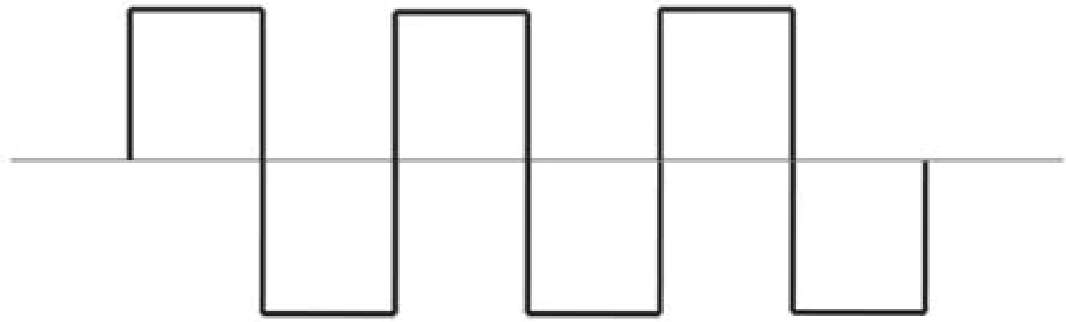
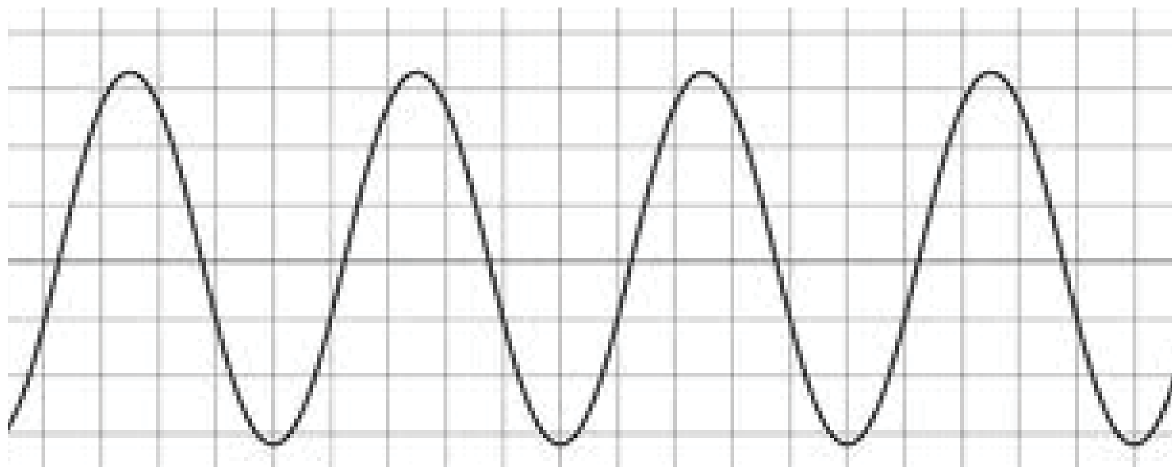
SOUND_QUICK RECAP

ARDUINO_SOUND

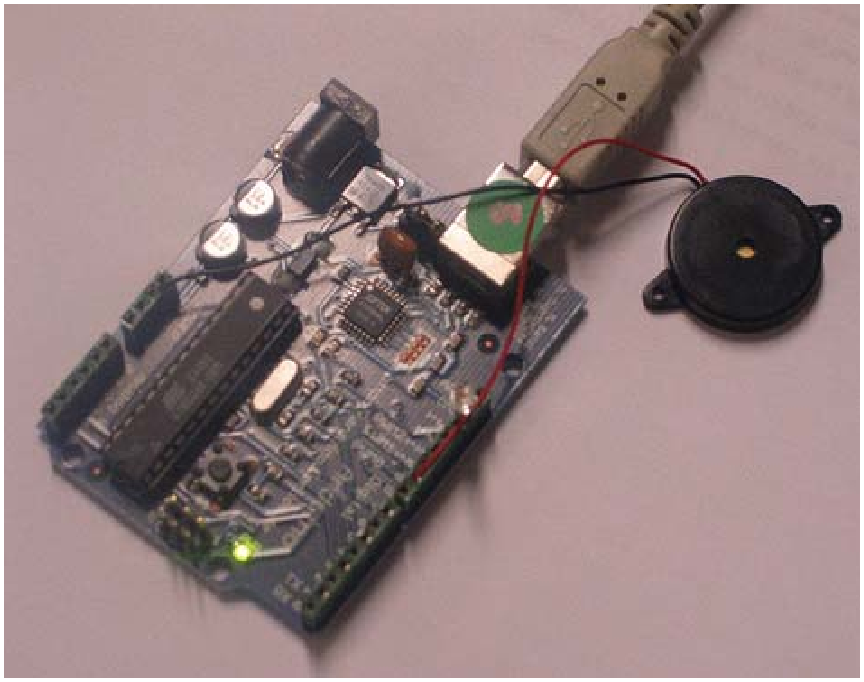
SQUARE WAVE



SINE WAVE(SOUND)



ARDUINO_SPEAKER



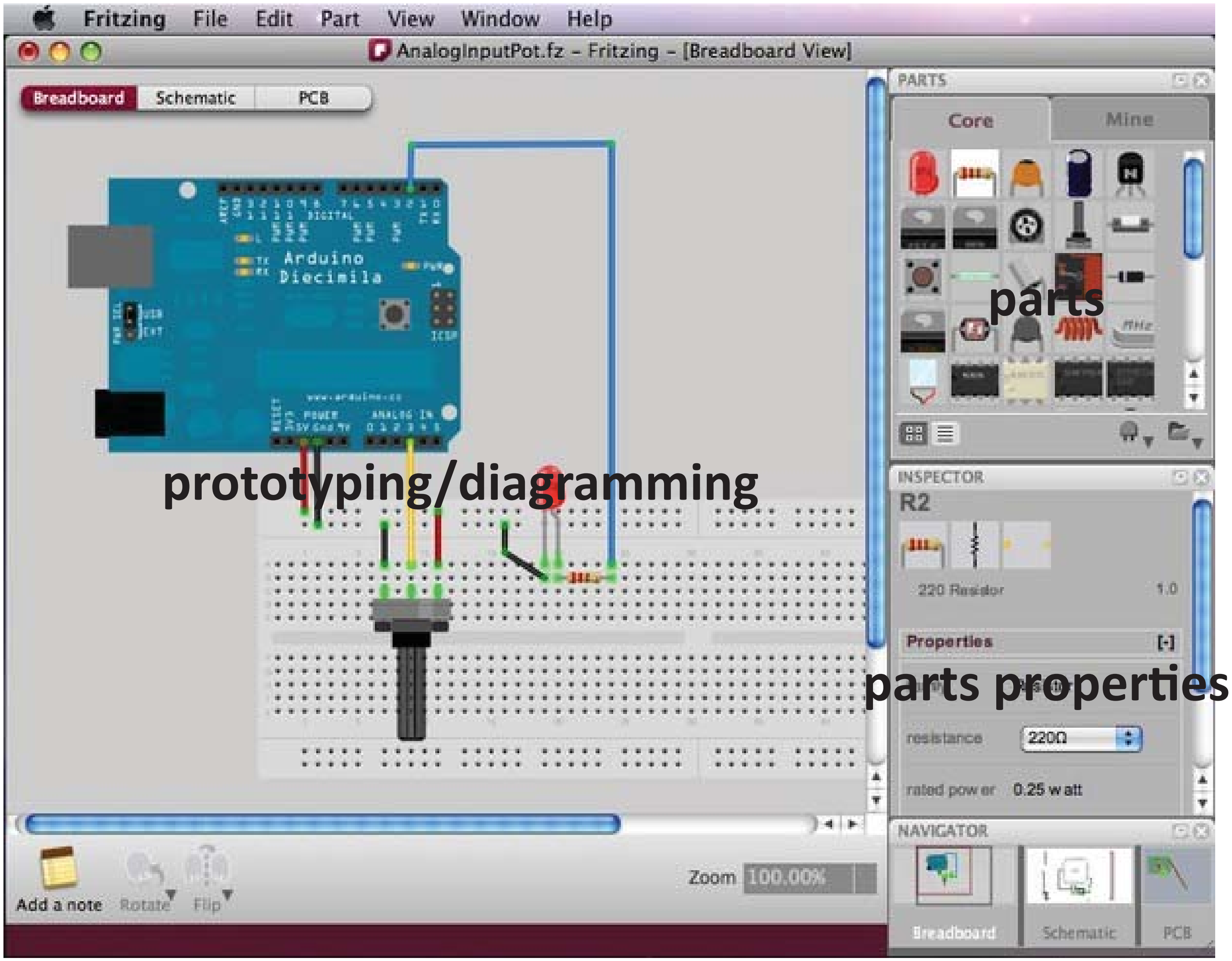
Constant Name	Frequency (Hz)	Constant Name	Frequency (Hz)
NOTE_B2	123	NOTE_GS5	831
NOTE_C3	131	NOTE_A5	880
NOTE_CS3	139	NOTE_AS5	932
NOTE_D3	147	NOTE_B5	988
NOTE_DS3	156	NOTE_C6	1047
NOTE_E3	165	NOTE_CS6	1109
NOTE_F3	175	NOTE_D6	1175
NOTE_FS3	185	NOTE_DS6	1245
NOTE_G3	196	NOTE_E6	1319
NOTE_GS3	208	NOTE_F6	1397
NOTE_A3	220	NOTE_FS6	1480
NOTE_AS3	233	NOTE_G6	1568
NOTE_B3	247	NOTE_GS6	1661
NOTE_C4	262	NOTE_A6	1760
NOTE_CS4	277	NOTE_AS6	1865
NOTE_D4	294	NOTE_B6	1976
NOTE_DS4	311	NOTE_C7	2093
NOTE_E4	330	NOTE_CS7	2217
NOTE_F4	349	NOTE_D7	2349
NOTE_FS4	370	NOTE_DS7	2489
NOTE_G4	392	NOTE_E7	2637
NOTE_GS4	415	NOTE_F7	2794
NOTE_A4	440	NOTE_FS7	2960
NOTE_AS4	466	NOTE_G7	3136
NOTE_B4	494	NOTE_GS7	3322
NOTE_C5	523	NOTE_A7	3520
NOTE_CS5	554	NOTE_AS7	3729
NOTE_D5	587	NOTE_B7	3951
NOTE_DS5	622	NOTE_C8	4186
NOTE_E5	659	NOTE_CS8	4435
NOTE_F5	698	NOTE_D8	4699
NOTE_FS5	740	NOTE_DS8	4978
NOTE_G5	784		

PROCESSING _MINIM LIBRARY: LOAD SAMPLE

```
import ddf.minim.*;
Minim minim;
AudioSample kick;
void setup()
{
 size(512, 200, P2D);
 // always start Minim before you do anything with it
 minim = new Minim(this);
 // load BD.mp3 from the data folder with a 1024 sample buffer
 // kick = Minim.loadSample("BD.mp3");
 // load BD.mp3 from the data folder, with a 512 sample buffer
 kick = minim.loadSample("BD.mp3", 2048);
}
void draw(){
 background(0);
 stroke(255);
 // use the mix buffer to draw the waveforms.
 // because these are MONO files, we could have used the left or
right       buffers and got the same data
 for (int i = 0; i < kick.bufferSize() - 1; i++)
 {
  line(i, 100 - kick.left.get(i)*50, i+1, 100 - kick.left.get(i+1)*50);
 }
}
void keyPressed()
{
 if ( key == 'k' ) kick.trigger();
}
void stop()
{
 // always close Minim audio classes when you are done with them
 kick.close();
 minim.stop();
 super.stop();
}
```

EagleCad and Fritzing_CUSTOM PCB FABRICATION

FRITZING



Additional Fritzing Parts:
http://code.google.com/p/fritzing/issues/detail?id=875

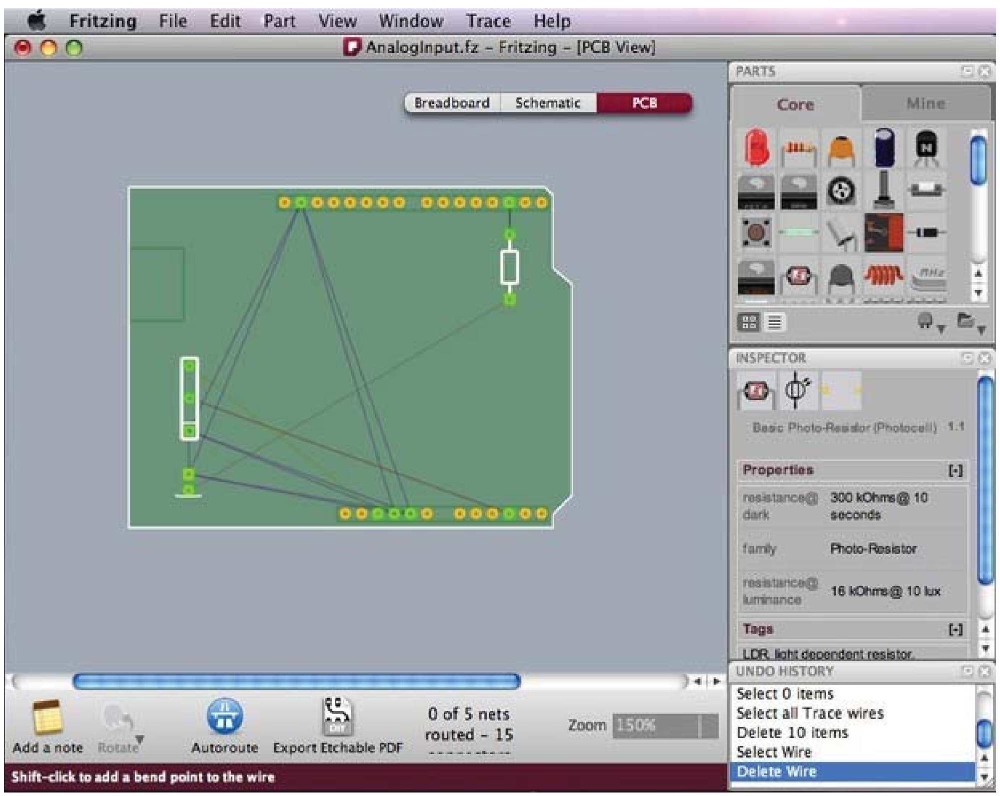
FRITZING_CUSTOM PCB BOARD

The first step in designing a PCB layout is arranging the parts on the board.
There are some very important issues to consider here, because the location of parts on the board will have a great effect on how successful the routing process will be.

Follow these guidelines:

- Place the parts with the most connections in the middle of the board.
- Notice that Arduino's footprint should also be positioned on the board, just like other parts (new in version 3.0).
- Rotate and position parts, leaving enough space between them (don't forget their actual size!).
- If the board is too small, redefine its width and height in the Inspector or alternatively resize the board by dragging its corners. Learn how to design a PCB with a custom shape.
- Don't place parts too close to the edges of the board.
- To avoid short circuits, don't place parts too close to the USB connector outline on the Arduino Shield.
- When designing stack shields, parts' heights should also be considered.

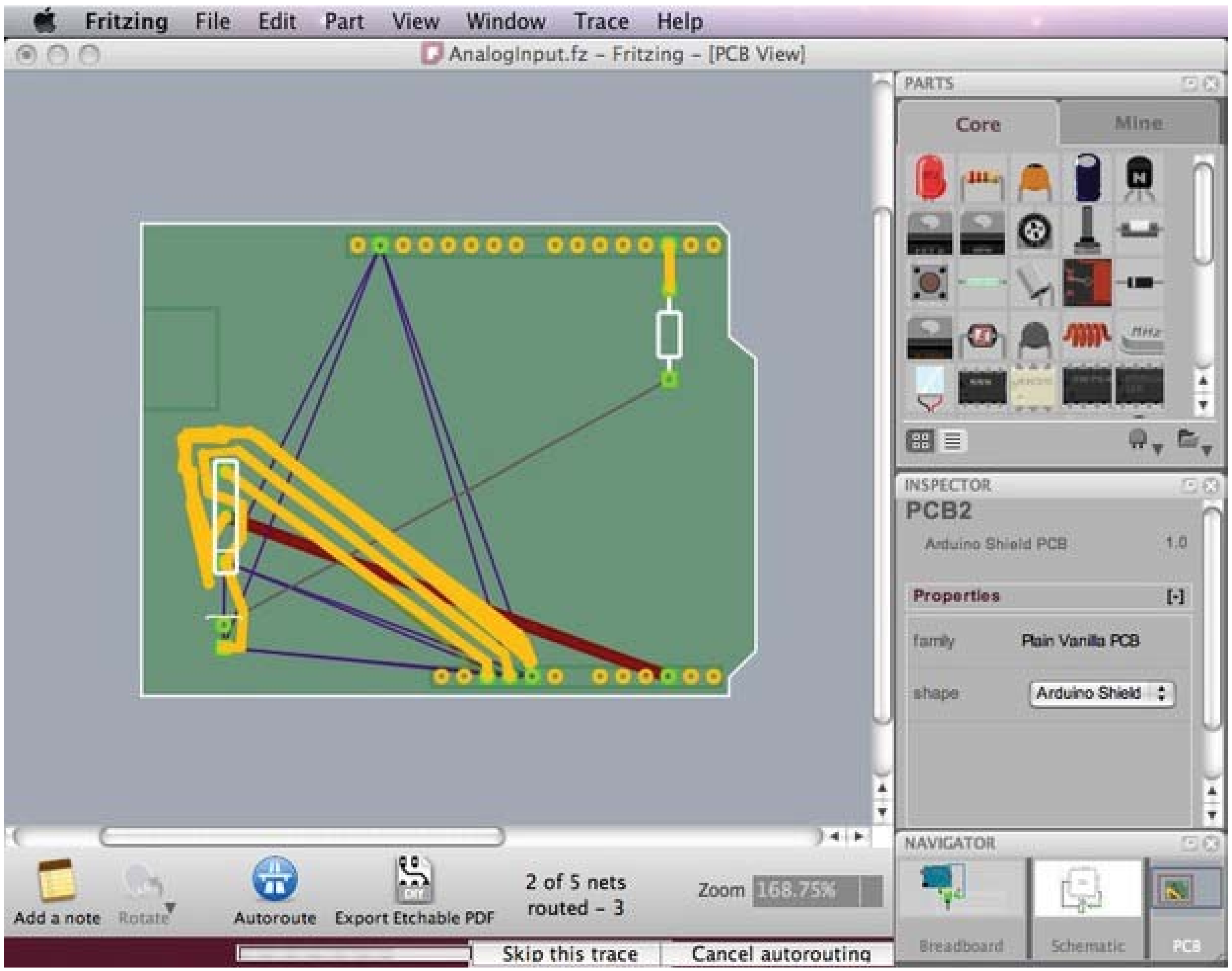
The following screenshot shows one out of many possible part arrangements for the given circuit:



Auto-routing

After positioning all parts on the board, be aware that parts are not really connected to each other yet. The thin connecting lines that you see (Rat's Nest Layer) only act as a guideline. We would now want Fritzing to automatically generate the connection traces between parts. Click the Auto-route function from the bottom menu bar.

If you notice that Fritzing is struggling trying to generate a connection, you can press the "Skip this Trace" button or "Cancel Auto-routing" in the bottom menu while in process.



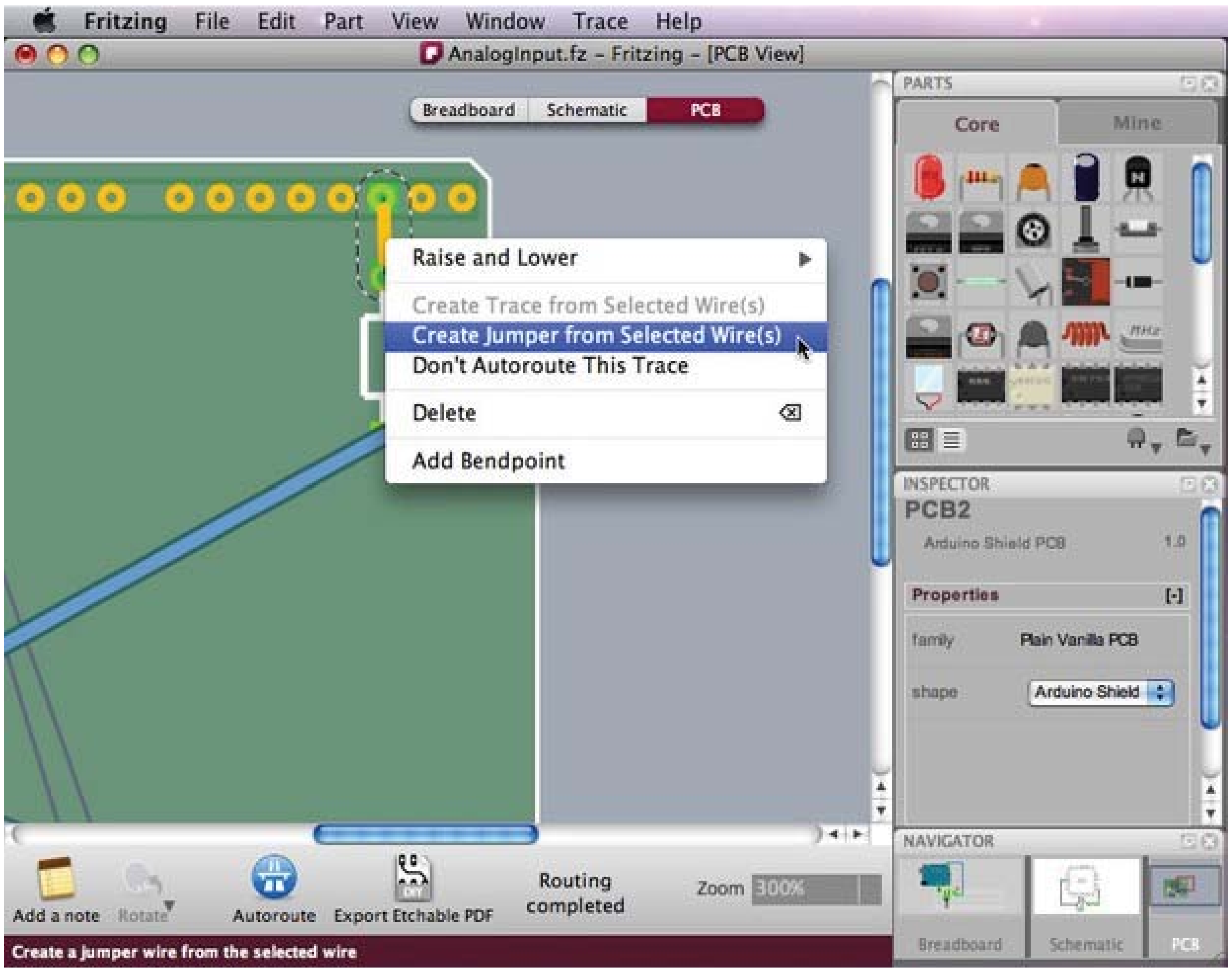
Hand-routing

Use any of the following methods to hand-route traces and jumpers:

The safest way is to right-click a Rat's nest wire and choose "Create Trace from Selected Wire(s)" or "Create Jumper from Selected Wire(s)". This will avoid making any changes in the circuit that you built in Breadboard View.

Another way is to simply click a part's connector, and drag to make a connection. A trace will be created. To create a jumper, just right-click on the trace and choose "Create Jumper from Selected Wire(s)". To avoid incorrect wiring, we strongly recommend you follow the Rat's nest wire connections while using this method.

Note that while clicking and holding on a connector, all equipotential connectors are highlighted (in yellow). This shows the whole set of connections attached to this particular connection, and can really help to make hand-routing decisions. Once again, take good care not to cross wires!



Guidelines for better routing

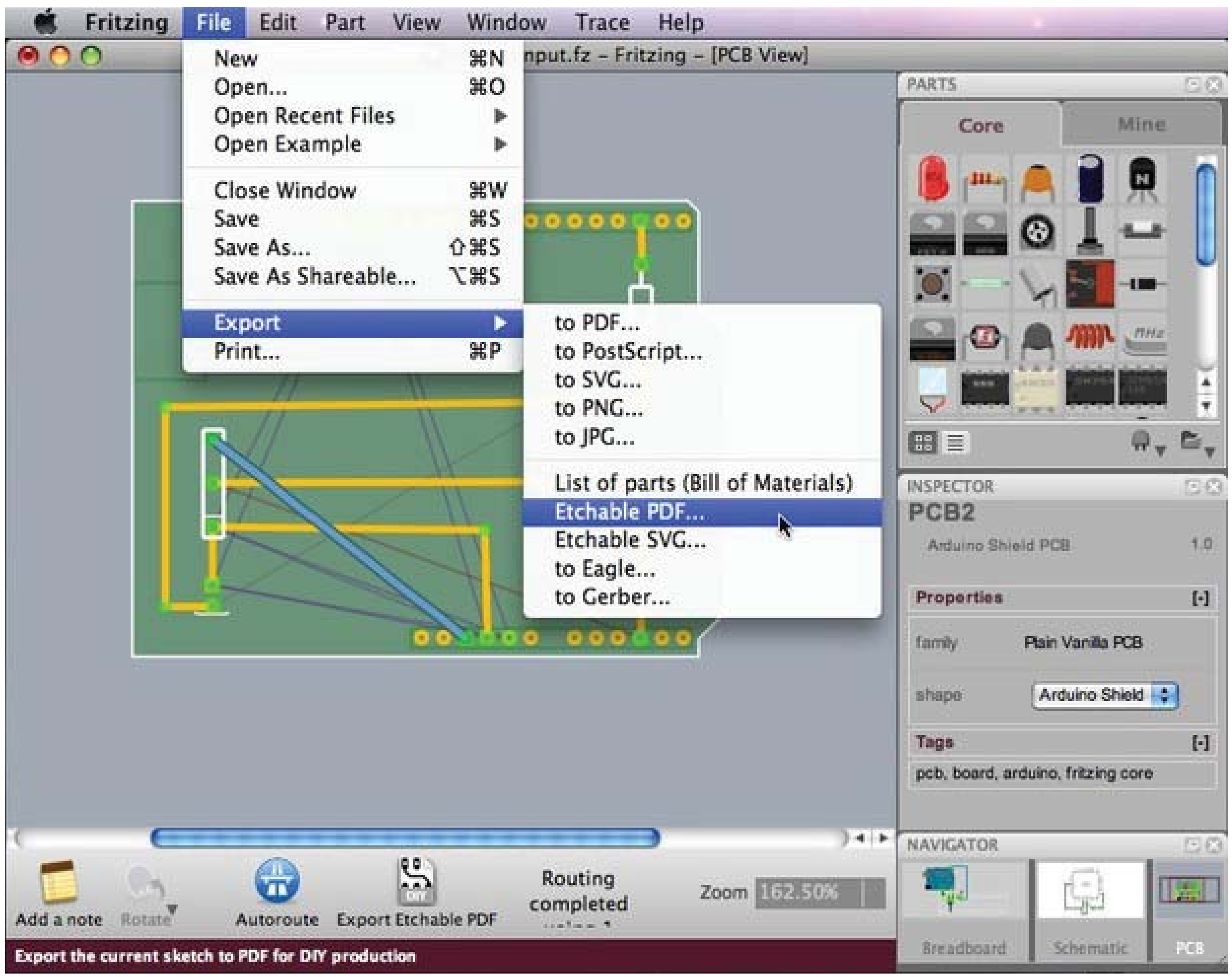
For both auto- and hand-routing, follow these guidelines:

-Place the parts with the most connections in the middle of the board.
-Try to get short connections by moving and rotating parts.
-Use the highlighting of equipotential connectors feature.
-Add bend points for tidy routing and so that lines do not cross.
-Don't forget the traces can go under parts like resistors.
-Use jumper wires instead of watching the auto-route go crazy.

Export Options

Fritzing features a variety of export options. When you are happy with your PCB design, you can choose to export JPG, PNG, etchable PDF and even Gerber files (for sending a professional PCB manufacturing service). The Bill of Materials option generates a list of all parts in the circuit.
From the menu bar choose File > Export > and the desired format.

-For DIY PCB production, use the Etchable PDF option which exports only the necessary design for etching.
-When exporting Gerber files, create a folder for the gerbers, and zip. it before sending to a manufacturer.

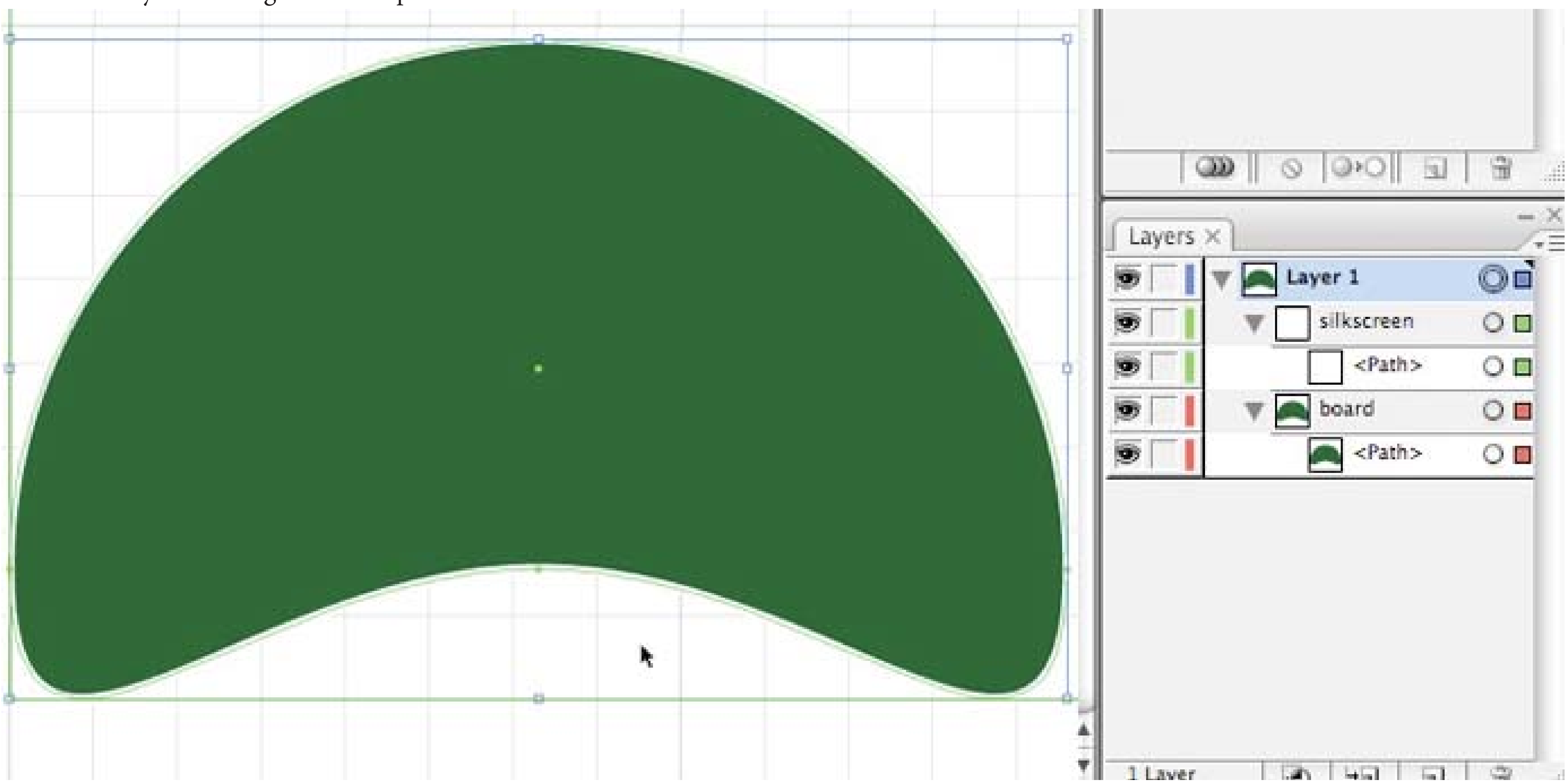


FRITZING_CUSTOM PCB BOARD SHAPE

Fritzing lets you design your PCB in any size and any shape. When you start working on a new sketch, a default rectangle board is placed in the PCB View. You can then define width and height in the Inspector or alternatively resize the board by dragging its corners.
To use boards with more complex shapes, create your board's shape with an external SVG editor (Inkscape, Illustrator etc.) and then import it to Fritzing. Follow these steps:

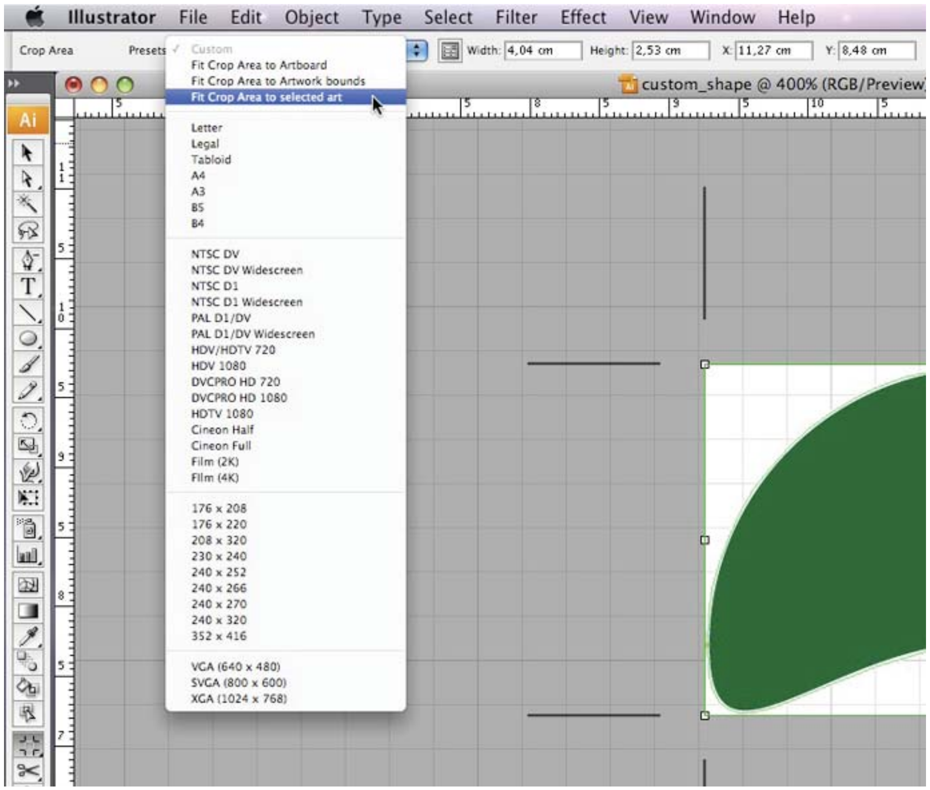
1. Design an SVG with two sub-layers:

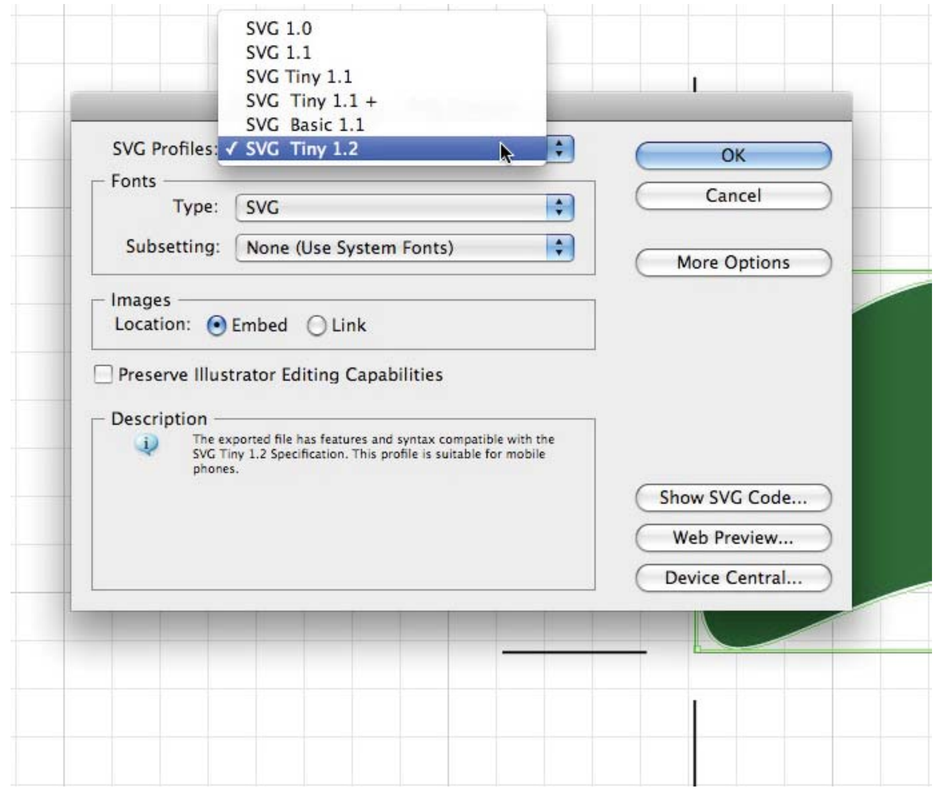
board layer with a path in any shape and size (NOTE: for gerber export, please use only polygons!). The shape should be filled green and with no stroke.
A silkscreen layer with a copy of the path you've just done, this time with no fill and with white stroke (for Inkscape users: stroke = 8 mil). Make sure the silkscreen layer is arranged at the top.



FRITZING_CUSTOM PCB BOARD

2. Illustrator Users: Align all elements to the center of the artwork
3. Illustrator Users: Use the crop tool and choose the preset “fit crop area to selected art”





6. Drag a plain PCB to the PCB View.
7. In the Inspector, select “import shape” from the shape drop down menu, navigate to the SVG file and press “Open”. Your custom PCB shape is ready to use!

EagleCad

Here are some of the basic quick keys:

Press **escape** at any time to stop the current action and return to the schematic 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)

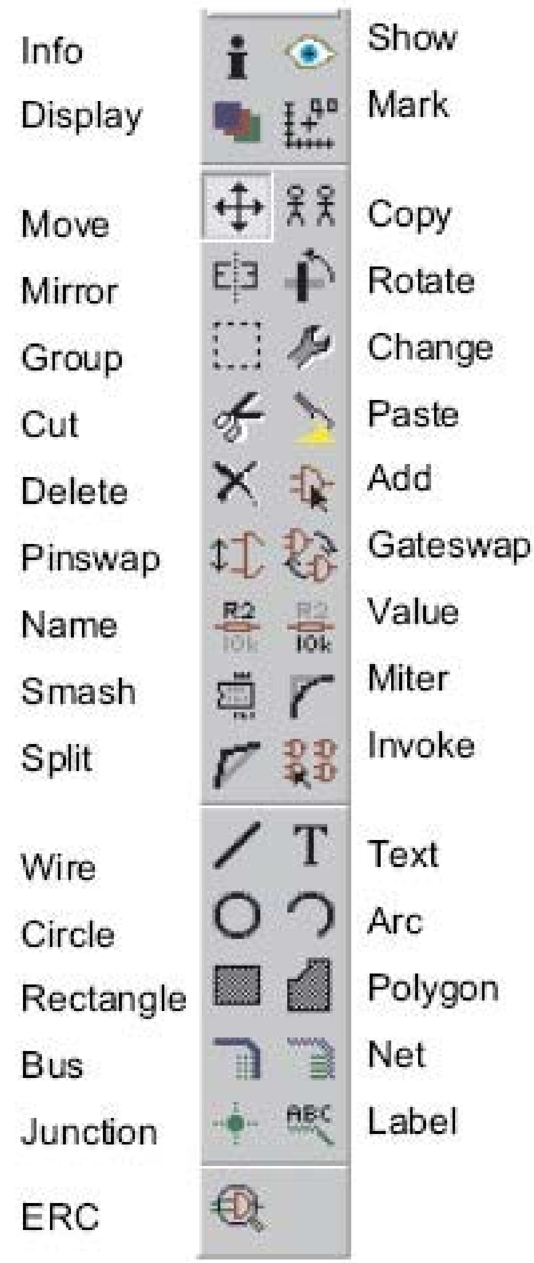
F5 to re-value a part (change 0.1uF to 10uF, etc)

F6 to smash a part (be able to move the name and value tags)

F9 to start a wire

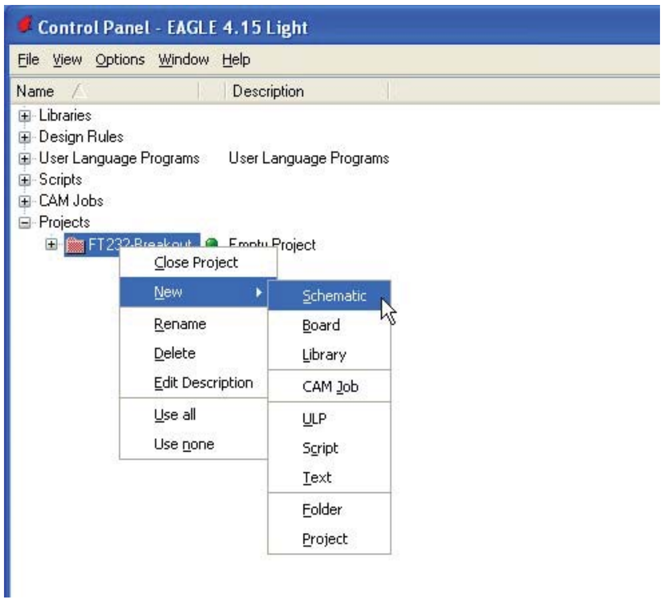
Alt+F9 to add a label to a wire

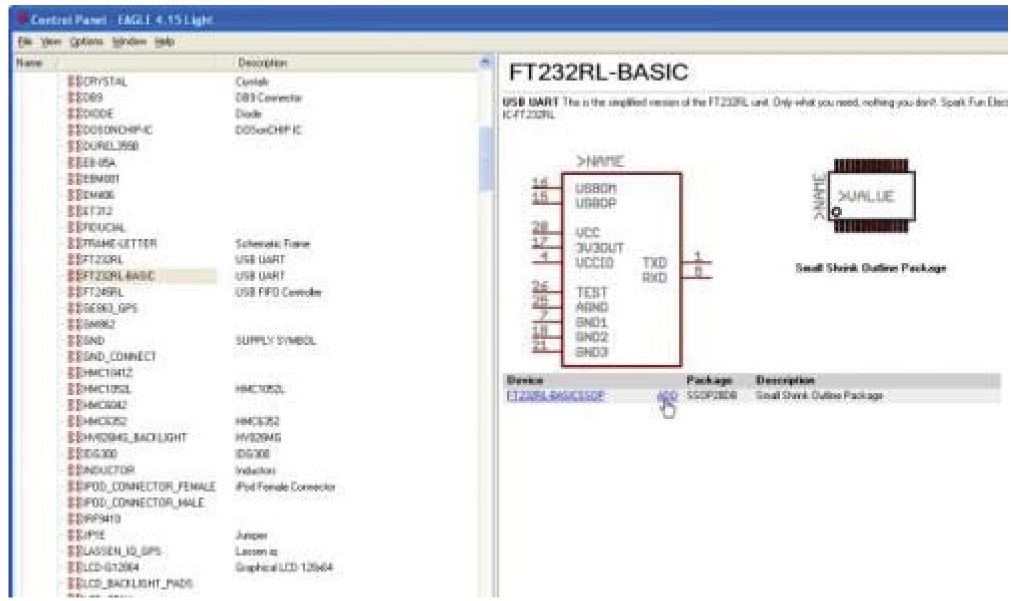
NEVER change the grid size in the schematic editor. Leave it on 0.1inch steps and don't use the alternate 0.01 step. If you do, you won't be able to hook wires to the pin tie points.



EagleCad_schematics

Now let's add the FT232RL part to our schematic. Close the library editor and go back to the Eagle Control Panel. Click on File->New->Project. Name this new project 'FT232-Breakout'. Right click on the FT232-Breakout project and create a new Schematic:





The schematic editor should open. Now go back to the Eagle Control Panel and expand the SparkFun Library:
You should see a long list of parts. Highlight the FT232RL-Basic part and in the right screen click on ADD. The schematic editor will pop up allowing you to place the FT232RL.

Now add these other items to your schematic:

1 x FRAME-LETTER : This will add a nice frame to your schematic. Add all parts inside this frame.

3 x CAP (Device name CAP0603) : 0.1uF/0.01uF 0603 capacitors

1 x CAP_POL (Device name CAP_POL1206) : 10uF tantalum capacitor

1 x INDUCTOR (Device name INDUCTOR0603) : Ferrite bead

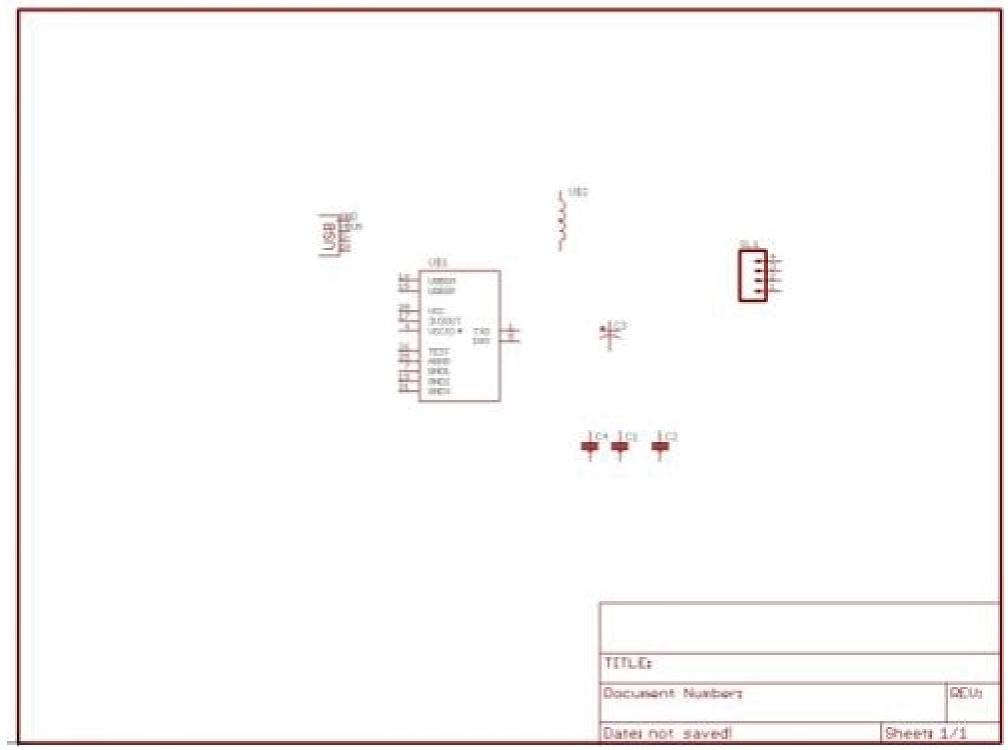
4 x STAND-OFF(Device name STAND-OFF): This part will add a hole and a keepout ring for a #4-40 screw. These can be used to raise your board up off a surface or to mount your board to an enclosure.

USB (Device name USBPTH) : USB Type B through-hole connector

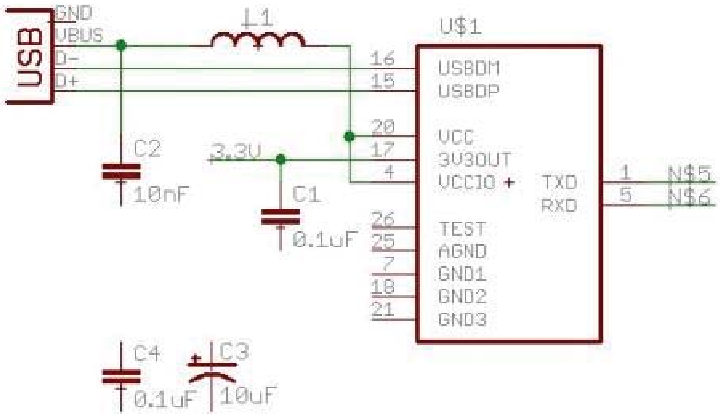
M04 (Device name M04PTH) : Four pin 0.1” connector

GND (Device name GND) : Ground connections

VCC (Device name VCC) : Power connections

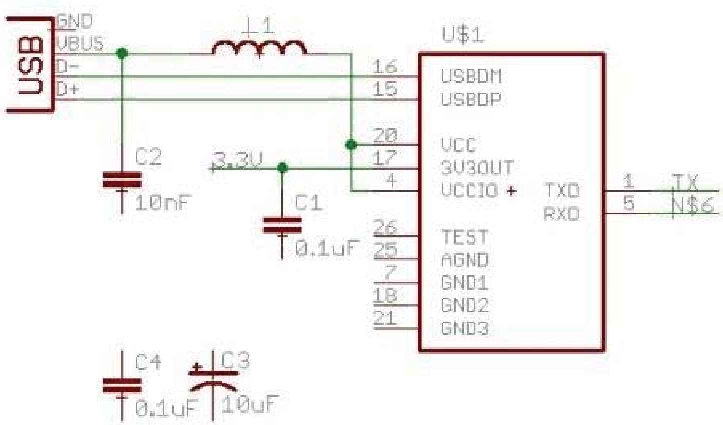


Now we just need to begin wiring nets. Arrange the pieces so that there is as little net overlaps as possible.

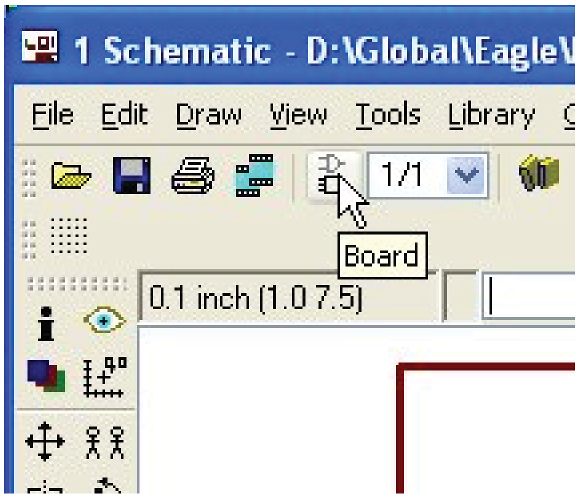


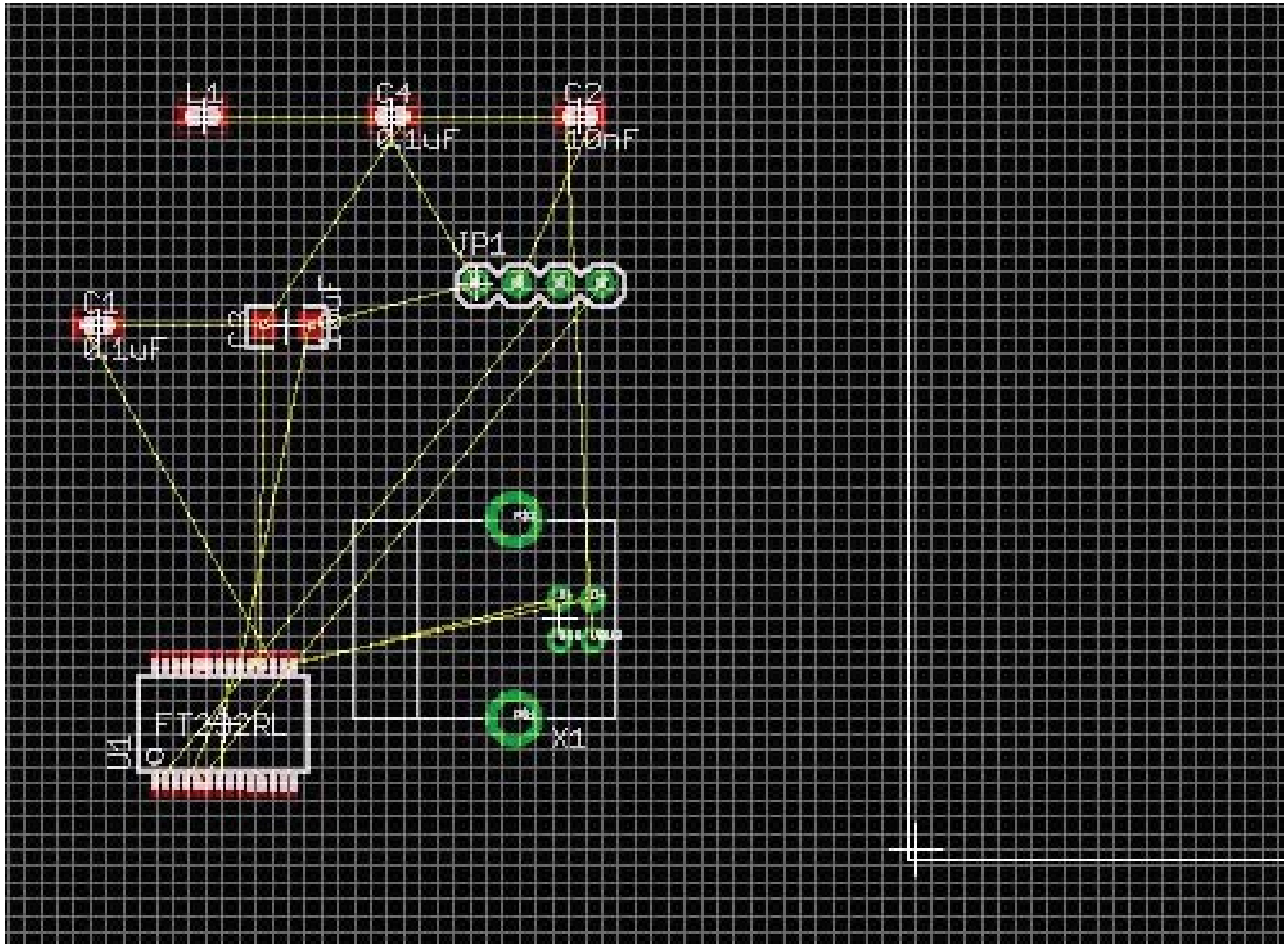
To wire a pin (TXD) to a far point (the 4-pin connector for example), instead of sending a wire half way across the page, we use net names. The green wire is not physically seen on the schematic, but Eagle knows to connect the two points on the layout because the two green wires have the same name.

Press F9 and click on pin 1 (TXD). Bring out the net a couple square widths and left click again to end the net. Press Alt+F9 to name the net. Click on the wire you just created. You should see a net name (like N$5) appear and be floating. Anchor it to the wire and TX pin:



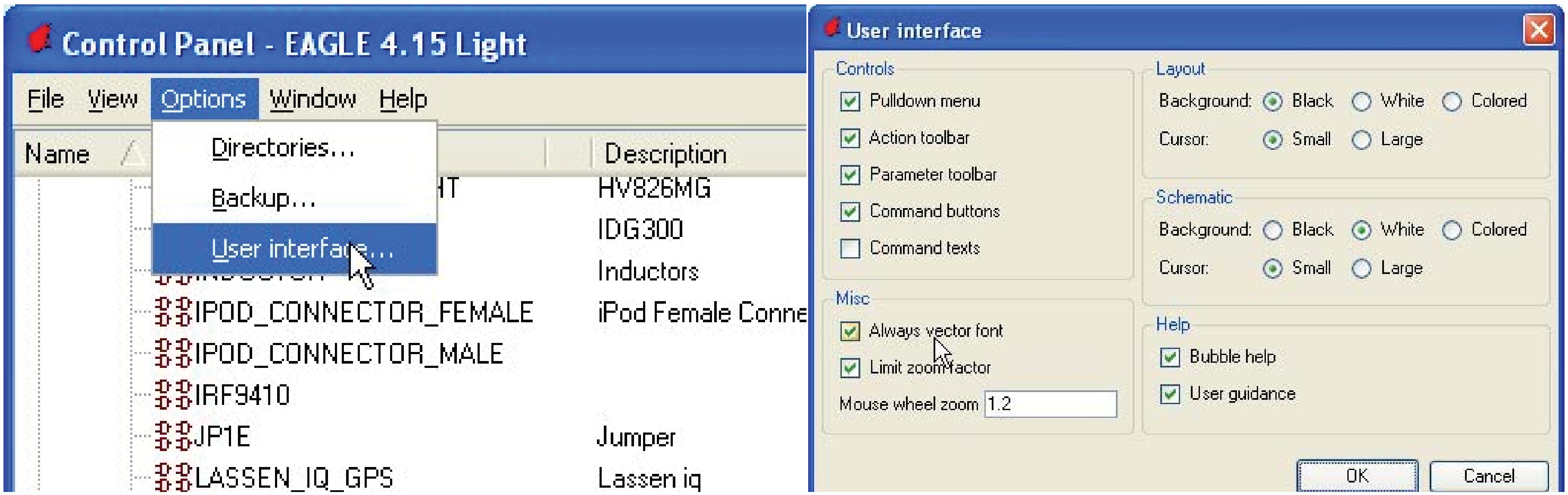
I probably could have wired JP1 directly to the various pins but I wanted to demonstrate the net/name properties. This will also make it easier to label the pins on the PCB. Speaking of which, if you have not already, click on the 'Board' button to open the PCB editor:

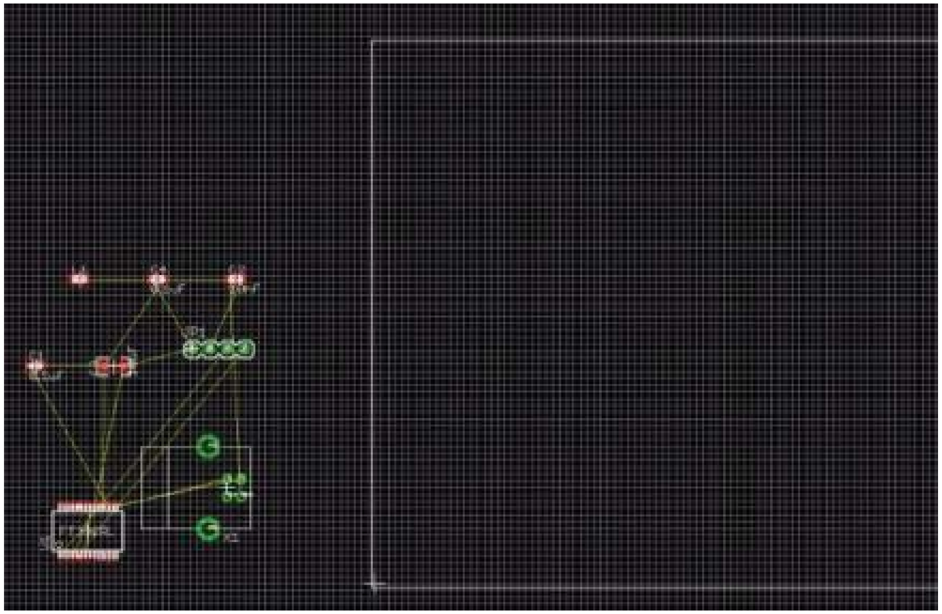




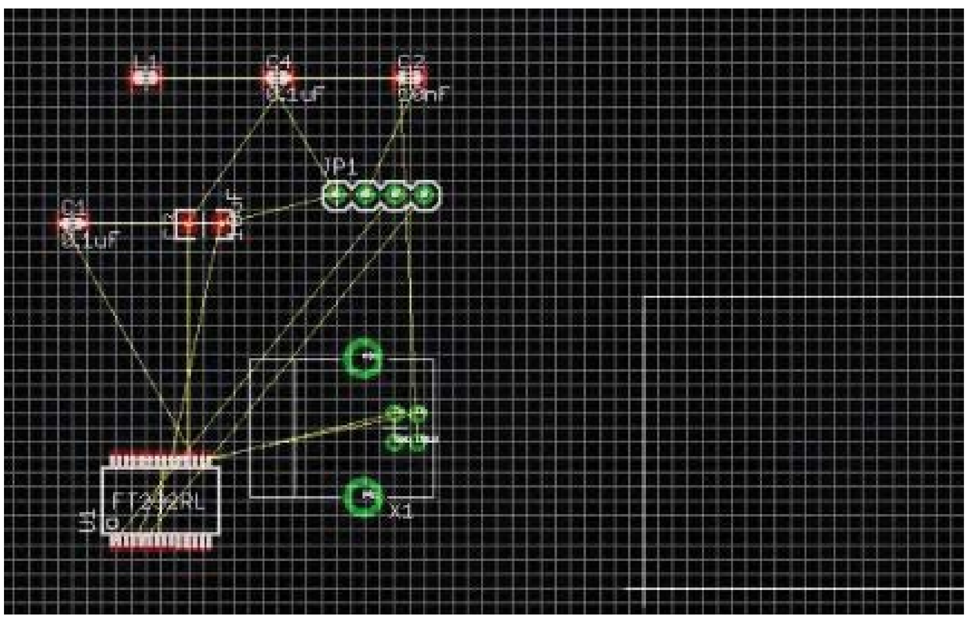
EagleCad_PCB Layout

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





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.

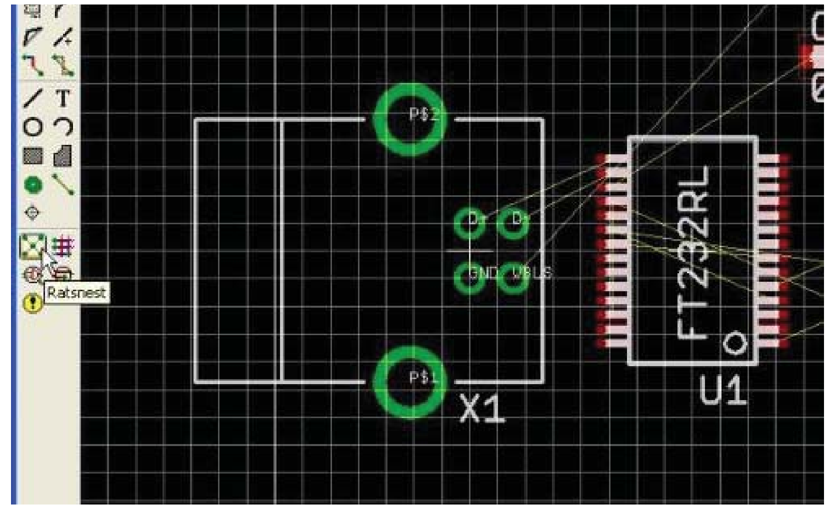
Remember:

Press F7 to move a component

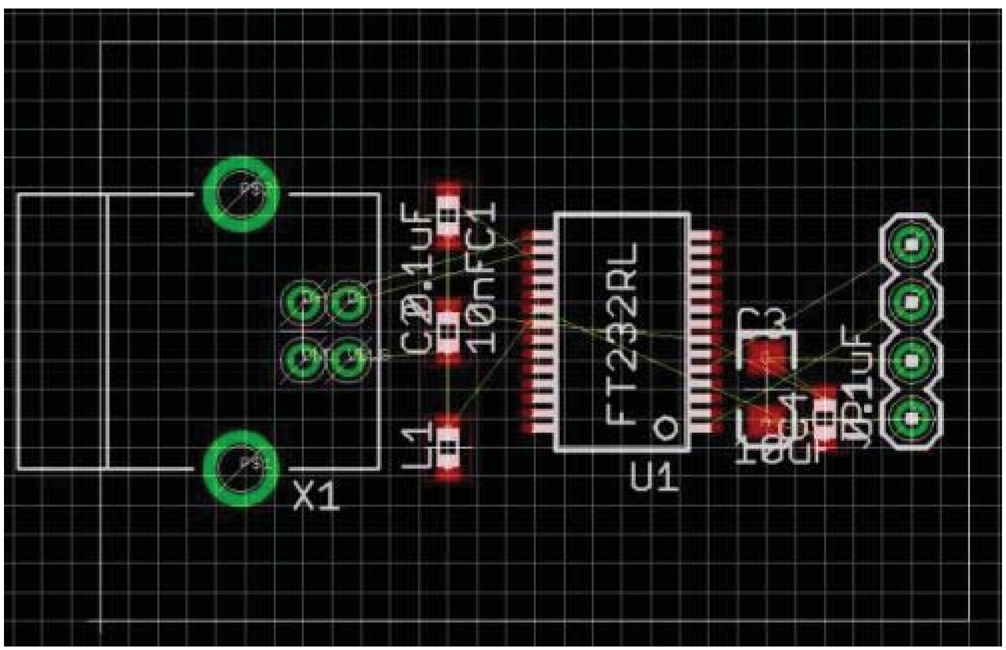
Right click to rotate

Hold control to grab a component at its origin

Scroll wheel to zoom in/out

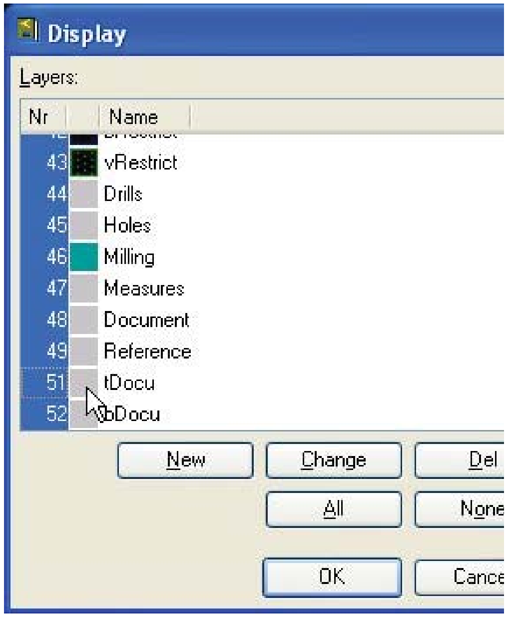


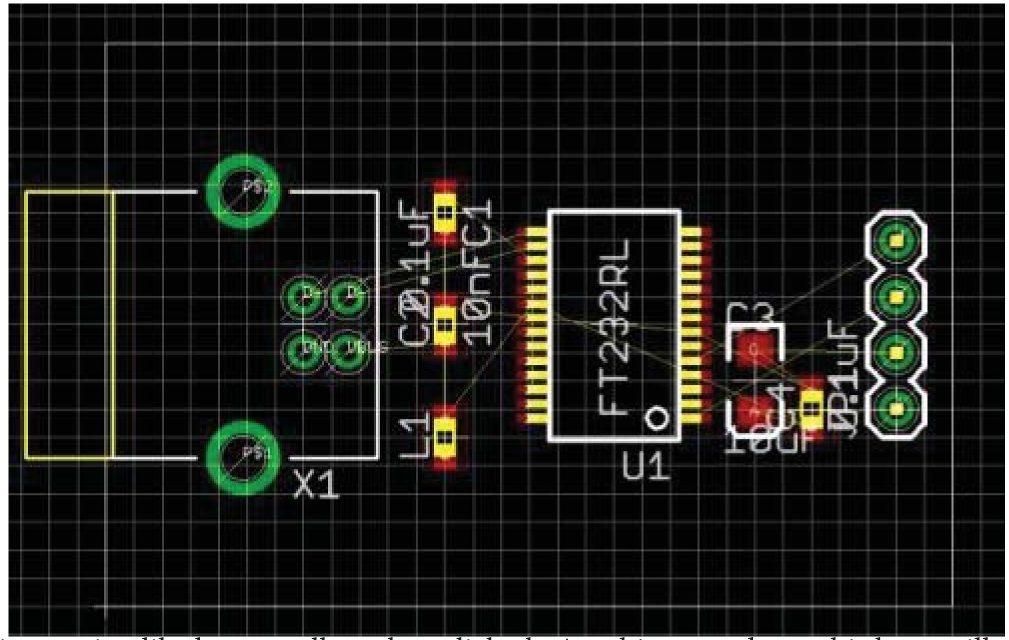
Hit the Ratsnest button from time to time to recalculate the air wires.



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':



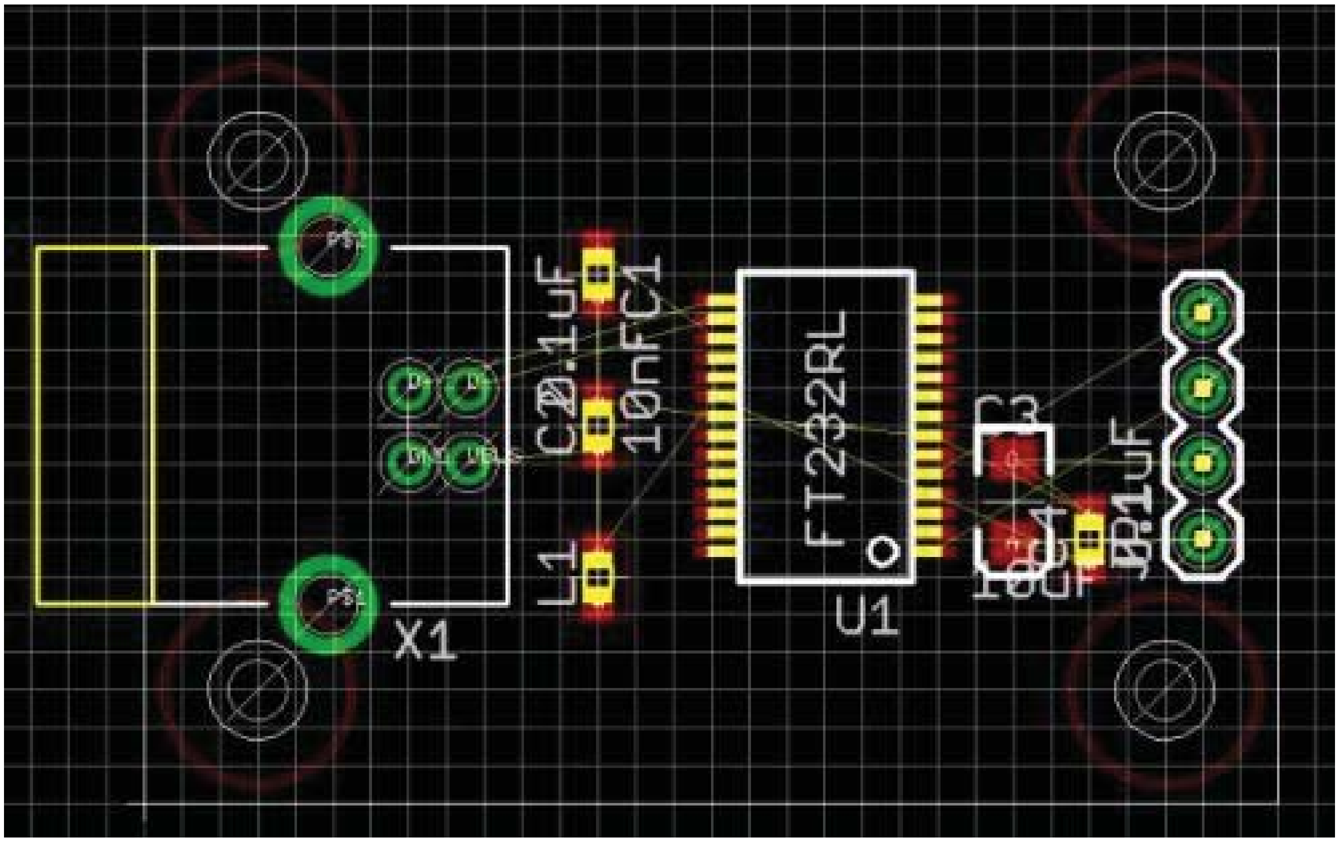


Then click on the gray 'Color' box and change it to something interesting like lemon yellow, then click ok. Anything you do on this layer will now be yellow. Do the same for layer 21 'tPlace'.

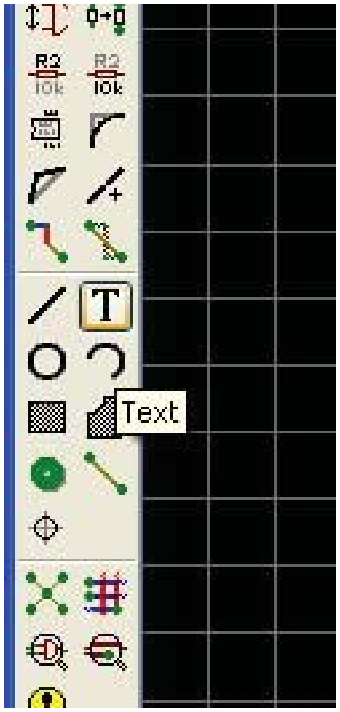
The yellow part of the USB connector is only there to indicate that the connector sticks over the edge that far. Only the white part of the USB connector footprint will actually show up on the silkscreen print on our PCB. Anything in light gray (tNames and tValues layers) will not print on the silkscreen 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.

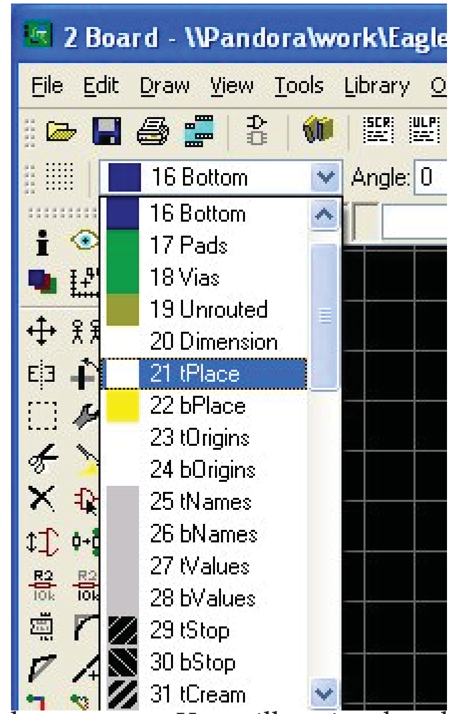
I like to use 4-40 screws and 0.25” diameter plastic standoffs on everything. These 4-40 screws need a 0.13” diameter hole and the standoffs have a 0.25” outside diameter that we will need to take into account. If you have not already done so, add four of the ‘Stand-Off’ components to the schematic (and therefore PCB). This component was created to couple the 0.13” drill with a keepout ring. This keepout ring helps show were 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.

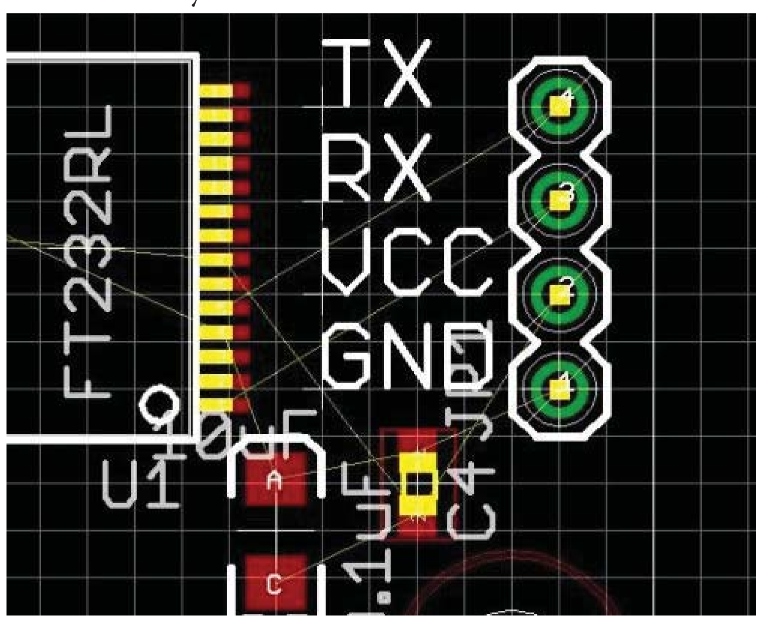


A ‘C2’ label is handy when you’re populating a board or when you’re troubleshooting a complex circuit, but on a day to day basis, you probably won’t need to know where C2 is. On the other hand, the TX and RX pins will probably be used every time you use the board! You really should label anything that will be connected to the outside world. To add a text label to a pin, click on the ‘Text’ button:

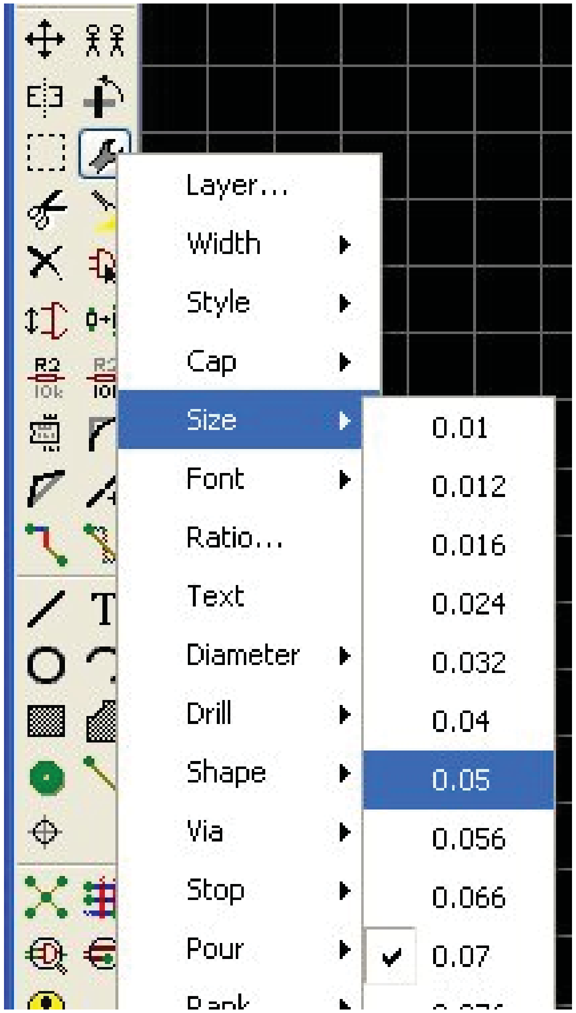




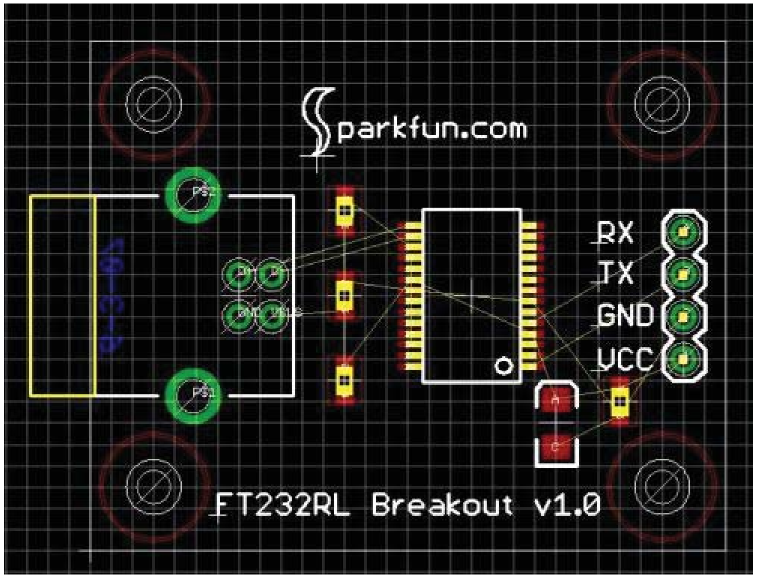
A window will pop up asking you what text you would like to add. Type ‘TX’ and press enter. You will notice that the text may be appearing on an odd layer. Be sure that you add text on the tPlace layer.



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

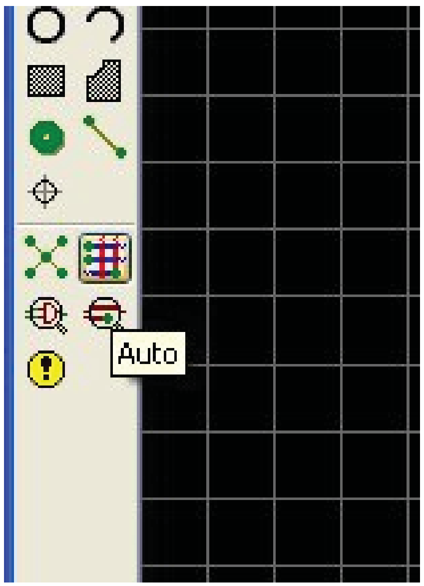


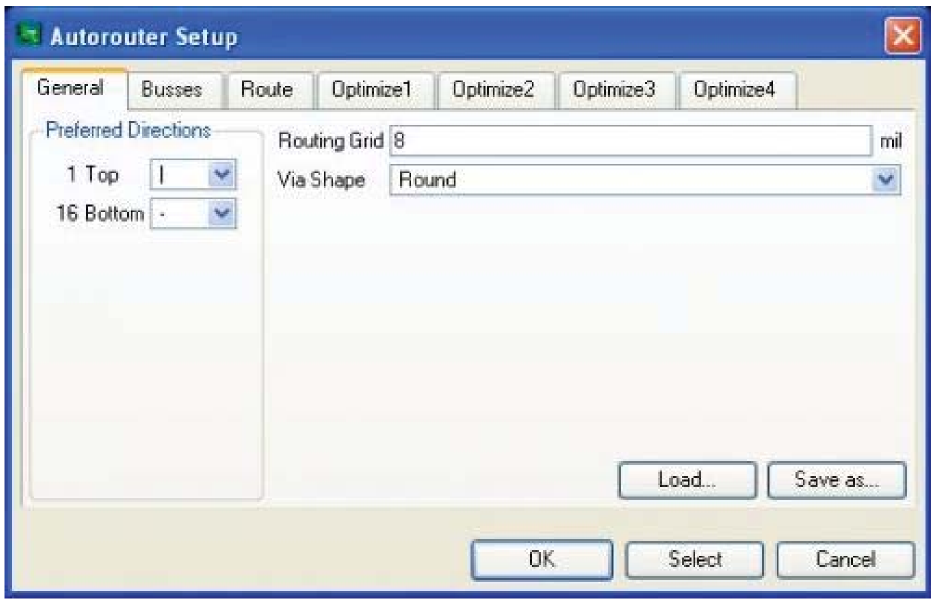
To change the font size, click on the Wrench ('Change' button), select 'Size', then 0.05. Now click on each text that you want to change the size on.



Above is the completed board ready for trace routing.

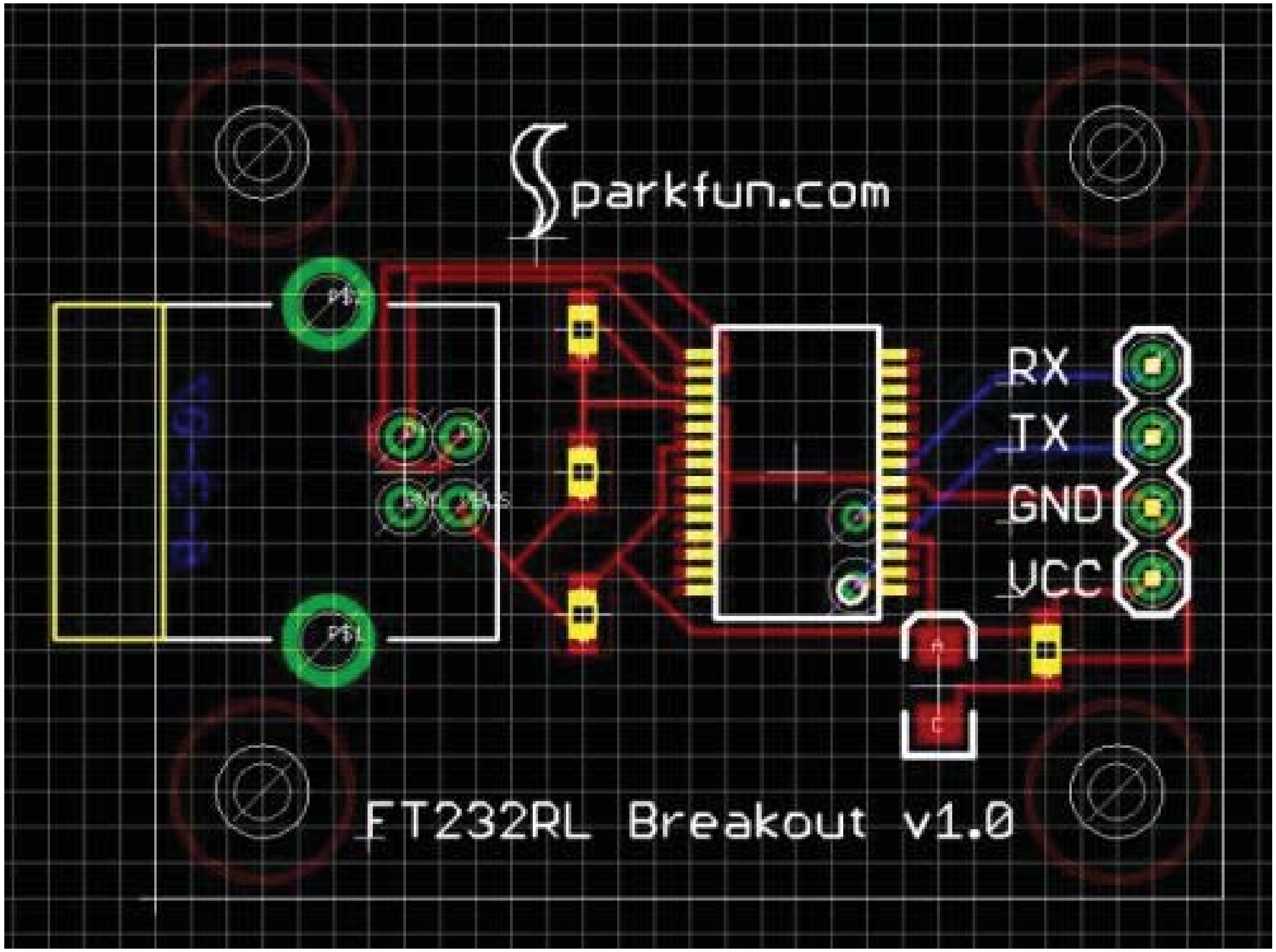
Many people swear up and down that an auto-router is a bad idea. It may be, but if we're not concerned about trace impedance or high speed signals, an auto-router is a great way to whip up protos. Spend your time innovating, not routing mundane traces.





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

Change the Routing Grid



Sparkfun.com
RX
TX
GND
VCC
FT232RL Breakout v1.0