, and is typically made of FR-4 fiberglass laminate. Although most circuit boards appear green, the FR-4 material itself is actually yellow (more later).
* The **copper layer** is what makes the electrical connections. This is typically a thin layer of copper foil bonded to the substrate, etched to the designed pattern.
  + Features on the copper pattern also have names. A **pad** is a copper shape to which component pins are soldered, and a **trace** is copper "wiring" connecting the pads. Sometimes, a **copper pour** is also used to connect components - this is just an area filled with copper. Usually, a copper pour is used as a ground.
* The **solder mask** is a layer of material which solder does not adhere to. This is typically applied over the entire board minus the pads, and prevents solder from flowing onto traces.
  + Soldermask is typically green, and is what gives circuit boards their characteristic color. However, soldermask is also available in other colors.
* The **silkscreen** is a layer of material for "artwork" - text and drawings on the circuit board. This usually contains component labels (or **reference designators**), the name of the board, and possibly component outlines.

Note that the board in the picture has copper on both sides, and as such, it is called a **two-layer board**. **Vias or through-holes** are holes which pass through from one layer to another. More often than not, they are **plated**, which makes them electrically conductive and allows them to electrically connect copper on the top and bottom layers.

Boards with more layers are possible, in which case there will be **internal layers** sandwiched between insulating substrate layers. However, cost also increases, so think really hard before deciding on a 12-layer design. Unless you're doing a serious commercial design, 4 layers is probably the cap. CalSol usually does 2-layer boards, since we get those sponsored by Advanced Circuits.

When there are internal layers, they typically include a dedicated **ground plane** and **power plane**, a solid copper layer either connected to ground or power. Signals can be routed on both external and internal layers.

## Electronic design automation (EDA)

Today, circuit diagrams and board layouts are both designed on a computer using **electronic design automation (EDA)** or **electronic computer aided design (ECAD)** tools. This is usually a suite of tools which includes, at the very least, schematic capture (to enter your circuit as a circuit diagram) and board layout (where you lay out the components, traces, etc on a board) tools. Advanced features may also automatic routing, where the software will attempt to **route** (lay out traces to connect components) for you - however, the results are only as good as the tools.

Note: The category of EDA tools also include those intended for integrated circuit (IC) or chip design. These cannot be used to lay out a circuit board.

### Workflow overview

The general workflow overview for designing a circuit board is as follows:

1. **Idea**: You need to know what you want to do first.
2. **Circuit**: Figure out how to implement your idea using electronic parts. Prototype the circuit on a breadboard (if possible) to ensure that it works.
3. **Schematic capture**: Enter your circuit schematic into the design suite.
4. **Board layout**: Convert your schematic into a circuit board layout, then place and route the components.
5. **Verification**: Ensure that your circuit will work as intended and is within any design constraints you may have.
6. **Fabrication**: Generate the design files, and send them off for manufacture.

## Schematic capture

Schematic capture is where you entry the circuit into the design suite. Typically, this is done interactively on a graphical representation of a schematic diagram. Basically, you insert components onto the page, and wire the components together accordingly.

Once you get into a schematic editor and learn the basic controls, this should be highly straightforward and intuitive.

### Components

|  |  |  |
| --- | --- | --- |
| C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_555.png | C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_resistor.png  C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_pot.png | C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_switch.png  C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_led.png |

**Figure 3 Example schematic components & symbols**

A **component** is the schematic symbol for a physical electronic component. It typically has one **pin** connection for each terminal on the device. For example, a resistor would be a component with two pins, and a 555 timer IC would be a component with 8 pins.

Additionally, each component has an **reference designator (refdes)**. This uniquely identifies the component within a schematic, and is usually in the format of a letter followed by a number (for example, R1, D22, and U4). The letter indicates the type of component, and the number is just a unique identifier within that type. Common letters and their meanings are listed below in Table 1. These are usually printed as silkscreen on the final circuit board so you know what component does there.

Table 1 Reference designator letters & meanings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Letter** | **Meaning** | **Letter** | **Meaning** | **Letter** | **Meaning** |
| **C** | Capacitors | **L** | Inductors | **TP** | Test points |
| **D** | Discrete diodes, including LEDs | **Q** | Transistors | **U** | Integrated circuits and misc. |
| **F** | Fuses | **R** | Resistors | **X** | Crystals |
| **J** | Connectors | **S / SW** | Switches |  |  |

In most programs, picking a component also includes picking its **package** or **pattern**, which is the form a component comes in. For example, a resistor can be in several different sizes (like 1/4 watt, 1/2 watt or even surface-mount 0603 - each requiring different copper and drill patterns), and the package determines which one it is. For integrated circuits, the same chip often comes in many different packages, possibly including a mix of through-hole and surface-mount. Common device packages are listed in Table 2. Google and Wikipedia should have many pictures.

**Through-hole** components are designed to have their pins inserted into a plated hole. Then, the hole is filled with solder which electrically and mechanically connects the component to the board.

**Surface-mount** components don't have "pins" in the traditional sense. They are designed to be placed on top of the board with their terminals soldered to a pad. These components are typically much smaller than their through-hole counterparts - and may be more difficult to manually solder.

|  |  |
| --- | --- |
| C:\Users\Ducky\Desktop\quackers\EE4\pcb through hole.png  Figure 4 A through-hole resistor | C:\Users\Ducky\Desktop\quackers\EE4\pcb surface mount.png  Figure 5 A surface-mount resistor |

Table 2 Non-exhaustive listing of common device packages as seen by CalSol / hobbyists – industry goes much smaller

|  |  |  |
| --- | --- | --- |
| **Device** | **Common through-hole packages** | **Common surface-mount packages** |
| **Discrete capacitors** | various | 0402, 0603, 0805, 1206  (for small capacitances) |
| **Discrete resistors** | 1/4 watt, 1/8 watt | 0402, 0603, 0805, 1206 |
| **Inductors** | various | various |
| **LEDs** | T 1¾ (5mm), T 1 (3mm) | 0603, 0805, 1206 |
| **Diodes** | various | various |
| **Transistors** | TO-92, TO-220 | SOT-23, SOT-89 |
| **Integrated circuits** | DIP-x  (where x is the number of pins) | SOIC-x, SSOP-x, TQFP-x, QFN-x  (where x is the number of pins) |

### Basic concepts

Schematics can be organized into **pages** - each corresponding to a physical sheet of paper which contains part of a circuit.

Then, there is the issue of connectivity, which is represented by **nets**. These basically state what pins on what components are connected together. You can think of a net as a wire - all pins which are part of a net are electrically connected together

On some design suites, nets can be named. However, whenever you create a new wire, they are typically given some generic name, like "Net 42."

Individual wires on a schematic can also be combined into a **bus**. A bus is essentially a bundle of wires, and individual wires can branch off of it. Busses are used to keep a schematic clean - instead of trying to route 10 different wires, you can have just one thick wire with individual wires branching off at the source and destination.

To simplify wiring, there are also **tunnels** - all wires entering a tunnel with the same name are connected. This allows you to connect components on different pages, or even connect components on the same page without needing a wire. Good use of tunnels can keep a schematic clean.

The implementation of this feature varies by software. Sometimes, you may need to add an actual tunnel object which can be named. On others, this is simply implemented by adding wires to a common net.

Finally, there are the power supplies - usually the symbols C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_+v.png for a positive voltage source and C:\Users\Ducky\Desktop\quackers\EE1\circuits\inline_gnd.png for ground (0v). Like tunnels, these are all implicitly connected to the same net.

### Good practices

Following good practices helps ensure readability and correctness.

* Convention says that a circuit should "flow" from left to right - inputs on the left and outputs on the right.
* Modularize your schematic - keep related parts together. Make use of different pages if your schematic is large enough.
* **Keep your circuits legible and clean**. I cannot stress how important this is. Messiness can conceal bugs - stuff you might not notice until after you receive and assemble your boards, only to figure out it does not work and you have a very expensive brick.
  + Crisscrossing wires may impair legibility. Although the rule is that crossed wires are only connected if there is a dot present, it may not be obvious at a quick glance.
  + Consider labeling portions of your schematic with a brief description of what it does.
  + Avoid having wires that go nowhere.
  + Consider using tunnels. This can also help modularize your schematic as well, eliminating the need for wires running across the schematic. When using tunnels, make sure that they are named intuitively. A tunnel named “Net 42” is not helpful; while a tunnel named “SPI\_MISO” is clear in what it does.
* Aesthetics: Schematics are not art, but it is usually a good idea to make them look nice.
  + For example, if you have a row of LEDs, place them such that they are aligned and evenly spaced.
  + Consistency is another point under aesthetics. If you decide to do something one way, stick with it. For example, if you put the current limiting resistors on the low side of a LED, do that across your schematic - don't mix low-side and high-side resistors.
  + However, this is usually a matter of personal taste. Use your best judgment.

## Board layout

The board layout is where you actually lay out the circuit board design. Typically, this is done in a different program than the schematic editor. However, you should be able to convert your schematic into a board design. Connectivity data and components will be forwarded to the board design program, but that's about it - current software cannot automatically create a full circuit board layout based off a schematic.

### Basic concepts

Similar to schematic capture, you again have components and wires. However, now, you are physically locating the component **footprint or pattern** - the pads which the components will be soldered to - on a board. Be careful not to make your design impossible to assemble - in particular, components should not overlap. Often, components have an outline on the silkscreen layer, and this should help prevent conflicts.

You can move components around the board, and on double-sided boards, you can also flip components to the back side. Flipping is recommended only for surface-mount components, and usually only when you cannot fit all the components on one side. Double-sided boards increase the complexity (and thus cost) of reflow soldering,

|  |  |
| --- | --- |
| C:\Users\Ducky\Desktop\quackers\EE4\pattern through hole nomask.png  Figure 6 Through-hole resistor: pattern (top), actual layout (bottom) | C:\Users\Ducky\Desktop\quackers\EE4\pattern surface mount nomask.png  Figure 7 Surface-mount resistor: pattern (top), actual layout (bottom) |

You will also **route** your board here - lay out the traces that will connect the pads on the components. You can route connections on any copper layer (and connect the different layers using vias), but all traces on a single layer forms a solid piece of copper. Therefore, if you have intersecting copper traces on the same layer, they are electrically connected. If you do not want that to happen, you must route the paths on different copper layers.

When connectivity data is forwarded from a schematic, a **rats nest** (consisting of individual **ratlines**) is used to indicate unrouted connections. Once all these disappear, you have routed your entire board.

Finally, you will need to define your **board outline**, which indicates the physical edges of the board.

### Design rules

Unfortunately, there are limits to what a board house can do - and these manufacturing capabilities are captured in the design rules. A quick listing is in Table 3.

Table 3 Common design rules, their descriptions, and common values

|  |  |  |
| --- | --- | --- |
| **Rule** | **Description** | **Common values** |
| **Minimum trace width** | The minimum width of any copper shape | 6 mil, 8 mil |
| **Minimum trace spacing** | The minimum spacing between two non-connected copper shapes | 6 mil, 8 mil |
| **Minimum drill size** | The minimum size of any via or hole | 13 mil, 20 mil |
| **Maximum drill size** | The maximum size of any via or hole | big |
| **Annular ring** | The minimum copper ring width around any plated via or hole | same as trace width |

\* 1 mil = 1/1000 in = 0.0254 mm

So, no, you can't make 22nm circuits on a circuit board.

### Practical considerations

Although you can design for the limits, doing so may affect reliability. In general, you want to design for loose tolerances, keeping trace sizes and spacings wider than the minimum.

Additionally, traces are made of copper, which is *not* a perfect conductor. They have resistance, which causes voltage drop and heat. A wider trace or a shorter trace will help reduce resistance and voltage drop. In particular, power traces are generally wider than signal traces.

There are online tools which can calculate the minimum trace width given the current going through it and maximum temperature rise.

### Good practices

* Do not design for the minimums unless you absolutely have to. Doing so can increase both failure rates and cost.
* Plan, plan, plan. Proper component placement can make routing prettier and easier.
* Place components such that they are aligned to the grid.
* Route such that the traces are aligned to the grid. For this, you may want to select a finer grid than the component placement grid.
* Avoid overlapping components unless you really, really, REALLY, know what you are doing. This generally makes assembly and rework difficult.
  + You MUST NOT overlap surface-mount components - this can make the board impossible to solder.
* Ensure the board is easy to assemble and rework.
  + For example, if you have a short component surrounded by tall components, you will have one heck of a time trying to replace that component should it be defective.
* If space permits, consider adding a ground plane and a power plane. On 4-layer boards, the two inner layers usually are a power plane and a ground plane.
* All traces should be angled at some multiple of 45°.
* Route boards such that all angles between traces are at 45°.
  + 90 degree angles are permitted, but discouraged. Sharp angles may cause interference and distortion on high speed signals.
* Consistency and patterns. Your board should look aesthetically pleasing.
* For high frequency circuits (well into the MHz range), poor layout can result in coupling between adjacent signals, which can cause corruption and data loss.
  + In particular, signals laid out in parallel can be subject to **crosstalk**, where one signal affects another line. The specific details are outside the scope of this document.

## Fabrication

So you've laid out your board. Now what?

### Verification & design rule checking (DRC)

Before you put down money to have your board fabricated, you want to make sure it works.

Your design software will probably have tools to **check net connectivity** - verify that everything that you want connected is indeed connected.

Your design software will probably also have a **design rules checker (DRC)** - run this to make sure that the board layout meets the design rules. However, your board house will also run their own DRC before they begin fabrication

Also consider getting your board **design reviewed** by another person (or send out a request to the entire team!)

### Gerbers

Gerbers ... as in baby food? Not quite.

Gerbers (RS-274X) are the standard file format for PCB manufacturing data, and all board design software should be able to export in this format.

Each Gerber file defines a 2d vector image, and you will have one Gerber file for each board layer. The common layers are listed in Table 4.

In addition to the Gerbers, you also need a file to define the drilled holes in your board. This is typically accomplished in a NC Drill (or Excellon format) file.

Table 4 Fabrication data - note that extensions do vary, and additional fabrication data exists - this is the bare minimum

|  |  |
| --- | --- |
| **Extension** | **Function** |
| **.gbr** | Generic Gerber file |
| **.gtl** | Top copper layer Gerber |
| **.gbl** | Bottom copper layer Gerber |
| **.gto** | Top silkscreen layer Gerber |
| **.gbo** | Bottom silkscreen layer Gerber |
| **.gts** | Top soldermask layer Gerber |
| **.gbs** | Bottom soldermask layer Gerber |
| **.drl, .txt** | NC Drill file |
| **.oln, .outline** | Board outline Gerber |

Once you have generated all these files from your design, you send these to the manufacturer. Once they have ensured that your board meets their design rules, they will produce the board and ship it to you.

## Assembly

Once you get your boards in the mail, you can assemble them by soldering components onto the board. However, that is a subject for another lab on another day.

## Advanced: Component creation

Most board design suites come with a **library** of predefined components. These include schematic symbols and patterns for the most common components and packages.

However, every once in a while, you may want to use a component that is not already available. To do so, you will have to create your own schematic symbol and pattern.

### Schematic Symbols

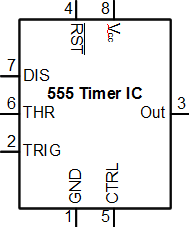


Figure Example schematic symbol for a 555 timer IC

The **schematic symbol** represents your component in the schematic editor. At the simplest level, it is a connect of pins (representing the physical device pins) which you can connect wires to on your schematic.

In addition to the pins, it is common practice to include the component's electrical symbol. While symbols exist for common parts like resistors, capacitors, inductors, and diodes, there are often no symbols for the pletora of chips. In those cases, a the component is represented by a box with pins on the edges (like in the 555 timer above).

Additionally, it is common practice to arrange pins such that things "flow" from left to right - with inputs on the left and outputs on the right. However, it is also common to lay out component symbols so that pins correspond to the locations on the actual device.

### Component footprints / patterns

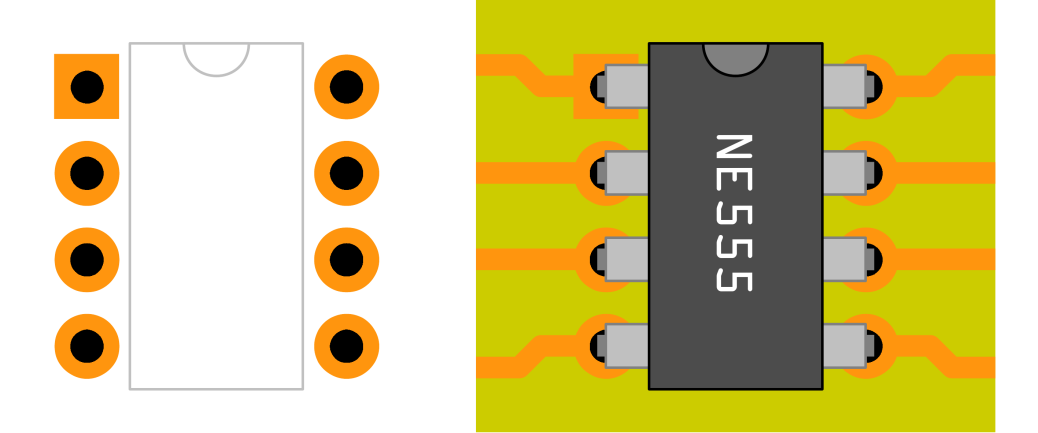


Figure 9 Example 555 timer pattern (left, silkscreen in grey) and actual layout (right)

The **component footprint or pattern** defines the arrangement of pads and silkscreen patterns necessary to attach a component to the circuit board.

**Pads** define the board features (copper and drill holes) to which the component will be attached. Usually, there is one pad for each component pin.

Pad holes are usually only necessary for through-hole components or mounting pins. When a through-hole pad is required, the copper pattern is repeated on both the top and bottom layer. Circular or ellipsoidal copper patterns are the most common for these, although a square patterns are commonly used to indicate pin 1 for through-hole components.

Drill holes in pads are always plated. It is possible to get non-plated holes (**mounting holes**) on components, but this usually increases fabrication cost and is replaced with a plated hole where possible.

The silkscreen layer usually includes a place for the component's refdes. Any information for aligning polarized components are also included - usually this is a dot next to the first pin for ICs, and a bar on the cathode terminal for diodes.

# Parts List

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Image** | **Function** | **Notes** |
| Your laptop | C:\Users\Ducky\Desktop\quackers\EE4\computer with duck.png  (please *don't* bring a desktop) | To run the software. | This isn't a parts-heavy lab, since it only deals with design |

# Lab Manual

What follows is a quick crash course in using DipTrace. Not all features will be explored - rather, only the basics will be touched upon. For more information, consider working through the official DipTrace tutorial: http://diptrace.com/books/tutorial.pdf (218 pages, lots of pictures.)

Also, note that a lot of this tutorial depends on your intuition with design software. Things like the mouse button and some keyboard commands (especially the delete and escape button) do what you would expect them to do.

## Lab 3.0: Pre-lab: setup and circuit overview

In CalSol, the PCB design suite we use is DipTrace (<http://www.diptrace.com/>), as we have the full version sponsored. All of the boards on Impulse were laid out using DipTrace, and if you have Dropbox access, you can view them.

1. Install DipTrace

## Lab 3.1: Circuit design

If you did the Extra for Experts during EE1, you will be familiar with the 555 timer circuit. Basically, it uses a 555 timer chip (along with some supporting components) to blink an LED.

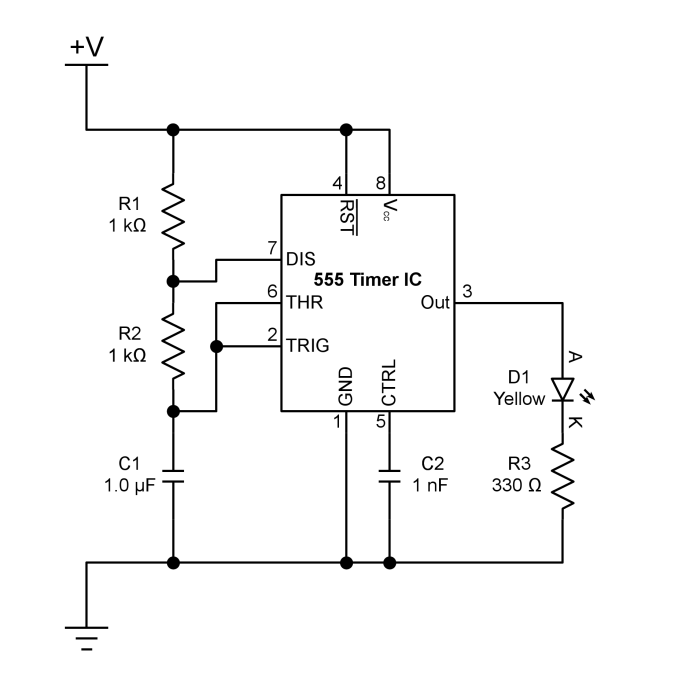
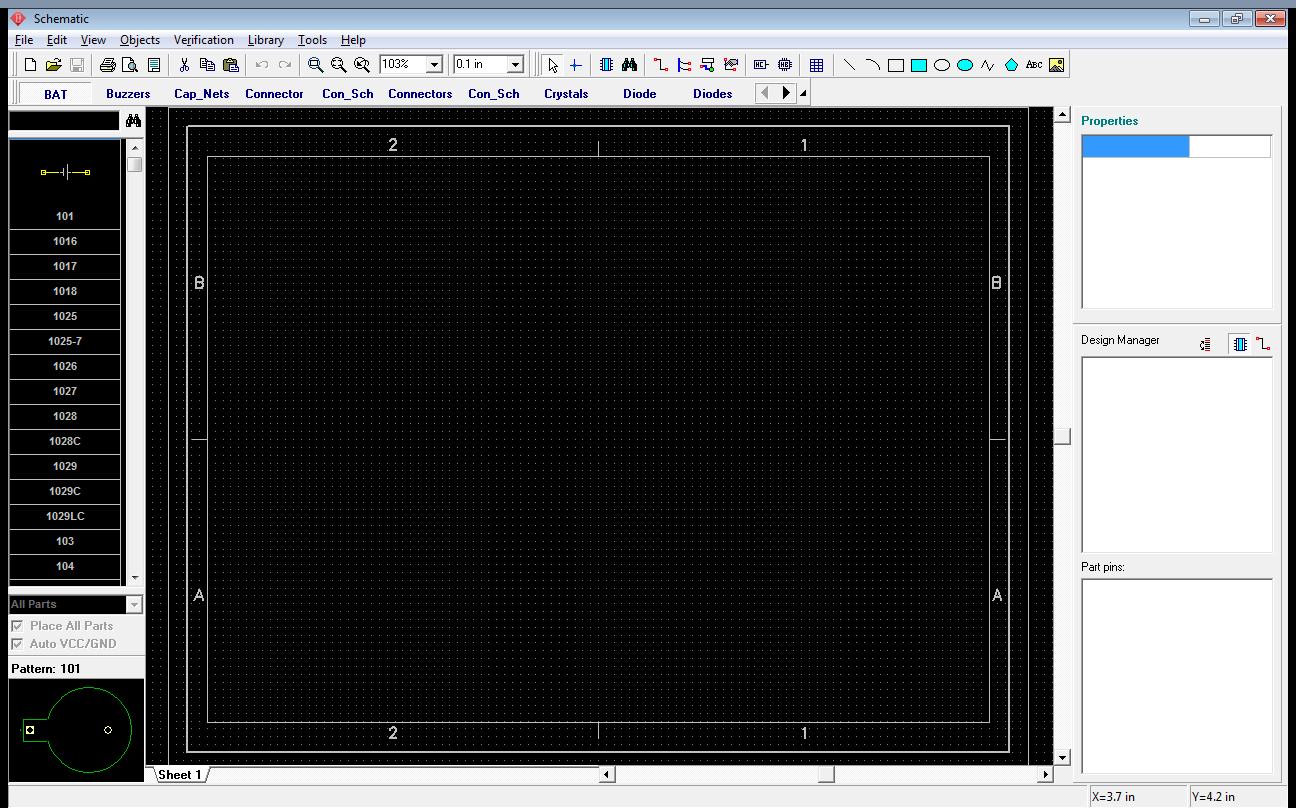
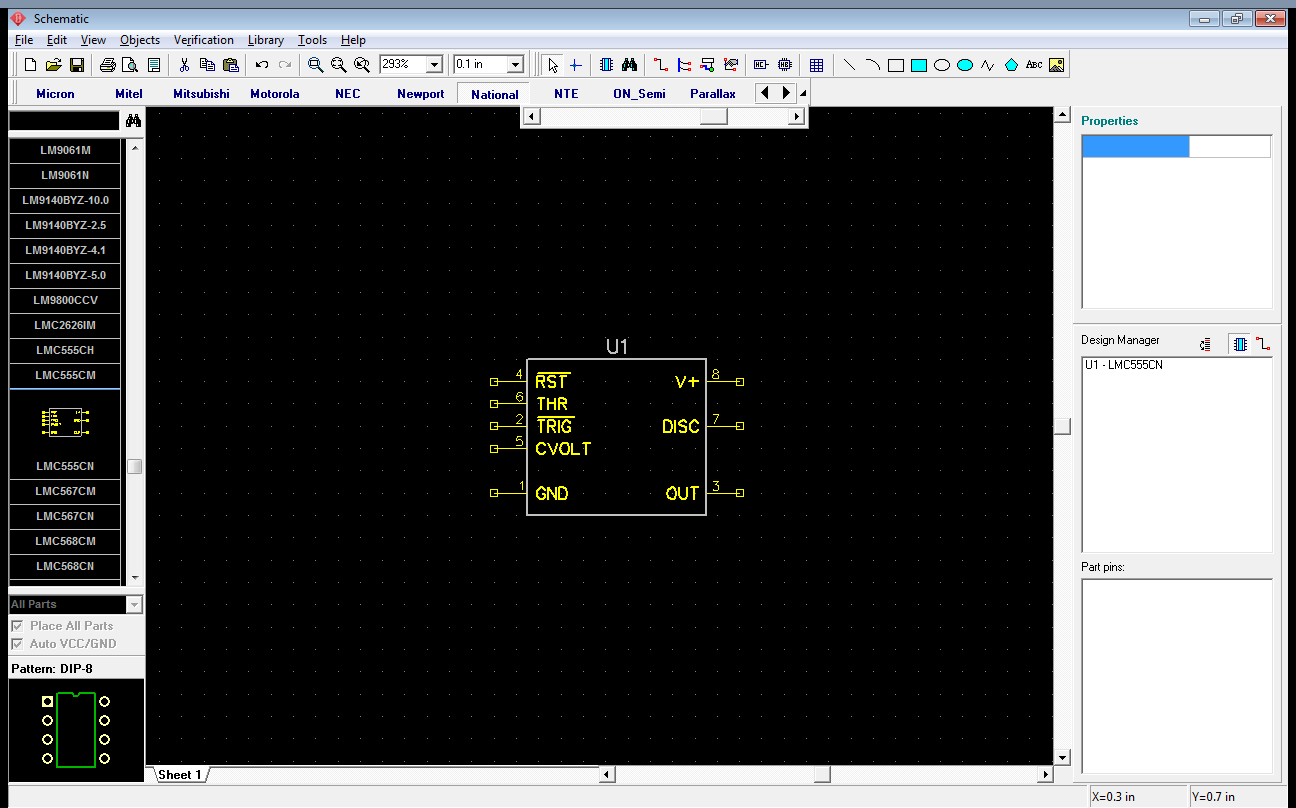
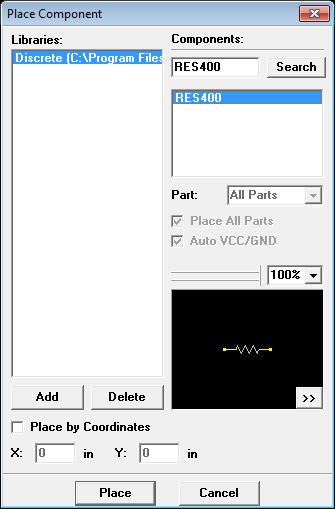
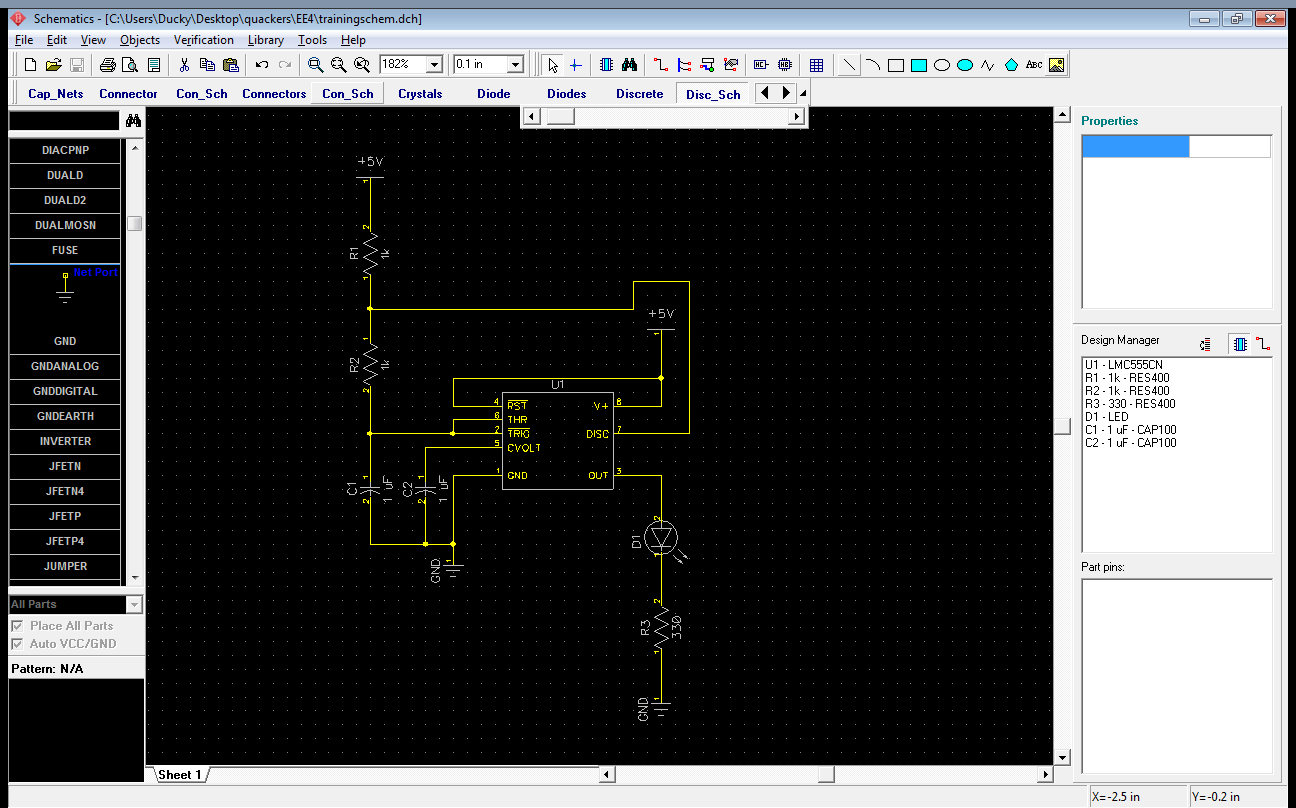


Figure 10 The 555 timer circuit

Today, we will be designing a board around this circuit.

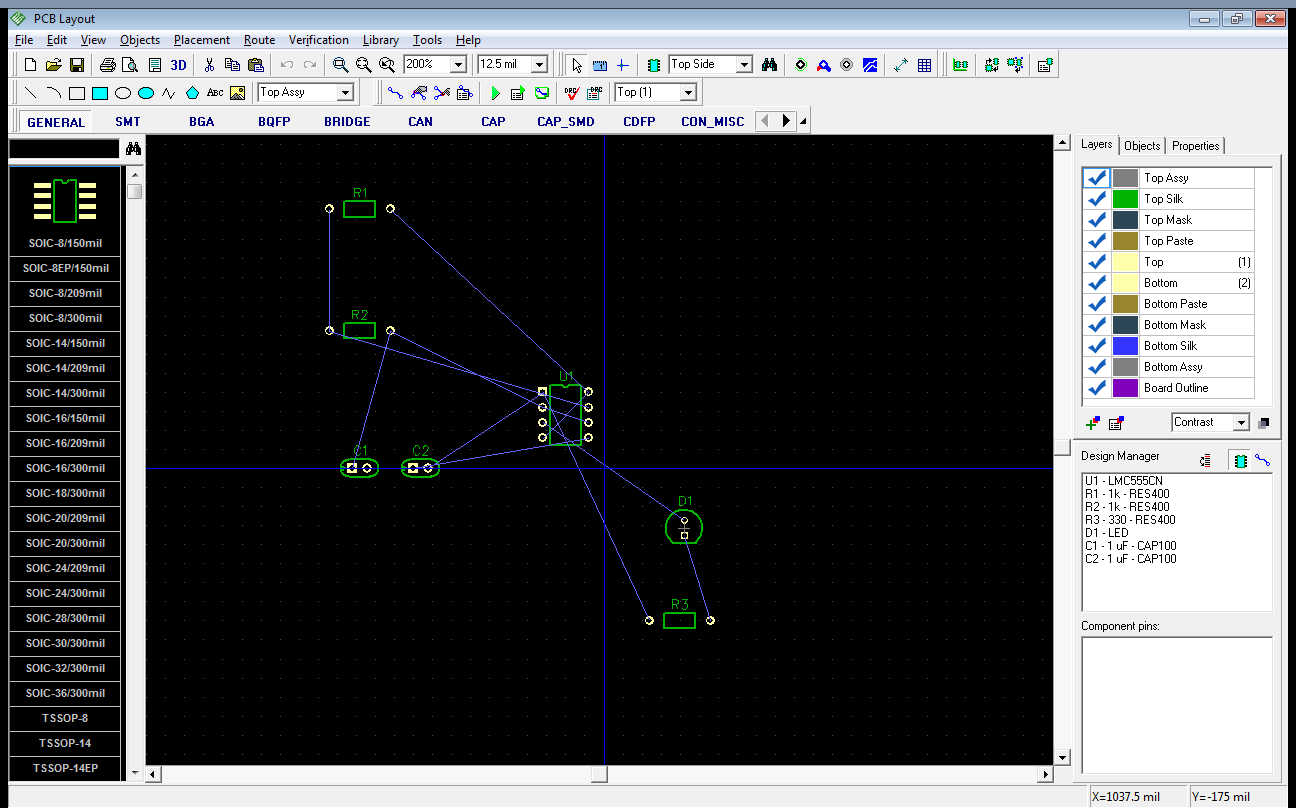
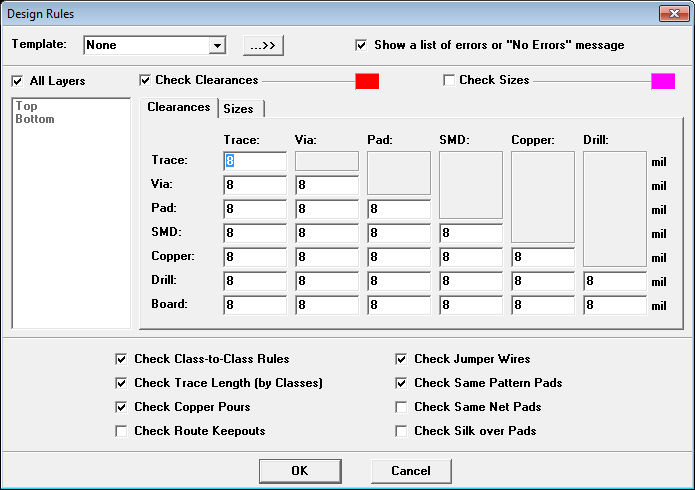
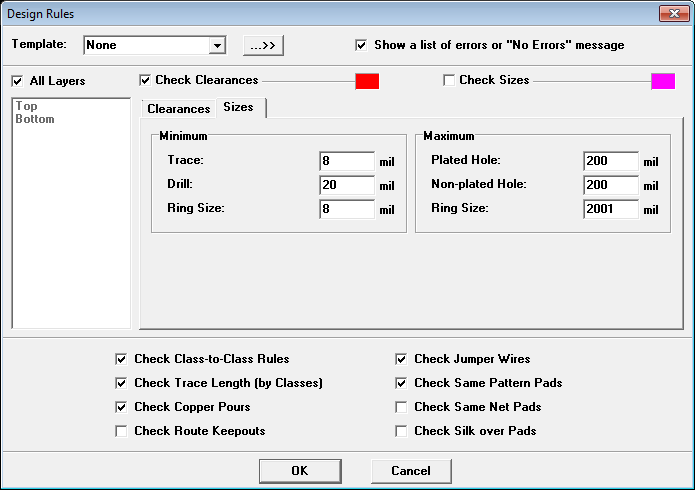
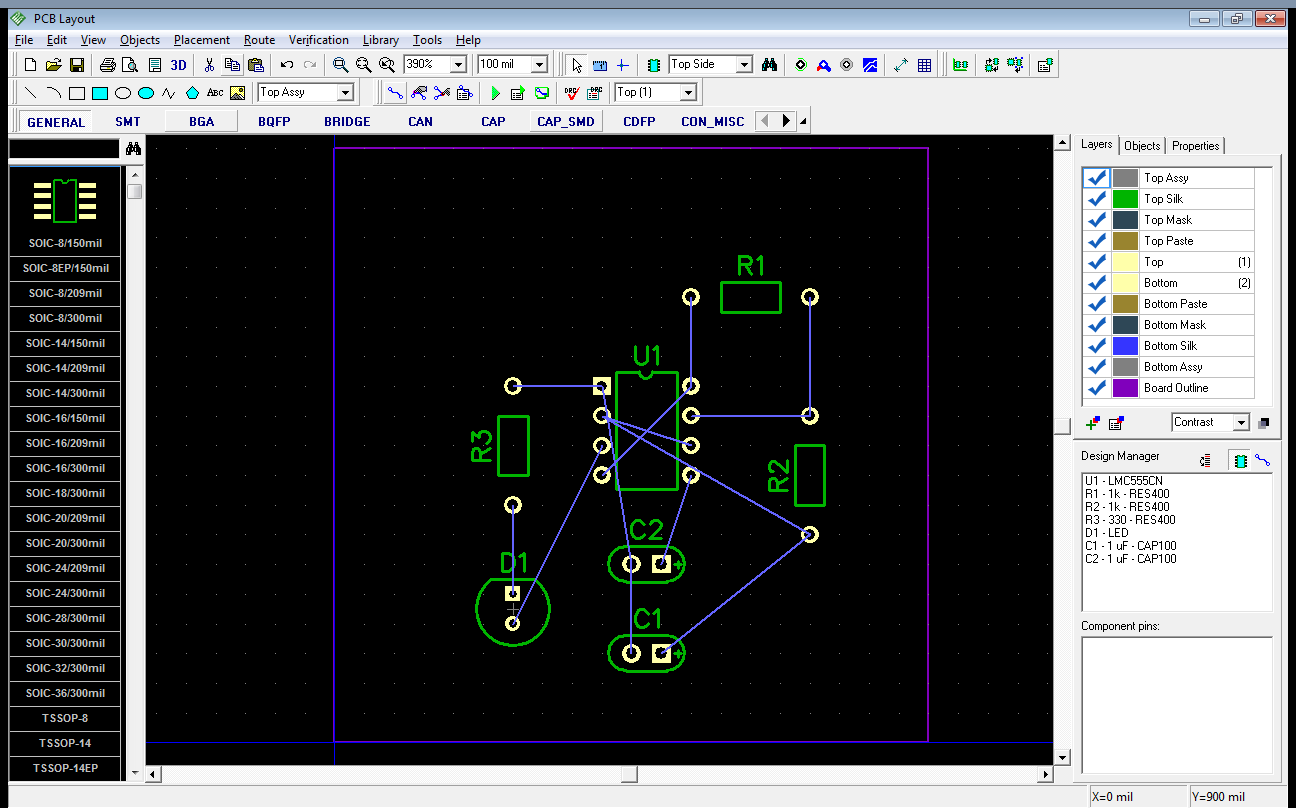
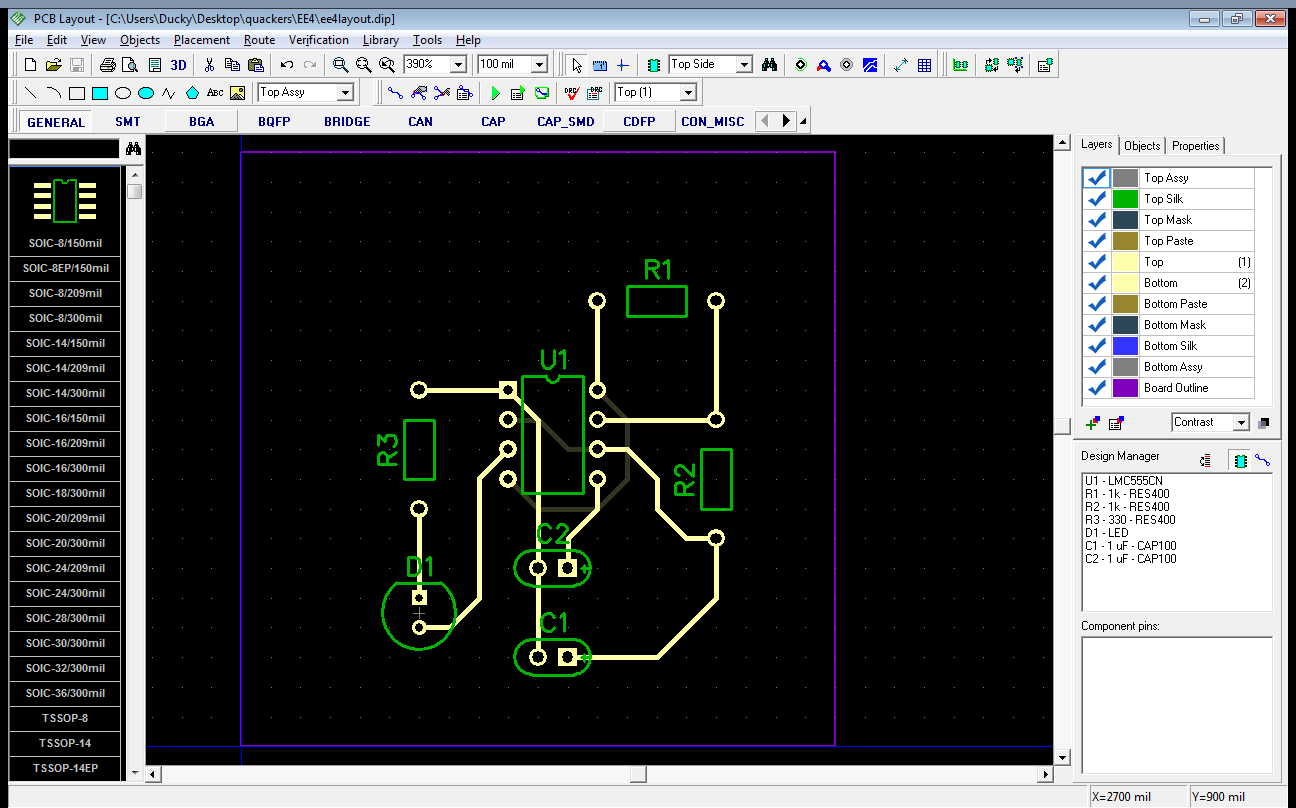
## Lab 3.2: Schematic entry

Enter the circuit into the DipTrace schematic editor.

1. Start up DipTrace Schematic.  
   
2. Play around with the interface.  
   In the main view of the schematic, you can use the mouse wheel to zoom in and out.  
   On the left side of the screen is where you can view components in the current library.  
   You can select the current library using the buttons below the toolbar.
3. On the library selection bar, scroll right and select the "National" tab. The listing of components should change accordingly.
4. On the listing of components, select the LMC555CN. This should allow you to place the component
5. Go to an empty space on your schematic, and place the component by clicking the left mouse button.  
   
6. However, there are other ways to select components. Go to the main menu, and select Objects->Place Component. The Place Component dialog should come up.
   1. The list of loaded libraries is initially empty, so click "Add", and select "\_discrete.eli" from the file selection window.
   2. Select the Discrete library from the list. This should populate the list on the right with the components in that library.
   3. What we're interested in is a resistor. Type "RES400" (a 400 mil spacing resistor) into the search bar, and click "Search." One component should be left.  
      
   4. Select that component, and click "Place."
   5. Place the component on your schematic. You may need several resistors.
   6. For each resistor, you need to set component specific information, like its value (resistance, in this case.) Double click the resistor, and set its Value in the component properties. Alternatively, you can select the component and edit its property sheet on the right.
7. You can wire your components by clicking on the small box on the component's terminal. Then, once you have reached the destination pin, click again to finalize the wire.
8. Also, place the required LEDs and capacitors. The LEDs are in the "Discrete" library and named "LED." The capacitor is also in the "Discrete" library, and named "CAP100."
   1. By now, you will probably have quite a few components. If you right click a component, you can see what you can do with them. Rotating and flipping them may help you organize your schematic.
9. Don't forget to add the power supplies. Remember from the technical manual that all the power supply symbols are connected. For this chip, add both a ground and a 5v power supply. The relevant symbols are in the "Disc\_Sch" library.
10. You may need to be creative with arranging your schematic to keep it readable. One possible schematic diagram is shown here:  
    

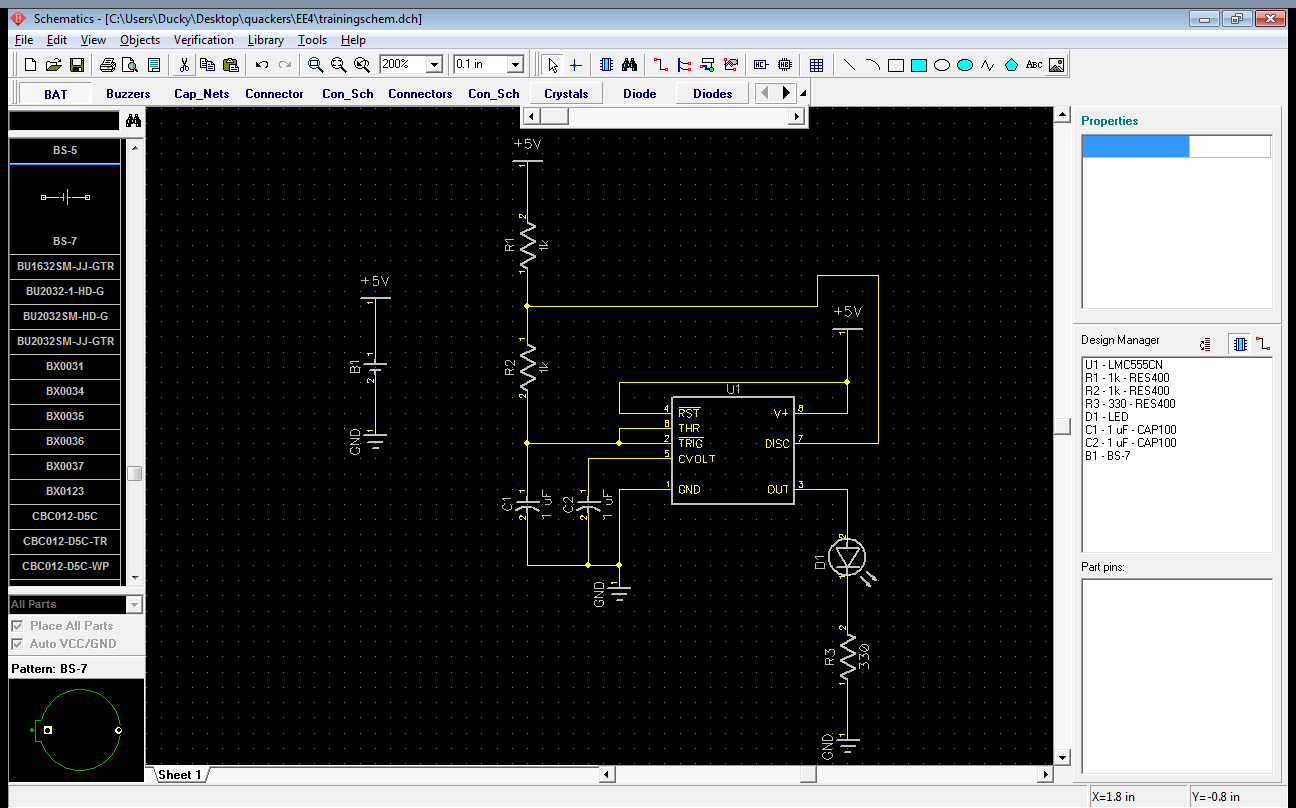
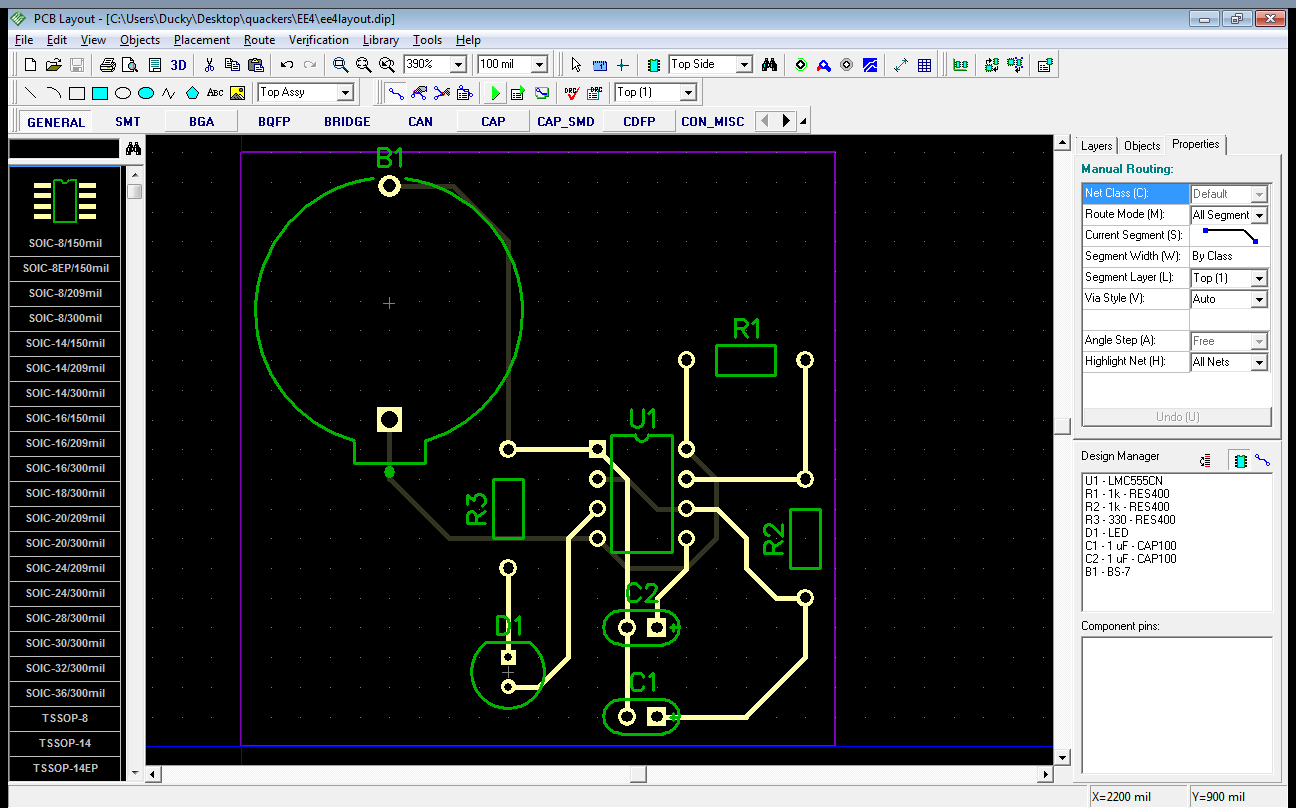
## Lab 3.3: Layout

Convert the schematic into a board, and layout the components and traces.

1. In the schematic editor's main menu, click File->Convert to PCB. Your circuit should now appear in the PCB Layout tool.  
   
2. Start off by checking the layout settings. For this project, you can set your grid at 100 mil. The grid settings are on the top bar.
3. Also, set up the design rules in Verification->Design Rules.
   1. The table you see defines the feature-to-feature minimum spacings. 8 mil should be adequate.  
      
   2. Also, check the sizes. 8 mil should be good for minimum trace width, and 20 mil should be good for the minimum drill size.  
      
   3. Yes, we can go lower if you really need it - 6 mil trace width / spacing and 15mil minimum drill size is the limit. But we won't - it's not a good idea to design at the limits.
4. Now, you should probably define the board outline. Let's say that you want a 2 square inch board. In the main menu, click Object->Place Board Outline. Click on the schematic to define the points on your board outline. Once you form a closed polygon, the board outline will have been defined.
5. Move components around in preparation for layout. Do this by clicking and dragging the components. Again, you can rotate components by right clicking on them and selecting "Rotate," or by pressing Ctrl + R.
6. Route your design.  
   
   1. Click the "Route Setup" button on the toolbar, and set your routing trace width. Let's set it at 15 mil trace width.
   2. Enter routing mode by clicking the "Route Manual" button on the toolbar.
   3. You can place routes by clicking on a pad. Click to set down a routed segment. Routing of a particular trace ends on either a double-click or when it hits a connected pad.
   4. You probably don't want to try routing everything on just one layer. Press (1) and (2) to switch between the two copper layers. Doing this in the middle of a routing operation will create a via to connect the two layers.
   5. An example routed design is below:  
      

## Lab 3.4: Forward annotation

Rarely do you get your schematic right on the first try. **Forward annotation** is updating an existing layout with a new schematic version.

1. It turns out we forgot our power supply! Uh-oh! Go back to the schematic editor, and let's correct this major oversight.
2. In the component library "BAT," place the component "BS-7" (a CR2032 coin cell holder) on your schematic. Connect +5v and ground to it. Note that batteries are polarized devices, and the longer line on the schematic symbol indicates the positive terminal.
3. A possible schematic looks like this:  
     
   Notice how the battery is only implicitly connected to the rest of the schematic using the power components.
4. Now, update your circuit board design with the new schematic. In the PCB layout editor, click File->Renew Design from Schematic->By Component, and select your schematic file.
5. Route the new battery holder component. Feel free to change your board dimensions as necessary (you can click on the board outline and drag the points.) One possible layout is as follows:  
   

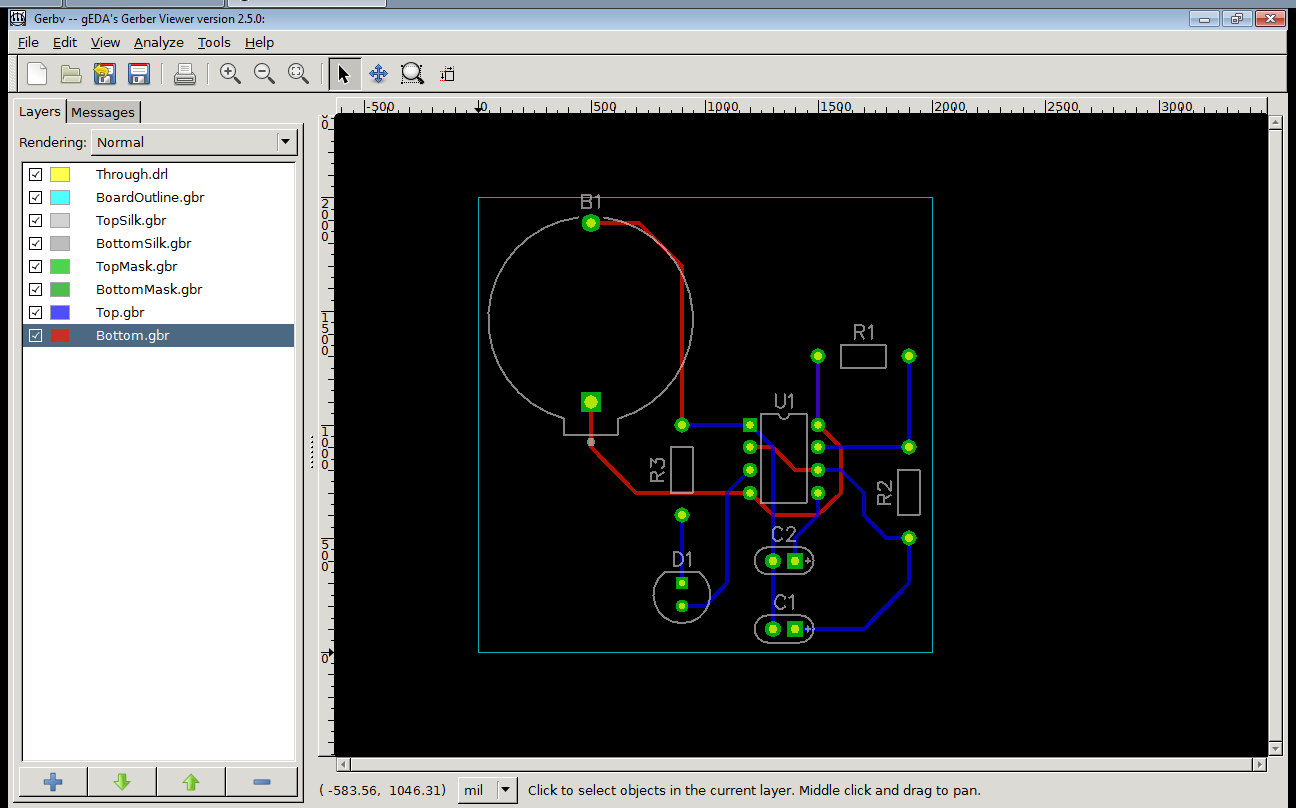
## Lab 3.5: Design verification

Your design is complete. But, does it work?

1. First, check that your design is completely routed. In the menu, click Verification->Check Net Connectivity. Accept the default settings (the checker will take into consideration copper traces, shapes, and pours), and let it check your layout. If you routed your board correctly, it should say "no errors found." Otherwise, you need to go back and fix any mistakes you made.
2. Similarly, check that you have satisfied all the design rules - click Verification->Check Design Rules, or F9. If it passes, it will tell you "no errors found," otherwise, it will indicate where you messed up.
3. On a real design, it's always a good idea to get both your schematic and layout checked by another person.

## Lab 3.6: Fabrication data

Most board houses will only accept Gerber files for fabrication - they do not have the time to deal with all the different file formats created by all the different programs. You need to generate these files once your board design is complete.

1. Export a complete set of Gerbers. Go to File->Export->Gerbers.
   1. The default settings should work fine. Click "Export All," and give those Gerber files a home.
   2. DipTrace will export more Gerber files than you need. The only essential ones for PCB fabrication are the copper layers, the soldermask layers, the silkscreen layers, and the board outline.
2. Export the NC drill file. Go to File->Export->N/C Drill.
   1. Again, the default settings should be fine. Click "Export," and save the file in the same folder.
3. Usually, you want to verify that your Gerbers are, in fact, sane.
   1. Dedicated Gerber viewer programs exist. Check out gerbv and see if you can get it to open your freshly created Gerber files.  
      
   2. Otherwise, if your board house has an online submission process, it *may* display your board files before your confirm your order.
4. Congratulations. You've created your first circuit board.

## Lab 3.7: Extra for Experts: Component creation & surface mount

Say that you don't like that boring 5mm LED. You want something a little more powerful. Scratch that, you want something *blindingly* powerful - 100+ lumens powerful. Something like a 1 watt LED[[1]](#footnote-1) [[2]](#footnote-2).

**Digikey link:**

<http://search.digikey.com/us/en/products/MX3AWT-A1-R250-000CE3/MX3AWT-A1-R250-000CE3CT-ND/2356708>

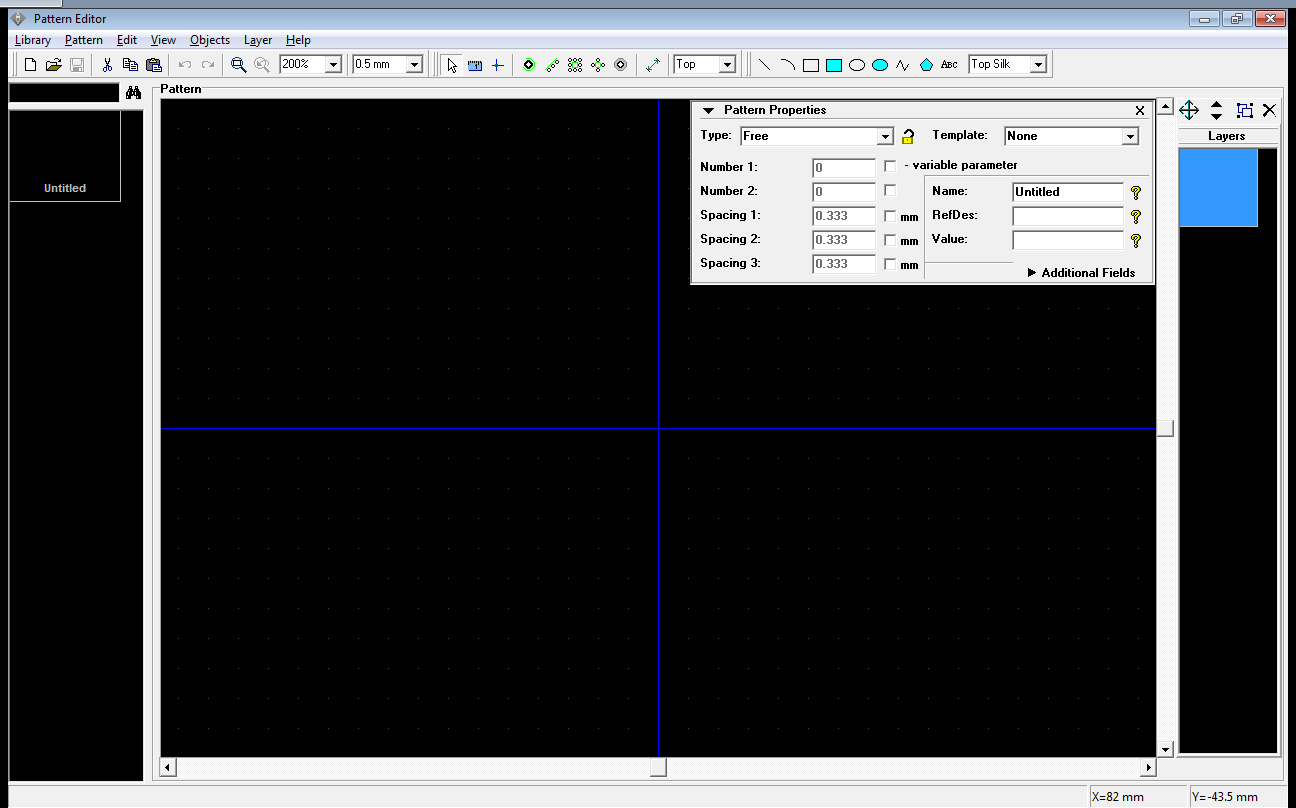
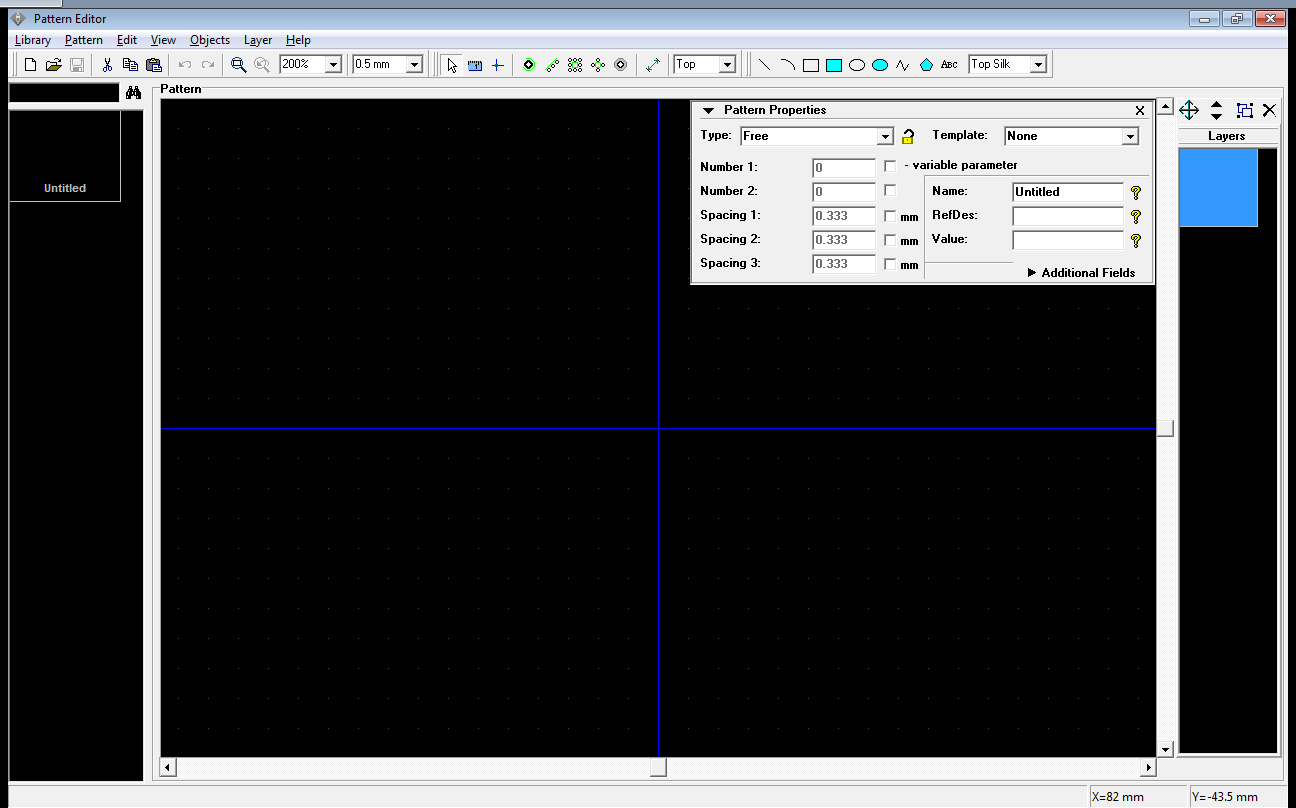
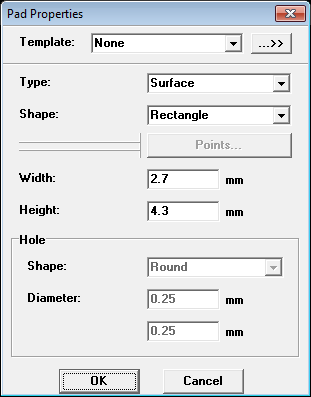
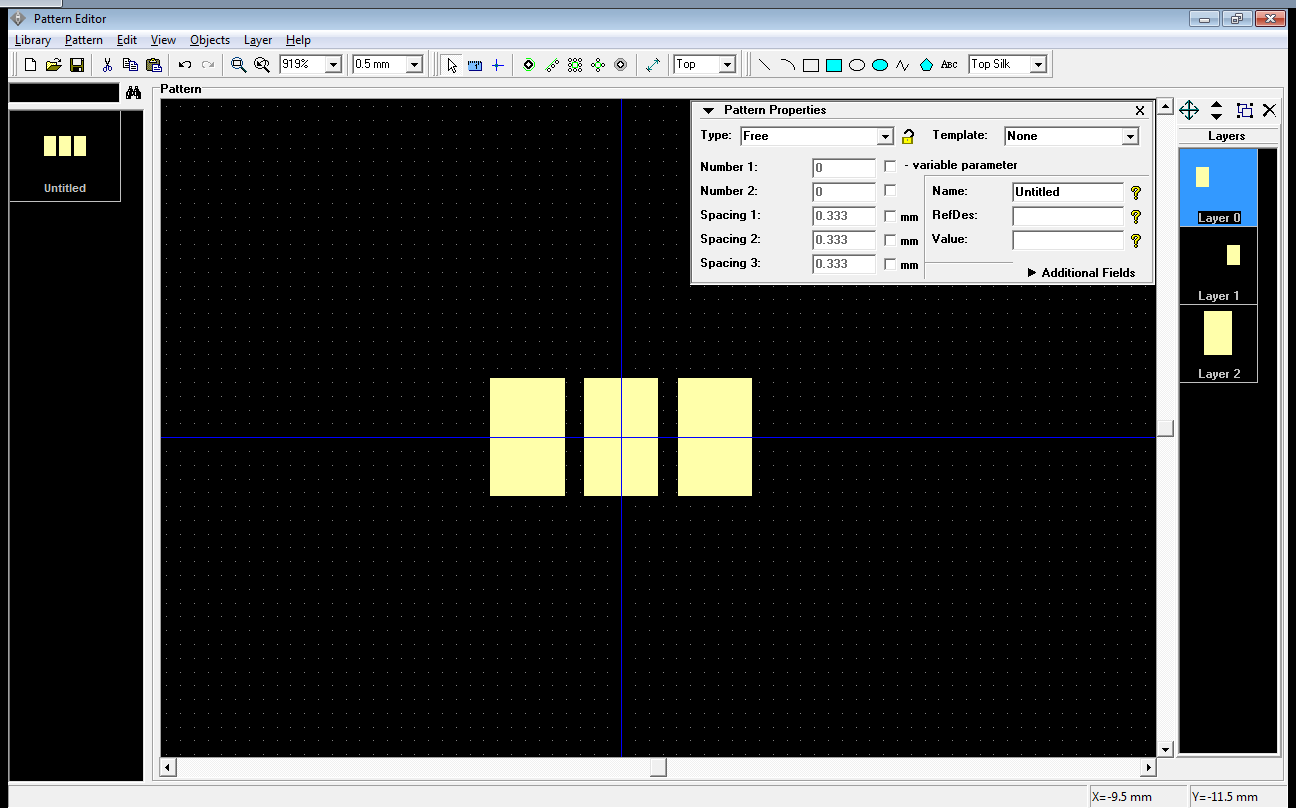
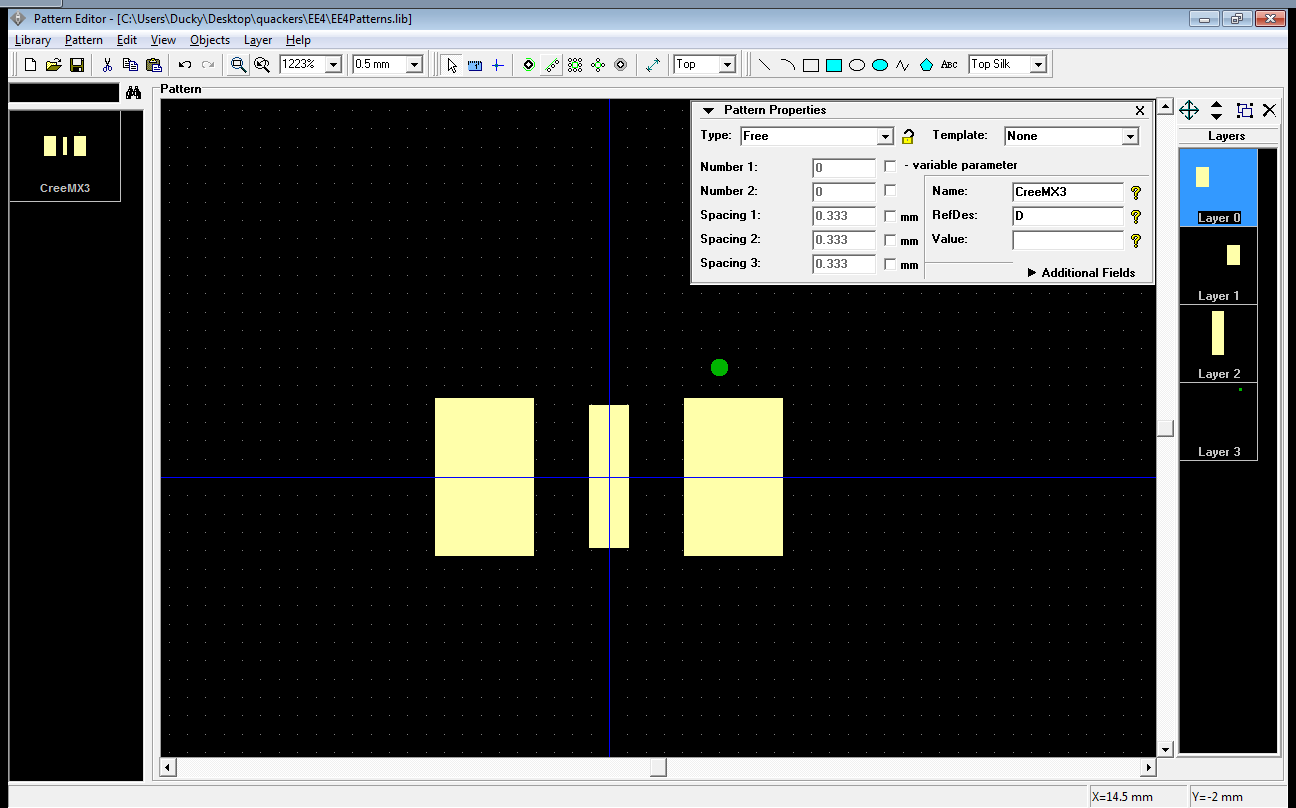
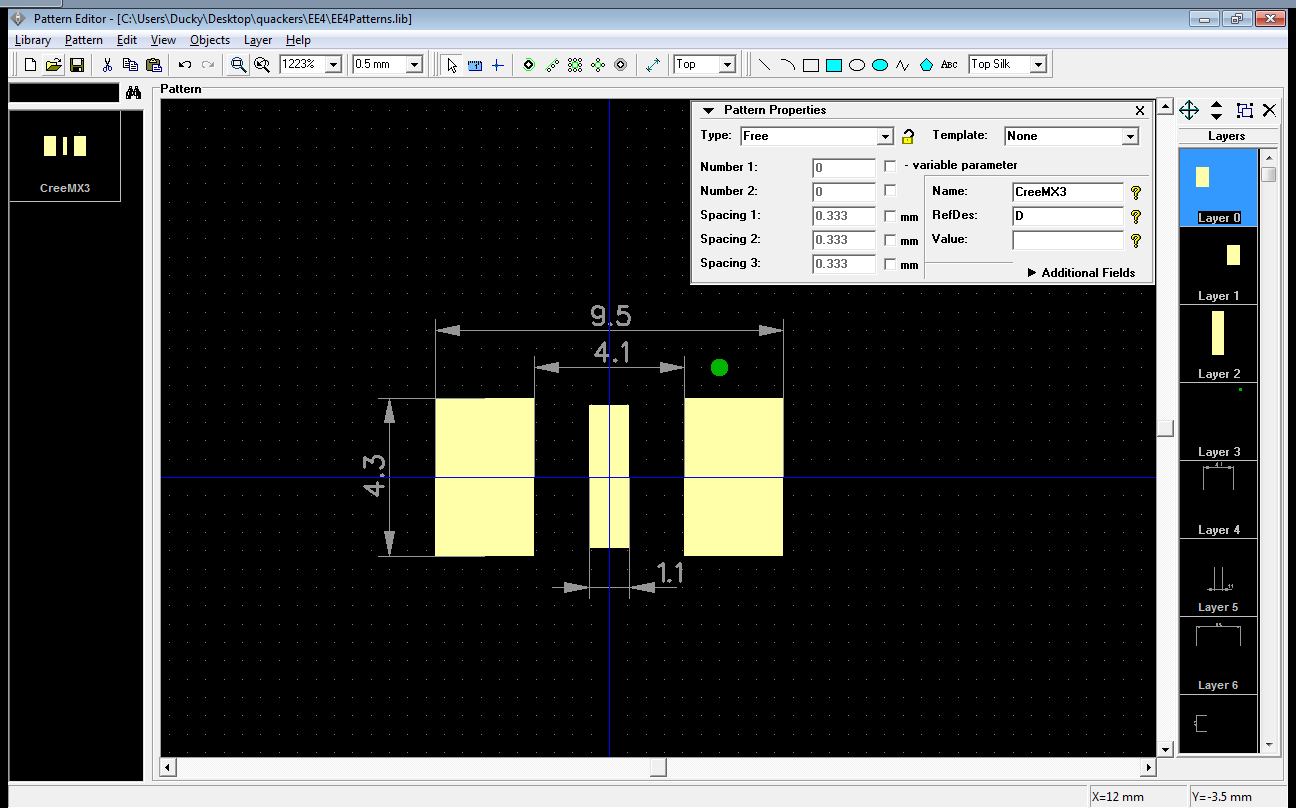
**Datasheet:**

<http://www.cree.com/products/pdf/XLampMX-3.pdf>

However, no such component exists in the standard DipTrace libraries, so you'll have to create your own.

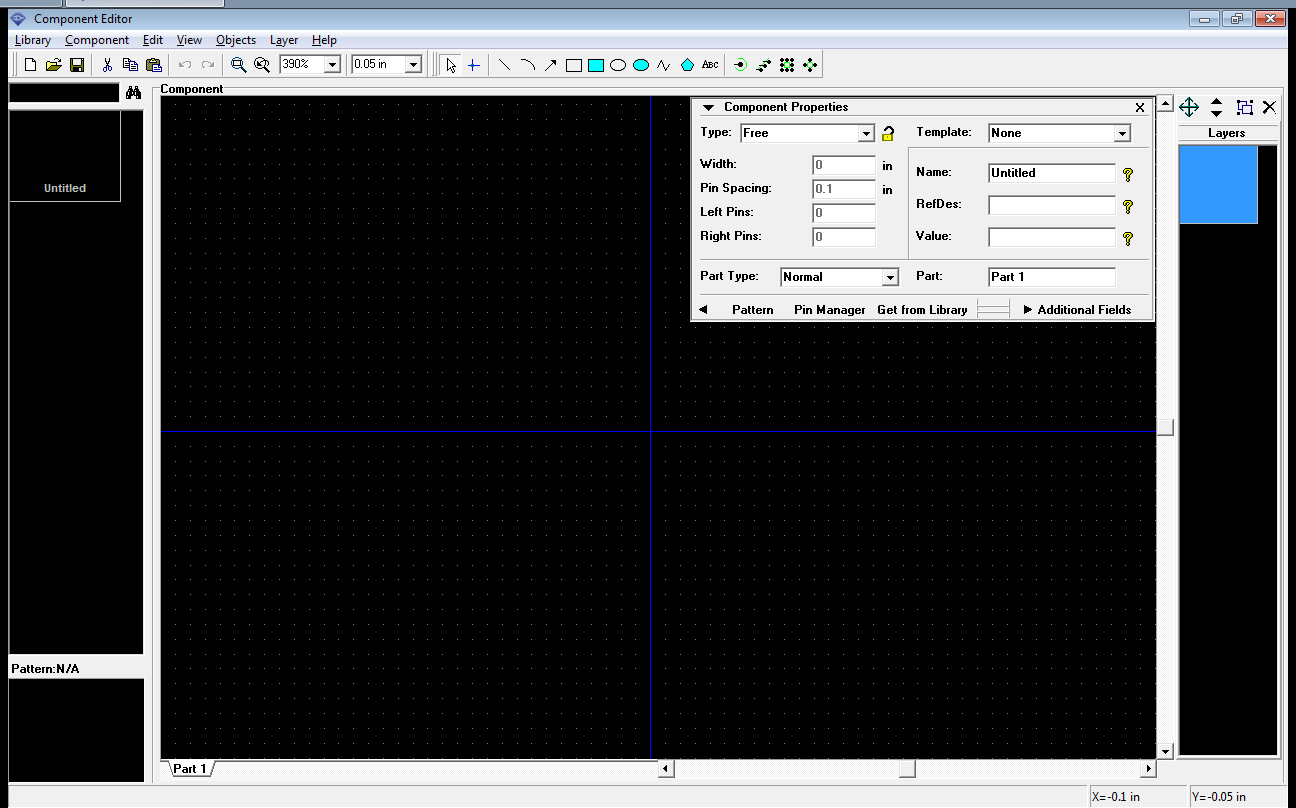
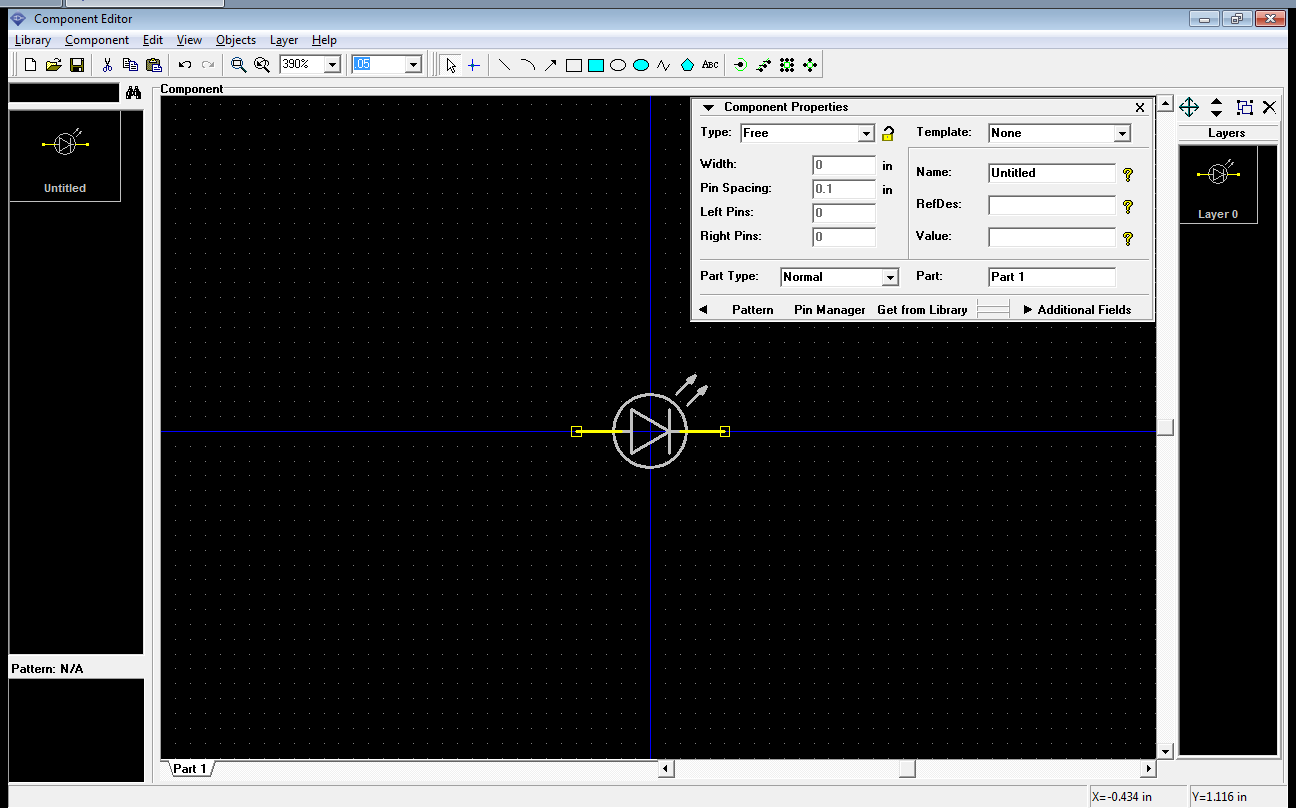
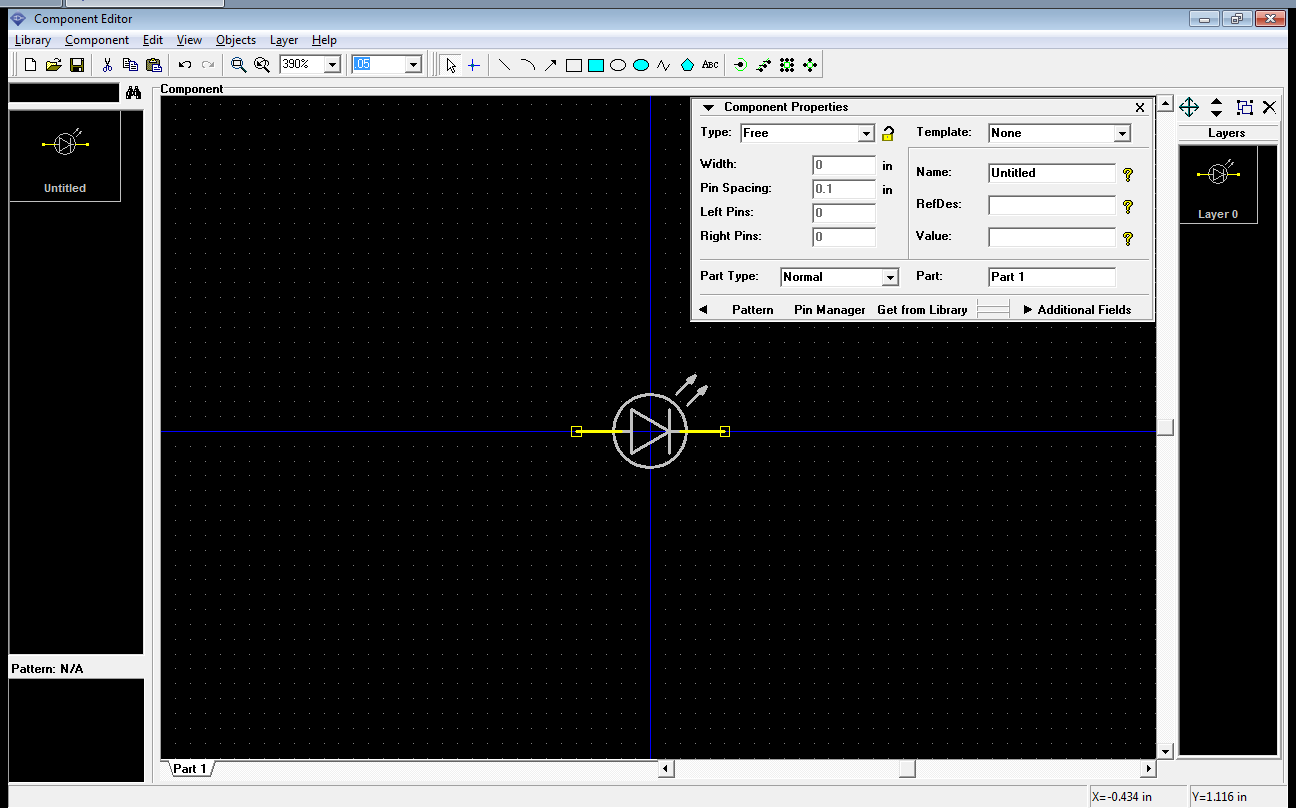
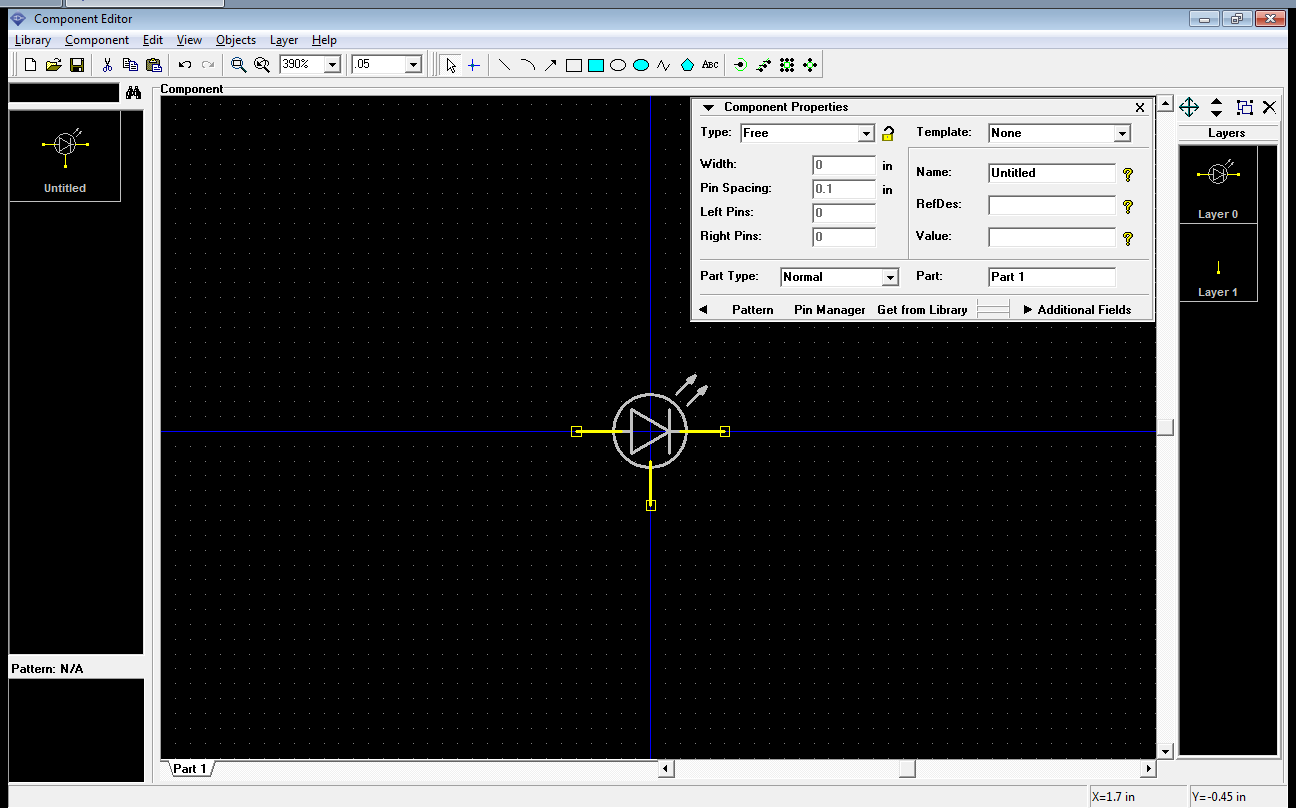
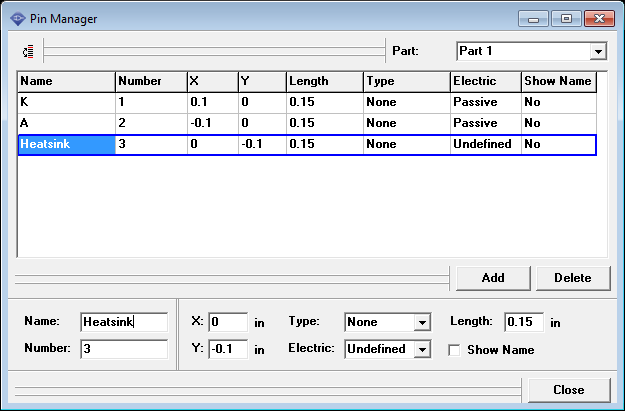
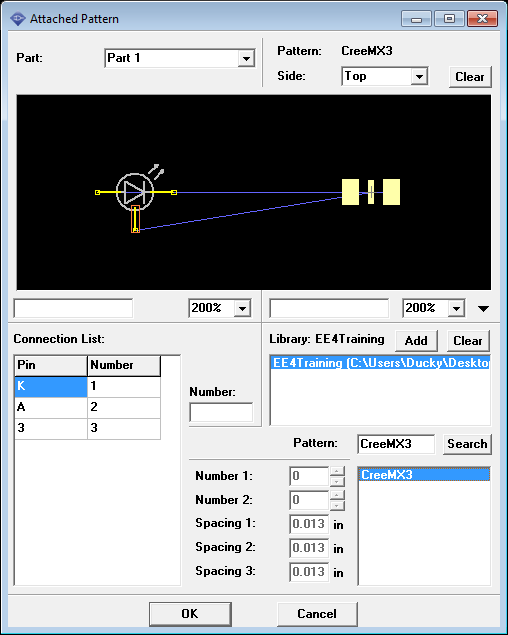
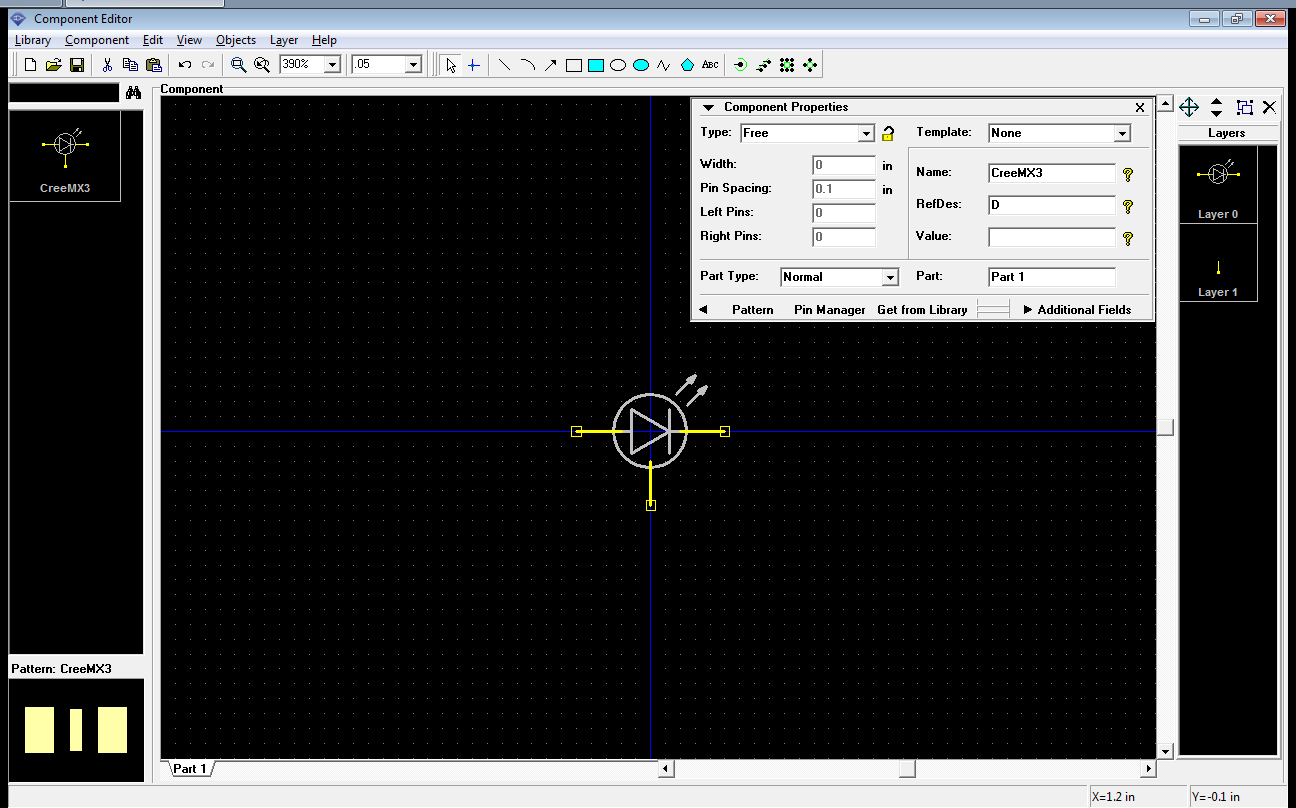
### Pattern creation

So, where to start? With the recommended solder pattern, of course! Usually, on datasheets for parts, they will have a recommended pattern, complete with dimensions. Your job is them to accurately input their drawing into the pattern editor.

1. Take a look at the recommended pattern. It is on page 8 of the component's datasheet.
   1. Note that this is a surface-mount device. There are 2 electrically connected pins (anode and cathode) and a heatsink pad.
   2. The drawings may look a bit confusing at first - they have the hatched and clear boxes on the drawing. I believe the hatched drawings are used for the stencil (indicates where to put solder paste) while the clear boxes are used to indicate copper.
2. Start the DipTrace pattern editor.  
   
3. Ensure the pattern editor's units are set consistently with your component datasheet's units. The LED datasheet uses millimeters, and you can set DipTrace's units by going to View->Units.
4. The pattern editor defines a component's pattern with pads and shapes. You should already be familiar with pads from the technical manual. As for shapes, it is just a geometric shape which can be placed on any layer, including silkscreen.
5. On the toolbox, click "Place Pad," and place three pads anywhere on the schematic. Note that your newly placed pads will NOT match what you want to do - this is fine (for now!)  
   
6. DipTrace defines pads with an coordinate, a shape (rectangle, oval, circle, and polygon), and sizes. Define your pads according to the component datasheet.
   1. Note that the recommended pattern is more or less symmetrical along the horizontal axis. Therefore, you only need to calculate the pad size for one side.
   2. Calculate the pad size for either of the two outer pads.  
      You should get mm for the width and 4.3mm for the height. Make sure you understand how to get these numbers from the dimensioned drawing.
   3. Choose any pad, and double-click it to bring up the properties. On the Type/Dimensions tab, ensure "Use Pattern's Pad Properties" is checked, and edit the Pattern's Pad Properties. This sets the common pad definition for the pattern.
   4. Set the dimensions consistently with the numbers you got. You should be using a surface, rectangular pad without a hole.  
        
      If you were designing a through-hole component pad, you select a through-hole type pad, and set the hole size. In this case, you must ensure that the hole is larger than the component pin and there is enough copper surrounding the hole to satisfy design rules.
   5. OK out of everything. All of your pads should have changed size now.
7. Calculate the X and Y position for the two outer pads. If you defined the origin of the pattern as the center of the component, the coordinate should be 0.  
   You should get mm as the distance between centers and mm as the coordinates.
   1. Double-click the relevant pads (1 and 2) and input the and coordinates. Your pattern should now look like this:  
        
      The pads are numbered from left to right as follows: 1, 3, 2
   2. Now, repeat for the center heatsink pad. If you set up the origin as defined above, you should get for the pad position, 1.1mm for the width, and 3.9mm for the height. Since this pad's dimensions are not consistent with the rest of the pattern, you will want to uncheck "Use Pattern's Pad Properties" and enter custom properties.
8. Add a polarity indicator. If you look at the LED's mechanical drawing, you will see a chamfer on the cathode side. Add a silkscreen dot on the pattern by drawing a circle. Adjust the circle parameters to your liking.
9. Give your component a name and refdes letter. Since this is a LED, use D for the refdes letter.  
   
10. You can also add dimensions (using the Place Dimension button on the toolbar) to compare your pattern with the datasheet. Dimension edges will automatically snap to relevant features like pad edges and centers.  
    

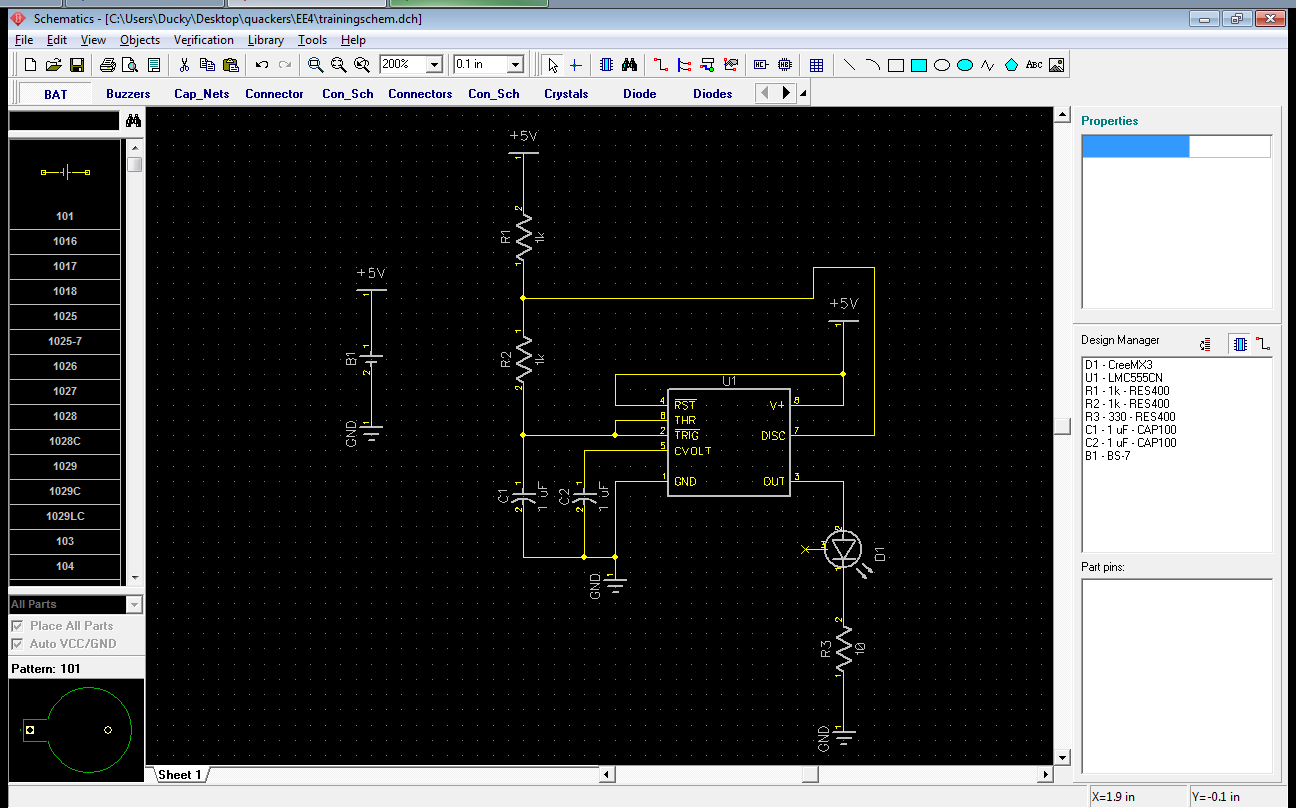
### Component creation

You have a pattern, now you need a schematic symbol.

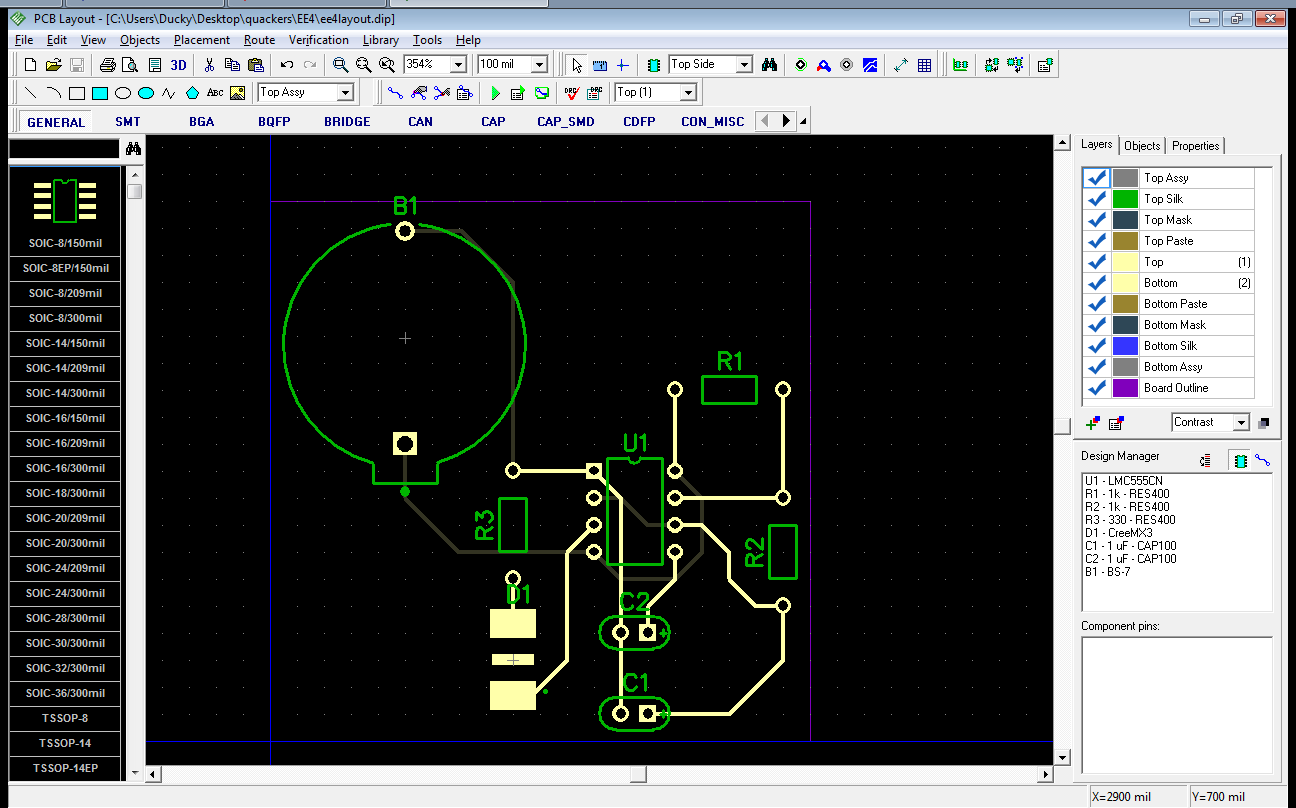
1. Start the DipTrace component editor.  
   
2. Often, it's easier to base your part off an existing schematic symbol, especially for complex parts like LEDs.
   1. Open the existing "discrete.eli" library.
   2. Select "LED" from the list of patterns. It should open in the main view.
   3. Select all (Ctrl+A) and copy (Ctrl+C).
3. Create a new library by clicking Library->New.
4. Paste the copied LED schematic. Note that, for whatever reason, the schematic is not aligned. Nudge it up and to the right by 0.031 in. A simple way to do this is to set the grid to 0.031 in, select everything, and hit the up arrow and right arrow once.  
   
5. The original LED has only two pins: anode and cathnode. However, we also have a heatsink pin. Click the "Pin" button on the toolbox and add a new pin.  
     
   Position the pin such that it is below the LED and faces downward. You can rotate components by using the right click menu, or pressing Ctrl+R.  
   
6. Go into the Pin Manager and define the new pin:  
     
   Here is where you can also activate the "Show Name" option for each pin. Experiment with it. Although not particularly useful for a LED, this is invaluable on chips.
7. Go into the Pattern dialog and define the associated pattern (what you just created before.) You will need to load your pattern library by using the "Add" button.  
     
   Ensure that the pins match as expected.
8. Give your component a name and refdes letter.  
   

### Revising the schematic

You build the component, now put it into your schematic.

1. Remove the old LED in your schematic and add your new one.
2. You may also want to scale down the resistor to the LED. If the input voltage was 5v[[3]](#footnote-3) and you wanted to run the LED at 150 mA, the resistor would be set at .
3. There is the issue of the Heatsink pad. We will not electrically connect it[[4]](#footnote-5), so right click the pin and select "Not connected." This is typically good practice as it eliminates any ambiguity as to whether a pin is intentionally not connected, or left unconnected by mistake.  
     
   

### Revising the board

1. Forward annotate your schematic modifications to your PCB layout.
2. By default, DipTrace snaps components to the grid based on the center of their Pin 1. This works well for through-hole components, but can cause strange behavior for surface-mount components whose dimensions aren't always a multiple of 100mil. Right click the new LED and choose Snap to Grid->By Origin.
3. Route your new component.  
   
4. You're done. You can re-export Gerbers if you want the extra practice.

# Design Software

CalSol uses DipTrace, however, there are more. A brief listing follows (for a more complete listing, see Wikipedia: <https://en.wikipedia.org/wiki/Comparison_of_EDA_software>):

## Free & Open Source

In general, free & open source software, especially in the niche EDA area, is less well-developed than their commercial counterparts. However, they are definitely viable, and you do have the advantage of getting access to the source code and there is no risk of vendor lock-in or ridiculous DRM schemes.

* gEDA and PCB - a GPL-licensed board design suite.
* KiCAD – another GPL-licensed board design suite.

## Commercial

### Software you can afford

Hobbyist-grade and above. Most have free trial versions, either time-limited to pin-limited.

* DipTrace - what CalSol uses.
* EAGLE - pretty much the hobbyist standard, but has unpleasant DRM[[5]](#footnote-6).

### Software for the 1%

You probably can't afford it. However, businesses are willing to put down the dough for the features and support offered. In some cases, the software may even replace a human board designer, which can save the company serious money.

* Altium Designer
* Cadence Allegro, Cadence OrCAD
* Mentor Graphics PowerPCB

# PCBs and You

CalSol gets its circuit board manufacturing sponsored by Advanced Circuits. However, there are many board houses out there, each with its advantages and disadvantages. A brief listing follows of places you can get one-of boards relatively cheaply. If you are looking for volume production, you will have many more options.

## Advanced Circuits

<http://www.4pcb.com>

Advanced Circuits is a US-based commercial board house. They are more focused towards business customers and volume production, but do have a special student offer.

<http://www.4pcb.com/index.php?load=content&page_id=134>

* Turn-around: 5 days, then shipping
* Design rules: 2 layers, 6mil/6mil trace width/spacing, 15mil minimum drill
* Cost: $33 each, maximum board size is 60 sq in
* Many other options available at different pricing structures

## BatchPCB

<http://batchpcb.com/>

BatchPCB is a SparkFun PCB service. They take peoples orders and combine them onto one order, which they send out to manufacture.

* Turn-around: about 2-3 weeks
* Design rules: 2 layers, 8mil/8mil trace width/spacing, 20mil minimum drill
* Cost: $2.50 per sq. in., plus $10 setup fee
* 4 layer boards also available, for an additional fee

## Dorkbot PDX / OSH Park

<http://oshpark.com/>

* Turn-around: about 2 weeks
* Design rules: 6mil/6mil trace width/spacing, 13 mil minimum drill
* 2-layer cost: $5 per sq. in., you get 3 copes
* 4-layer cost: $10 per sq. in., you get 3 copies

## SeeedStudio

<http://www.seeedstudio.com/depot/fusion-pcb-service-p-835.html?cPath=185>

* Turn-around: about 1 week for manufacturing, then shipping from China
* Design rules: 2 layers, 6mil/6mil trace width/spacing, 0.3mm minimum drill
* Cost: $9.90 for 10 copies of a 5cm x 5cm board, other sizes also available

... and many others - Google if you are truly interested ...

# Sources & Further Reading

These references were used in the creation of this document. If you want to learn more, take a look at these sites!

<https://en.wikipedia.org/wiki/Printed_circuit_board>

<https://en.wikipedia.org/wiki/Schematic_capture>

<http://www.alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf> - contains many good tips on board layout

<http://batchpcb.com/index.php/Faq>

1. You will not reach the full power output of the LED by using only the 555 timer - it sources at most 150mA. You will need a transistor amplifier to use the LED to its full, blinding, potential. [↑](#footnote-ref-1)
2. It is also unlikely that the small coin cell battery can supply all the current required for the LED. The design will likely need a larger battery or a dedicated power supply. [↑](#footnote-ref-2)
3. The CR2032 battery in the holder actually runs at around 3.0v, but that's not even enough voltage to overcome the forward voltage drop of the high-power LED. [↑](#footnote-ref-3)
4. A heatsink does not need to be electrically connected, but it should be connected to a large area of copper to help dissipate heat. Typically, heatsinks can be electrically connected to ground. This is especially useful if your board has a ground plane. You should double check the datasheet to see if a heatsink pin is internally connected anywhere. [↑](#footnote-ref-5)
5. <https://groups.google.com/forum/?fromgroups=#!topic/comp.arch.embedded/95ToLSa1nhg> [↑](#footnote-ref-6)