

# Turn your EAGLE schematic into a PCB

by [westfw](#) on September 29, 2006

## Table of Contents

|  |    |
|--|----|
| Turn your EAGLE schematic into a PCB .....                         | 1  |
| Intro: Turn your EAGLE schematic into a PCB .....                  | 2  |
| Step 1: Starting from the schematic... ..                          | 2  |
| Step 2: Menu commands used .....                                   | 3  |
| Step 3: The untouched PCB design .....                             | 3  |
| Step 4: About Board "layers" .....                                 | 4  |
| Step 5: Move the components into the legal area .....              | 6  |
| Step 6: Shrink the board outline a bit .....                       | 6  |
| Step 7: Start placing the components .....                         | 7  |
| Step 8: Check signals to see how they'll route .....               | 8  |
| Step 9: Load design rules .....                                    | 9  |
| Step 10: Fix incorrect package .....                               | 9  |
| Step 11: Try the autorouter .....                                  | 10 |
| Step 12: Route remaining tracks manually .....                     | 10 |
| Step 13: Add power plane polygons .....                            | 12 |
| Step 14: Add V+ Polygon .....                                      | 13 |
| Step 15: Neaten up: smash package text .....                       | 14 |
| Step 16: Neaten up; move traces .....                              | 15 |
| Step 17: Fixing an OOPS! .....                                     | 16 |
| Step 18: Neaten up: Allow for alternate packages and options ..... | 17 |
| Step 19: Do Design Rule Check .....                                | 18 |
| Step 20: Output using Exported images .....                        | 18 |
| Step 21: Other useful menu icons .....                             | 19 |
| Step 22: Useless commands .....                                    | 20 |
| Related Instructables .....  | 21 |
| Comments .....   | 21 |



Author:westfw

Middle aged geek

username also works at yahoo.com, mac.com, comcast.net, wharton-10.arpa

## Intro: Turn your EAGLE schematic into a PCB

In a previous Instructable, I provided an intro to schematic entry using CadSoft's EAGLE editor.

In this instructable, we'll make a printed circuit board from that schematic

I guess I should say that we'll make a PCB **DESIGN** ; making the physical board is a different task, and there are a lot of tutorials on the net (and even some instructables) on making the board after you have the design.

Cadsoft EAGLE generic information:

Cadsoft EAGLE is available from <http://www.cadsoftusa.com/>

Cadsoft is a German company that is a veritable mecca of software distribution enlightenment. In addition to the reasonably-priced professional PCB design packages (\$1200), they have freeware, lite, non-profit, and other intermediate licenses. Their software runs under windows, linux, and MacOSX. It's slightly quirky, with a steep (but not too high) learning curve on the front end, but from most reports it is not any more so than other professional CAD packages. They have online support forums that are active from both the company and other users, the package is under current development and gets better with each release. A number of PCB fabricators will accept their CAD files directly. It's good stuff.

Use it. Propagate it. Buy it when you "go pro."

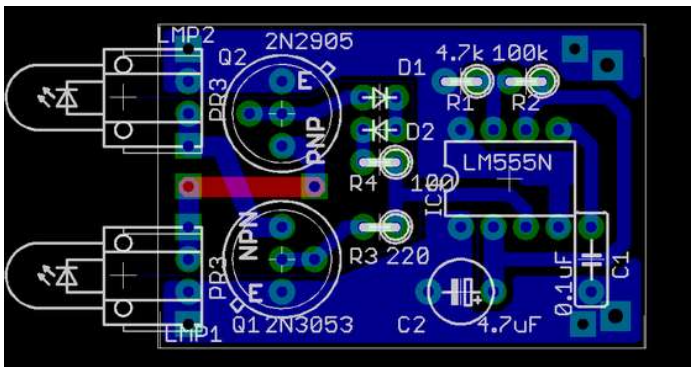
See also:

[Schematic Entry](#)

[Creating Library parts](#)

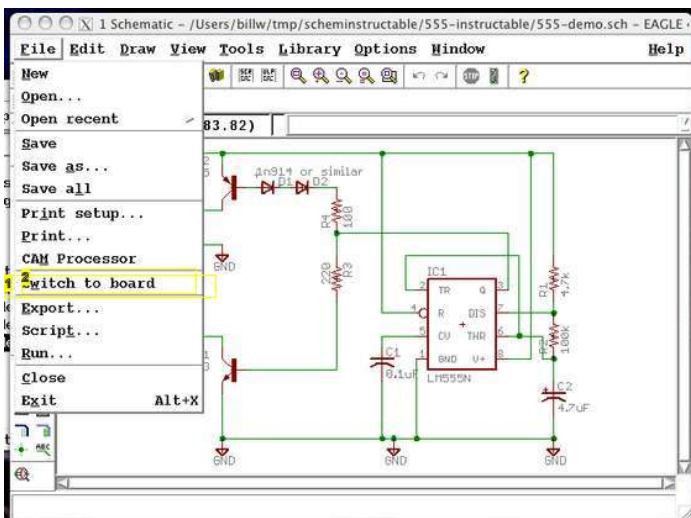
[Design rule modification](#)

[Send CAD Files to manufacturers](#)



## Step 1: Starting from the schematic...

So this is the schematic we have from the Schematic Instructable . Up in the file menu, there's a "Switch to board" selection. If we do that from a bare schematic, it will offer to create the board from the schematic for us (say "yes"), and then leave us sitting in the Board Editor.



### Image Notes

1. Create board if it doesn't exist.
2. Switch to Board

## Step 2: Menu commands used

The Board Editor looks a lot like the schematic editor, with some different commands. Here's a summary of the iconic commands that I use in this instructable, and some brief summaries:

**INFO** Shows information about an object (component, signal, trace, etc.)

**MOVE** Allows components to be moved (same as schematic.)

**GROUP** Groups a collection of objects into a "group" that can be manipulated simultaneously.

**DELETE** Delete an object. Items created in schematics need to be deleted there.

**SMASH** Separate the text labels of a part from the part itself, so they can be moved independently.

**BREAK** Add a corner to a line (or trace.)

**ROUTE** turn an airwire into a trace

**LINE** draw lines (usually in non-copper layers. ROUTE is for drawing copper.)

**VIA** create a hole and pad associated with some signal. (actually, we'll use a text command.)

**HOLE** a hole that isn't associated with a signal, ie for mounting.

**RATSNEST** recomputes airwires and polygons, eg after components have been moved.

**CHANGE** changes an object's properties.

**RIPUP** changes a routed trace back to an airwire. Sorta equivalent to "delete" for traces.

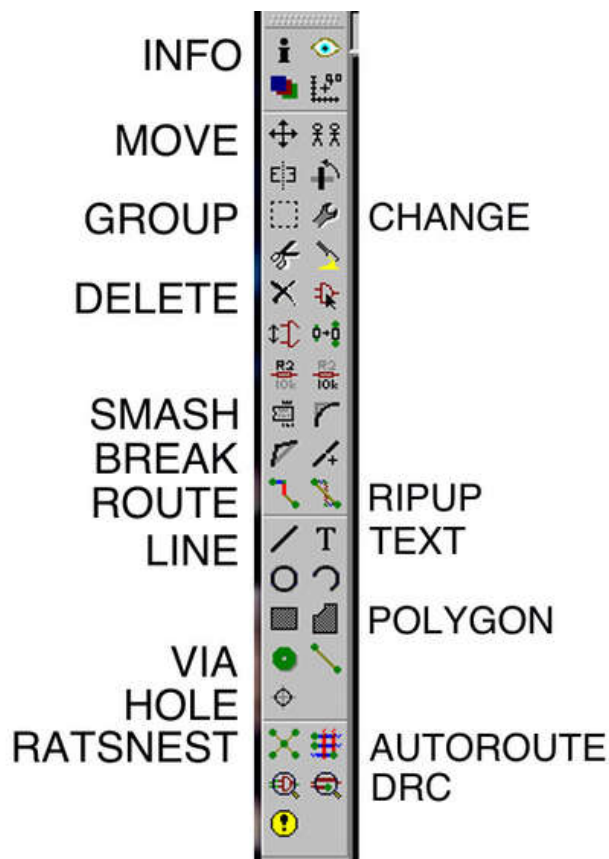
**TEXT** add text

**POLYGON** create a polygon (actually, we'll use a text command.)

**AUTOROUTE** invoke the autorouter.

**DRC** invoke the Design Rule Check and parameter setting.

I'll describe the remaining icons toward the end, and assign them "useful" or "useless."



## Step 3: The untouched PCB design

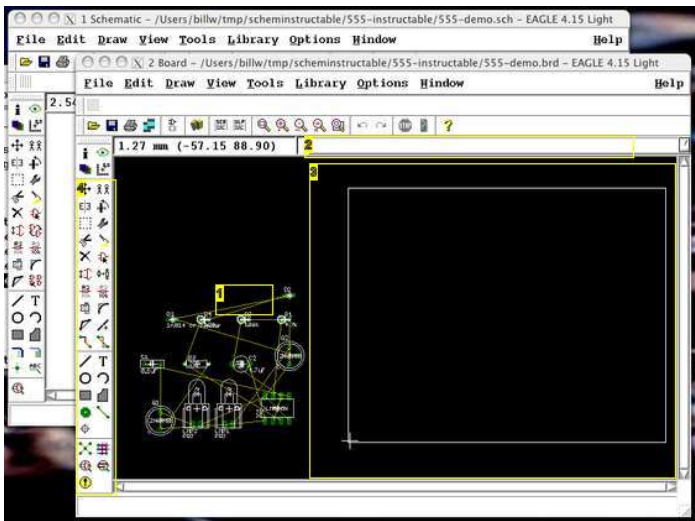
This is what the newly created board design will look like. All your components will be in a clump over to the left of the origin, and there will be a frame that marks the allowed size of a board when using the freeware or "Lite" versions of EAGLE (80x100mm). All the component pads will have to be inside that outline when you move them around, although you can cheat a bit and have traces or board outlines that exceed the board size limit. This has the annoying side-effect that if you pick up a component from it's original location, you can't put it back down outside the outline (however, you can use ESC to abort the move, and the component will revert to its original location.)

Ok, a few definitions are in order

All the signals you created in the schematic are currently **AIR WIRES** ; thin yellow lines that are drawn in the shortest possible way, crossing each other as needed. They stay connected to component pins even when you move the component around. The **RATSNEST** command recomputes and redraws these after you move things around (and, say, make two connected pins closer together than they used to be.)

**ROUTING** a signal consists of turning an airwire into an actual copper trace on some layer(s) of the board, and positioning that trace so that it doesn't short against other traces on the same layer of the board. The Freeware version of Eagle only supports a TOP and BOTTOM layer, and as hobbyists we have motivation to try to use only ONE layer. A signal can transition from one layer to another using a **via** , which is a conducting hole, sorta like a jumper (and we'll use jumpers to implement the top level of the board if we can make the board *mostly* single-sided.)

Creating the PCB design consists of placing all the components in logical places, and routing all the airwires in a way that allows the design to work.



#### Image Notes

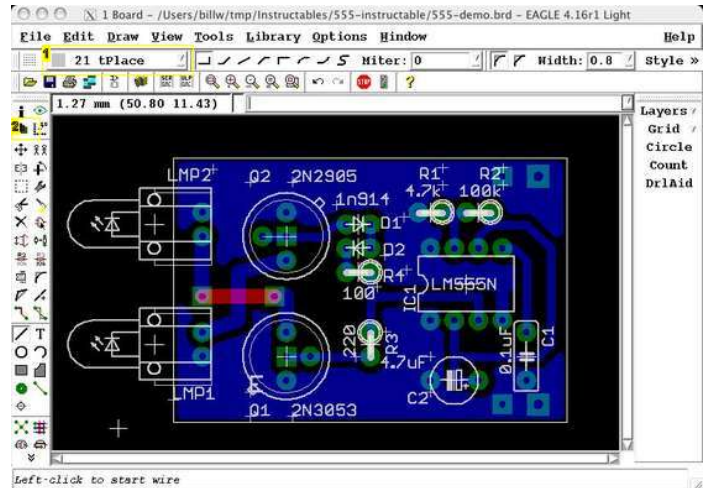
1. AIR WIRES
2. Text command area
3. Freeware (or "lite") legal board area.
4. Icon commands

### Step 4: About Board "layers"

The Eagle Board editor has MANY more layers than the schematic editor. A confusing multitude of layers. Most of the drawing commands have a layer-selection pulldown menu that you can use for specifying which layer you want to draw on (exceptions include objects like vias that span multiple layers.)

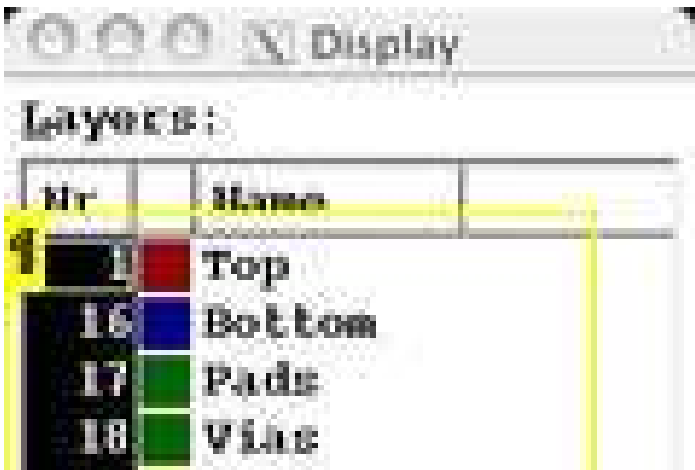
Here are some of the more important layers:

|            |                         |   |
|------------|-------------------------|---|
| 1          | Top                     | Top (component) side copper traces              |
| 2-15       |                         | Other copper layers (not supported in "lite")   |
| 16         | Bottom                  | Bottom ("Solder") side copper traces            |
| 17         | Pads                    | Pads for component mounting (all copper layers) |
| 18         | Vias                    | Holes with copper that span multiple layers     |
| 19         | Unrouted                | Airwires that haven't been routed yet           |
| 20         | Dimension               | The Board outline                               |
| 21, 25, 27 | tPlace, tNames, tValues | Top layer silkscreen                            |
| 22, 26, 28 | bPlace, bNames, bValues | Bottom layer silkscreen                         |
| 23, 25     | tOrigins, bOrigins      | "origins" for top and bottom components         |
| 51, 52     | tDocu, bDocu            | like xPlace, but not copied to real silkscreen  |

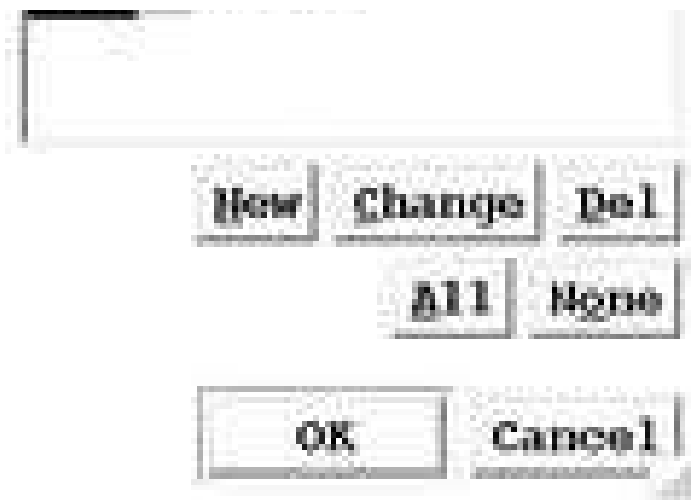


#### Image Notes

1. Layer pulldown controls which layer you are currently drawing on
2. LAYERS command controls which layers are currently displayed



|    |           |
|----|-----------|
| 19 | Unrouted  |
| 20 | Dimension |
| 21 | tPlace    |
| 22 | bPlace    |
| 23 | tOrigins  |
| 24 | bOrigins  |
| 25 | tNames    |
| 26 | bNames    |
| 27 | tValues   |
| 28 | bValues   |
| 29 | tStop     |
| 30 | bStop     |
| 31 | tCream    |
| 32 | bCream    |
| 33 | tFinish   |
| 34 | bFinish   |
| 35 | tGlue     |
| 36 | bGlue     |
| 37 | tTest     |
| 38 | bTest     |
| 39 | tKeepout  |
| 40 | bKeepout  |
| 41 | tRestrict |
| 42 | bRestrict |
| 43 | wRestrict |
| 44 | Drills    |
| 45 | Holes     |
| 46 | Milling   |
| 47 | Measures  |
| 48 | Document  |
| 49 | Reference |
| 50 | dxf       |
| 51 | tDocu     |
| 52 | bDocu     |



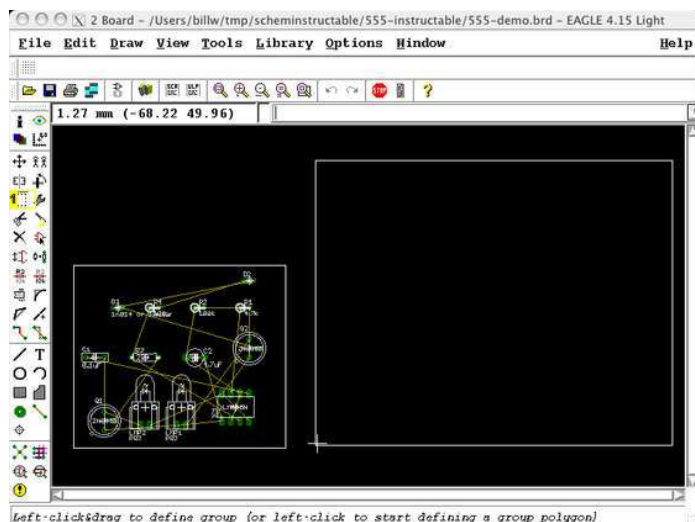
#### Image Notes

1. Important layers for drawing.
2. Layers normally used within components, or generated automatically

### Step 5: Move the components into the legal area

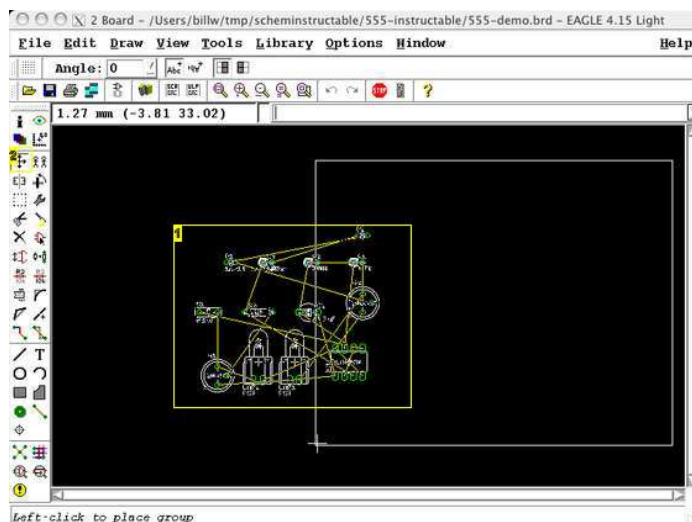
The first thing we want to do is move at least some components into the legal board area where we can work with them. If you have a particularly large board with many components, you might want to do this a section at a time. For this sample board, we have plenty of room and we can move them all at once, using the group-move feature.

Select the GROUP icon, then click and drag to make a rectangle that goes all the way around the components. Then select the MOVE icon and RIGHT click (right clicking selects the group instead of a single component) and drag the set into the board outline. Use the ZOOM button to tighten the view.



#### Image Notes

1. GROUP icon



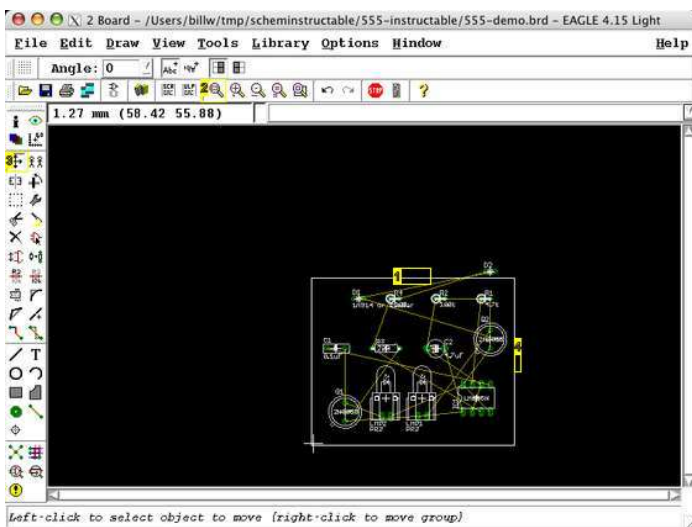
#### Image Notes

1. Right click and drag the group of components.
2. MOVE icon

### Step 6: Shrink the board outline a bit

The full legal side of the board is bigger than we need. Shrink the outline by using the MOVE tool. Click on the center of the top horizontal line (which selects the whole line instead of an endpoint) and move it down, Then click on the center of the rightmost vertical line and move it leftward. Clicking near the center of a line moves the whole line. Clicking near a vertex moves only the point.

It doesn't have to be perfect at this point; we're mostly looking for a better view for the next steps. (Oh yeah - click the zoom button to re-zoom the window on the smaller outline.)



#### Image Notes

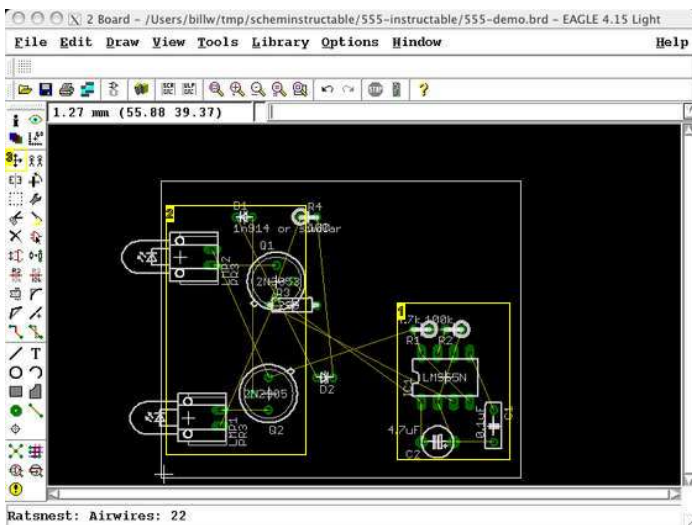
1. click here and drag
2. ZOOM to fit
3. MOVE the outline
4. click here and drag

### Step 7: Start placing the components

Now we need to move the components to (near) where we want them on the final board. OR we want to move them to sensible places that will make the placement of traces easier. A lot of the "ART" of making PCBs (and especially Single Sided Boards) lies in finding "good" places for the components.

In general, you can start by placing the components similar to how they appear on the schematic. (This breaks down when a chip has multiple gates, or the diagram in the schematic symbol has vastly different pin placement than the actual chip, but it's a good place to start for discreets and simple components. The worst that will happen is that you'll have a layout that makes sense, even if it doesn't route well...)

In this case, I put the power output transistors near lamps that they're associated with, and I looked on the web for a 555 layout that would work well (for the longest time, I tried to do boards with the timing cap placed near the timing resistors, and I always needed a jumper. Sigh.) ("Let no one else's work evade your eyes.")



#### Image Notes

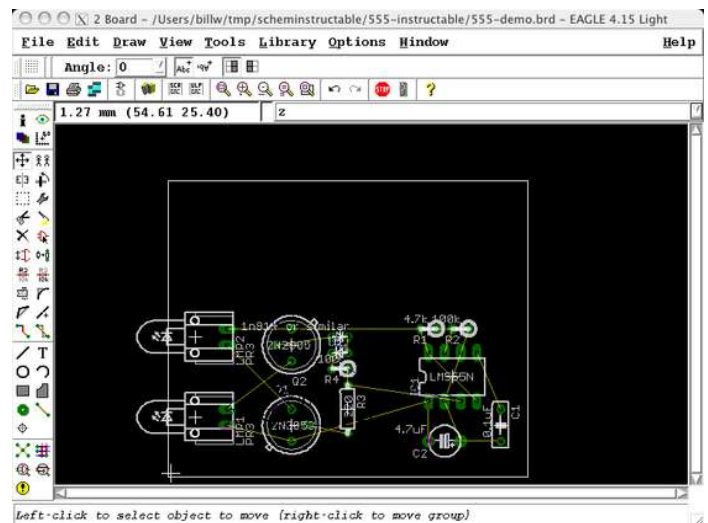
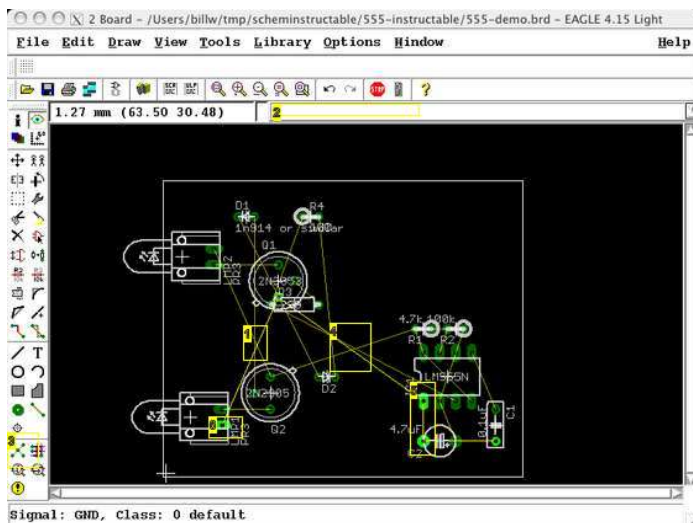
1. Find this placement on the web :-)
2. Lamps and transistors placed next to each other.
3. MOVE command



## Step 8: Check signals to see how they'll route

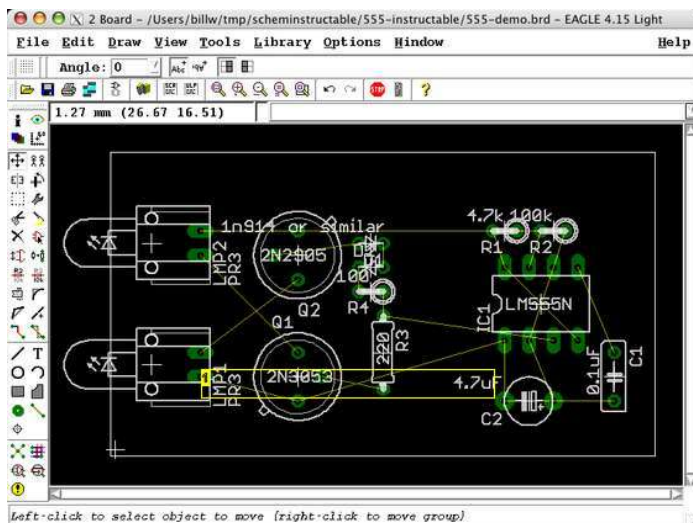
One way to get hints on part placement is to look at some significant signals to see whether they have nice straight paths, or whether they zigzag all over the board. First use the RATSNEST icon/command to have EAGLE recompute the airwires. The way things are now, I have nice straight connections from the transistors to the lamps, but if I type "show gnd" in the command line, I see that this is at the expense of making the ground signal zigzag. So I swap the transistors because GND is more important to have straight. (IMHO, YMMV, etc.) (This ends up putting the transistors near the supplies that they switch, rather than near the lamps that they switch to, so that still makes sense from a circuit point of view too.)

After the rest of the components are placed in ok-looking relative locations, I can squeeze them together again (manually, moving them one at a time; no magic command for this!) and shrink the board outline some more.



### Image Notes

1. highlighted GND signal path
2. type SHOW GND
3. RATSNEST command
4. highlighted GND signal path
5. highlighted GND signal path
6. highlighted GND signal path



### Image Notes

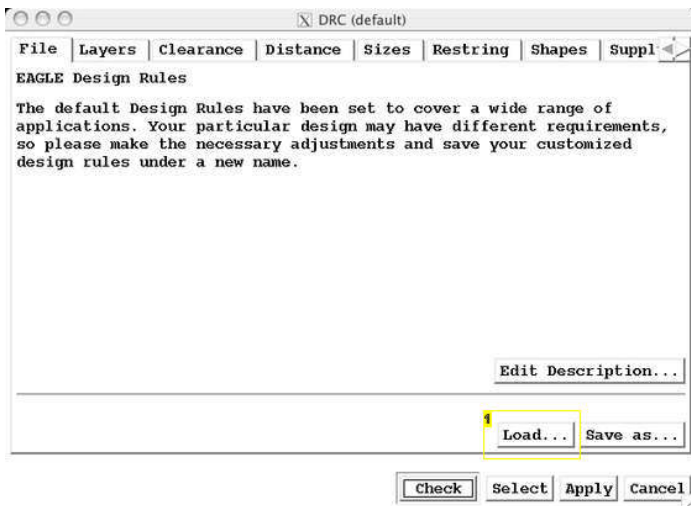
1. New GND path



## Step 9: Load design rules

Since we're hobbyists, we want to make our board with wide traces and big spaces (see <http://www.instructables.com/id/EZVIGHUBGCEP287BJB/>) So we'll load up that set of hobbyist design rules before we start laying out tracks.

Click the Design Rule Check icon and use the LOAD button to load `hobby.dru` from my other instructable. Or you can modify values manually and individually, of course. Or leave them as is...



### Image Notes

1. Use the LOAD command to load hobby design rules.

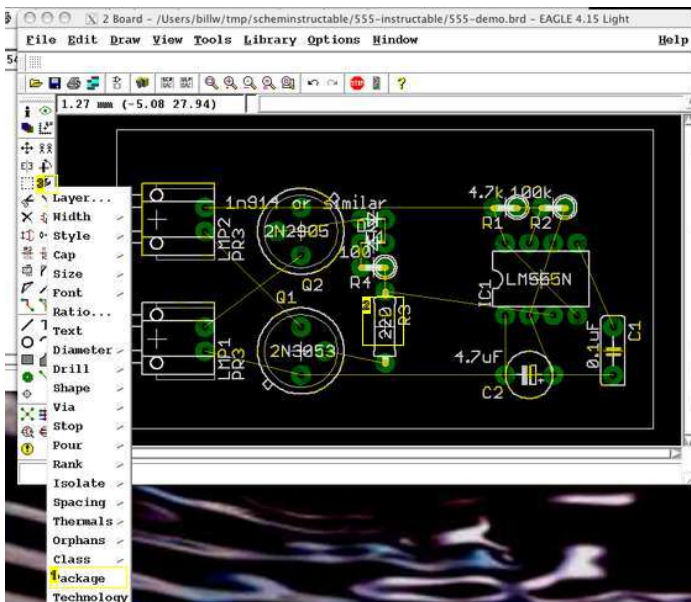
## Step 10: Fix incorrect package

You can see how the design rule change has already modified the board. Pads are bigger, and they're all round.

You'll also notice that one of the resistors is set as a non-vertical package, unlike the rest. This was probably an error in the schematic entry, and it didn't matter when all we had was the schematic. Now that we are making the board, we want to change the package as appropriate.

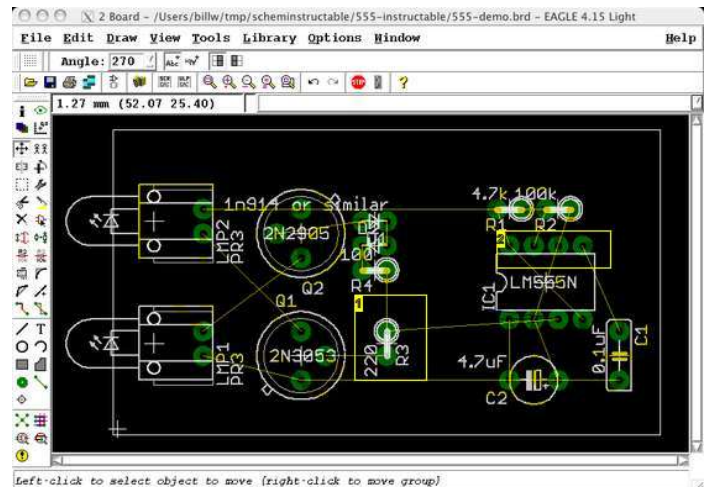
When you select the change->package tool and click on the part to change, you'll be shown a list of all the legal packages for that part (these should be the same ones that showed up in the schematic "add" dialog)

There are other ways to enter a "change" command in the text command entry area that you'll want to look into if you need to change a lot of devices to a particular package, so you can skip going through the list for each one. Something like "change package 'R-US/0207/2V'", and then just click on each component.



### Image Notes

1. Change package
2. Then click this resistor
3. CHANGE



### Image Notes

1. The package changed!
2. Pads round now. And BIG.

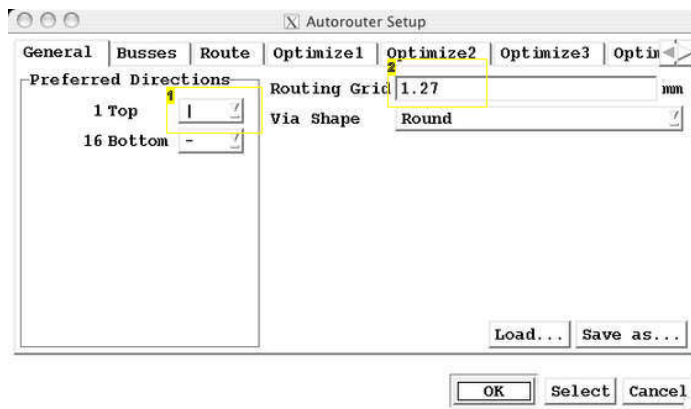
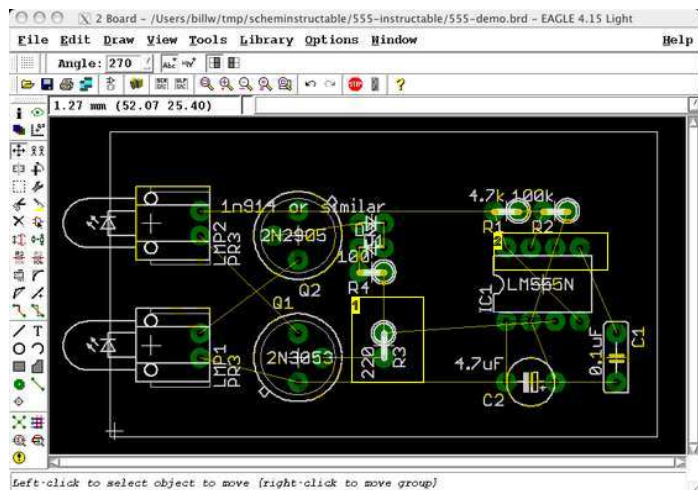
## Step 11: Try the autorouter

Now we'll see if the autorouter can do some of the work for us. The EAGLE autorouter isn't the best in the world, but even when it does a "bad" job, it will give us some general hints on how things need to look, or where the trouble spots are.

Click the AUTOROUTE icon, and a dialog box will pop up. The default parameters will produce a double sided board, and we want to at least TRY to make a single-sided board, so the first thing to do is set the preferred direction for the TOP layer to NA (Not Applicable.)

The other thing you may need to change is the routing grid. This defaults to the same default grid as the board layout editor in general: 0.05 inches (1.27mm, since I have my editor set up in metric.) Since this particular board has big parts, and we haven't moved any off the default grid, we're ok with that value. If you have SMT components or have moved things around on a finer grid, you might have pads that are not on the routing grid, which the autorouter doesn't like much ("unreachable pad", etc) You can make the grid very small, but it will take longer. IMO, it's better to start with a coarse grid and halve it each time it looks like routes fail because the grid is too large.

Also note that the autorouter obeys the board dimension lines, so if you haven't moved them close to your components, you might have traces travel all over the board. Or if you've moved the outline too close to the pads, you may have prevented traces from going places they need to go.

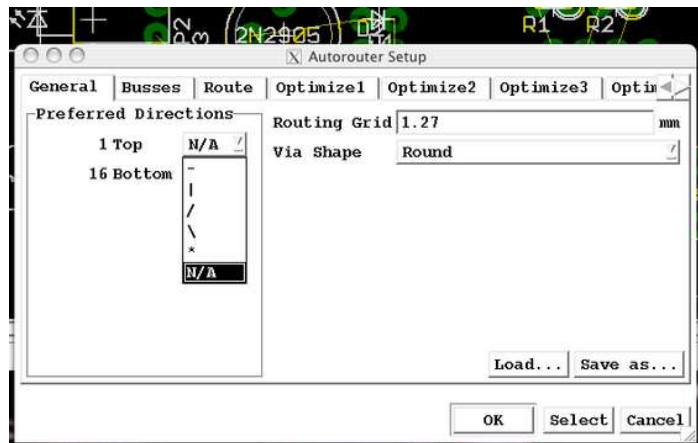


### Image Notes

1. Change to NA.
2. Maybe change grid too.

### Image Notes

1. The package changed!
2. Pads round now. And BIG.

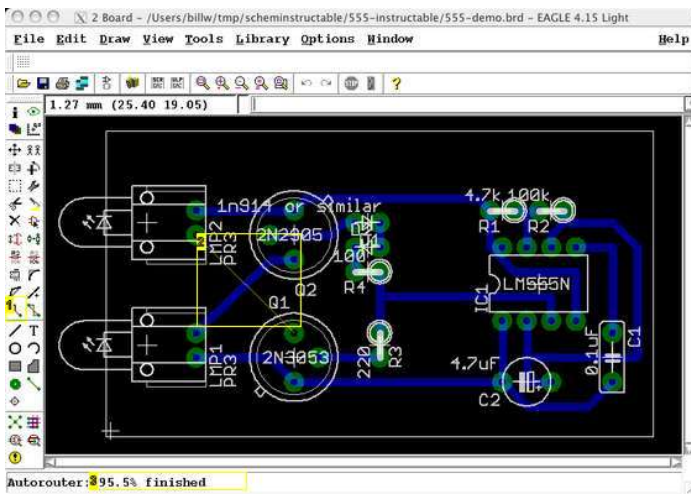


## Step 12: Route remaining tracks manually

The autorouter did a pretty nice job here. There's only one trace left.

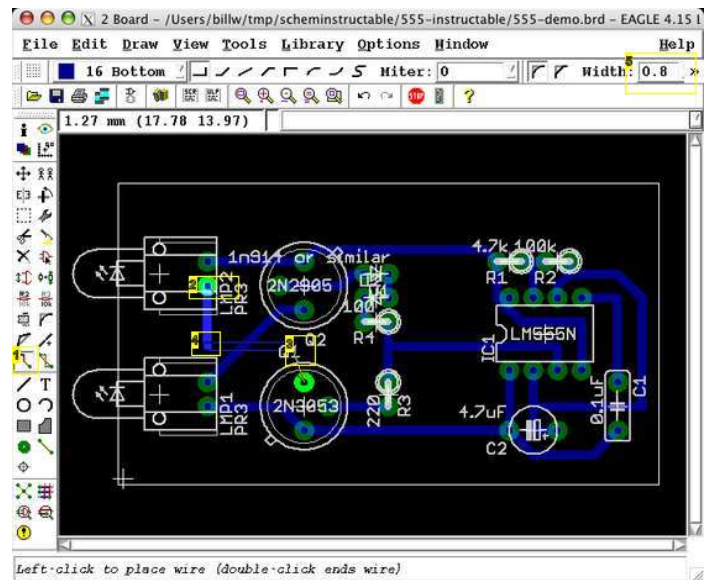
There are a couple of ways we could route this signal manually, including some snakey routes between transistor pins that the autorouter didn't use because of the design rules we specified. This is a relatively high current trace, and I decided that I won't manually violate the design rules either. Instead, I'll use a jumper wire on the component side, which I can model in EAGLE as a top-side trace.

Select the ROUTE tool and click on an endpoint of an unrouted (yellow) airwire, and you can position a trace pretty much anywhere you want, selecting width, layer, and type of bend from the menu bar as you go along. This is shown in the succession of pictures in this step.



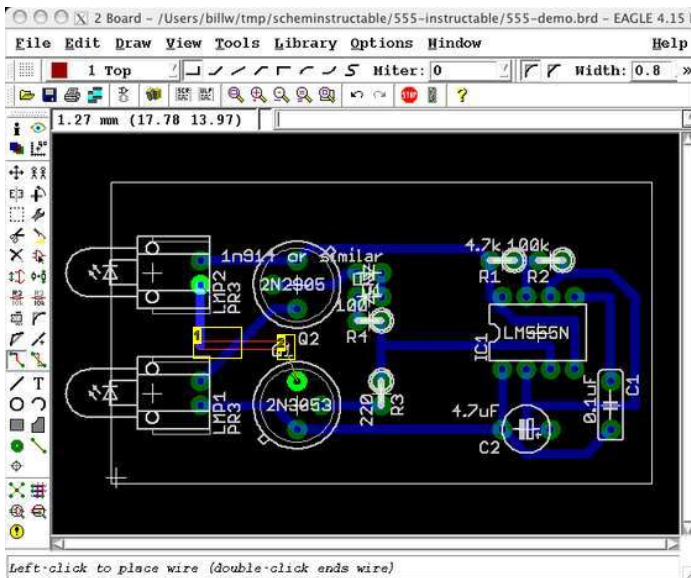
#### Image Notes

1. ROUTE icon
2. One signal couldn't be routed
3. One signal couldn't be routed



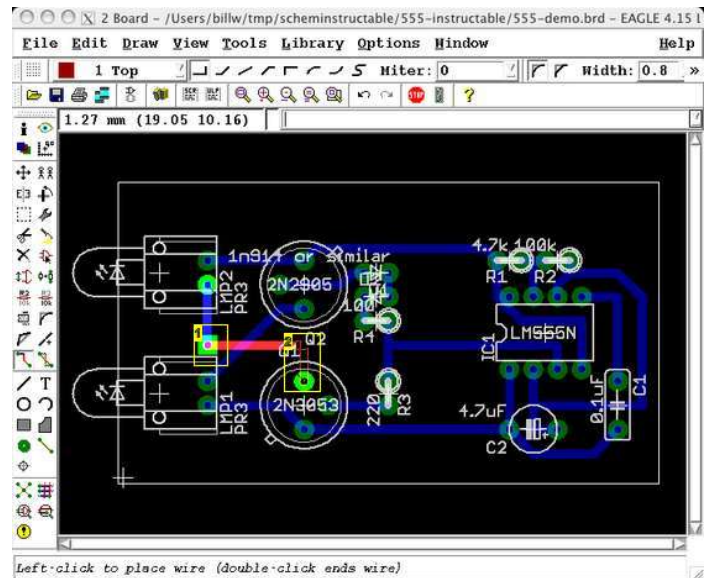
#### Image Notes

1. ROUTE command
2. Click here to start route.
3. Move to here and click middle button
4. click here to terminate segment of route that is on the bottom layer.
5. You may need to set this.



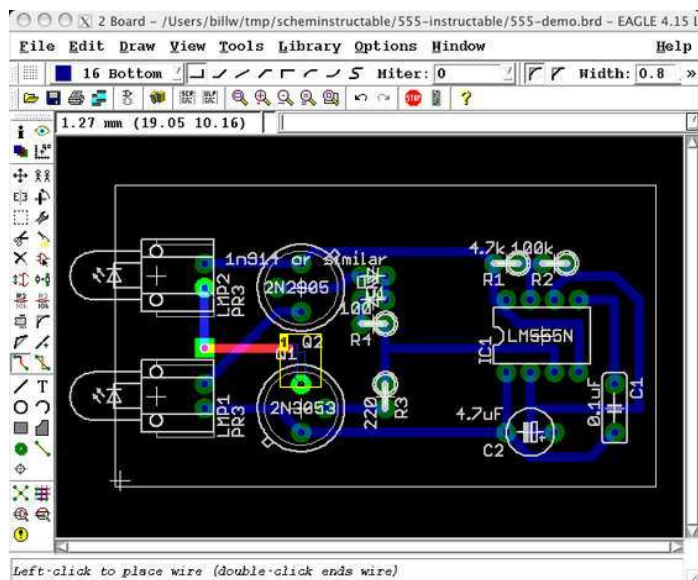
#### Image Notes

1. trace is now moved to top.
2. Click here to terminate the top-layer segment.



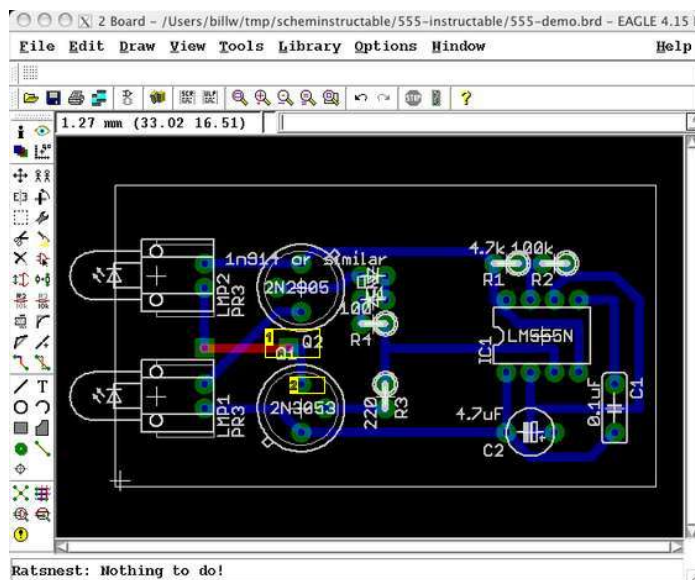
#### Image Notes

1. via created automatically!
2. move to final destination, use middle button again.



#### Image Notes

1. Now our trace is on the bottom again



#### Image Notes

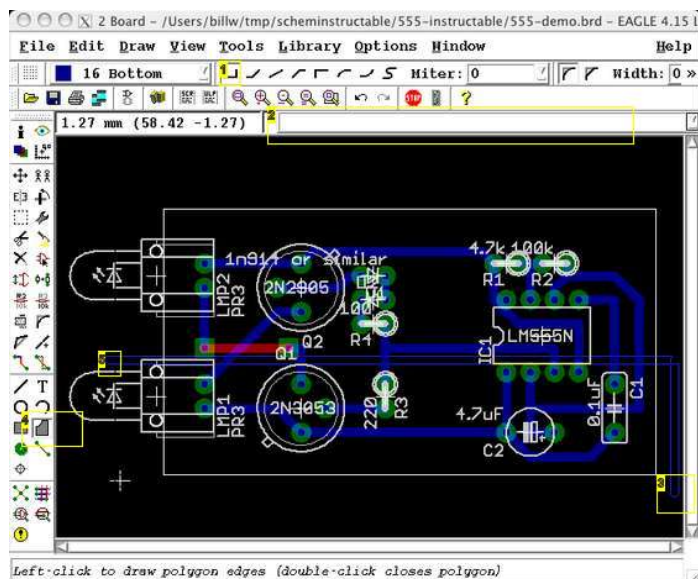
1. Another automatically create via
2. trace terminates here

### Step 13: Add power plane polygons

"Power planes" are large areas of copper that carry an actual signal, usually power and ground. On multi-layer boards, it's common to have entire layers mostly dedicated to such a power plane. Even on a single layer board there are some advantages to doing something similar:

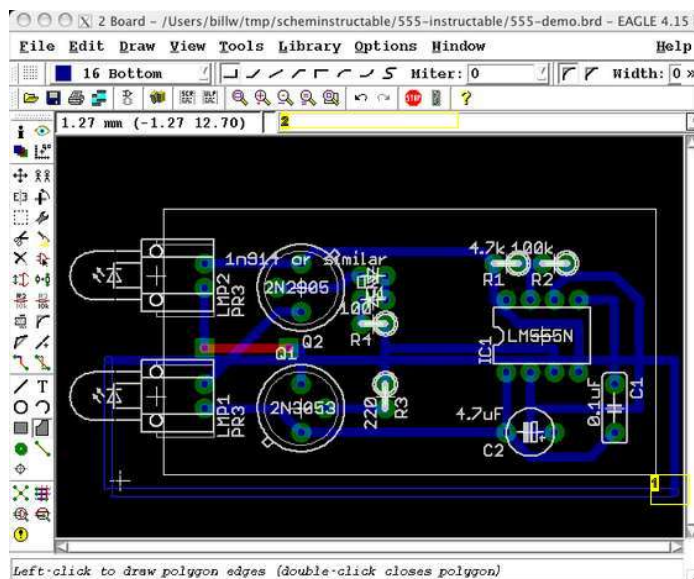
- 1) Use less etchant
- 2) carries heavier current, just in case
- 3) makes it easier to attach test leads
- 4) acts as a sort of "static barrier" to fingers

In EAGLE, such large signal areas are drawn with the **"polygon"** command. There is an icon on the toolbar for drawing polygons, but it will create polygons associated with a new signal, and I find that when creating a polygon for an existing signal, it's easier to type the text form of the command in the text command area. To create a polygon attached to a signal named 'gnd', type **"poly gnd"**. By giving it a signal name in the command, the polygon will automatically be connected to that signal. (If you draw a polygon with the icon, you can connect it to a signal later by using the "name" command to rename the polygon. (however, after this, you can't rename the polygon again without renaming the signal as well.))



#### Image Notes

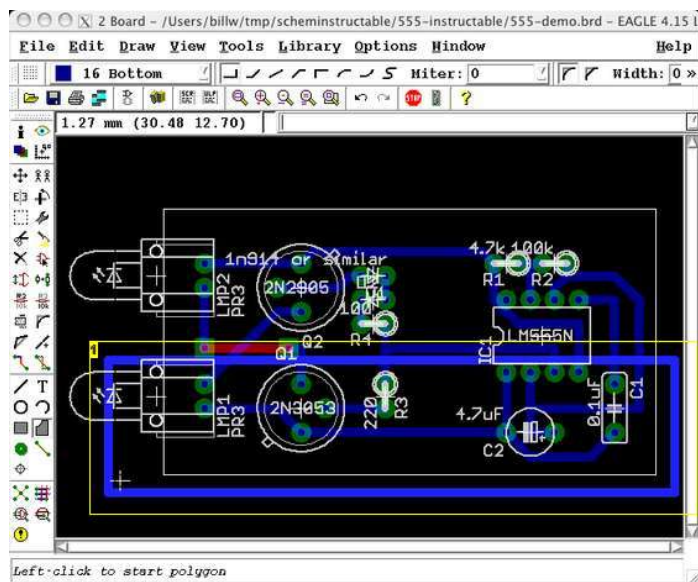
1. wirebend determines shape. Right click to change.
2. Type "POLY GND" here to start the polygon.
3. Click here to start polygon edge
4. This is the polygon tool, but it's easier to create polygons associated with a signal from the text command area.
5. Click here for next vertex



#### Image Notes

1. Move back to here and click to terminate polygon (cause you intersected yourself)
2. type "polygon gnd"





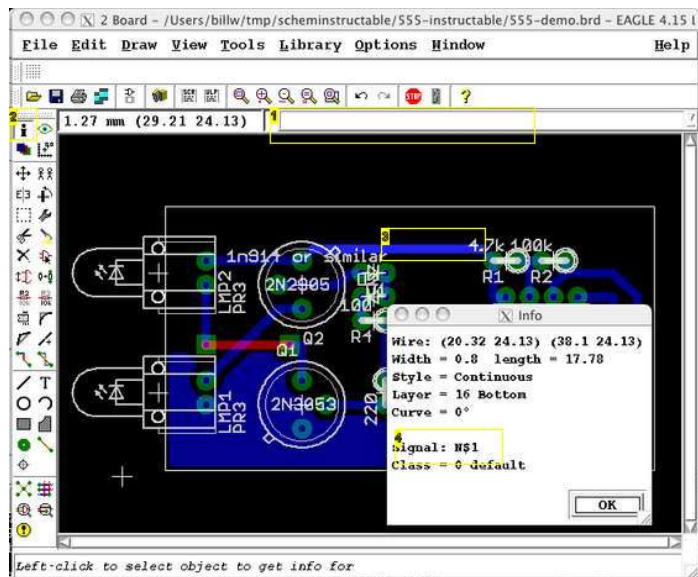
#### Image Notes

1. Only the polygon outline shows up.

### Step 14: Add V+ Polygon

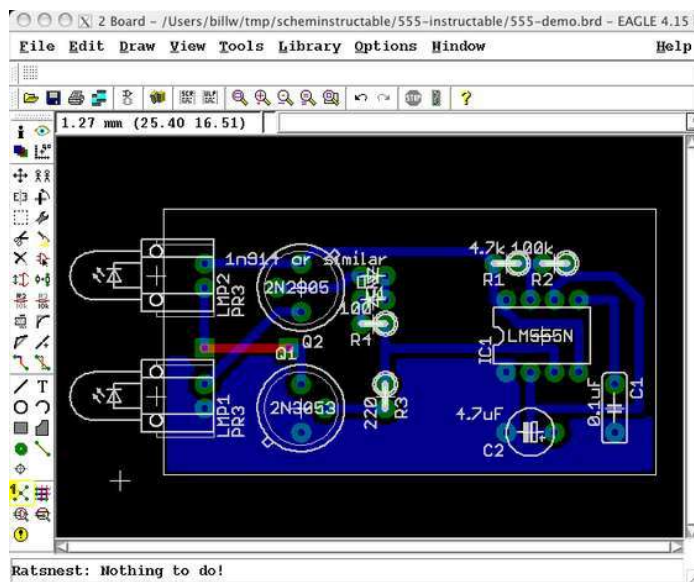
Now we'll repeat the process for the positive voltage. However, we never named that signal when we drew the schematic, so it will have some random name like "N\$23"; We can use the "INFO" command to find the signal name to use when we draw the polygon, after which it's the same as drawing the GND polygon.

In this case, the V+ signal is named n\$1, so we type "poly n\$1 "



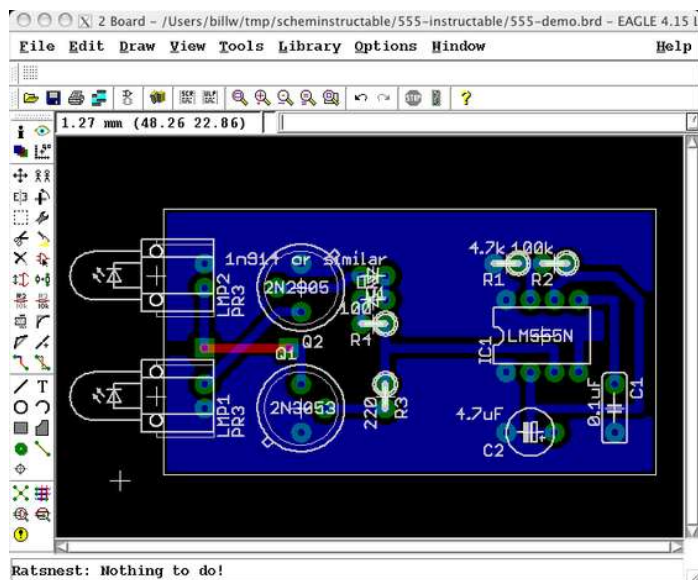
#### Image Notes

1. Now that we know we want signal n\$1, type "POLY N\$1" here.
2. INFO command
3. click on signal
4. This tells us the signal name. The other info is sometimes useful too.



#### Image Notes

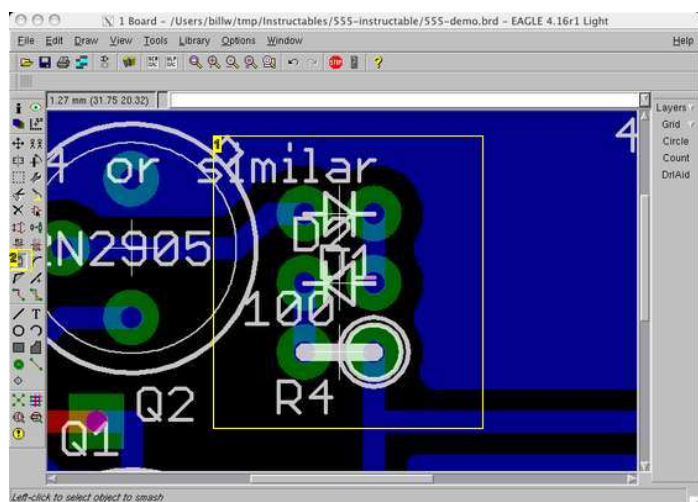
1. The AIRWIRE command fills in the polygon. Why? Who knows!



### Step 15: Neaten up: smash package text

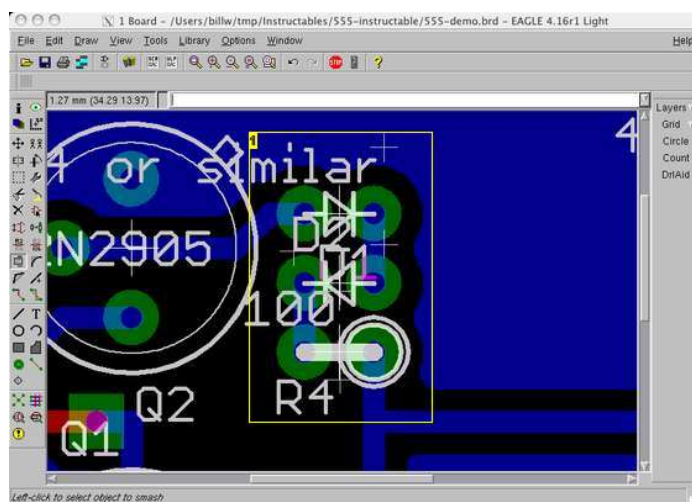
If we want the names of components to be legible on the top of the board (transferred via toner transfer), or just to look good on printouts, they names and values probably have to be moved from their default locations. In order to move the text separately from the device itself, we use the "SMASH" command. (Why is it called "smash"? I dunno!)

Select the SMASH icon from the menu, then click each component whose text you want to move. If this is ALL of the components, there's a ULP that will smash everything (but ULPs are a subject for possible future instructables. Or the EAGLE manuals.)



#### Image Notes

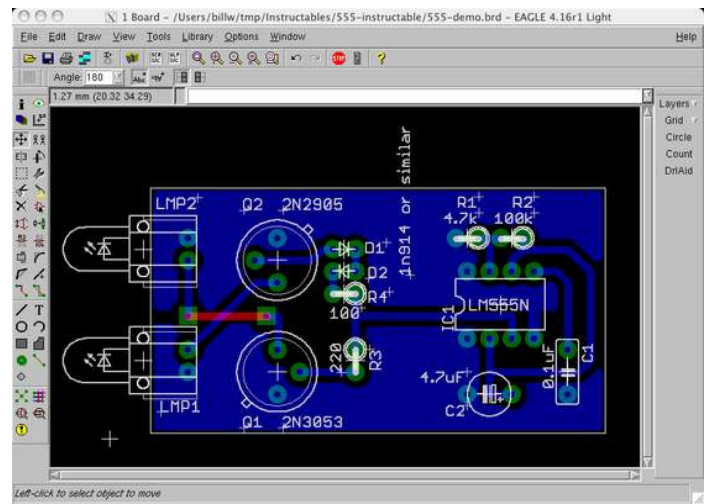
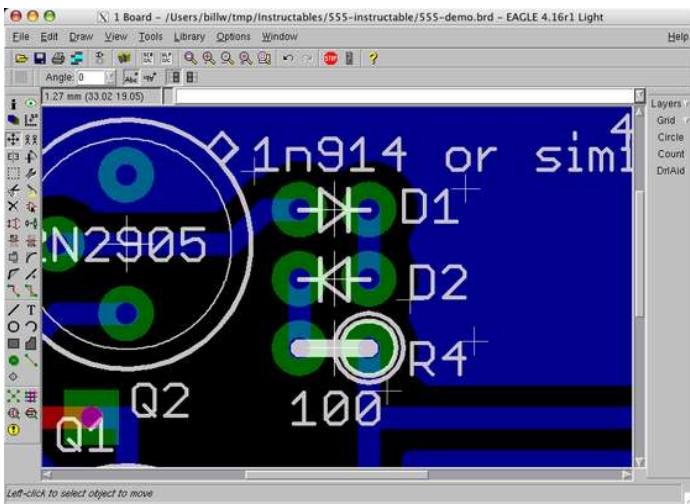
1. Particularly messy text
2. SMASH icon



#### Image Notes

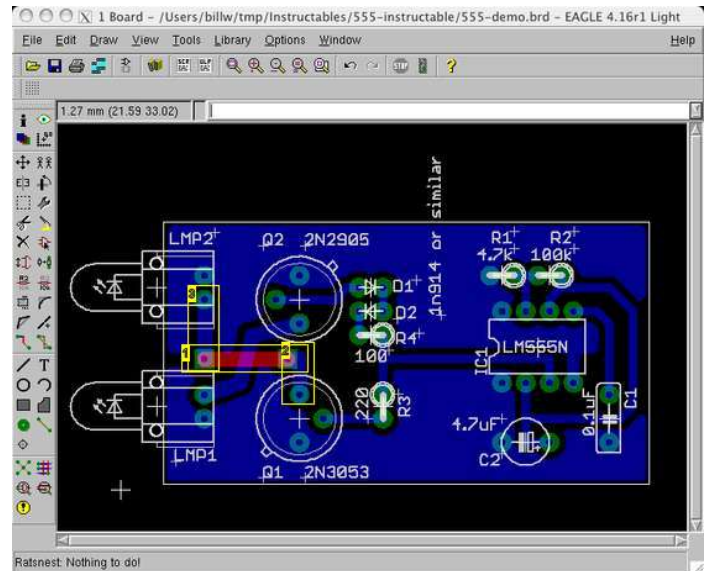
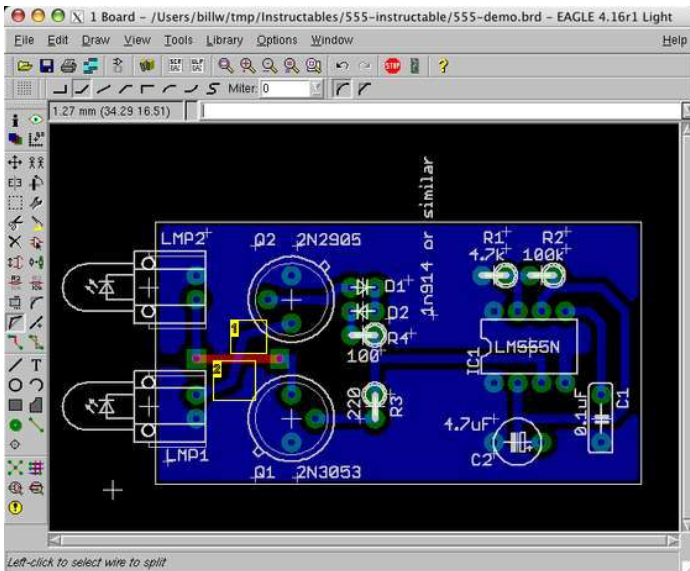
1. Now the text has its own origins so it can be moved independently of the component itself.





### Step 16: Neaten up; move traces

We can move some of the traces so they look neater, offer better clearance, etc.  
Also, we shrink the board to its final size by squishing the components together some more.

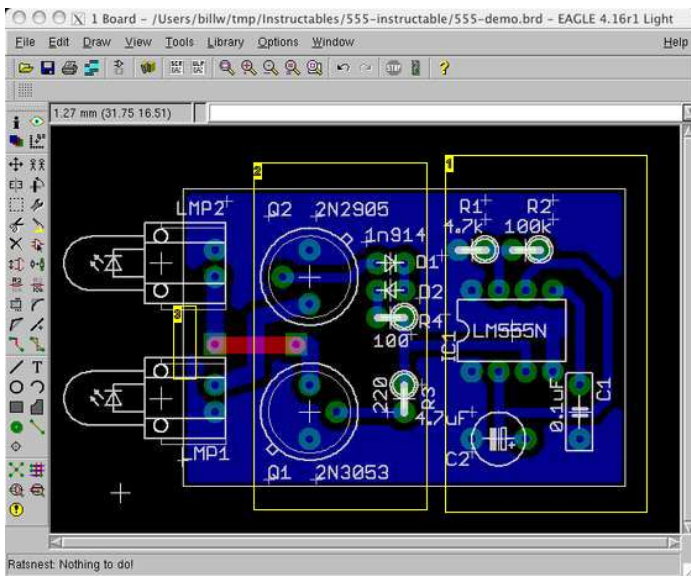


#### Image Notes

1. This corner moved slightly
2. This corner added with BREAK icon

#### Image Notes

1. Trace widened with "Change Width"
2. Trace widened with "Change Width"
3. Trace widened with "Change width"



#### Image Notes

1. Stuff moved closer
2. Stuff moved left
3. Edge moved in

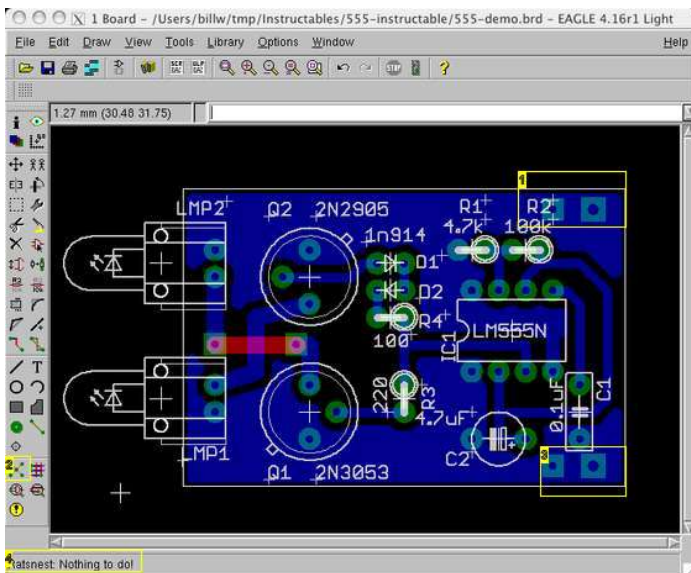
### Step 17: Fixing an OOPS!

Remember back in the schematic I mentioned that there were a couple of things that had been left out? You should be noticing them about now...

POWER connections; there's no way to connect a battery or power supply to this circuit board. Oh sure, you can just tack some wires onto the supply polygons, but how elegant is *that*!

We could go back to the schematic and add some actual power connectors or battery holders, but those are a bit rigid for a circuit that's probably going to be connected to a battery pack with some wires anyway. Instead, let's add some Vias to act as connection points for the power wires.

When adding Vias like this, it is convenient to use the text command entry area so we can name the signal at the same time we add the via. Type "via 'gnd'" (yes, you need the quotes here, unlike for polygons.) You can adjust the drill size and via shape, and plunk the via down in the appropriate supply polygon. I like to use two vias as a sort of strain relief (one is made larger so you can feed wire + insulation through it, the other is sized for just the wire.) A click on the RATSNEST icon will make sure the vias are connected to the polygon. Then do the same for the V+ signal (named NS1, you recall.)



#### Image Notes

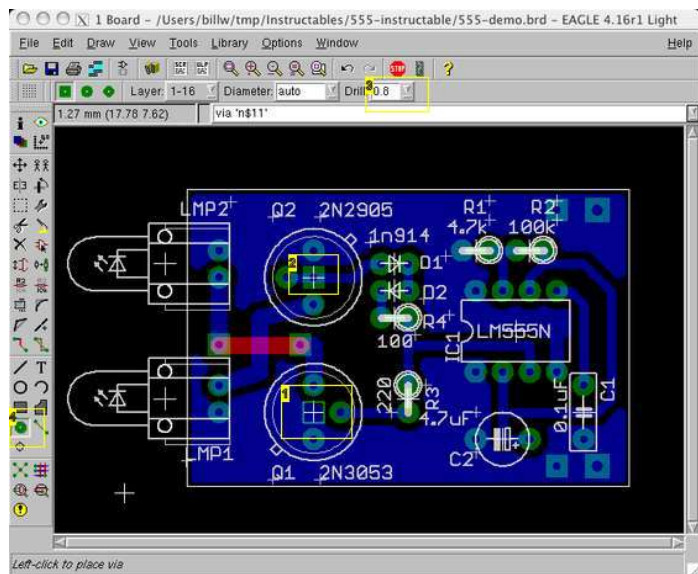
1. V vias
2. RATSNEST
3. GND vias
4. RATSNEST is happy with the state of the board.

## Step 18: Neaten up: Allow for alternate packages and options

We can drop some extra holes for mounting different packages. The transistors used in the published schematic that we entered apparently come in a sort of metal can package that has dropped in popularity. If we arrange for three in-line mounting holes, we can substitute a whole lot of different transistors whose package leads come that way (TO92 or TO220, to mention two popular modern packages.)

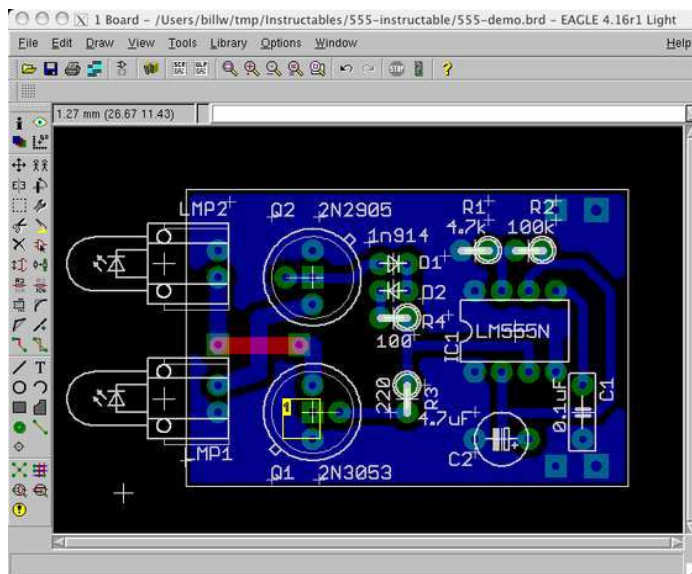
Use the info command to figure out the signal names, and then "via 'n\$X'" on the command line to create the via, followed by a manual route to the via if needed. In this case, one of the vias placed collides with a signal trace hidden by the GND polygon, so we have to remove that trace with the "ripup" command (the polygon will still connect to the pad.)

While we're at it, I'll add some text to the silkscreen to show where the emitter lead of the transistors should go. Use the "text" icon button, and change the layer to tPlace.



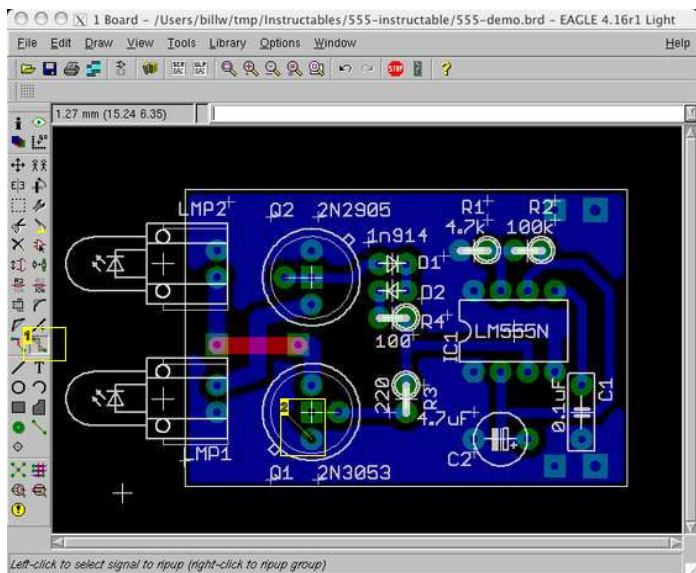
### Image Notes

1. New via
2. New via
3. Size of via
4. VIA command



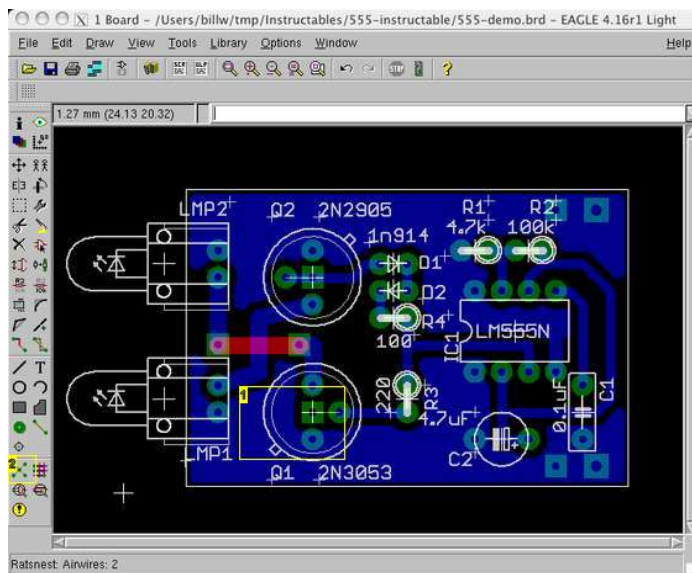
### Image Notes

1. The via collides with a trace that was hidden by the GND polygon



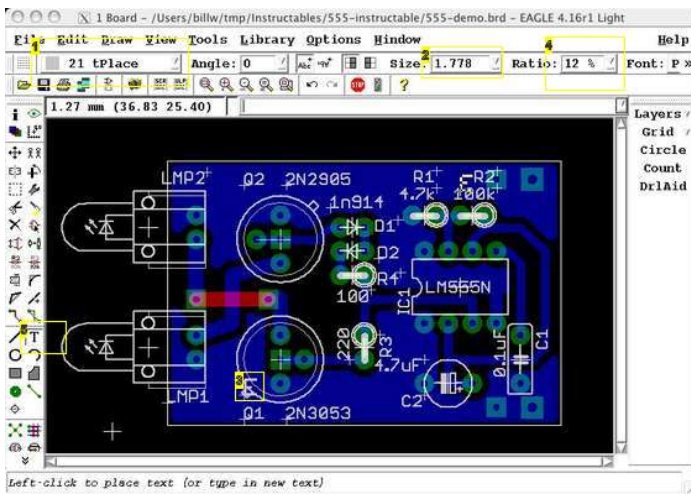
### Image Notes

1. RIPUP command
2. Click on trace to "un-route" it.



### Image Notes

1. A new RATSNEST command will recompute the polygon, and it stays connected to the pad.
2. RATSNEST

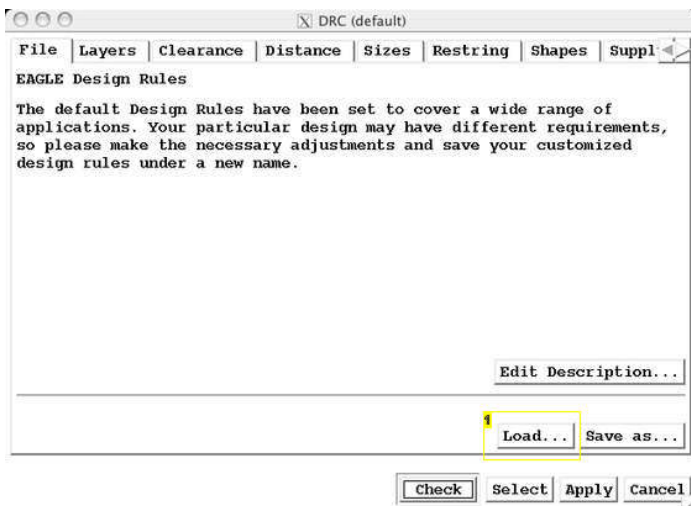


#### Image Notes

1. Select layer here. "tPlace" is the top layer silkscreen.
2. Text size
3. Added text.
4. text "Boldness."
5. Text command

### Step 19: Do Design Rule Check

We want to run a design rule check to make sure that none of the manual editing we've done violates the rules...



#### Image Notes

1. Use the LOAD command to load hobby design rules.

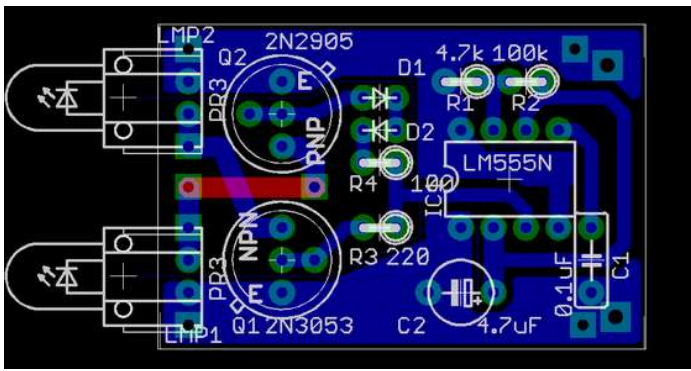
### Step 20: Output using Exported images

Save your work often. You've been doing that, right?

Now we're essentially done, and we should figure out how we're going to output our board for admiration on web pages, review by peers, transfer to physical PCB material, and so on.

One way to output the board is to "export" an image.





## Step 21: Other useful menu icons

Here are some other useful commands accessible from the menu icons

**LAYERS** Adjust which layers are displayed. Boards have many more layers than schematics!

**MIRROR** Move a component from being mounted on the top of the board to being mounted on the bottom of the board.

**CUT** COPY a selection, despite the name.

**NAME** Change the name of an object.

**CIRCLE** Draw a circle.

**RECTANGLE** Draw a rectangle.

**MARK** Place a measurement mark. Your info area will start showing distances relative to the mark as well as to the origin.

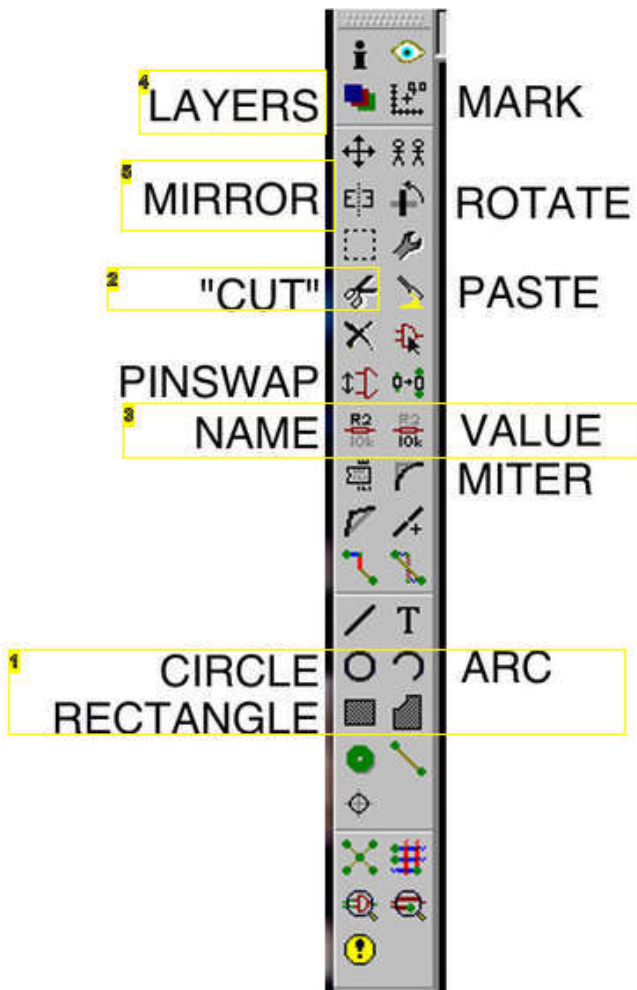
**ROTATE** rotate an object. This can rotate angles other than 90 degrees.

**PASTE** Paste some objects that were previously copied with CUT.

**VALUE** Change the value of an object.

**MITER** make signal corners rounded.

**ARC** Draw an arc.



### Image Notes

1. Might be used for drawing non-signal elements.
2. CADSoft calls this CUT, but it really does COPY
3. Change component name/value, just like in the schematic
4. Manipulate which layers are shown and/or printed.
5. Flip components to the other side of the board. Useful for putting SMT components on the copper side of Single sided boards, for example.

## Step 22: Useless commands

These are menu icons that I don't find at all useful in creating boards, at least not from schematics (and I feel that you should always make schematics to go with your boards; both for self-documentation and the error-checking capabilities that are added.)

**SHOW** SHOW is more useful from the text command area. I think.

**DUPLICATE** Duplicate an object. Usually done in the schematic.

**ADD** Add a component. Usually done in the schematic.

**REPLACE**

**JOIN** Happens automatically, usually?

**POLYGON** more useful from the text command area.

**SIGNAL** Create a signal. Usually done in the schematic



### Image Notes

1. Better used in schematic
2. Better typed in text cmd area with a signal name: "Poly gnd"
3. Happens automatically
4. Better done in schematic
5. Better used in schematic
6. SHOW command better typed in the text command area where you can provide a name, as in "show GND"



## Related Instructables



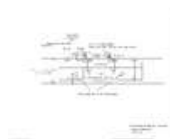
**Render 3D images of your PCBs using Eagle3D and POV-Ray** by ongissim



**Eagle-ize Leevonk's PIC protoboard** by westfw



**Arduino MIDI-in shield** by carkat



**Free CAD program using ExpressPCB** by botronics



**Make hobbyist PCBs with professional CAD tools by modifying "Design Rules"** by westfw



**Arduino FM radio receiver shield** by ZrvZ

## Comments

50 comments

[Add Comment](#)

[view all 114 comments](#)



**jonnyphenomenon** says:

Feb 9, 2011. 5:27 PM [REPLY](#)

Great instructable man. I just finished reading the "how to draw a schematic" one and loved it. you helped me figure out what i was doing wrong. :)

I sure would like to be able to download that hobby.dru file, but the site keeps telling me I need to be logged in. (I am logged in, or else I wouldn't be able to post this) Anyway, guess I will have to follow your instructable and just make my own :)



**alr206** says:

Jul 22, 2010. 2:14 PM [REPLY](#)

How do you get it to incorporate contacts that are connected to gnd? Everytime I try, it excludes those areas (I am using your design rules file). Thanks -- your instructables have been VERY helpful!



**westfw** says:

Jul 22, 2010. 3:16 PM [REPLY](#)

make sure that both the signal and the polygon have the same name (ie "gnd") The easiest way to name the signal is to attach one of the "gnd" symbols to it, and for the poly it's easiest to use the "poly gnd" to create it, but you can also name them with the "name" command/button.



**abraxas2** says:

Aug 19, 2010. 9:50 PM [REPLY](#)

Every time I try to rename a signal path or pad, in board view, it says "Cannot Back Annotate ...Change in Schematic View" So after many attempts I finally manage to change one of these items names to "Base", in schematic view, since it connects to a transistor base. It prompts me with an invite to connect Base and Gnd, I respond "no". The adjacent signal path which I do want renamed gnd, keeps on resetting to the name Base and won't let me change it to gnd...no matter how many times I try. Worse yet, when I do the gnd plane, both the Base and Gnd signals get tied to the plane. It's driving me nuts.



**abraxas2** says:

Aug 20, 2010. 1:28 AM [REPLY](#)

The only thing I can see is that I have some very faint phantom lines connecting the Base and GND traces. These are lines I believe left over from a Rip Up repair. I'm guessing the program still thinks these items are connected.



**alr206** says:

Jul 23, 2010. 7:11 AM [REPLY](#)

I didn't realize that what I actually had was thermals, so the pads should appear a little bit different under a closer view than in the pictures above. Thanks again!



**westfw** says:

Jul 24, 2010. 12:15 AM [REPLY](#)

Ah. Yes, I have thermals turned off in my init files, which makes my normal behavior different than the default behavior. I should be more careful about that!



**abraxas2** says:

Aug 19, 2010. 7:52 PM [REPLY](#)

I can't thank you enough for this excellent tutorial. I thought I'd never get the hang of it and it's not a very intuitive program. Do consider turning these tuts into a book. Now hopefully one last question : after drawing the ground plane with the ratsnest command I got a nice big plane but also a bunch of tiny little islands that serve no purpose but to make the board look dumb. I tried erasing these islands with the rip up but it just ripped up the closest vital wire instead. Is there away to zap these tiny little island ground planes ? Thanks MUCHO !!



**westfw** says:

Aug 20, 2010. 12:26 AM [REPLY](#)

you can draw lines and/or rectangles in the tRestrict or bRestrict layers, which will prevent the polygon fill from entering those areas. You can see this in the PCB that goes with <http://www.instructables.com/id/Single-Sided-Really-Bare-Bones-Board-Arduino-in-EA/> - I use it to prevent copper between the pads of the bottom-side SMT capacitors. Note that the restrict layers are not displayed by default; you'll have to turn them on in the layer menu (they might get turned on automatically when you draw on them, too.)



**abraxas2** says:  
It mysteriously fixed itself. Thanks.

Aug 20, 2010. 1:25 AM [REPLY](#)



**abraxas2** says:  
Now that the autoroute has destroyed my hours of work, how do I undo it ? Undo doesn't work !!!

Aug 17, 2010. 9:43 PM [REPLY](#)



**westfw** says:  
You can't undo the autorouter. It's a good idea to save your work often, especially before large changes like autorouting. The autorouter should not have destroyed any of the manual routing you did; it just added a bunch of "other" traces. You can get rid of individual traces, or sections, or everything, with "ripup", but there isn't specifically a "ripup the bits you just autorouted."

Aug 18, 2010. 12:02 AM [REPLY](#)



**Naved Ahmad** says:  
Hello Every body: This is N-A-G from Afghanistan, I need help If any one has the basic, intermediat or advanced notes about making remote control, motors, tank treads, PCB [in pdf format or any other format] please send them to any one of my accounts i will be thankful of that person. or if any body has any link or some ones email, gmail, hotmail, skype, yahoo that knows about the thinks i wrote above please send them. i will be thankful of that person again. thanks alot buddies. Contact: [0093 (0) 794 50 63 43] - [0093 (0) 706 71 98 93] E-Mail [nghafori@af.mercycorps.org] Gmail: [navedahmadghafori@gmail.com] Yahoo: [navedahmadghafori@yahoo.com] Skype: [naved.ahmad.ghafori]

Jul 15, 2010. 1:49 AM [REPLY](#)



**nicolo86** says:  
Really useful, but i have a question. Is it possible to make two different PCB connect with each other. if yes how. thanks

Jul 5, 2010. 2:41 AM [REPLY](#)



**tomtortoise** says:  
yo when i did this step the two non-polar capacitors disappeared from the whole thing? i have know idea what happened to them and i still barely know how to use the program.

Jun 10, 2010. 5:55 PM [REPLY](#)



**amando96** says:  
swett, helped me a lot  
\*faves\*

Apr 7, 2010. 7:00 AM [REPLY](#)



**abraxas2** says:  
Sorry but I'm missing something here. I've got a nice picture of components and lines between them. I need artwork, that is to say a black and white transferable image for final etching. I'm not seeing that under export. When I get the export image it's just a copy of the board layout with all these component images included. I just want the conductive lines and pads ???.

Mar 29, 2010. 11:58 PM [REPLY](#)



**westfw** says:  
There was supposed to be a follow-on instructable on getting PCB-ready output files of various sorts, but real life sorta delayed that one. A couple of other helpful Instructables have shown up from other authors, though:

<http://www.instructables.com/id/Automating-Eagle-export-and-preparing-for-printing/>  
<http://www.instructables.com/id/Professional-PCBs-almost-cheaper-than-making-them-/>

Apr 1, 2010. 1:17 AM [REPLY](#)



**abraxas2** says:  
I'm using 5.6.0

Mar 29, 2010. 11:59 PM [REPLY](#)



**puffyfluff** says:  
Just to let you guys know, if you want cheap factory-made PCBs, try BatchPCB. It's a service run by Sparkfun for PCB manufacture. The prices are \$2.50 per square inch on 2-layer boards or \$8.00 per square inch for 4-layer boards and a \$10 setup fee (all USD). There's no minimum quantity. It takes a few weeks depending on when you place the order, but it's still pretty slick.

Mar 9, 2010. 5:32 PM [REPLY](#)



**simplicio** says:  
What I do to get onboard power connections is to drop "test pads" onto the schematic from the testpad library (e.g. TPSQ/TPSQPAD1-13). That way you get thermals around the pads which makes them easier to solder.

Jan 23, 2010. 10:57 PM [REPLY](#)



**Dimitrios** says:  
Really nice instructable. I am actually looking for someone who can make a pcb layout. Who can help me? It's a small schematic with 13 components (resistor/diodes/transistors) Like 2" x 2". I am not knowledgeable enough to use a cad program. Anyone who can help me, pls message me with your email so I can email the schematic.

Dec 4, 2009. 7:51 AM [REPLY](#)



**westfw** says:  
(did I set some kind of record for time gap in between publication and "featuring" ? About 3 years! (To be fair, there was no such thing as "featured" back when this was first written. It makes me wonder what other really good instructables are languishing unnoticed mostly because of their age...))

Nov 13, 2009. 8:44 PM [REPLY](#)



**joeuhlik** says:

Why does Eagle want to draw the polygon as line segments rather than as a rectangle as you show?

Apr 7, 2009. 12:32 PM [REPLY](#)



**westfw** says:

It'll draw the outline as per the current settings for line-bending and etc as specified for "wires"; to get a rectangle, I usually set the line bend to right angles and draw two "L" shapes.

Apr 7, 2009. 5:06 PM [REPLY](#)



**diyuser** says:

How would you tell Eagle not to draw polygons/power planes when auto routing. Just plain wiring.

Nov 13, 2009. 4:47 PM [REPLY](#)



**westfw** says:

I don't think Eagle ever creates polygons during autorouting, though it may use any existing polygons as THE route for whatever signals they represent. I always end up spending a fair amount of time adjusting exactly how the polygons fill; I don't like the way it tends to create "spikes" in the areas between pads, for example. It all has to be done manually.

Nov 13, 2009. 8:42 PM [REPLY](#)



**keithongrq** says:

i've done every thing here and the end result look good but when i save the file and reopen it, the polygon becomes a rectangle made of dotted lines. some help please thanks

Aug 31, 2009. 12:13 AM [REPLY](#)



**westfw** says:

This is normal. EAGLE doesn't "render" the polygon on the screen until you issue a "ratsnest" command. I'm pretty sure that gerber output would be correct anyway, and having the polygons displayed as outlines makes it easier to route additional tracks.

Aug 31, 2009. 7:49 AM [REPLY](#)



**robnee** says:

How does one get the vias added for +V and GND to "unfill"? When I export the artwork these are filled in and I can't see where to drill in the finished board. On my screen too they appear filled where the ones in the photo above are unfilled and would presumably output correctly.

May 1, 2009. 1:14 PM [REPLY](#)



**brainiac27** says:

One could use the drill-aid ULP. This script puts a hole of a chosen diameter in everything that would require drilling, including the vias. I hope that helps.

Jul 23, 2009. 9:04 AM [REPLY](#)



**westfw** says:

What are you using to export ? I see the real holes via "export image" and "print", and I can't imagine what would leave the pad holes but fill in the via holes... (also, which version of Eagle are you using?)

May 1, 2009. 2:04 PM [REPLY](#)



**platitudes** says:

Is it more practical as a rule to have power planes or ground planes?

Jul 20, 2009. 12:41 AM [REPLY](#)



**westfw** says:

Huh. Somehow some of the pictures got disconnected from where they were supposed to be. This should now be fixed!

Apr 4, 2009. 1:25 AM [REPLY](#)



**jimmy dean** says:

When I try and draw a polygon, it's never filled in. When I finish the polygon it just makes a rectangle made of dotted lines.

Mar 5, 2009. 2:41 PM [REPLY](#)



**westfw** says:

Assuming the polygon is attached to the appropriate signal (it should contain a pad, smd, or via connected to the signal within its borders), it will fill in when you issue a "ratsnest" command.

Mar 6, 2009. 12:14 AM [REPLY](#)



**jimmy dean** says:

Oh, i didn't do the ratsnest, thank you.

Mar 8, 2009. 6:40 PM [REPLY](#)



**RedBinary** says:

Nice! Just letting you know that I really appreciate you going through the time && trouble to lay out the basics in all of your Eagle instructables. Really shortens the learning curve for me, speeding up the process! +5.0 && faved!

Jan 1, 2009. 11:11 AM [REPLY](#)



**Nawaz** says:

Dec 25, 2008. 10:01 PM [REPLY](#)

yo,nice instructable...well illustrated.Thanks for sharing...But i have a question.Do you know how to print a bunch of tiled PCB on toner transfer?That is printing many of the same circuit on the same paper...Thanks



**westfw** says:

Dec 26, 2008. 6:26 PM [REPLY](#)

Not really. You can copy and paste the whole eagle PCB layout into a new .BRD file, up to the limits of your license (good for really small boards.) You can export images, postscript, or gerbers and manipulate them with an external graphic utility (although I'm not familiar with any freeware that does automatic tiling.) None of these is very convenient.



**Nawaz** says:

Dec 30, 2008. 12:03 AM [REPLY](#)

Thanks for your quick reply and sorry for my late reply....Once more,nice instructable...keep it up. :)



**huxisme** says:

Nov 3, 2008. 6:33 AM [REPLY](#)

Do you mind to tell me how do you usually mirror the PCB board layout? Normally what im practicing is that i will print the PCB layout in pdf format then from there use whatever program to flip horizontally before trace them on the PCB board. So how's your opinion?



**westfw** says:

Nov 4, 2008. 8:01 AM [REPLY](#)

There is a "mirror" button in the print dialog. Although usually for a board with traces on the bottom, you don't WANT "mirror", because the printout is already mirrored by virtue of being viewed from the top, and/or from being flipped over for "iron-on" type application... (OTOH, I think printing to PDF or postscript and then using a high-quality PDF or Postscript printing utility may have other advantages, so I wouldn't say that the way you're doing things now is wrong...)



**huxisme** says:

Nov 4, 2008. 9:10 AM [REPLY](#)

You are really helpful. i found that mirror button. Thanks very much.



**thermoelectric** says:

Sep 17, 2008. 1:16 AM [REPLY](#)

How would I turn this to a board? Its already routed(is that the right word) but I dont know how to change it to the file to toner transfer?



usbtinyisp v2.0.brd73 KB



**westfw** says:

Sep 17, 2008. 7:47 AM [REPLY](#)

It's not an ideal design for a first effort at toner transfer; it has relatively fine traces that go between pins, it's double sided requiring plated through holes in places where it'd be difficult to solder on both sides of the board, and etc...

The easiest way to convert to a board is to take \$8.05 in the mail to AdaFruit and wait a couple of days. (Better yet, make in 12.22 and get the pre-programmed microcontroller needed by the project as well.)

(I don't think people realize in general what a great service it is to be able to buy reasonably priced pre-made PCBs for projects like this...)

I was planning on a future instructable covering the various ways to output PCB designs into the "real world", but it isn't moving very quickly :-)



**thermoelectric** says:

Sep 17, 2008. 1:50 PM [REPLY](#)

How would you get plated through holes?



**westfw** says:

Sep 17, 2008. 5:31 PM [REPLY](#)

In general, you don't. When designed a PCB for hobbyist fabrication, you can add extra Vias that let you make the top-to-bottom connections, instead of "assuming" that each component hole makes that connection. The board in this instructable is sort-of an example. If it were a two-layer professionally made board, the trace from Q1 to the top lamp would go over the top layer directly from pin to pin, instead of having the two extra vias. For a board like the usbtinyisp, there isn't a lot of room to add those extra vias without "stretching" the board in a couple of directions...



**thermoelectric** says:

Sep 17, 2008. 10:17 PM [REPLY](#)

So I should just buy the board, Thanks for the quick-reply



**westfw** says:

Sep 17, 2008. 11:59 PM [REPLY](#)

yeah, I think so. I did some playing with the board to see if I could make a larger but single-sided version, and it didn't go very well :-)  
That's an advantage of the whole open-source hardware thing; not only can you download the PCB design, but it's in a form that can be "easily" modified... But in this case, it didn't help much.

[view all 114 comments](#)