

[Skip to products menu](#) [Skip to content](#)

SparkFun Electronics



Beginning Embedded Electronics - 9

by [Nate](#) | June 19, 2008 | [113 comments](#) Skill Level: ★ Beginner

Lecture 9 - Eagle: PCB Layout

To learn how to use Eagle, we are going to create a simple breakout board for a popular USB IC. The FT232RL is a USB to TTL serial converter. This section covers the physical layout of the PCB.

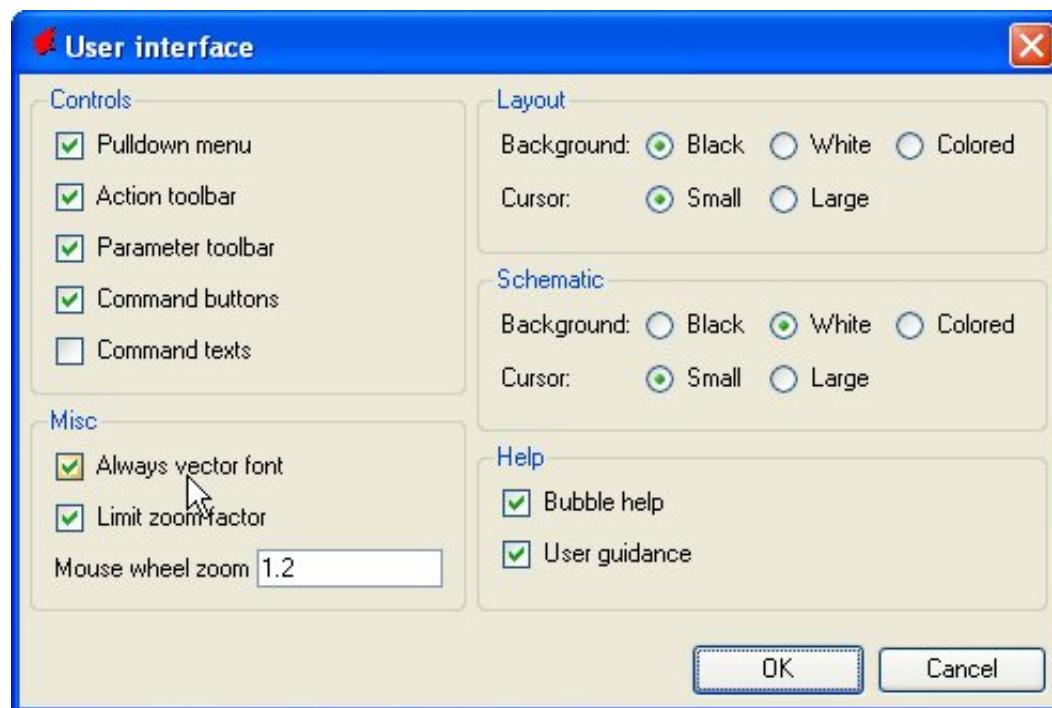
We've got the schematic captured and the connections should be made correctly. The next step is to arrange various components on a board and then send the board files out to a manufacturer (also called a fabrication or 'fab' house).

Layout is an art and engineers make bad artists. It's all about the small polishing - text labels, stand off holes, correct footprints. Just keep turning out PCBs and you'll see your layouts improve dramatically with practice.

Before you do anything, turn on vector fonts! If you don't, your silkscreen text will be off on every PCB you create:



From the Eagle control panel, click Options->User Interface



Select 'Always vector font'.



Above is a breakdown of all the different buttons in the Layout window. You can view the name of each button by hovering over the button with your mouse. Many of the short-cuts from the Schematic window still apply:

- Press escape at any time to stop the current action and return to the Layout window
- F7 to move a part
- Alt+F7 to group a bunch together
- F3 to delete a part
- F4 to rename a part (change C7 to C2)
- F5 to re-value a part (change 0.1uF to 10uF, etc)
- F6 to smash a part (be able to move the name and value tags)
- F9 to start routing a wire
- Alt+F9 to rip up a wire

If you get this error while moving bits around:



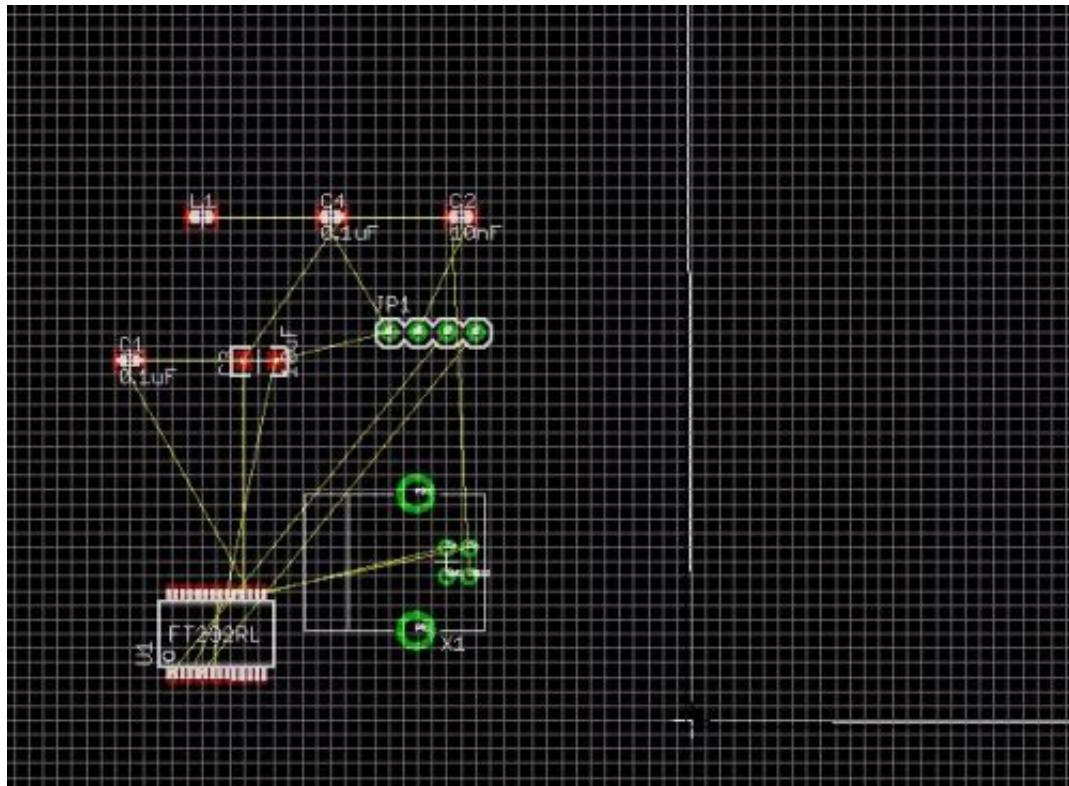
This just means that you are trying to put a component outside of the allowed area. With the Light edition of Eagle, you can only place components in quadrant I (upper right quadrant), whereas the components show up by default in quadrant II (upper left quadrant). Just move the parts to a positive X and Y coordinate and you should be ok.

Starting with our current layout:

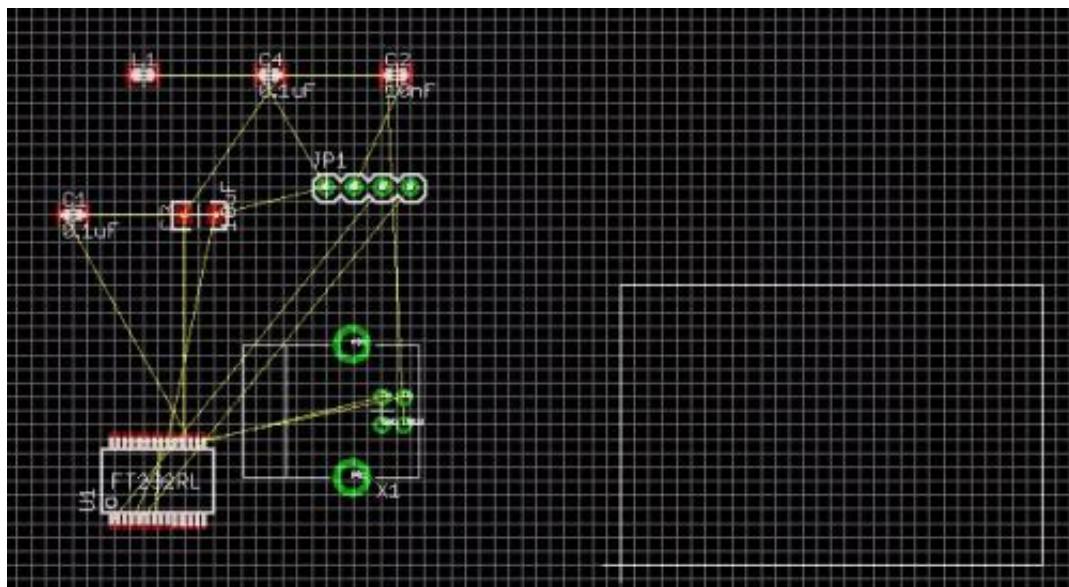


The first thing to do is correct the board outline. I don't know why Eagle slightly offsets the default border.

Make sure you're on a 0.1" grid by pressing F10. Then hit F7 and hold control while clicking near the origin. This will grab the frame corner and force it onto the 0.1" grid. Make the bottom left corner sit at (0,0):



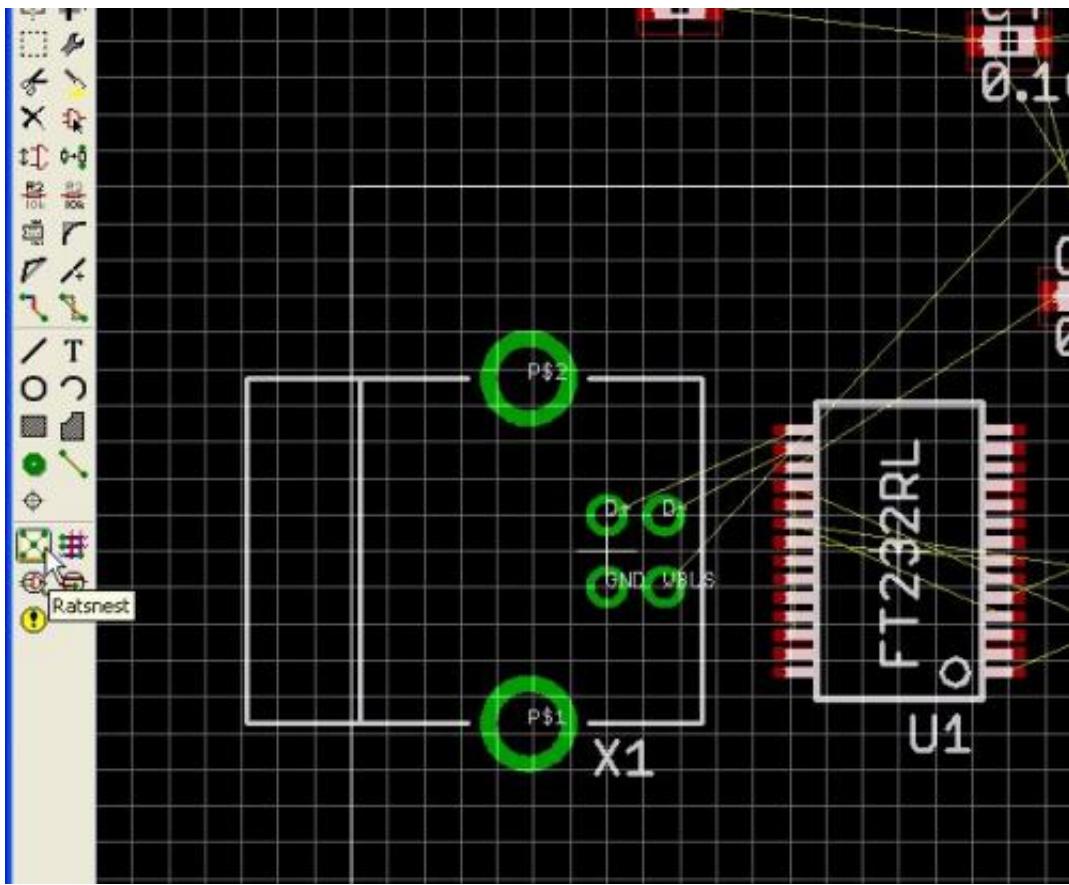
Do this for the other three corners bringing them in to make a 1.5x1.0" square board size.



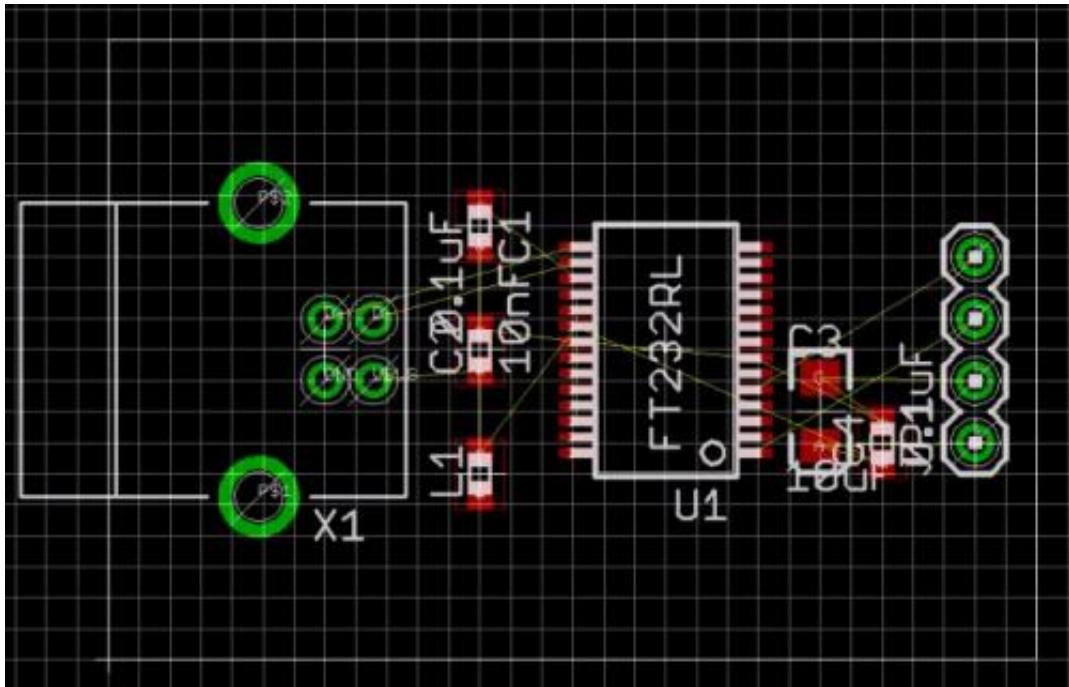
Now go to town bringing the components into the board area. Keep in mind the gold color un-routed 'air' wires. The less twisted you make these by creatively arranging your components, the easier the trace routing will be.

Remember:

- Press F7 to move a component
- Right click to rotate
- Hold control to grab a component at its origin
- Scroll wheel to zoom in/out

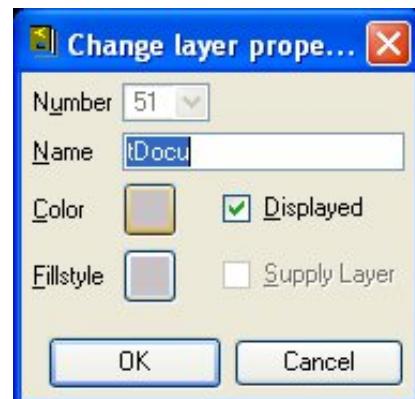
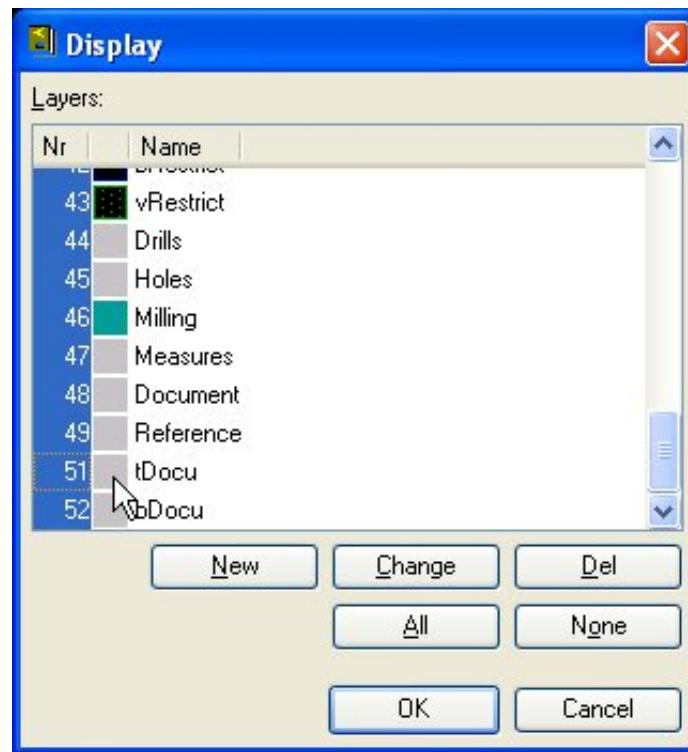


Hit the Ratsnest button from time to time to recalculate the air wires.

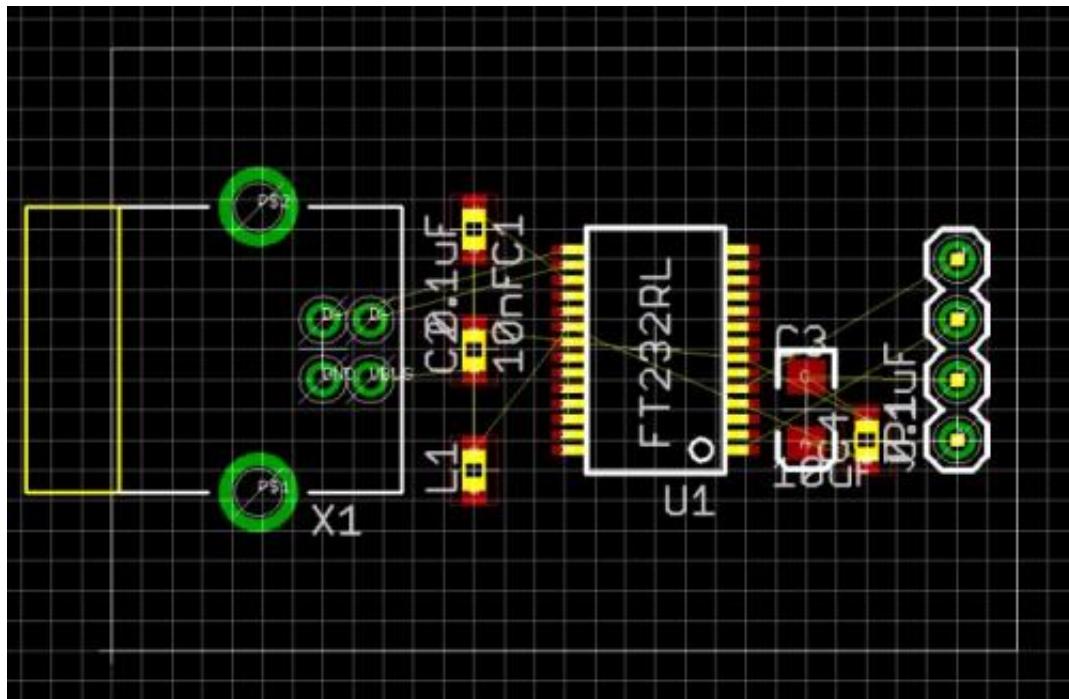


Here are the components arranged in a basic configuration. Another beef I have with Eagle is the default colors for the various layers make it impossible to see what is going to be printed on the silkscreen layer. Let's change the 'tPlace' layer to pure white and change the 'tDocu' layer to lemon yellow.

Click on the 'Display' button, scroll down to layer 51 and double click on the gray box next to 'tDocu':



Then click on the gray 'Color' box and change it to something interesting like lemon yellow, then click ok. Anything you do on this layer will now be yellow. Do the same for layer 21 'tPlace'.

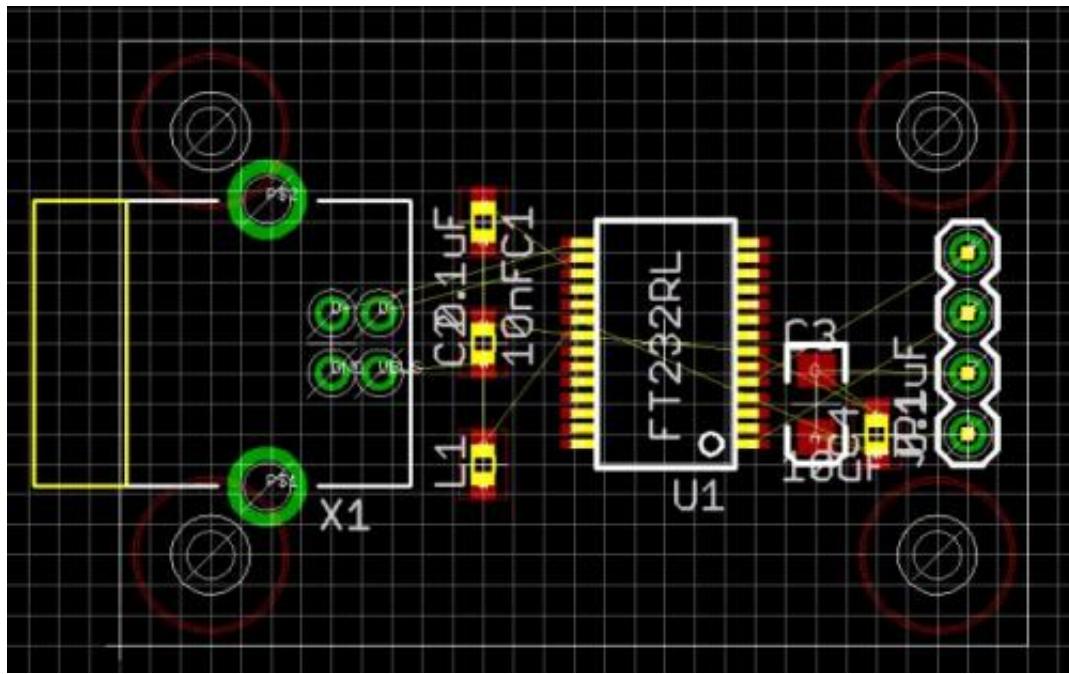


Now this is starting to make sense! Anything in yellow is just there to indicate physical size. The yellow part of the USB connector is only there to indicate that the connector sticks over the edge that far. Only the white part of the USB connector footprint will actually show up on the silkscreen print on our PCB. Anything in light gray (tNames and tValues layers) *will not print* on the silkscreen layer. They're just there for your own reference. We can of course change how the various layers are processed (and include the value and name layers on the silkscreen) but this can cause a lot of squeezing and hassle. It's up to you and your design but we will leave the part indicators and values out of this layout.

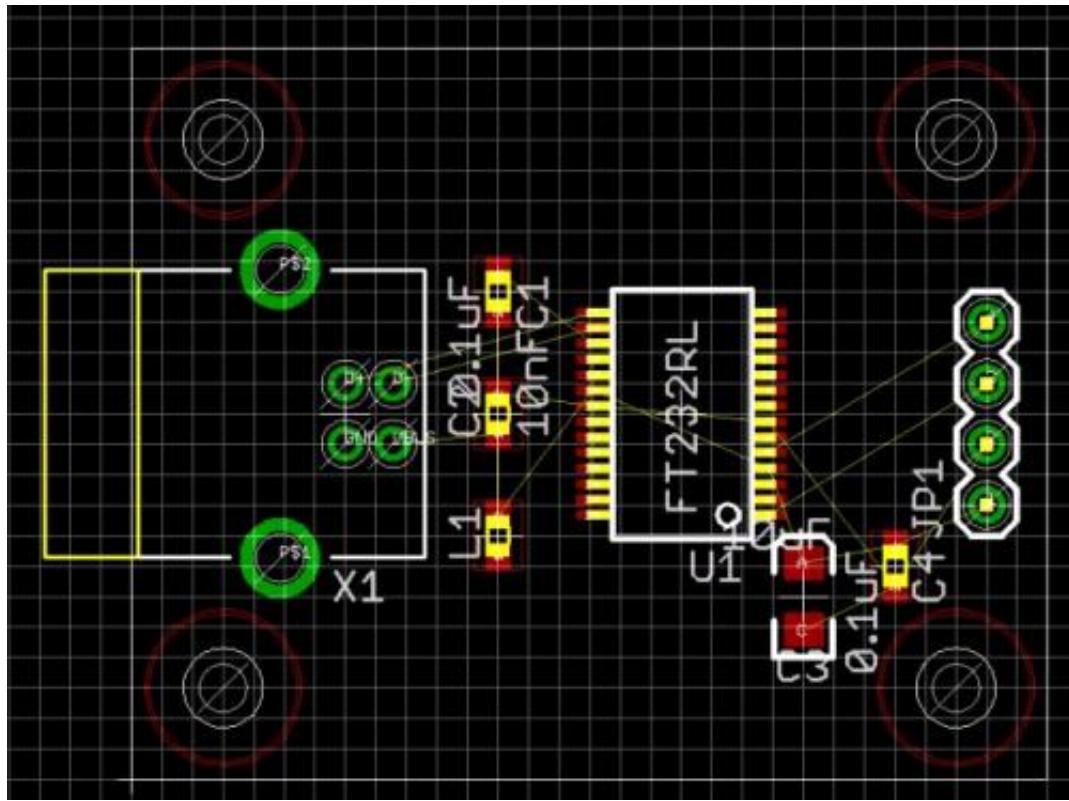
The next thing we need to do, for all PCB layouts, is to add stand-off holes. These holes will allow you to insert a simple screw and hold a stand-off in place. Without standoffs, PCBs will sit uneven against a flat surface (because of the bumpy solder joints protruding on the bottom of the board). Having a PCB sit flat against a surface is also a bad idea electrically - I've sent \$200 up in smoke because some bits of clipped wire shorted against the bottom of my board when I was troubleshooting it.

I like to use **4-40 screws** and **0.25" diameter plastic standoffs** on everything. These 4-40 screws need a 0.13" diameter hole and the standoffs have a 0.25" outside diameter that we will need to take into account. If you have not already done so, add four of the 'Stand-Off' components to the schematic (and therefore PCB). This component was created to couple the 0.13" drill with a keepout ring. This keepout ring helps show where the screw head will fall. If you fail to take this keepout layer into account, the screw will go through the hole, but the screw head may run into or short components.

Throw four standoff holes around the corners of your board.



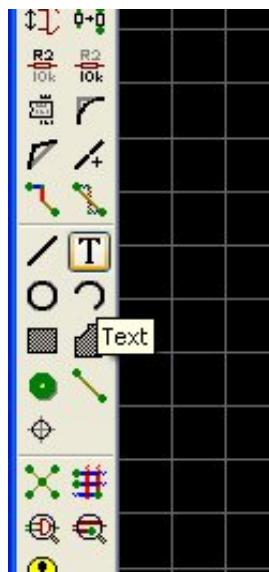
Ahah! Now I see why I made that keepout circle. You can see where the standoffs would have run into the USB connector. Looks like we've got some bumping to do...



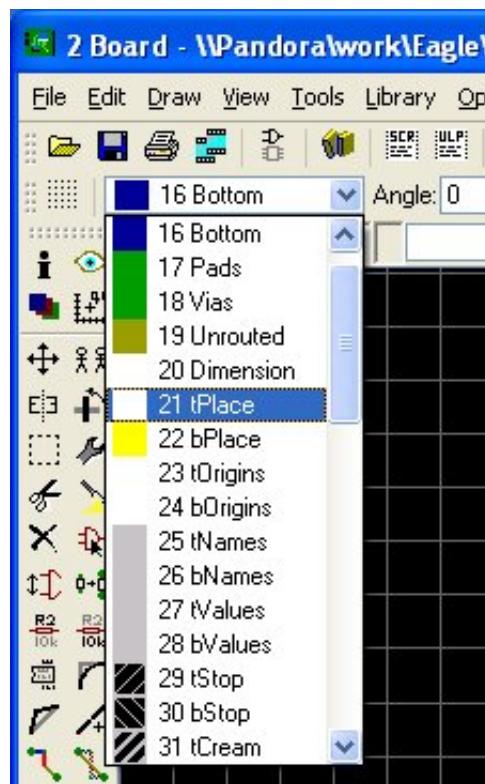
By using the group (Alt+F7) and the move (F7) commands, I increased the border to 1.5x1.2" which is a bit bigger than I would like, but for the purposes of this tutorial, we're not going to stress tight packing of components - rather we want to stress the basics for a good PCB. Notice how I flipped and dropped C3 and C4 down a bit? Time to add labels!

A 'C2' label is handy when you're populating a board or when you're troubleshooting a complex circuit, but on a

day to day basis, you probably won't need to know where C2 is. On the other hand, the TX and RX pins will probably be used every time you use the board! You really should label anything that will be connected to the outside world. To add a text label to a pin, click on the 'Text' button:



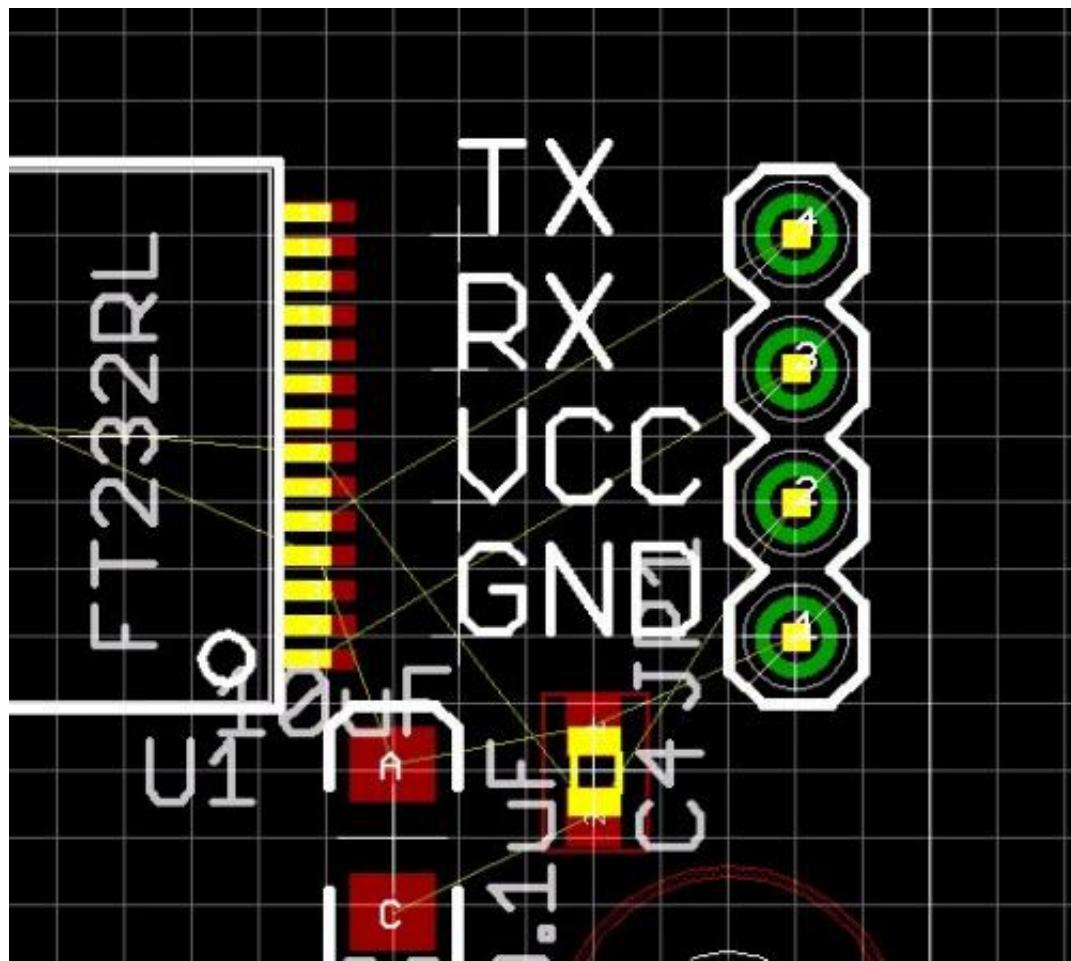
A window will pop up asking you what text you would like to add. Type 'TX' and press enter. You will notice that the text may be appearing on an odd layer. Be sure that you add text on the tPlace layer.



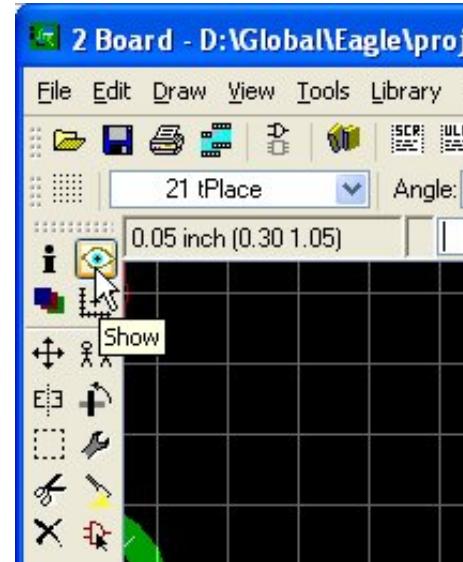
Drop down the layer menu and select tPlace for top silkscreen text

Once you've placed 'TX', press escape. Eagle will now show the text window again. Enter 'RX' and press enter. Repeat for RX, VCC, and GND. When done, press escape twice to return to the layout window.

In this case we have VCC/GND/TX/RX to label:



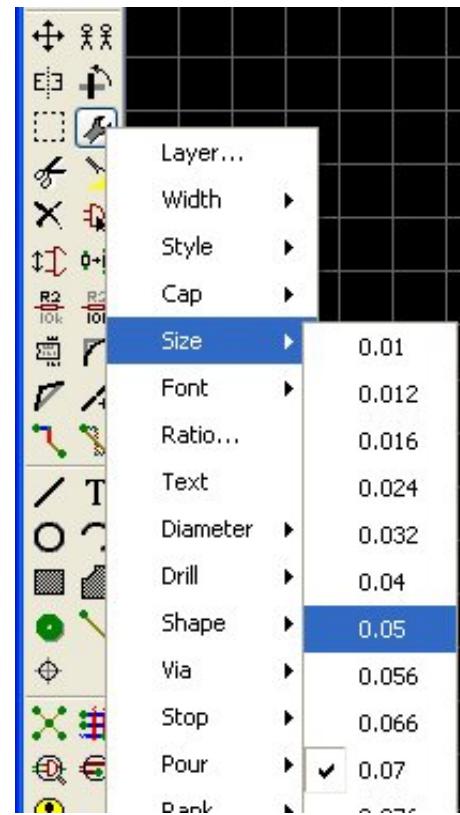
Make sure you add your labels to the 'tPlace' layer!



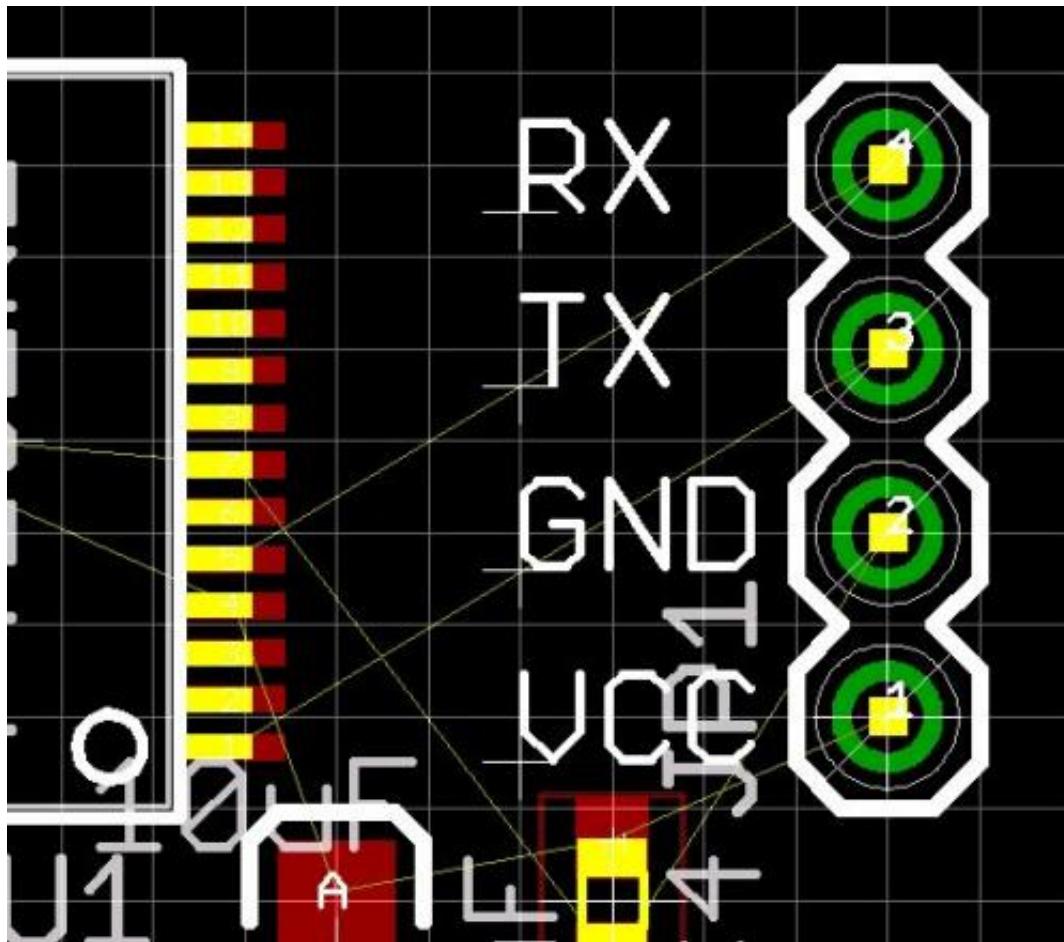
To check which pin is connected to which net, hit the eye button and click on a pin.



Checkout the text at the bottom of the Eagle window - pin 1 is VCC. Do this for all the pins and arrange the labels accordingly. You should also take the font size down to 0.05:



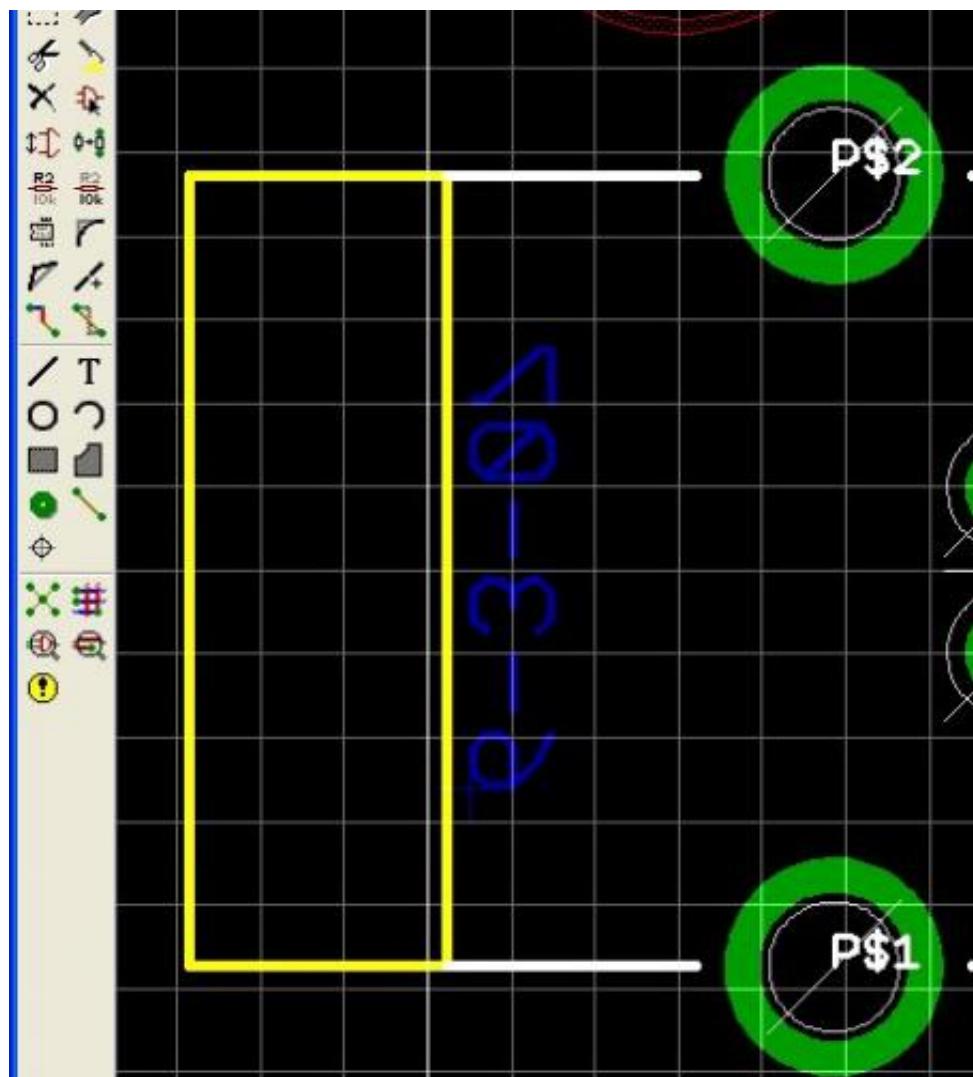
To change the font size, click on the Wrench ('Change' button), select 'Size', then 0.05. Now click on each text that you want to change the size on.



Labels in place and lined up

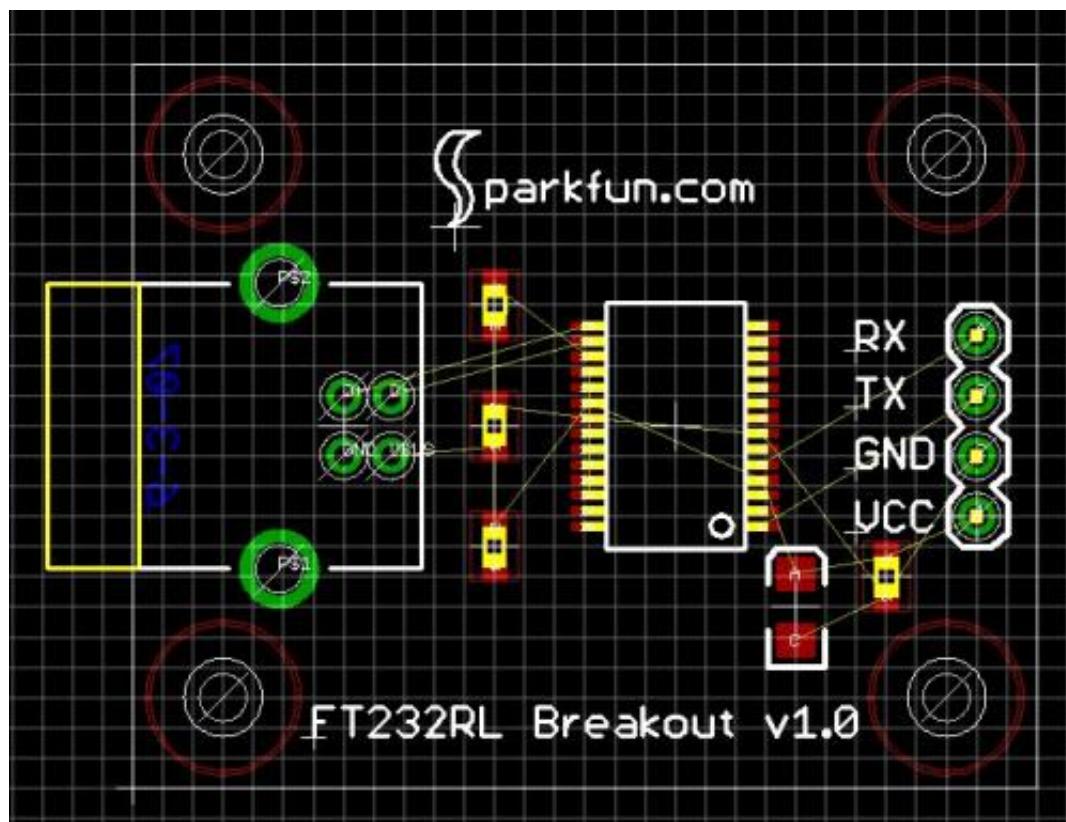
Try to get all the labels with the same vertical and horizontal alignment. This is a nit picky aesthetic thing, but it shows on the final board.

You will have many board revisions. It's always good to add a date code to the board so that you can match your files to the board version in hand. Add text to the bottom copper layer to an inconspicuous spot. The easiest way to do this is to add text to the top copper layer then hit the mirror button (you can also hit the scroll wheel on the mouse to move the component to a different layer). This will automatically mirror the text and drop it to the bottom layer.



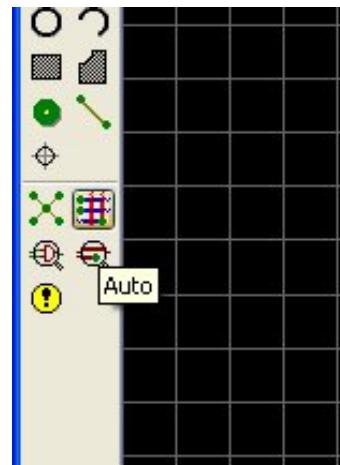
6-3-07 mirrored, on the bottom copper layer, and underneath the USB connector. There shouldn't be any signal traces in this part of the board so we're not wasting space. You could add text to the bottom silkscreen layer but some fab houses don't allow bottom silkscreen (it adds an extra printing step).

Also add some text to the top silkscreen layer indicating what the board is, what it does, who made it, etc.

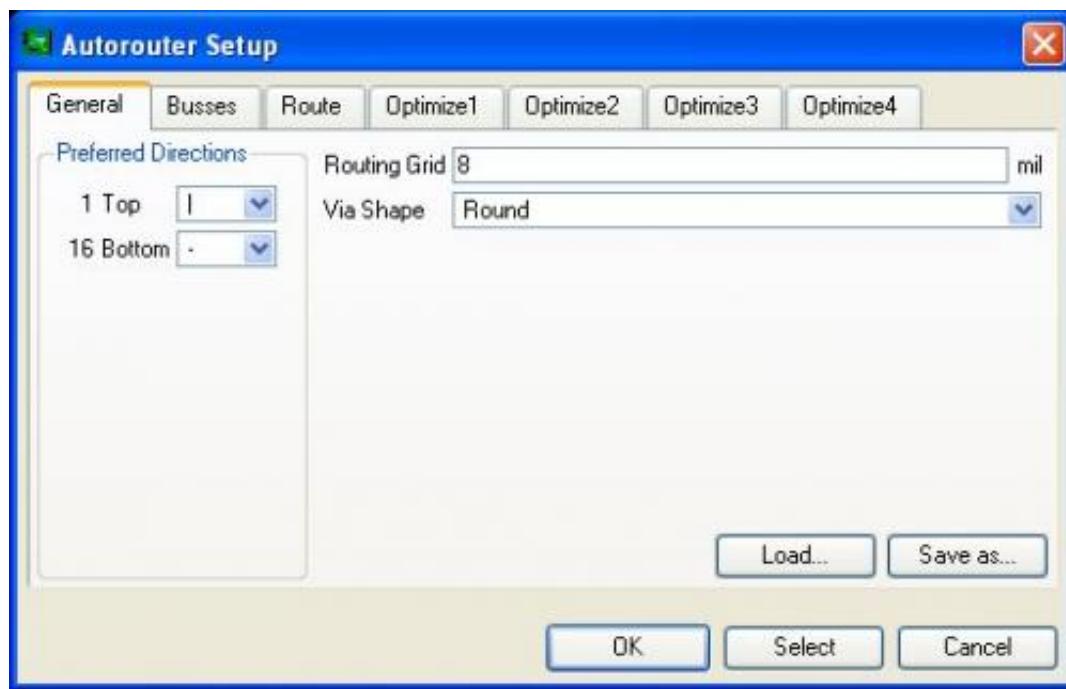


Above is the completed board ready for trace routing.

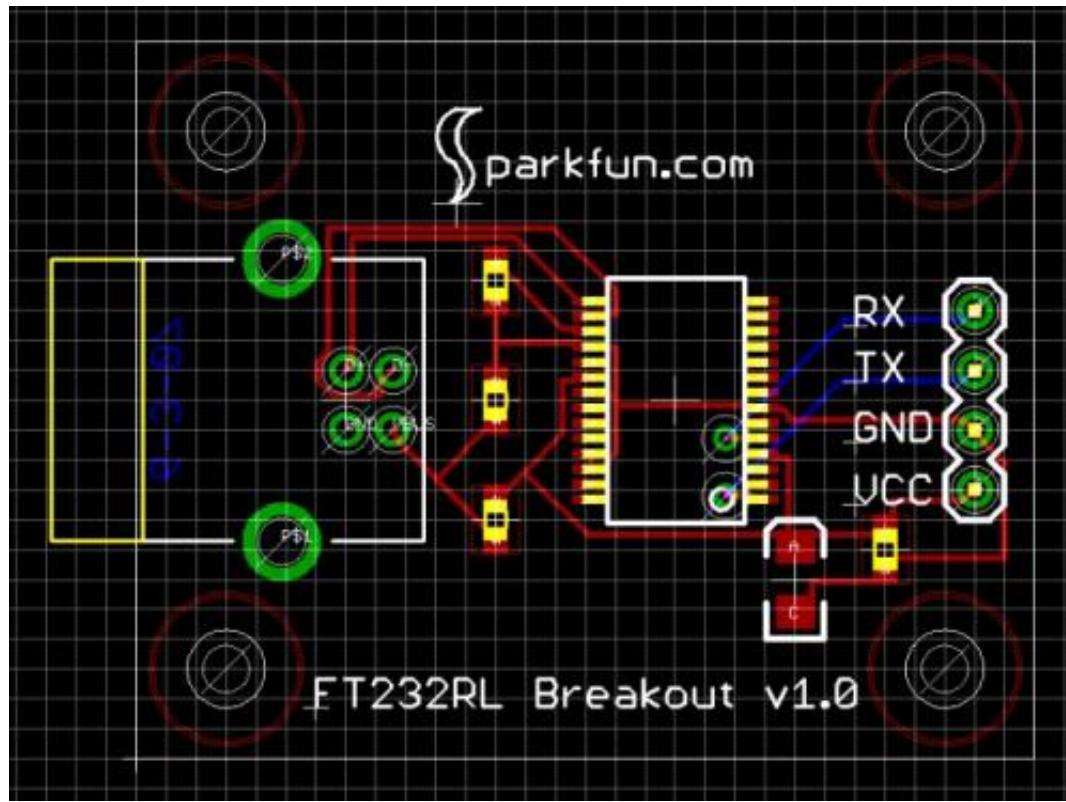
Many people swear up and down that an auto-router is a bad idea. It may be, but if we're not concerned about trace impedance or high speed signals, an auto-router is a great way to whip up protos. Spend your time innovating, not routing mundane traces.



To auto-route the board, click on the 'Auto' button. The defaults are all fine except for the 50mil grid:



Change the Routing Grid to 8 (our fab house uses 8 mil traces and spaces).



Demo board auto-routed. Eagle [files / PDF](#)

It's not immaculate, but it will work just fine and the router took under 2 seconds to route. It would probably take me 5-10 minutes by hand. I will hand route sensitive parts of certain boards, but this is a very simple proto.

Things to check on every layout:

- Date code

- Silkscreen title and pin labels
- Standoff holes
- All connections routed
- TX and RX routed correctly
- Print off 1:1 and check footprints

Routing TX and RX correctly? What does this mean? I can't tell you how many times I've heard newbies say 'well my PCB *would have* worked, but the manufacturer swapped the TX and RX pins'. No, the manufacturer did not swap them, the newbie neglected to actually *read* the datasheet (RTFD!). Sometimes an RX pin is an input. Sometimes an RX pin is an output. If you get it wrong, you'll look like a dunce. Read the datasheet and verify that everything is kosher. In the case of our FT232RL breakout, the TX pin is an output and RX pin is an input (pretty standard). I'm going to change the silkscreen indicators to read 'RX-I' and 'TX-O'. This should remove any doubt in my mind when I'm using the board a year from now - and so I don't have to go digging up the datasheet.

The last thing to check - print off 1:1 means to print on paper and 1 to 1 scale of the board layout. Then place the components on top the paper to verify that everything fits their associated footprint.

Another quick note about printing layouts - this can be a great way to create an assembly sheet. While you've got the PCB editor open, press Alt+F11. This will turn off all the layers that are not pertinent to assembly. Print this, I normally scale it up 3 times, and hand write in what parts go where. Quick and effective way to create a cheat sheet for PCB population. Pressing F11 will bring back the normal layers but you may want to activate/deactivate a few others.

Now that we've got the board laid out, stand-off holes in place, date code in place, and accurate pin labels, it's time to mash up the layers and create some gerber files.

What's a gerber?

There are lots of different layout packages out there (Protel, Orcad, Eagle, PCB, etc). One way or another all the PCB fab houses out there need a way to control their machines to work with your layout files. The universal format is something called **Gerber Files**. Basically these are txt files with coordinates that tell the PCB machines to go to location X,Y and do something (drill, expose, etch, print, etc). Because there are different layers to your PCB, you need to create different text files for the different layers. This is where the Eagle CAM program comes into play.



Once you're good and happy with your PCB layout (there is no turning back after you submit the files!), click on the CAM button to bring up the processor. This window will allow you to do different things to different layers. Eagle comes with a couple default *.cam files. The most common ones are the gerb274x.cam and excellon.cam. I got tired of running two separate processors. I am also a hold over from Protel and a different naming convention so I created my own single file CAM processor. You can snag it [here](#). The [sfe-gerb274x.cam](#) is based on the default Eagle file with a few tweaks:

- Layers are renamed for easier reading - top copper, bottom silkscreen, etc.
- All the mirroring is turned off - this will make gerber inspection much easier
- A top paste layer was added in case you want to create a solder paste stencil
- Excellon drill file is created along with the 6 magic layers
- Drill file is 2:4 Leading (remember this!)

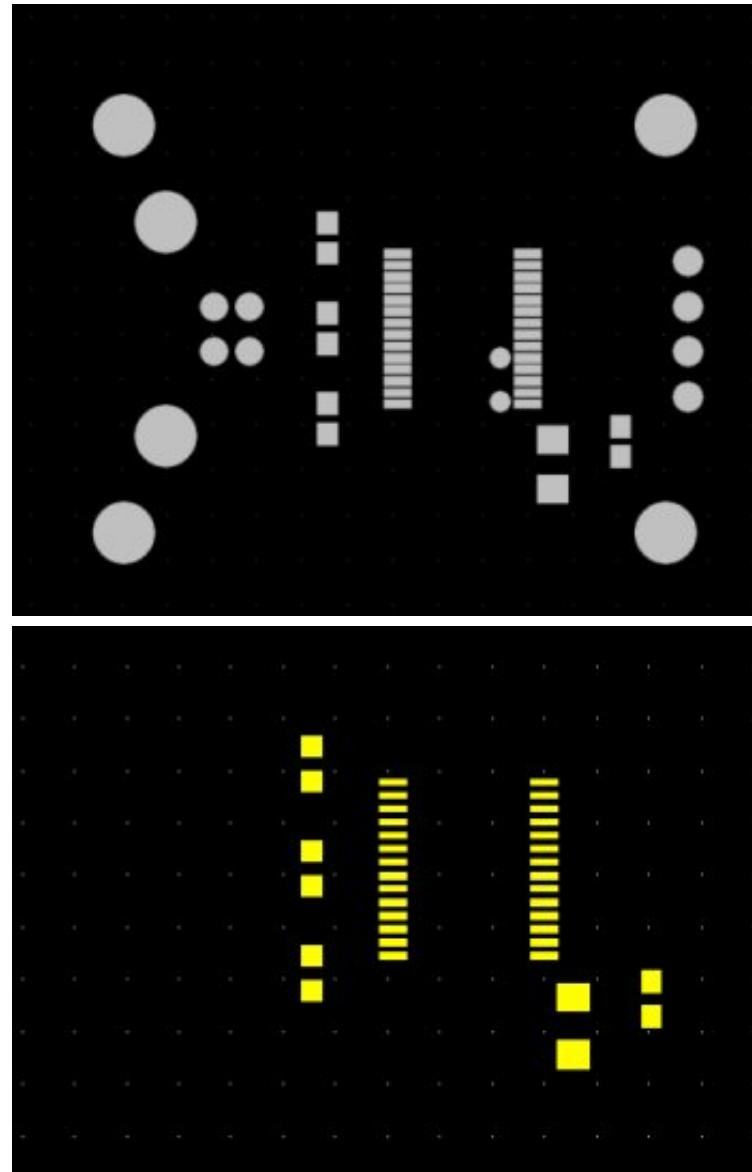
What are these magical layers you ask of? Anytime you transmit a PCB layout to a fab house, you need to pass them 7 files, and 7 files only:

- Top Copper (GTL)
- Top Soldermask (GTS)
- Top Silkscreen (GTO)
- Bottom Copper (GBL)
- Bottom Soldermask (GBS)
- Bottom Silkscreen (GBO)
- Drill File (2:4 leading - remember this)

What is this GTL, GTS? These are the file extensions that the CAM processor will produce. A silkscreen is also called an 'overlay' (hence, GTO). And for your reference, a soldermask is also called a solder 'stop layer' because the soldermask prevents solder from being where it is not wanted.

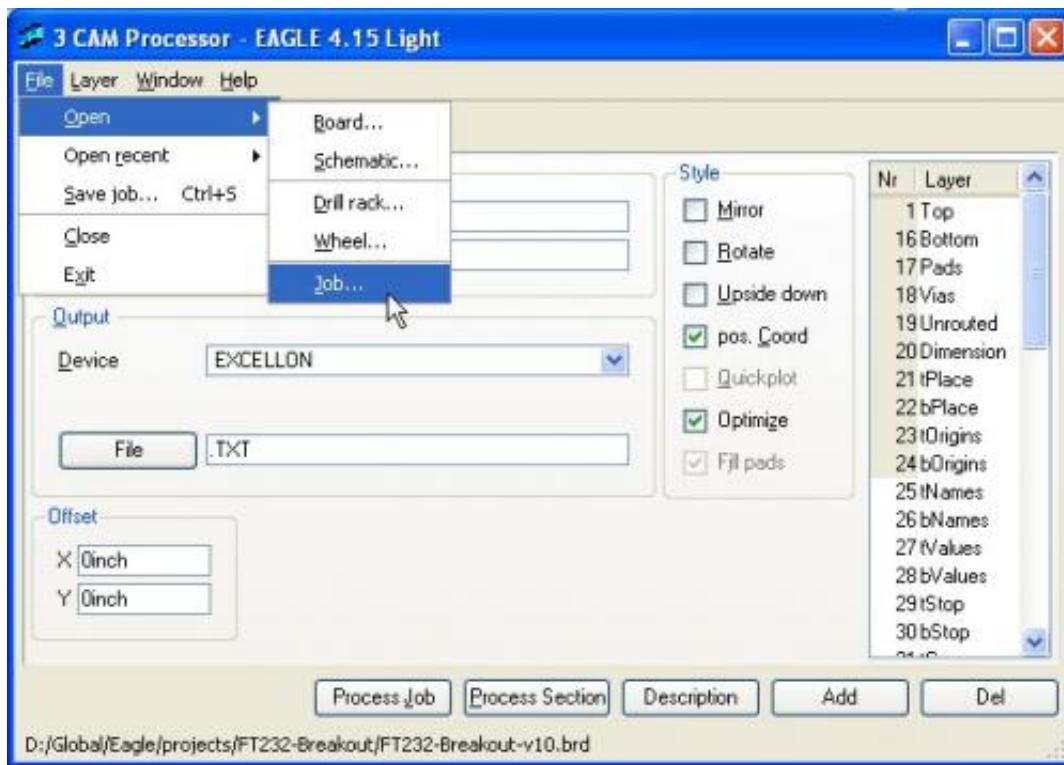
Some fab houses will charge extra for a bottom silkscreen layer. You can just ignore this layer if you need to. You will also see a GTP file extension. This is the Gerber Top Paste file. You can use this file to get a solder

paste stencil cut it you want. One would think that a GTP file would be the same as the GTS (top soldermask file). But no!



On the left, the soldermask layer. On the right, the top paste layer. The soldermask layer exposes the pads *and the vias*. You wouldn't want solder paste in vias! So the paste layer only has the SMD component pads exposed.

To create these layers, click on File->Open->Job



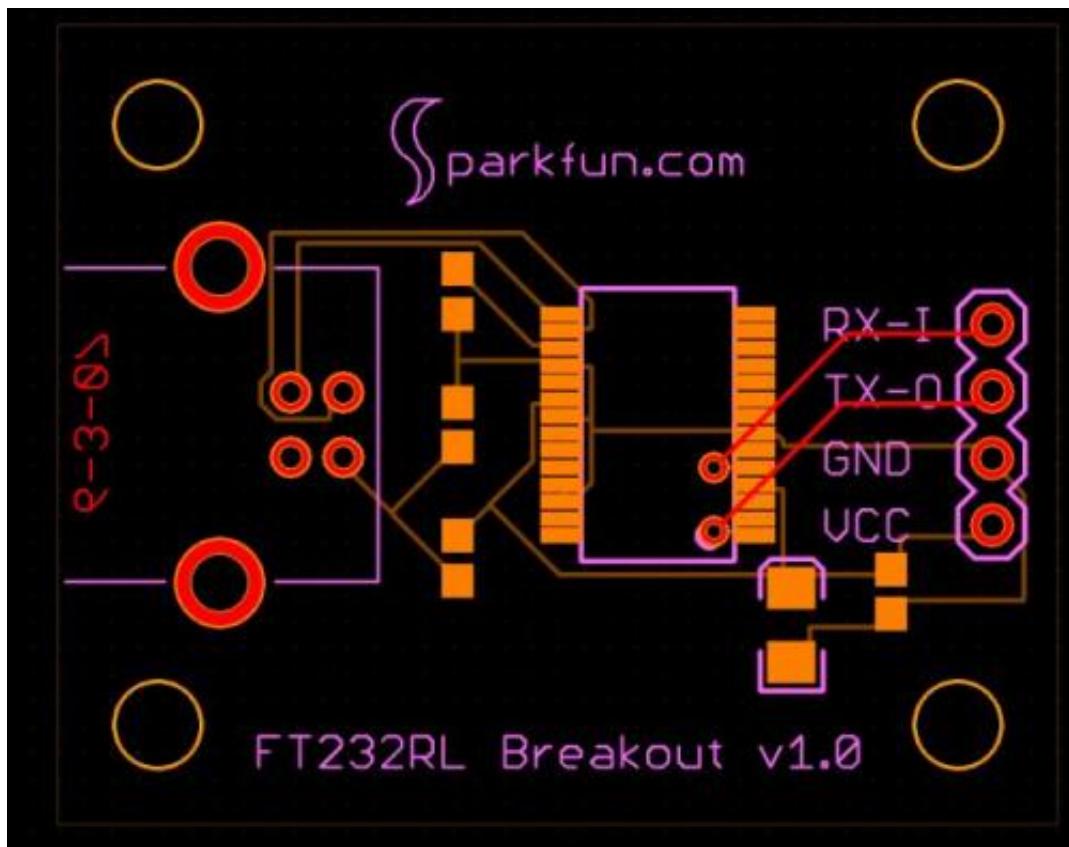
And select the sfe-gerb274x.cam file. Then click 'Process Job'. Some status bars will blink by, and within a few seconds, you should have a handful of extra files in your project directory:

eagle.epf	4 KB
FT232-Breakout-v10.b#1	9 KB
FT232-Breakout-v10.b#2	8 KB
FT232-Breakout-v10.b#3	8 KB
FT232-Breakout-v10.brd	12 KB
FT232-Breakout-v10.dri	1 KB
FT232-Breakout-v10.GBL	2 KB
FT232-Breakout-v10.GBO	1 KB
FT232-Breakout-v10.GBS	1 KB
FT232-Breakout-v10.gpi	2 KB
FT232-Breakout-v10.GTL	9 KB
FT232-Breakout-v10.GTO	10 KB
FT232-Breakout-v10.GTP	1 KB
FT232-Breakout-v10.GTS	2 KB
FT232-Breakout-v10.pro	2 KB
FT232-Breakout-v10.s#1	23 KB
FT232-Breakout-v10.s#2	23 KB
FT232-Breakout-v10.s#3	23 KB
FT232-Breakout-v10.s#4	5 KB
FT232-Breakout-v10.sch	24 KB
FT232-Breakout-v10.TXT	1 KB

The magic 7 gerber files

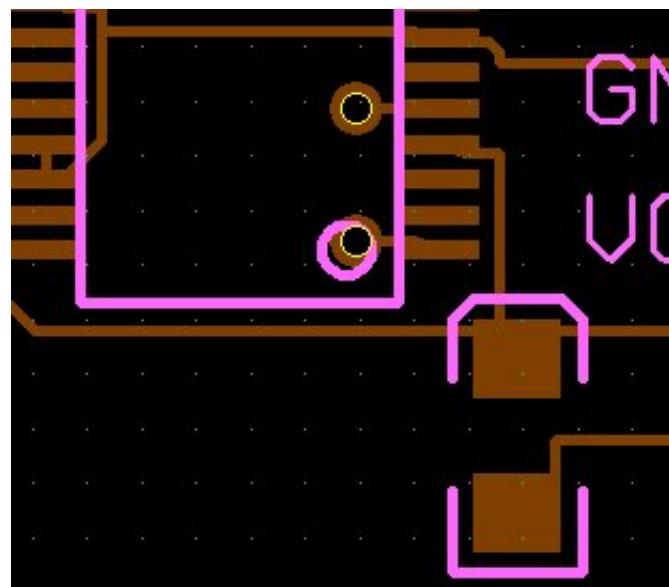
Something I highly recommend is to review the gerber files before submitting them for fabrication. Viewing just what is going to the fab house can expose potential problems that were shrouded before by all the extra layers and graphics in the Eagle layout window. There are some free viewers out there that will let you view the gerber

layers together. Most of the free viewers require you to enter an email, require 15 seconds for them to advertise at you, limited to one layer, or other really annoying limitations. Luckily, [Viewplot](#) still exists. This free program will let you open and look at your layers easily and you don't have to fill out any silly forms to get to the download link! Be sure to select the drill file type '2:4 Leading' to matchup the holes to the layers.



Simple

But oh wait, what is that?

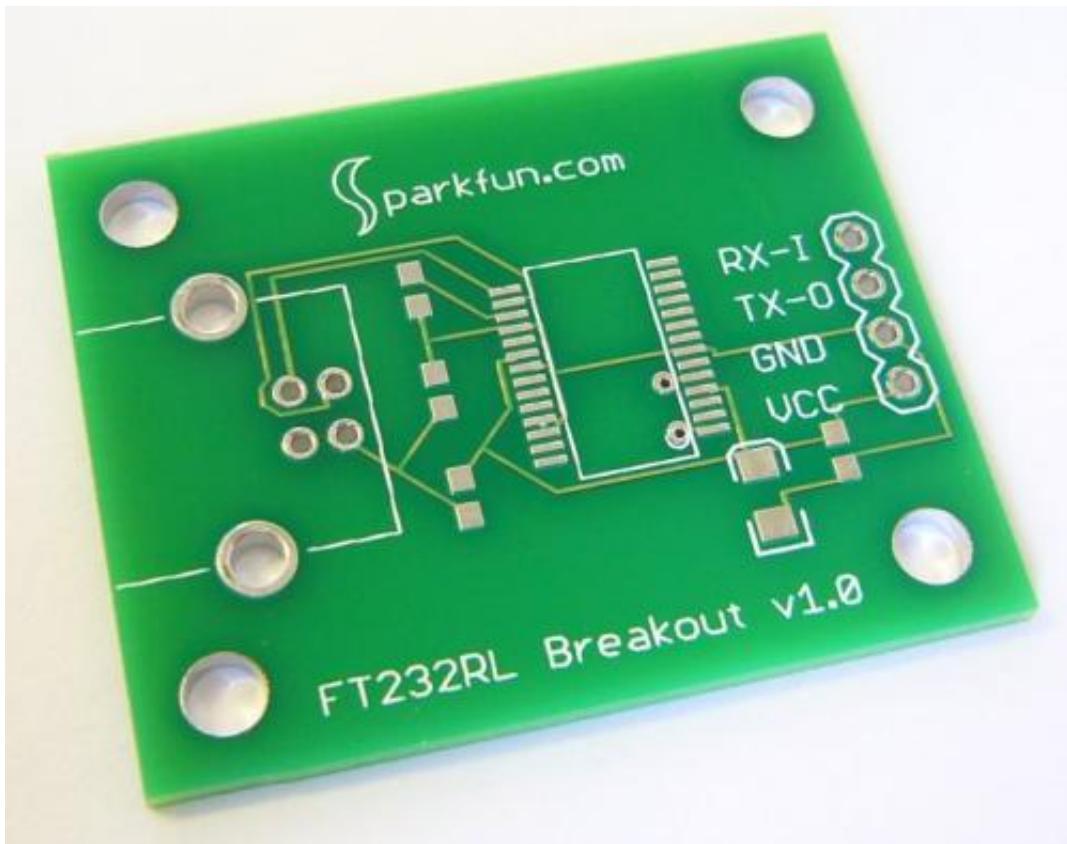


Silkscreen on top of a via

This silkscreen being broken up by the via is not a big deal. Mostly this is just to show how difficult it can be to detect problems from within the layout program. By viewing what you're actually submitting to the fab house, you can see exactly what they are going to deliver - be it good or bad. Review your gerber files before you submit! You'll save yourself time and monstrous amounts of money. If you want a \$50 coaster, I'll sell you a truck full.

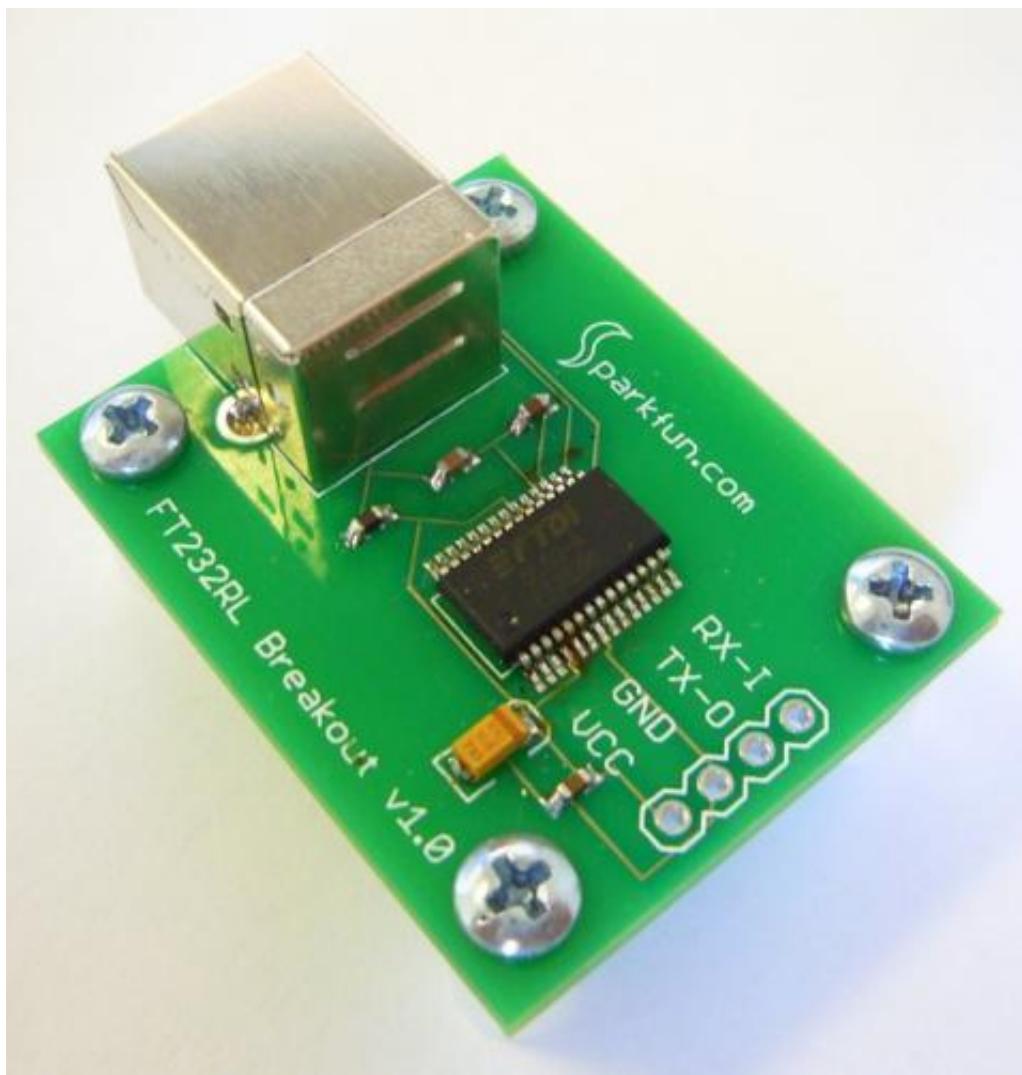
Zip these 7 files together and shoot them off to the fab house of your choice (Shameless Plug: we use [BatchPCB](#)). Depending on who you use and how deep your pockets are, you'll have to sit on your hands for 1 day up to 20 days. What do I do? I start other designs! While this board is being fabbed in 2 weeks, I create and submit a new design 7 days into it. That way I always have a new PCB proto coming in every few days. It's like Christmas every week! What am I getting this week? Is it going to work? I can't wait to test out my new cat tracker!

With the magic of television, I can jump 20 days into the future and show you...



In the flesh

It looks great! Time to whip out some parts and the soldering iron.



Assemble your new board!

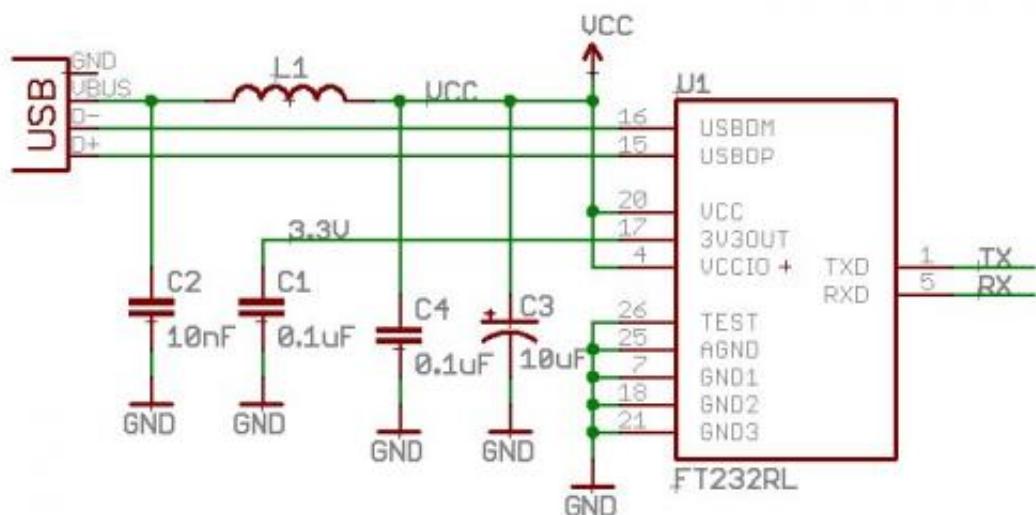
Soldering was not too bad. Silkscreen looks good. Standoffs look good. Time for testing!

Always assume a proto will short out the first time you use it. Be very cautious and be ready to kill power immediately. Plugging this board onto the computer - guess what happens?



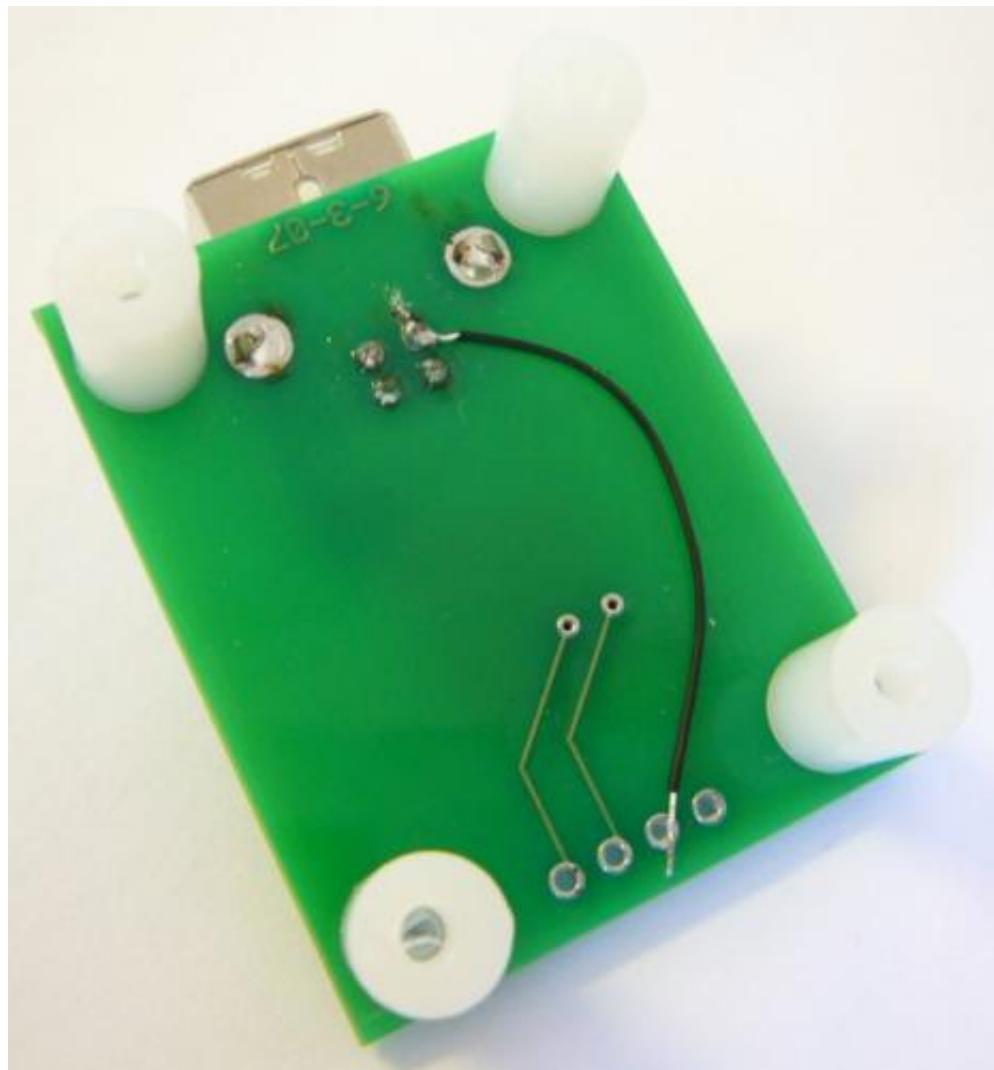
Uh-oh

What went wrong? Nothing is heating up. Nothing smoked or popped. The FT232 IC doesn't seem to be enumerating onto the USB bus. But why? Let's check the schematic one more time...



FT232 Breakout v1.0 Schematic

Son of a... This is why we prototype! Mistakes like this happens to the best of us. What is the problem you ask? Checkout the GND pin on the USB connector. It's not connected to anything. Without a ground connection to the board, no current can flow, the FT232RL will never enumerate. The board is shot! Or is it?



The green wire fix

(I know the wire is not green, but this is the nick-name for an oops! fix) This is my final thoughts on PCB layout - anyone can layout a PCB, but it takes a true magician to get a bad PCB working. Scratch, cut, splice, and otherwise modify your PCB until you get it working. This example was an easy fix. I've seen some really impressive fixes over the years.

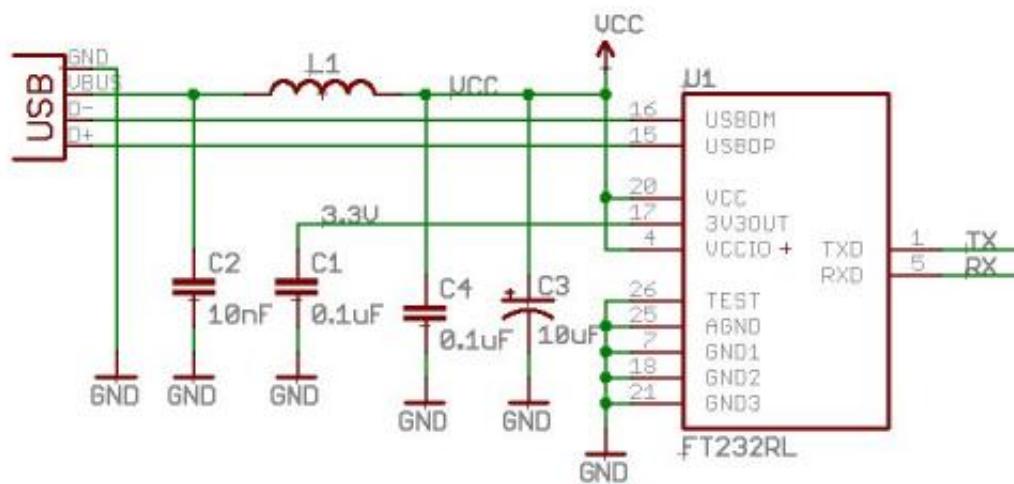


From the [SFE PCB History Museum](#)

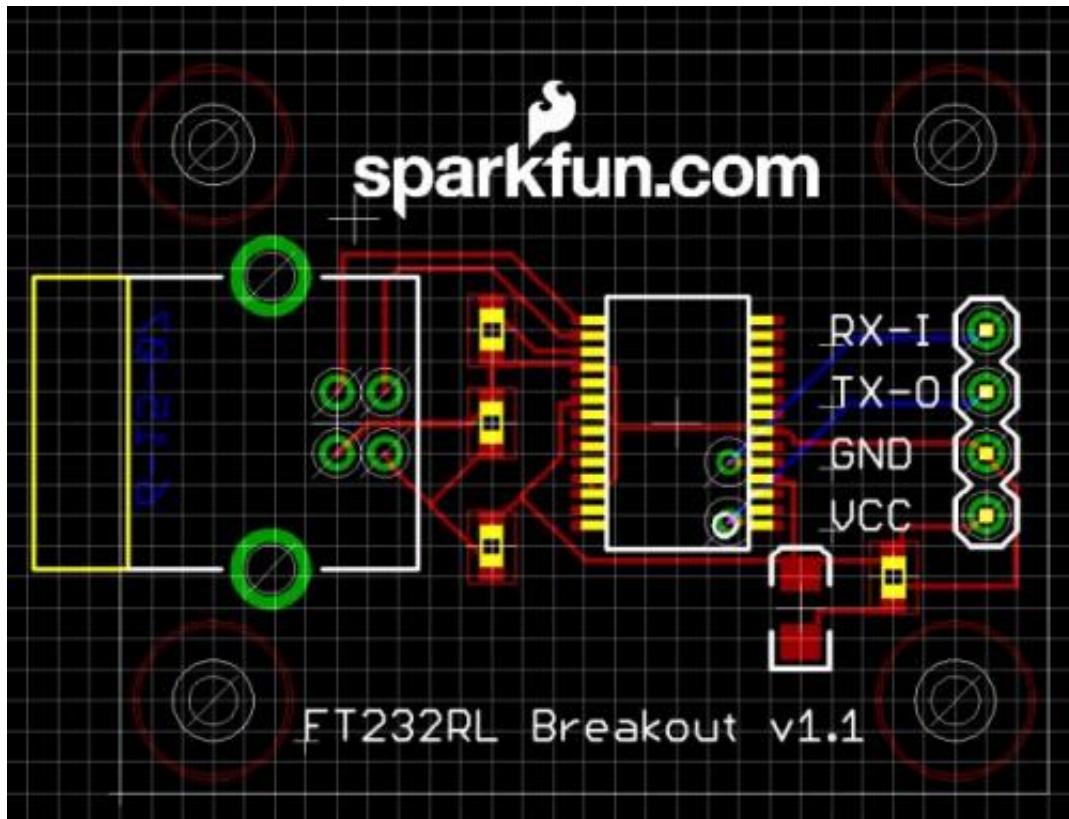
Don't immediately throw up your hands and layout a new board. Instead, make sure you get every bit of the functionality of your board working, by any means necessary, and then make all the revisions. Otherwise, you'll constantly spin your wheels with PCB revisions.

Things to remember:

- Be sure to save your schematic as a new file name: *-v11
- Be sure to update the date code on the board
- Be sure to update the silkscreen on the board



Updated v1.1 schematic with GND connection on the USB connector



Updated v1.1 board layout with new GND connection, updated silkscreen and date code

We love feedback! Please report typos, comments, or recommendations to spark@sparkfun.com.

[Lecture 1 - Background and Power Supply](#)

[Lecture 2 - How to Get Code Onto a Microcontroller](#)

[Lecture 3 - What is an oscillator?](#)

Lecture 4 - UART and Serial Communication

[Lecture 5 - AVR GCC Compiling](#)

[Lecture 6 - Soldering Basics](#)

[Lecture 7 - SMD Soldering](#)

[Lecture 8 - Eagle: Schematics](#)

[Lecture 9 - Eagle: PCB Layout](#)

[Lecture 10 - Eagle: Creating a new part](#)

[Common Mistakes, Tips and Tricks](#)

SparkFun Electronics ® | Boulder, Colorado | Customer Service