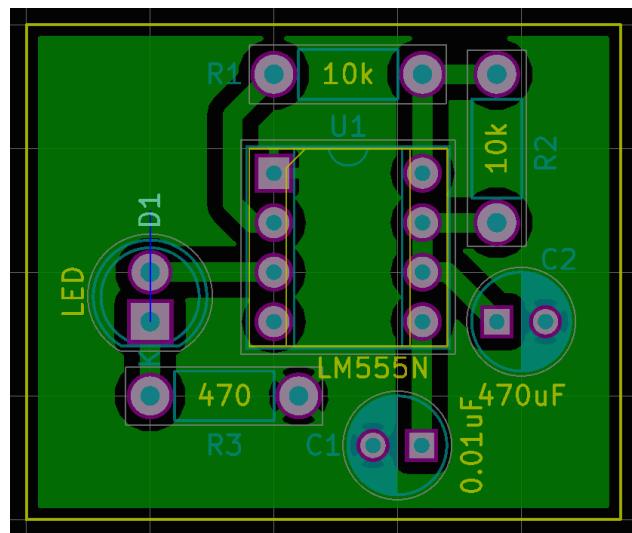


# Practical PCB Design

## A KiCAD Walkthrough

April 23, 2017



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Circuit . . . . .	2
1.2	Setup . . . . .	2
1.3	Overview . . . . .	2
<b>2</b>	<b>Create a Project</b>	<b>5</b>
<b>3</b>	<b>Schematic</b>	<b>6</b>
3.1	Page Settings . . . . .	7
3.2	Moving Around . . . . .	7
3.2.1	Important note for laptop users . . . . .	8
3.3	Add Components . . . . .	8
3.4	Power Ports . . . . .	12
3.5	Rearranging Components . . . . .	13
3.6	Adding Connections . . . . .	14
3.7	Defining Values . . . . .	16
3.8	Annotate Schematic . . . . .	17
3.9	Assign Footprints . . . . .	19
3.10	Generate Netlist . . . . .	20
<b>4</b>	<b>PCB Layout</b>	<b>21</b>
4.1	Design Rules . . . . .	21
4.2	Component Placement . . . . .	23
4.3	Board Outline . . . . .	25
4.4	Routing . . . . .	26
4.5	Vias . . . . .	27
4.6	Pours . . . . .	29
4.7	Export . . . . .	31

# 1 Introduction

The goal of this workshop is to give a quick guide to KiCad, from installation to schematic creation and PCB layout, via the design of a PCB for a basic 555 LED blinder circuit. **KiCad** is a powerful open source PCB design suite and a great tool for documenting circuits and producing high quality PCB's. This guide was written using version 4.0.6 of KiCad.

## 1.1 Circuit

The circuit we will be laying out in this workshop is the classic 555 timer LED blinder circuit. This circuit is taken directly from the Typical Application section of the **LM555 datasheet**, and simply blinks a LED on and off every few seconds.

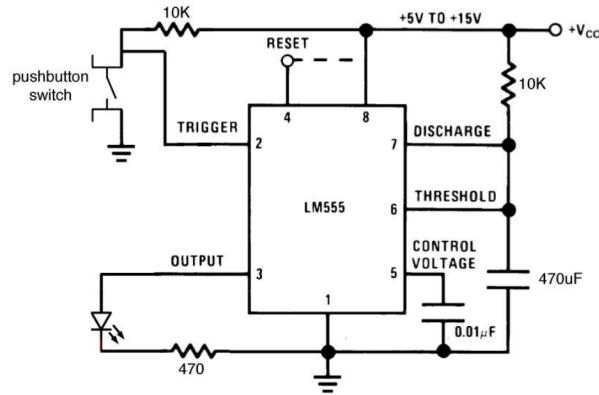


Figure 1: LM555 Example Circuit

## 1.2 Setup

KiCad is available for a wide range of operating systems. The download page is [here](#). For Mac, make sure to get the “extras” package as well. It contains a useful set of default components and footprints.

## 1.3 Overview

KiCad differs from other electronics design software in that it is extremely modular. Each section of the design process is divided into a distinct program. The KiCad getting started guide has this fantastic flowchart which is a great guide for figuring out what button to push next at any step of the design process.

KiCad has a bad habit of not labeling any buttons. However, if you hover over any button for a few seconds, a tooltip will come up explaining what the button does. Also, in any of the editors typing ‘?’ will bring up a list of keyboard commands.

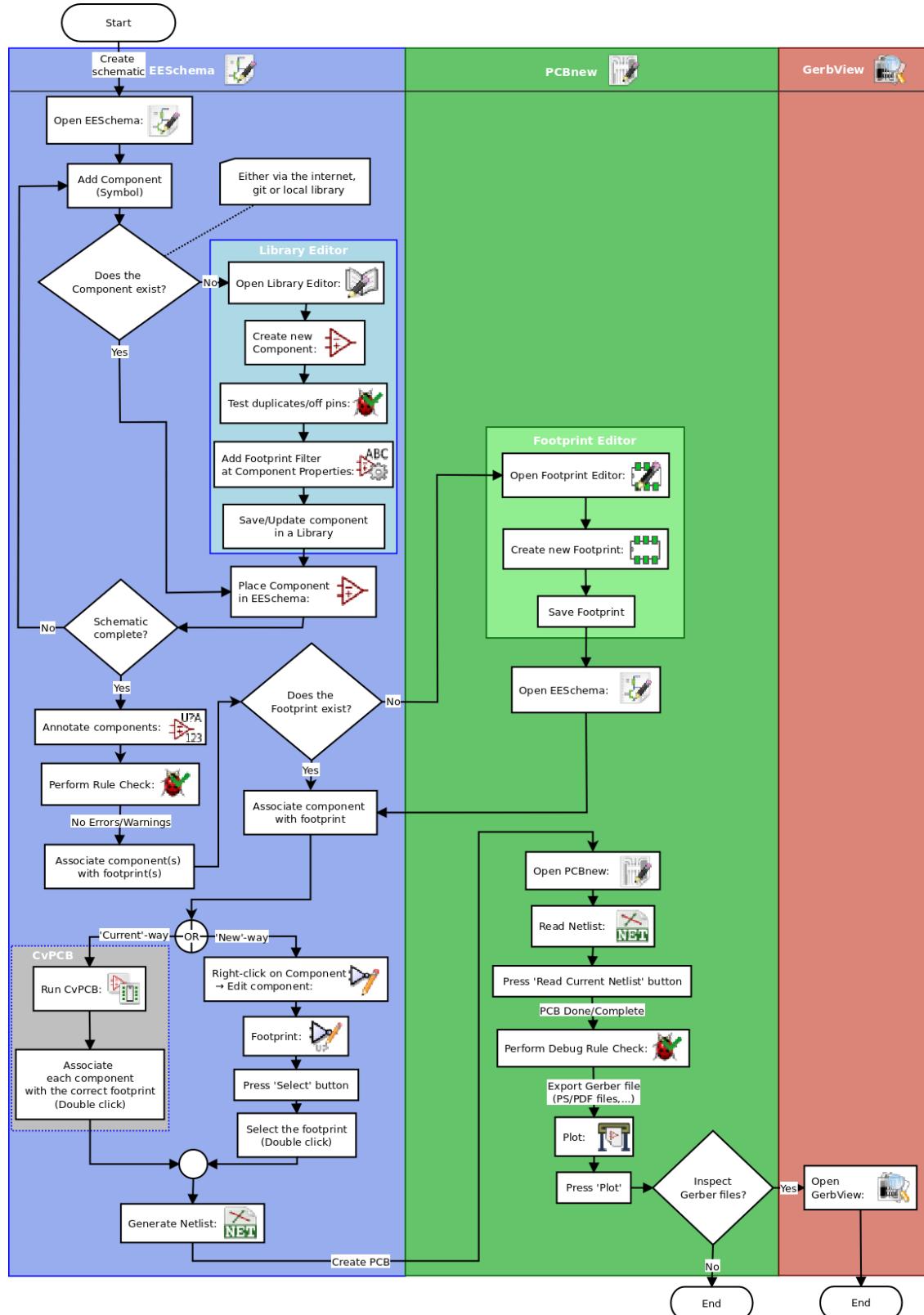


Figure 2: KiCad PCB workflow flowchart

## 2 Create a Project

The first step is to create a project, which will serve as a general container for KiCad to keep track of the mess of files that will make up your design. Open up the KiCad application. Select **File → New project**, and pick a nice name and location to save the project. For this tutorial, the project name is “LEDBlinker”. As the dialog will remind you, it is a very good idea to create a project in its own folder.

Once this step is done, you should see a **.pro**, **.kicad\_pcb**, and **.sch** file show up in your project directory and the KiCad application’s sidebar.

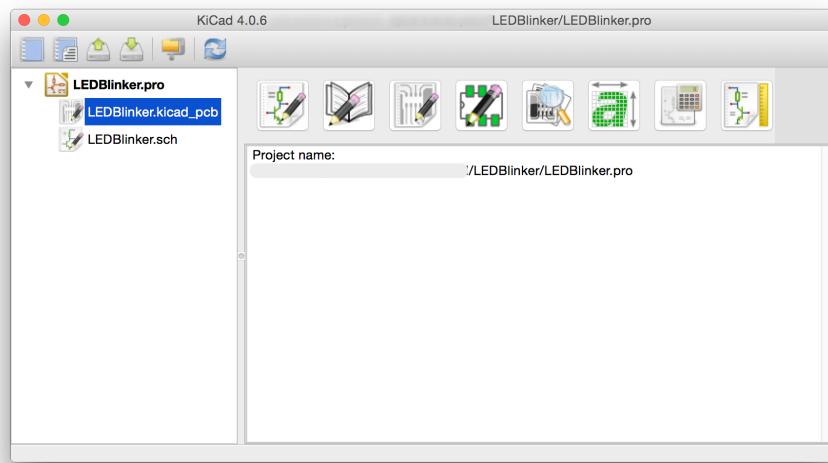


Figure 3: After project creation

From this screen, you can double click on the files to open them for editing. The large square buttons allow you to directly launch the various utilities within KiCad.

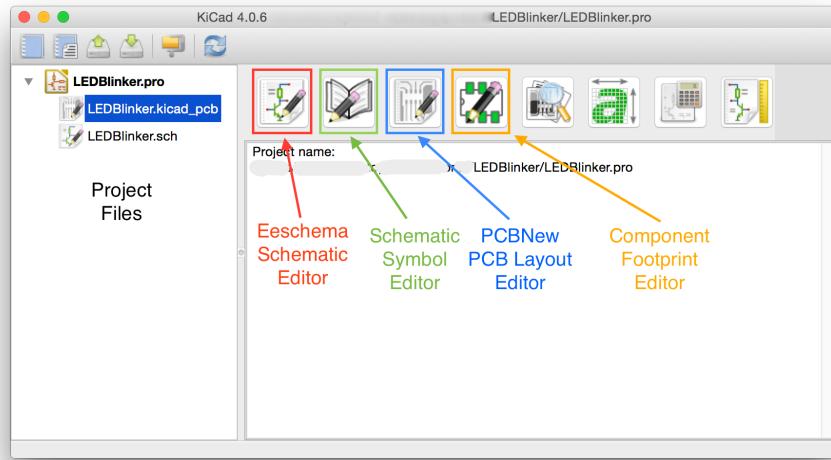


Figure 4: KiCad Main Window

### 3 Schematic

The schematic defines the components that make up the circuit and the connections between them at a high level. Double click on the schematic file in the side bar, or the schematic editor button to launch Eeschema, the schematic editor application. Once the file loads, you should be looking at a blank schematic sheet.

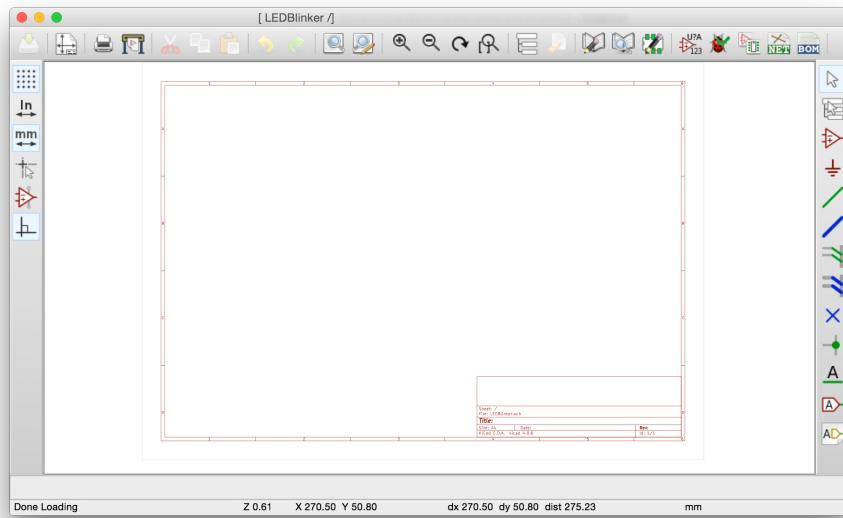


Figure 5: Blank Schematic

### 3.1 Page Settings

Just as in high school, you should always put your name on your work. So, the first thing to do is to adjust your page settings under **File → Page Settings**. Here are some options you should set for every design you make:

- Page size and orientation (left sidebar)
- Date
- Revision number
- Title
- Company (Good place to put your name if this is a personal project)
- Comment 1 (Put your name here if the Company field is filled)

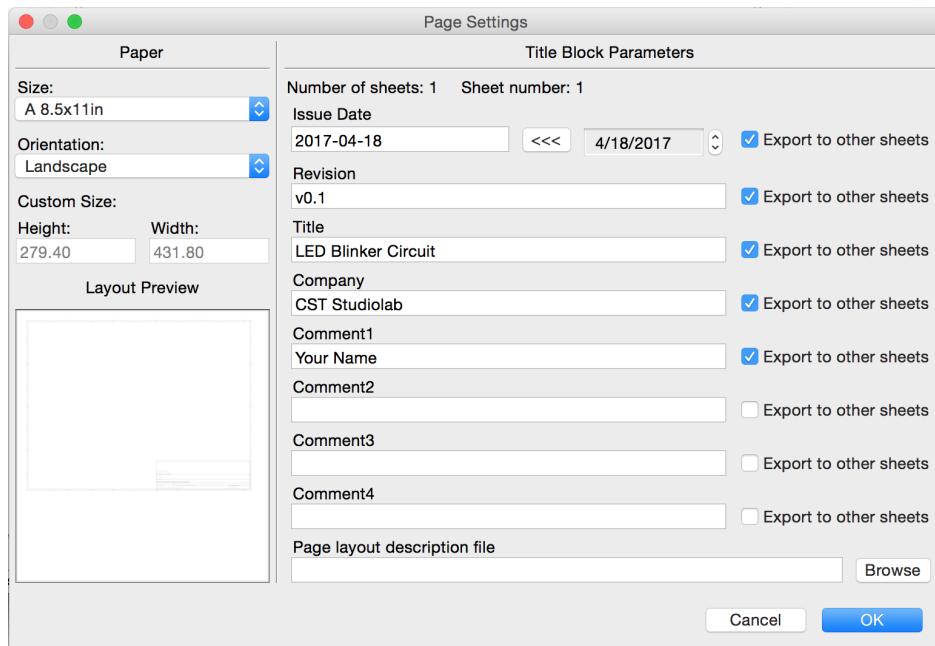


Figure 6: Schematic Page Settings

Checking the “**Export to Other Sheets**” option will copy these settings to other schematic documents in your project. Click OK to save your settings, and you should see the title block of the schematic update with the information you entered.

### 3.2 Moving Around

Now that the page is set up, we can start the business of drawing our schematic. To move around, use the scroll wheel on your mouse to zoom in. Holding the 3rd mouse button and dragging pans the canvas. To practice, zoom in on the title block and make sure all the options you set in the last section appeared correctly.

### 3.2.1 Important note for laptop users

If you are using the trackpad on a Mac, open up the main preferences (**KiCad → Preferences**) and check the option to “**Use Touchpad to Pan**”. This will enable touchpad support so that you can scroll and zoom with the trackpad just like any other application. If you’re using an external mouse, leave this option unchecked.

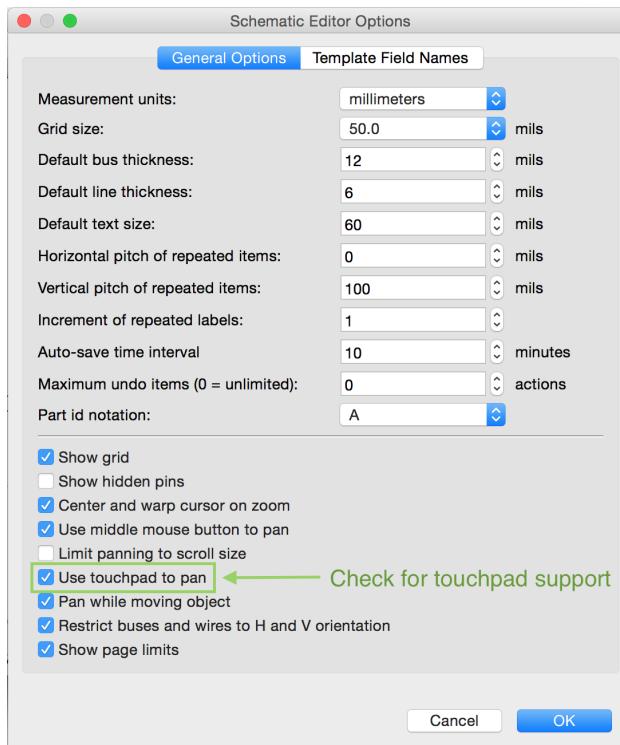


Figure 7: Touchpad Pan Option

### 3.3 Add Components

Let’s add in the components of this circuit. Switch to the add component tool by clicking on the Add Component button on the right toolbar or pressing the ‘A’ key. Click somewhere on the canvas to place a component.



Figure 8: Add Component Tool Button

A dialog box will pop up asking you which component to place.

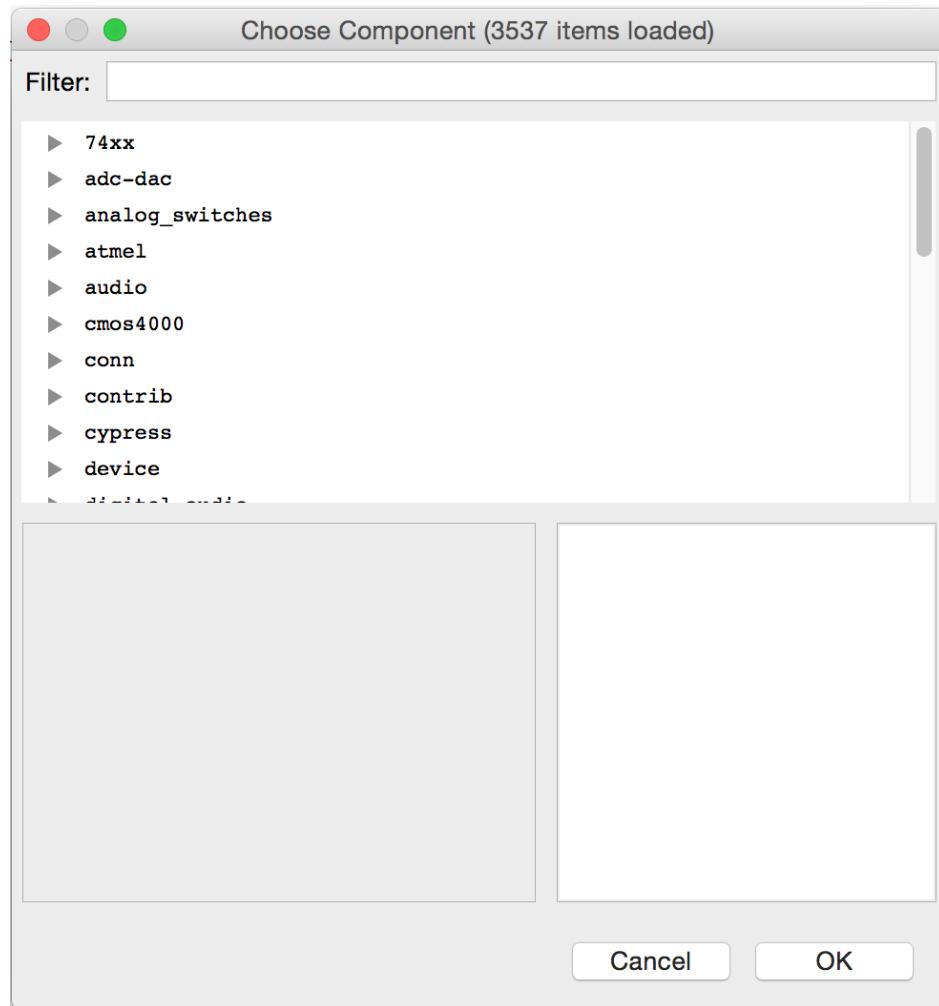


Figure 9: Add Component Dialog

You can search for specific components using the top dialog box. Let's find the LM555 timer chip at the center of the circuit. Type '555' into the search box and select the LM555 component.

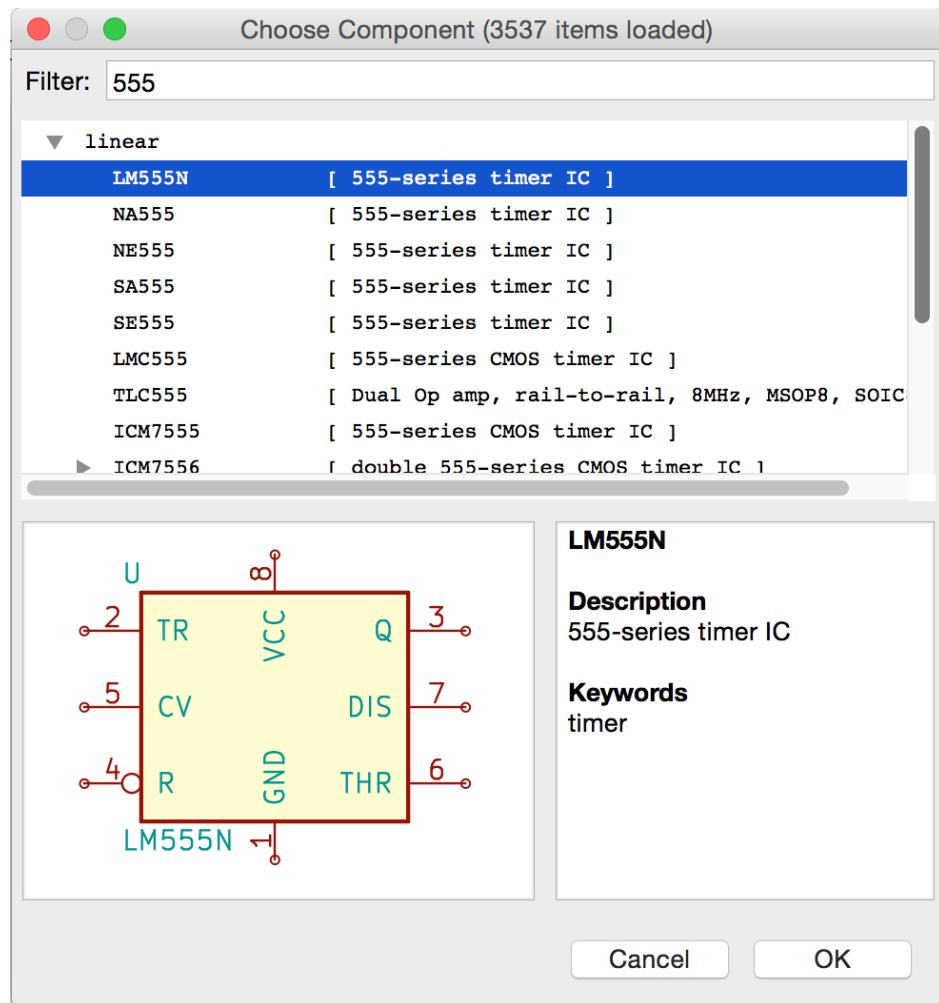


Figure 10: Add LM555

Click OK on the dialog to select the LM555. You should see a ghost of the component following your cursor. Click to place the LM555 in the center of your schematic.

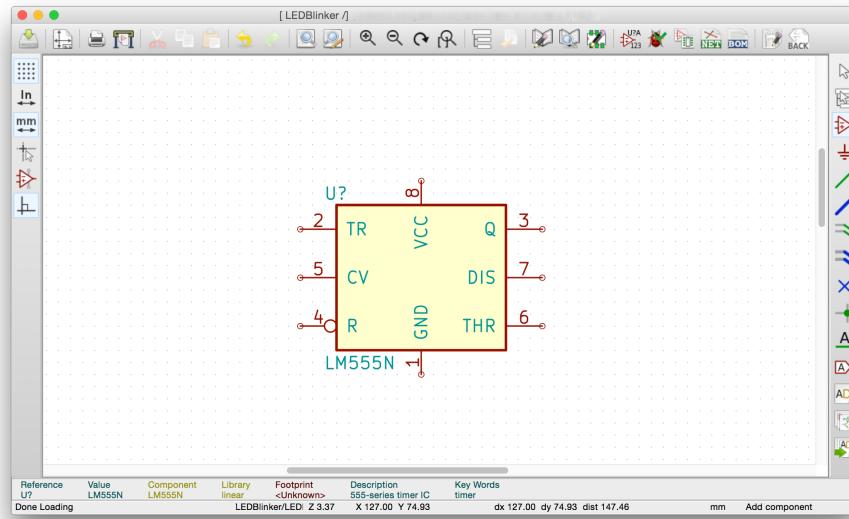


Figure 11: Placed LM555

If you click again, the select component dialog will reappear for selecting the next component to place. Try searching for and placing the rest of the components in the circuit. You'll need 3 resistors, 2 capacitors, and a LED. The Don't worry about exactly matching the same layout as in the example circuit diagram, we'll be reorganizing the components later.

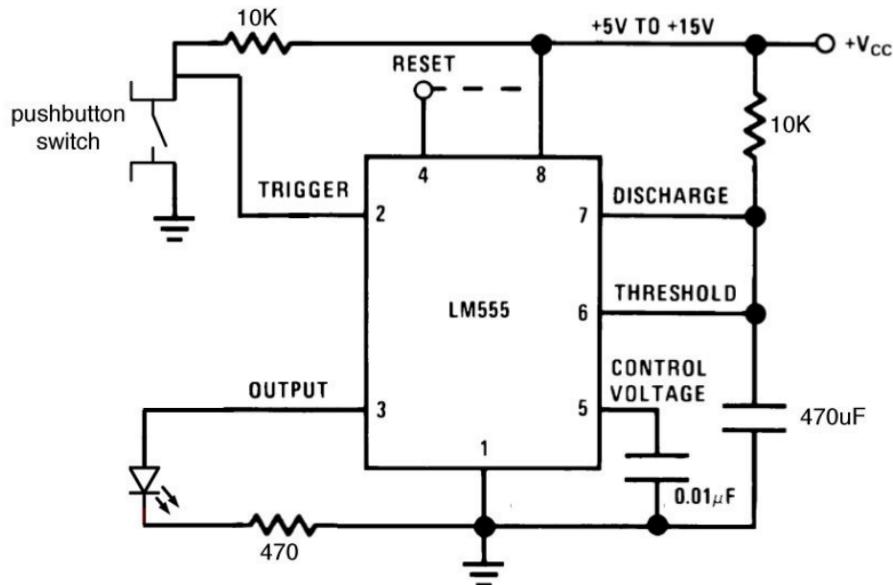


Figure 12: LM555 Example Circuit

Once you've finished adding the components, your canvas should look something like this:

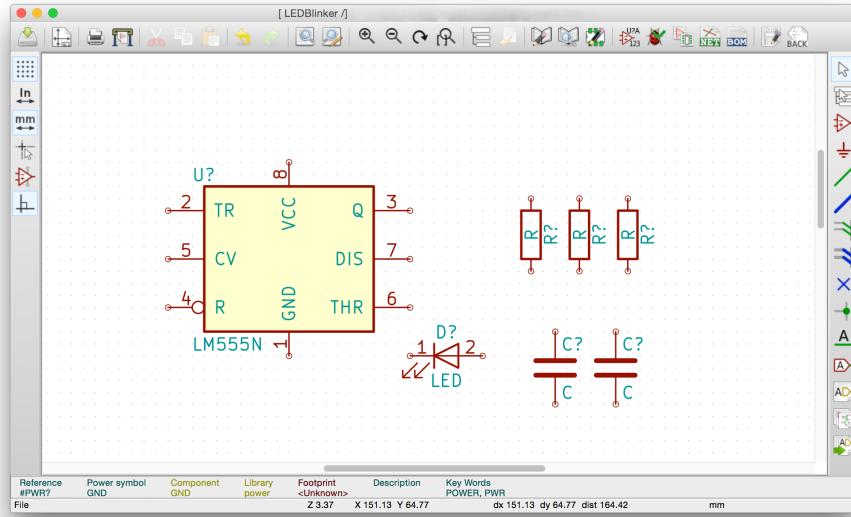


Figure 13: All Components Added

Remember to save your work, hit **Ctrl-S** or **Cmd-S** on Mac.

### 3.4 Power Ports

Let us also place labels for the power and ground connections for this circuit. Select the power ports tool by clicking on the button that looks like a miniature ground symbol.



Figure 14: Add Power Port Dialog

This tool acts in a similar fashion to the add component tool. Click to open the dialog to select symbols, search for a symbol, then hit OK and click to place your selected symbol. Add a VCC symbol and a GND symbol to your schematic.

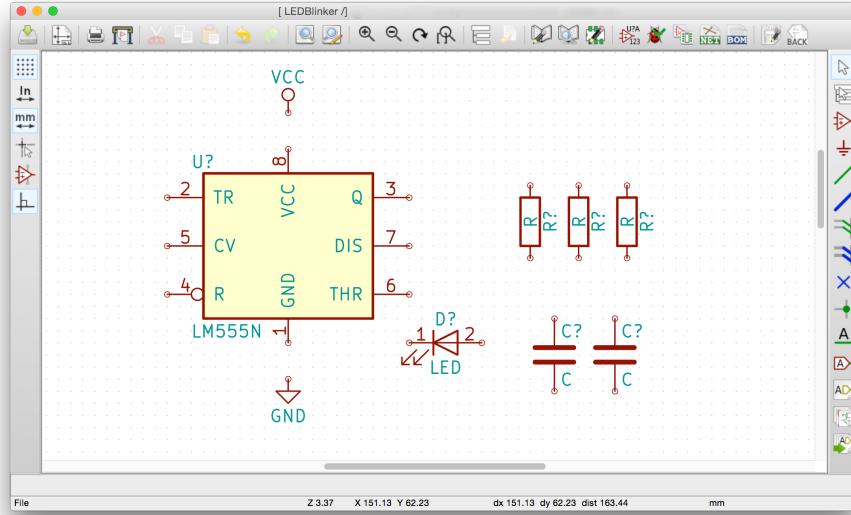


Figure 15: Power Symbols Added

### 3.5 Rearranging Components

KiCad tends to very heavily emphasize using keyboard shortcuts. In general, you will work with one hand on the keyboard and one hand on the mouse. What is different from most other programs is that when you press a shortcut to perform an action, that action will act on whatever is underneath your cursor at that moment. Explicitly selecting objects is optional.

To demonstrate this, let's try moving the LED over to the output pin (pin 3 on the LM555). The default shortcut for moving is '**M**', so hover over the LED with your cursor and press **M**. The LED will start following your cursor around the canvas. Click to place the LED down again.

Let's rotate the LED. The hotkey for rotation is '**R**', so hover over the LED and press **R** until it is facing down.

If there are multiple objects under the cursor, a small dialog will appear to clarify which object you actually want to work with. This often happens if there are labels or pieces of text overlapping a component. Simply select the correct object in the dialog to continue.

If you want to manipulate multiple objects at once, click and drag to create a selection box. When you release the mouse, anything touching the selection box will be moved. Right click to show other operations, which will be applied to the entire group.

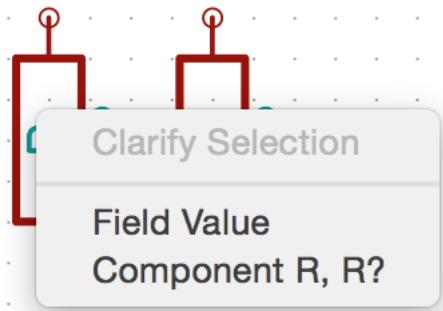


Figure 16: Clarify Selection Dialog

Rearrange the components around the LM555 so that they are roughly near the pins they need to be connected to. For reference, here is an example layout:

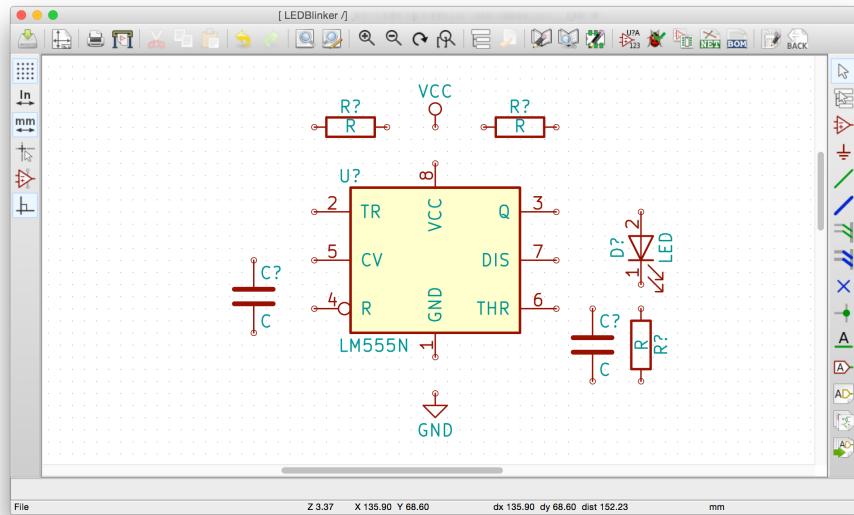


Figure 17: Components Rearranged

### 3.6 Adding Connections

Now that our components in place, we need to add the connections between them. Press the button with a green line on it to select the Place Wire tool. Alternatively, the default hotkey is ‘W’.



Figure 18: Place Wire Tool

Click on the pins (the small circles) of the components to start wires, then drag over to another pin and click to make a connection. Start by connecting the ground pin (pin 1 of the LM555) to the ground power symbol and the VCC pin (pin 8 of the LM555) to the VCC power symbol.

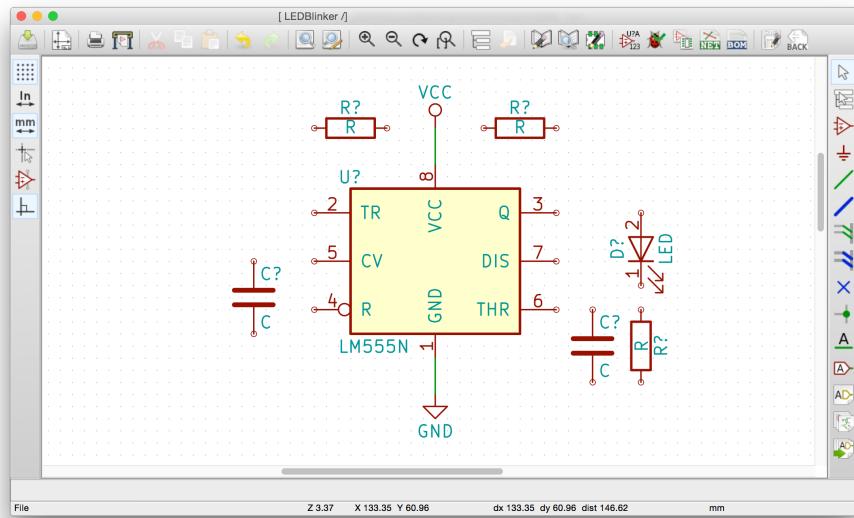


Figure 19: Add Power Connections

Follow the example schematic to make the rest of the needed connections. If you end a wire by clicking on another wire, KiCad will automatically make a junction. Wires that cross without a junction do not connect. See the example below for reference.

If you need to move a component that already has wires connected to it, the Grab tool (default hotkey ‘G’) may be useful. This acts similarly to Move, but tries to keep existing connections. The grab tool also works on wire corners and junctions, which is useful for tidying up connections.

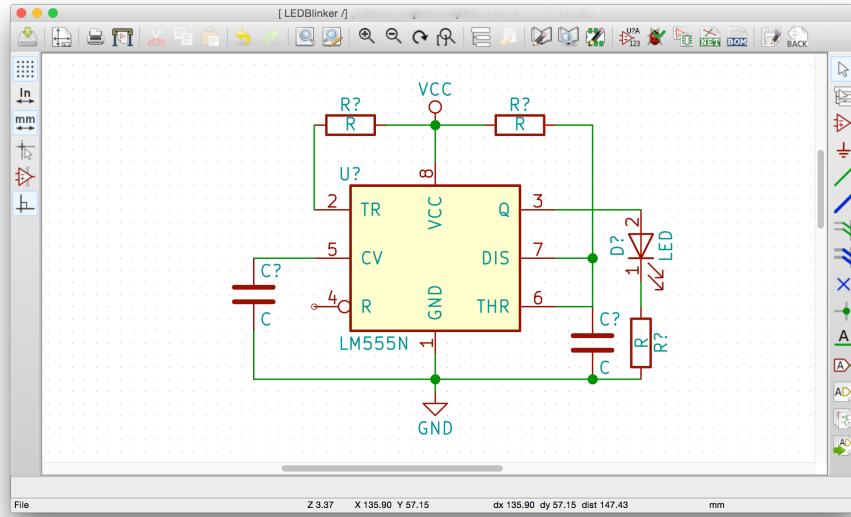


Figure 20: Circuit Connections Example

### 3.7 Defining Values

The components have been connected, but we still need to define the values of the passive components. Right click on one of the resistors or components and in the popup menu that appears, select **Edit Component → Value**. A dialog will appear where you can enter the appropriate value.

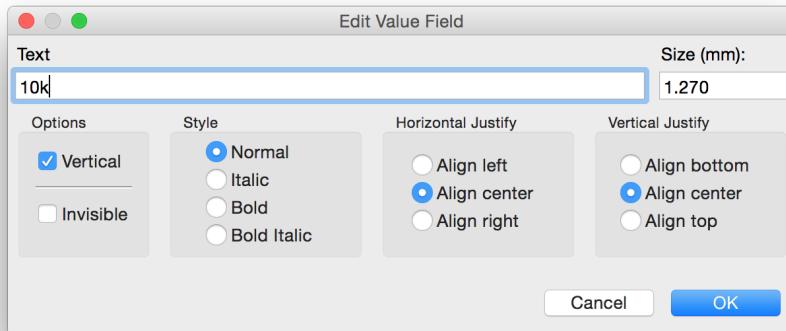


Figure 21: Edit Value Dialog

When you click OK, the label on the schematic should update with the correct value. Once you've finished labeling all the resistors and capacitors, the schematic should look something like this:

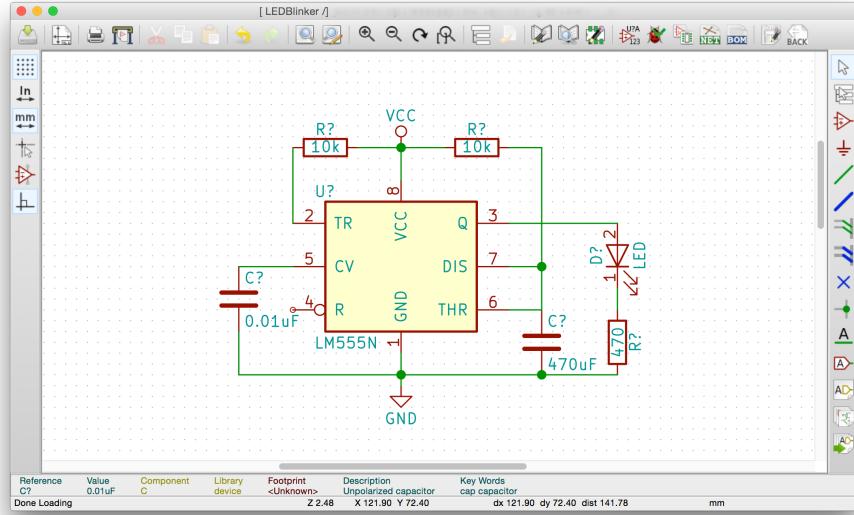


Figure 22: Schematic with Values Added

### 3.8 Annotate Schematic

Now all the information from the datasheet has been entered, and we just have a few final steps before we can start the physical layout. First, we need to give unique ID numbers to all the components. Notice that up to now, every component has had a placeholder identifier like ‘R?’ or ‘C?’.

On the top menu bar, locate the button labeled with an op-amp and the numbers 123.



Figure 23: Annotate Schematic Button

When you press it, a dialog will appear showing options on how to assign the identifiers. Choose the **‘Use the entire schematic’** and **‘Keep existing annotation’** options, but leave the rest at their default settings.

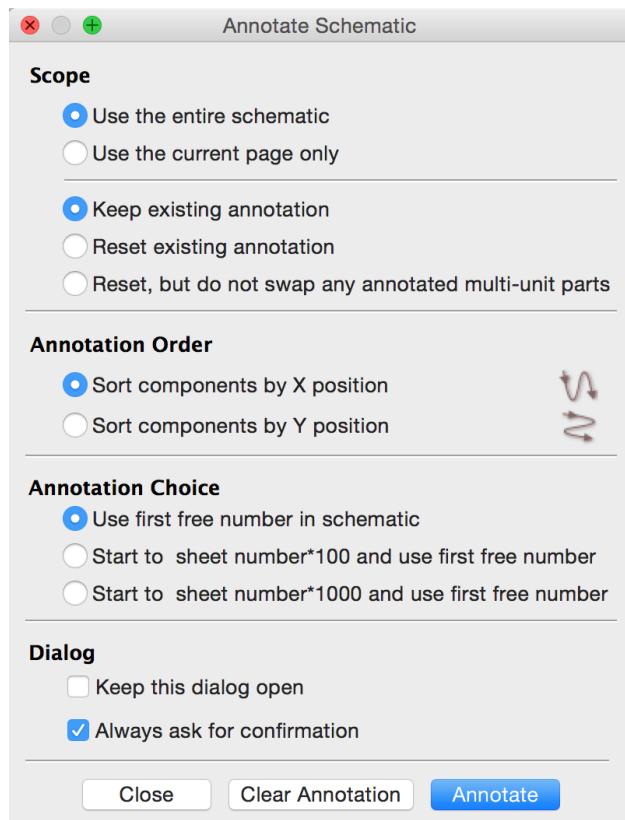


Figure 24: Annotate Schematic Dialog

Click the **Annotate** button to run the numbering utility. Now you should see that every component has a unique number assigned to it.

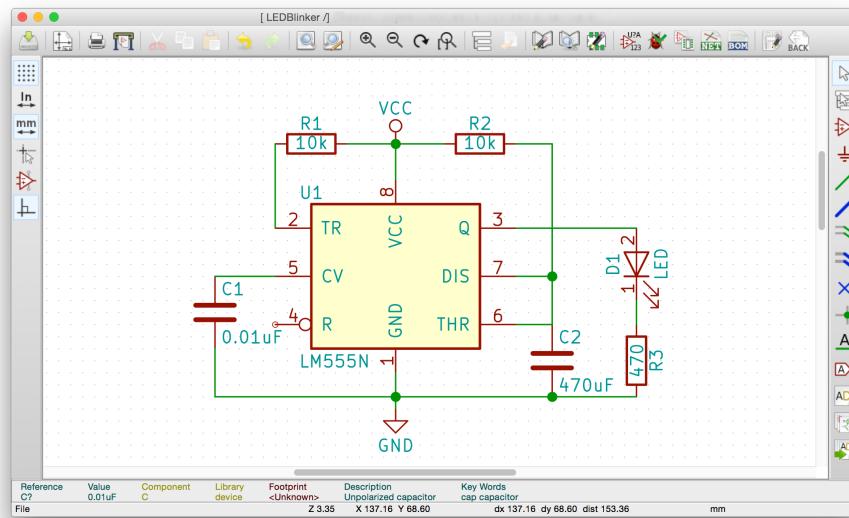


Figure 25: Annotated Schematic

## 3.9 Assign Footprints

Now we need to tell the PCB editor what footprints to use for each component. Since most components come in many different variants with different shapes and sizes, KiCad doesn't explicitly assign a footprint (physical hole/pad pattern) to each schematic symbol. Instead, you can choose a different footprint depending on what component style you have on hand or decide to order.

Click the button two over from the annotate schematic button, with the op amp and IC shapes on it. This is the CVPcb utility for matching components to footprints.



Figure 26: Footprint Matching Tool Button

A dialog should open up with three columns. On the left is a list of known component libraries. As you can see, footprints are organized by type of component. In the middle is a list of all the components in your schematic. On the right is a list of all the footprints within the currently selected library.

The rightmost three buttons in the menu bar represent filtering options. Usually, I check the middle one, which filters by number of pins, and the right one, which filters by the selected library. When you have a footprint selected, the icon with a magnifying glass opens a small window that shows you what that footprint actually looks like.

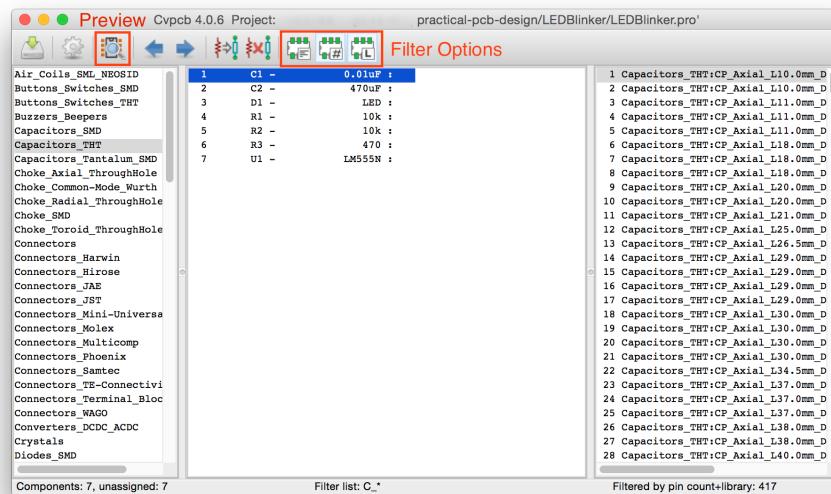


Figure 27: Selecting a Footprint

Some of the libraries or footprints are labeled with **THT**, representing Through Hole, or **SMD**, representing Surface Mount Device. These are the two most general categories

of component styles. Through hole components have wire leads as connections, which are threaded through holes in the PCB and soldered. Surface mount devices are soldered directly onto flat pads on one side of the PCB and tend to be much smaller than through hole components.

Click on the first component in the list, likely a capacitor. Then select the `Capacitors_THT` library on the left, and double click one of the footprints on the right to select it.

Assign footprints for each of your components. In practice, which footprint you select will depend on the dimensions and style of the actual component you will be using, but for this workshop I would recommend these footprints:

Component	Library	Footprint Name
Capacitor	Capacitors_THT	Capacitors_THT:C_Radial_D5_L6_P2.5
Resistor	Resistors_THT	Resistors_THT:Resistor_Horizontal_RM7mm
LED	LEDs	LEDs:LED-5MM
LM555	Housings_DIP	Housings_DIP:DIP-8_W7.62mm_Socket

When all the footprints have been assigned, save and close out of the dialog.

### 3.10 Generate Netlist

Now we have specified all the information we will need in the PCB editor. Click on the button labeled NET to generate the netlist.



Figure 28: Generate Netlist Button

This netlist file contains the schematic specifications in a compact format that the PCB editor can understand. Make sure PCBNew is the format selected, and hit the Generate button. Save the netlist in the same directory as the rest of the project.

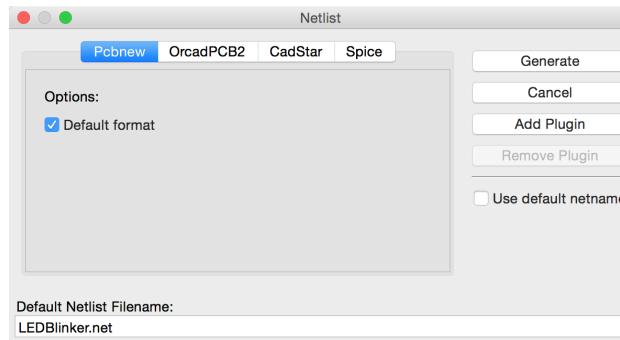


Figure 29: Generate Netlist Dialog

We are done with our schematic! Save and close out of the schematic editor.

## 4 PCB Layout

Let's get this PCB started! Double click on the `kicad_pcb` file or press the PCBNew button. You should see an empty canvas with a standard frame. The page settings do not copy over from the schematic, so open up **File → Page Settings** and update everything as needed.

If you are on a laptop, remember to enable the “Use Touchpad to Pan” option, just as in the schematic editor. Also, if you are on a Macbook or other computer with a high resolution screen, you may find that the default editor engine runs rather slowly. If this is the case, select **View → Switch Canvas to OpenGL**. The OpenGL renderer runs much faster, but it does have a different, more Altium-like color scheme.

Also note that on the left toolbar there are buttons for switching between inches and millimeters. Depending on what components you are using, it may be helpful to switch between the two while designing. For example, pin headers usually have 0.1” spacing, but some screw terminals have spacing measured in millimeters. For this workshop, select inches.

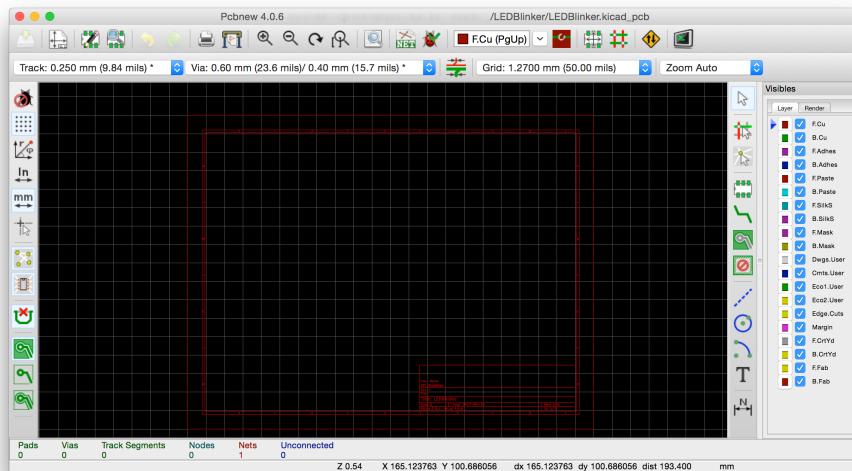


Figure 30: Blank PCB Window

### 4.1 Design Rules

Before we get started with layout, we need to define a set of design rules for our board. These rules are usually provided by the PCB manufacturer and provide constraints to make sure that all the geometry you draw on your PCB can actually be produced. If the smallest bit the manufacturer's PCB mill has is 1/16”, you wouldn't want to be drawing 1/32” holes. Many manufacturers will provide specific instructions, for instance, see **OSHPark**.

For today's workshop, we will be working with a set of design rules suitable for an **Othermill PCB mill** like the one in the MAE shop. Open up **Design Rules → Design Rules** and select the **Global Design Rules** tab. Make sure that your units are inches! Quick note: the commonly used unit of “mils” refers to one-thousandth of an inch (ie 20 mils is 0.020”). Fill out the design rules as show below:

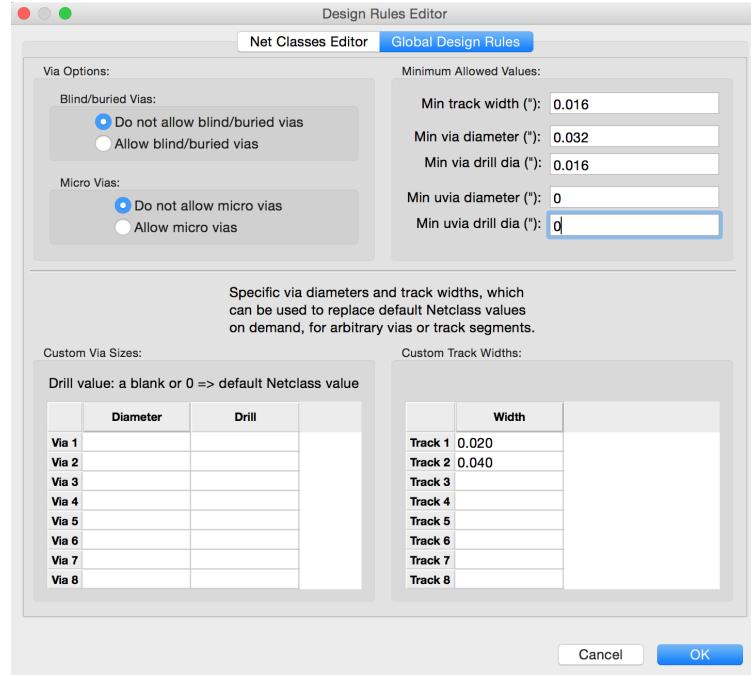


Figure 31: Global Design Rules

Next, switch to **Net Classes Editor** and update the Default net class dimensions as shown below:

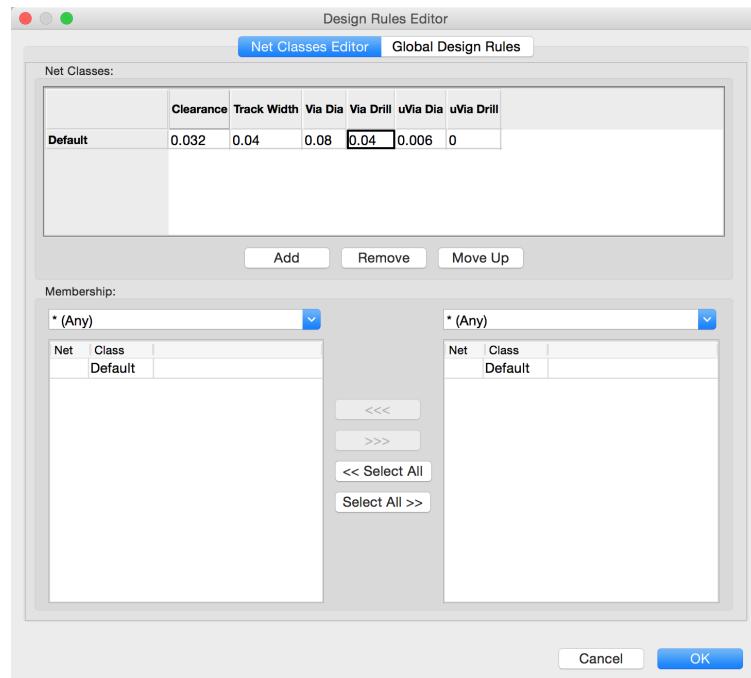


Figure 32: Net Class Design Rules

Net classes allow you to assign different constraints to different parts of your circuit. For

instance, it is often good to have extra wide traces for power supply traces, so you could define a special net class for all your supply connections.

Next, let's set up our layers. Open **Design Rules → Layer Setup**. A PCB is made up of layers of copper and fiberglass laminated together, kind of like a sandwich. On the top left of the dialog, there is a dropdown box for selecting various presets. For the Othermill, choose **Two Layers, parts on Front only**.

You will often hear about “2-layer” or “4-layer” boards. This refers to the number of copper layers in the board, with more layers giving more flexibility but also being more expensive. As you can see, there are also many other layers. Masks and paste are used to define where component’s pads are for automated soldering. The silkscreen layers let you add labels or artwork to the PCB. There are also a number of hidden drawing layers used to make notes or comments for the manufacturer.

Click OK when you are finished, and you should see your board stackup update in the right panel of the editor.

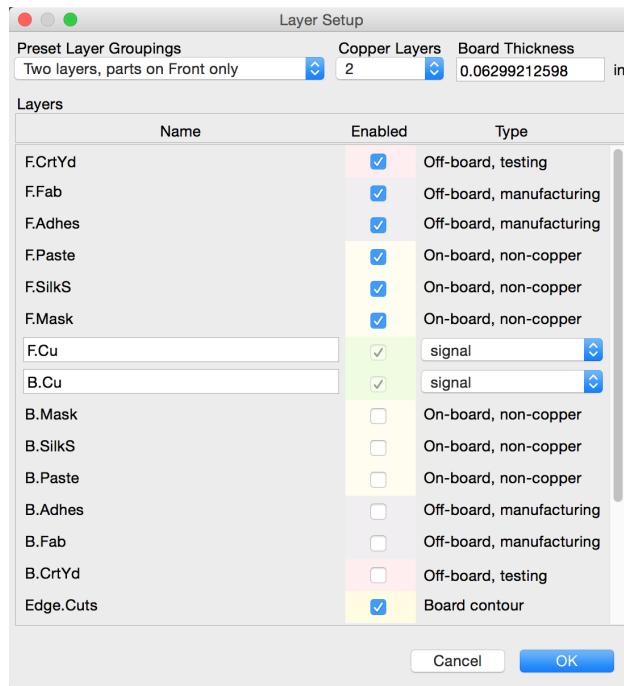


Figure 33: Layer Setup Dialog

## 4.2 Component Placement

Now we can import the circuit layout from the netlist we generated from our schematic. Locate the import netlist button on the middle of the top toolbar.



Figure 34: Import Netlist Button

Click the **Read Current Netlist** button, and you should see a list of components being imported into the PCB. After you close out of the dialog, you should see all the components in a giant pile in the middle of your PCB.

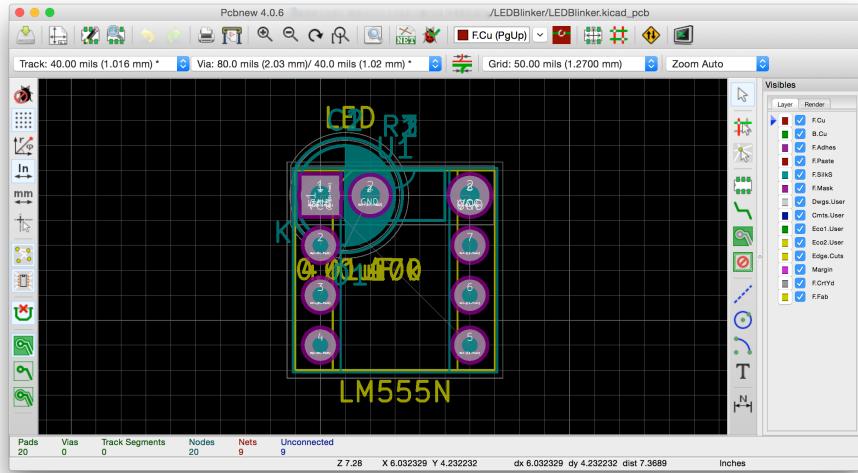


Figure 35: Components Imported from Netlist

Use the move and rotate commands just like in the schematic editor to separate out all the components. Make sure that no components are overlapping. Notice the faint grey lines running between the pads of the components. This is called the “ratsnest.” It shows all the connections from the schematic that we will have to make in the PCB. Try to rotate and move the components such that the lines are as short as possible, and connected pads are close together. Also, do not worry about the GND net, we will be doing something special for the ground connection later.

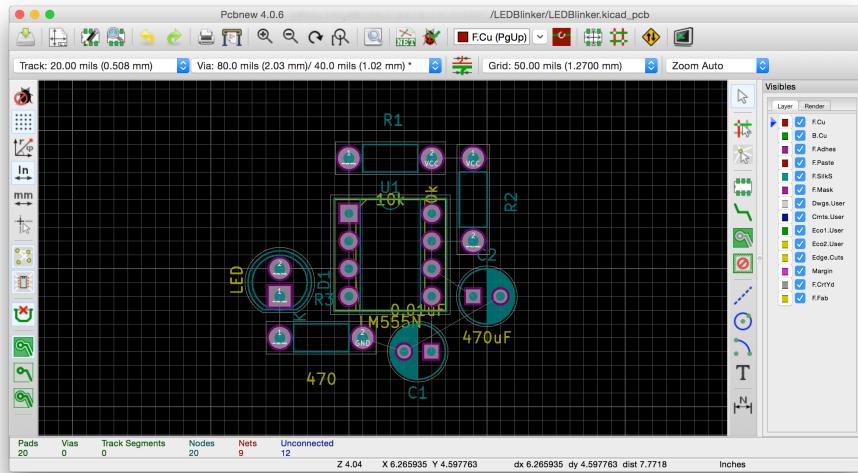


Figure 36: Example Component Placement

### 4.3 Board Outline

Now let us define the shape of our board. On the right layer selection panel, click on the row labeled **Edge.Cuts** with a Yellow icon. The blue arrow should swing down to show that it is selected.

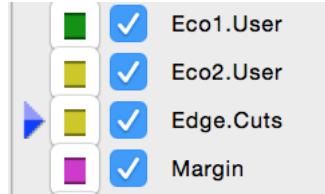


Figure 37: Selecting Edge.Cuts Layer

Next, select the Graphic line tool by pressing the button with the blue dashed line on the right toolbar.

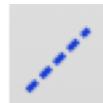


Figure 38: Graphic Line Button

We want to draw a rectangle around our components as our board outline. Click to start drawing the board outline, then move your mouse to define line segments. Click to set corners, then double-click on the starting point to finish the drawing. Note that it is totally possible to have non-rectangular boards, but not all board manufacturers will support them.

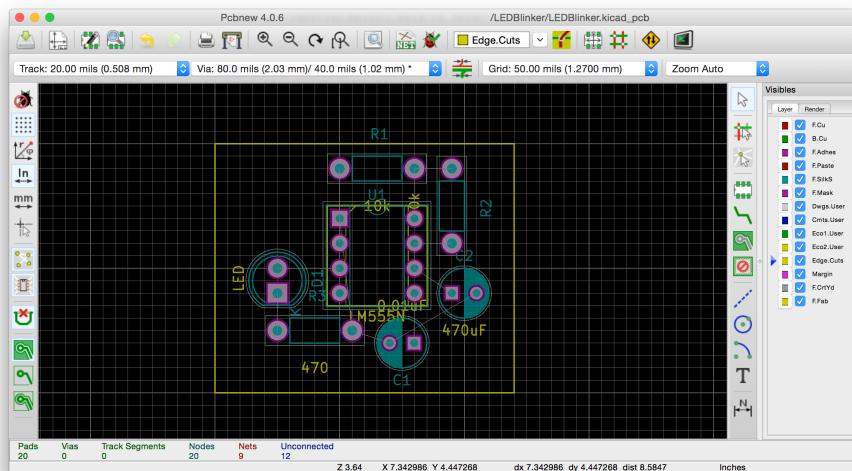


Figure 39: Example Board Outline

## 4.4 Routing

Now to draw the actual traces. We need to switch back to one of the copper layers, F.Cu or B.Cu. We are simulating designing a board for the Othermill, so we want to be routing on the back layer B.Cu.

This might seem counter-intuitive, but for the one-layer boards the Othermill creates, we want the copper on the opposite side as the components. If you were creating a board with surface-mount components, the traces will need to be on the same side as the components.

To enter routing mode, either click on the routing button, or press the default hotkey ‘X’. Then, click on a pad to start a trace. KiCAD will highlight the pads on the same net, so simply move your mouse over to nearby pads and click on them to create traces. You can press escape to exit out of the routing tool mid-trace.

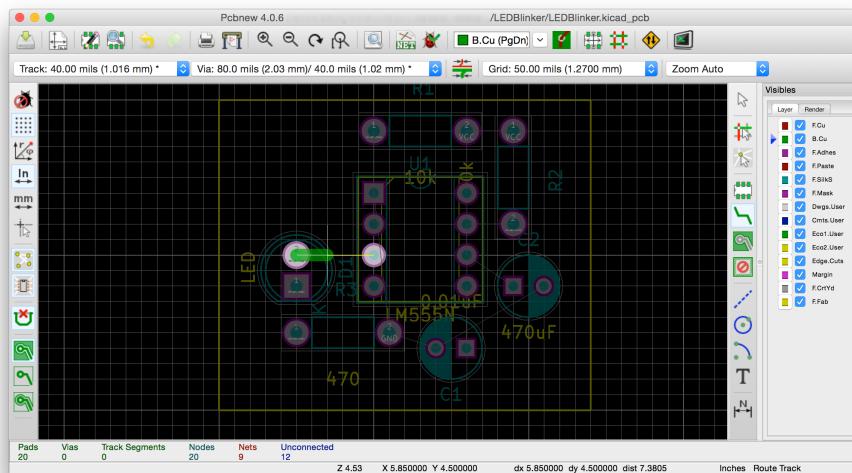


Figure 40: Routing a Trace

The routing tool will do its best to make nice clean bends, but sometimes it can get confused or have trouble fitting through small gaps. In those situations, you can manually drag segments around by pressing ‘D’ while hovering over a trace. Note that you have to be in routing mode for this to work! Also, for fine routing work you will likely have to adjust the grid spacing.

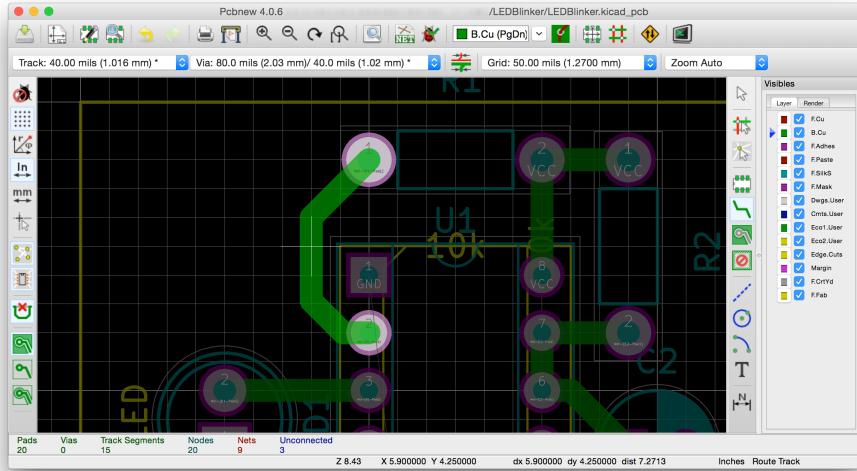


Figure 41: Dragging a Trace

Practice routing traces by finishing the connections for the rest of the board, except for the GND connections.

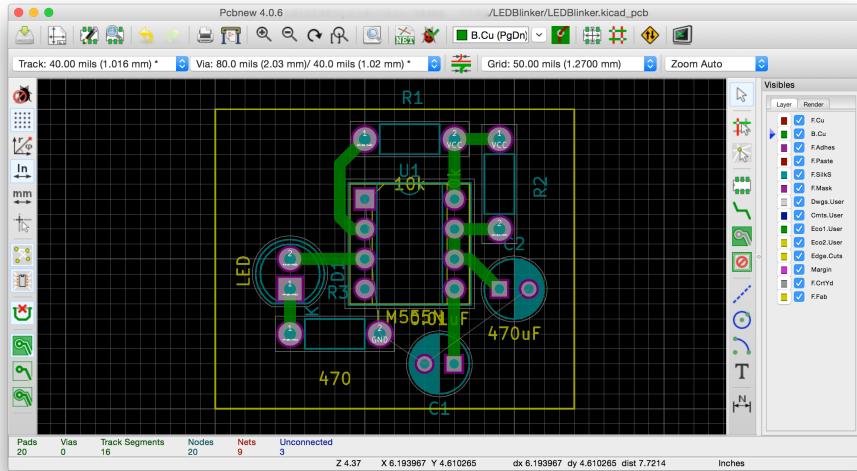


Figure 42: Example Routing

## 4.5 Vias

While this circuit is pretty simple, when routing more complicated circuits it is often the case that you need to have signals cross over each other, or switch layers. To do this, you can use vias. Vias are holes drilled through the entire board and plated with metal to allow signals to travel between layers.

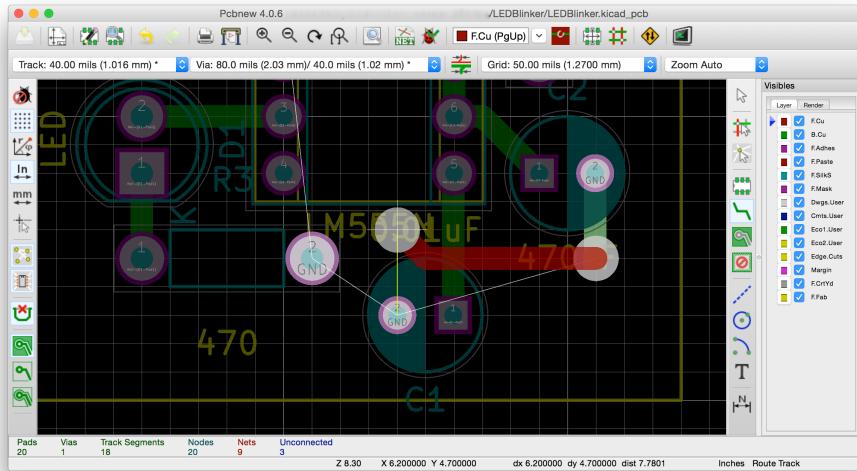


Figure 43: Placing a Via

To place a via, press ‘V’ while routing a trace in routing mode. When you click, a via will be placed and the trace will switch to the other layer. Now you can safely cross any traces that were in the way and place another via to return to the original copper layer.

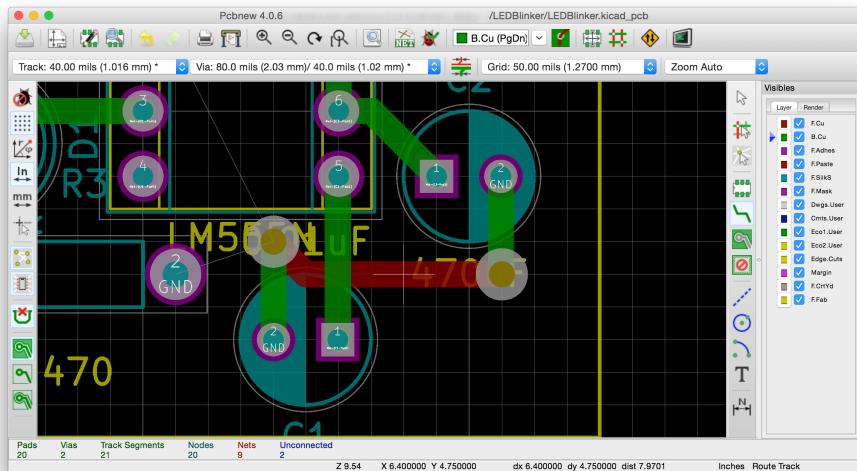


Figure 44: Finished Via Bridge

For a one sided board, vias can be used to represent jumper wires that will be hand soldered after the board is milled. As long as the via is physically large enough to accommodate jumper wire, it can be helpful to map out these ad hoc connections on the top layer, even if they will not be actually milled out.

## 4.6 Pours

We've been treating the ground connections specially because instead of just creating traces for ground, it is typical to create a ground plane that fills in all the empty spaces in the board. This helps improve the electrical performance of the board by ensuring that all ground connections are kept at the same voltage level.

In addition, the capacitance between other traces and the ground plane can help to reduce the amount of noise running through your board. This is especially noticeable for 2 and 4 layer boards, where sensitive signal traces can be sandwiched between multiple ground planes, forming a box blocking out external signals.



Figure 45: Pour Tool Button

Find the pour tool on the right toolbar, just under the routing mode button. To use this tool, we draw a polygon around all the places we want the pour to cover, then KiCAD will programmatically fill in the empty spaces with copper. Click on the top left corner of the board to start the pour.

A dialog should pop up to specify the properties of this pour. Select GND as the net to pour over and make sure the pour is on the bottom (B.Cu) layer. Most of the other settings are usually alright by default.

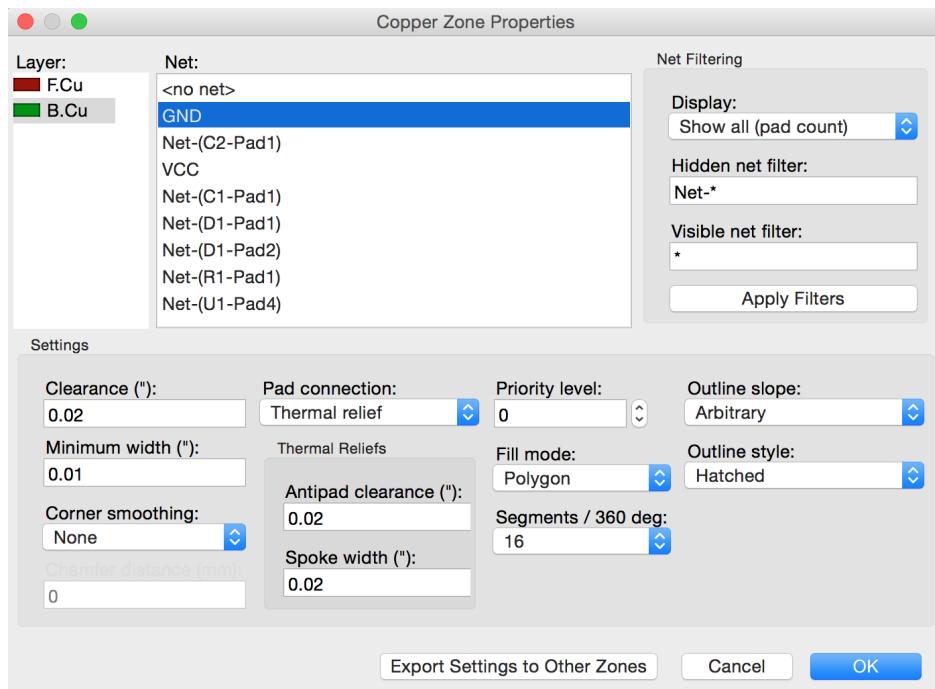


Figure 46: Pour Options Dialog

The one that might be useful to adjust is the “Pad connection” option. Since such a large solid piece of copper would be difficult to solder to, KiCAD will generate thermal relief patterns around pads. If you are making a homemade PCB on a mill, these thermal relief patterns can be fragile and difficult to mill. In those cases, it might be better to select the Solid option.



Figure 47: Thermal Relief Styles

Click OK to confirm the settings, then click on the corners of the board to define the pour region, similar to when we defined the board shape. Double click on the first corner to complete the pour. KiCAD will then automatically fill in the pour. If you later make changes to the board and need to update the pour, right click on the outline of the pour region and select the **Zones → Fill** option in the menu that appears. Or just press the ‘B’ hotkey.

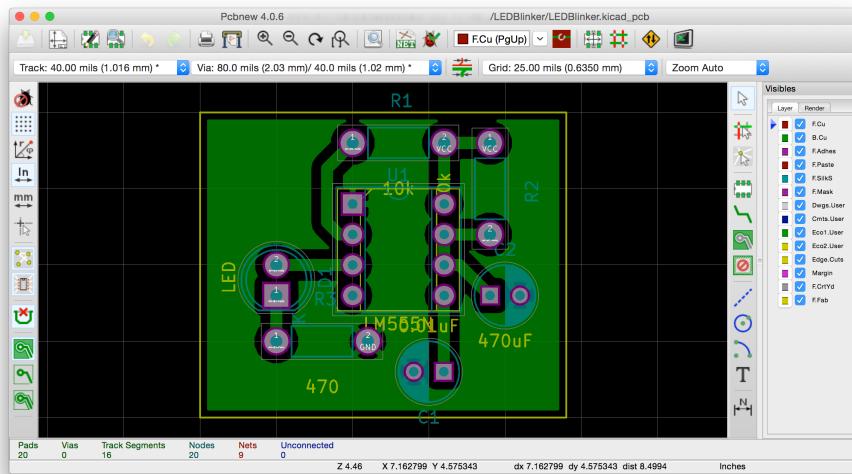


Figure 48: Example Ground Pour

## 4.7 Export

With that, the PCB is finished! Check that the bottom info bar shows 0 unconnected nets. You can also click the red ladybug in the top toolbar to run a design rule check (DRC), which will also check for clearance issues or footprint problems.

At this point, it is also a good idea to clean up the silkscreen layers and labels. Select the **F.Fab** Layer for value labels and **F.SilkS** for silkscreen labels. The general rule is to make sure that labels are close the component they describe, but more importantly that labels are unambiguous.

If a component has an obvious orientation, orienting the component identifier the same way is a good idea. Another style is to place all identifiers facing in toward the center of the component, so that the bottom of the text faces toward the component it describes. There are many ways of doing this, but above all consistency and unambiguous identification are key.

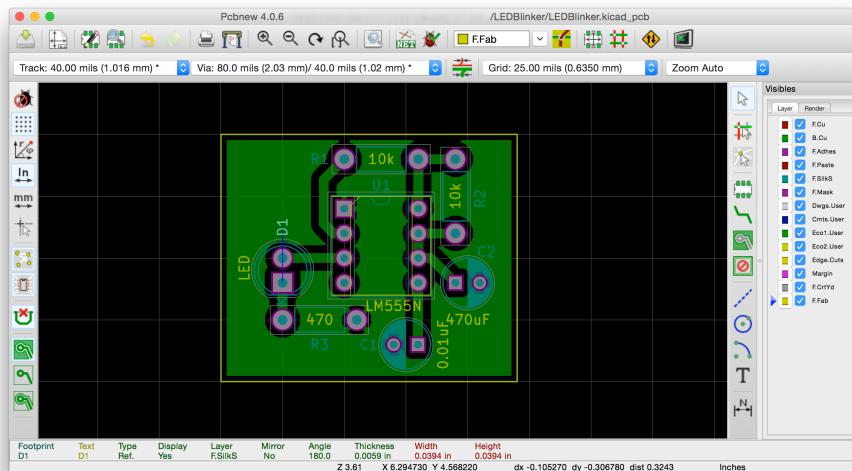


Figure 49: Cleaned up Silkscreen

The final output step will depend on what your manufacturer wants. Most will have explicit instructions on their websites. The typical output format is called the Gerber format. We will walk through the Gerber export process for the Othermill. Select the proper layers and options as shown below. To keep things organized, also specify an output directory for the files. Click the **Plot** button to generate a set of Gerber outputs.

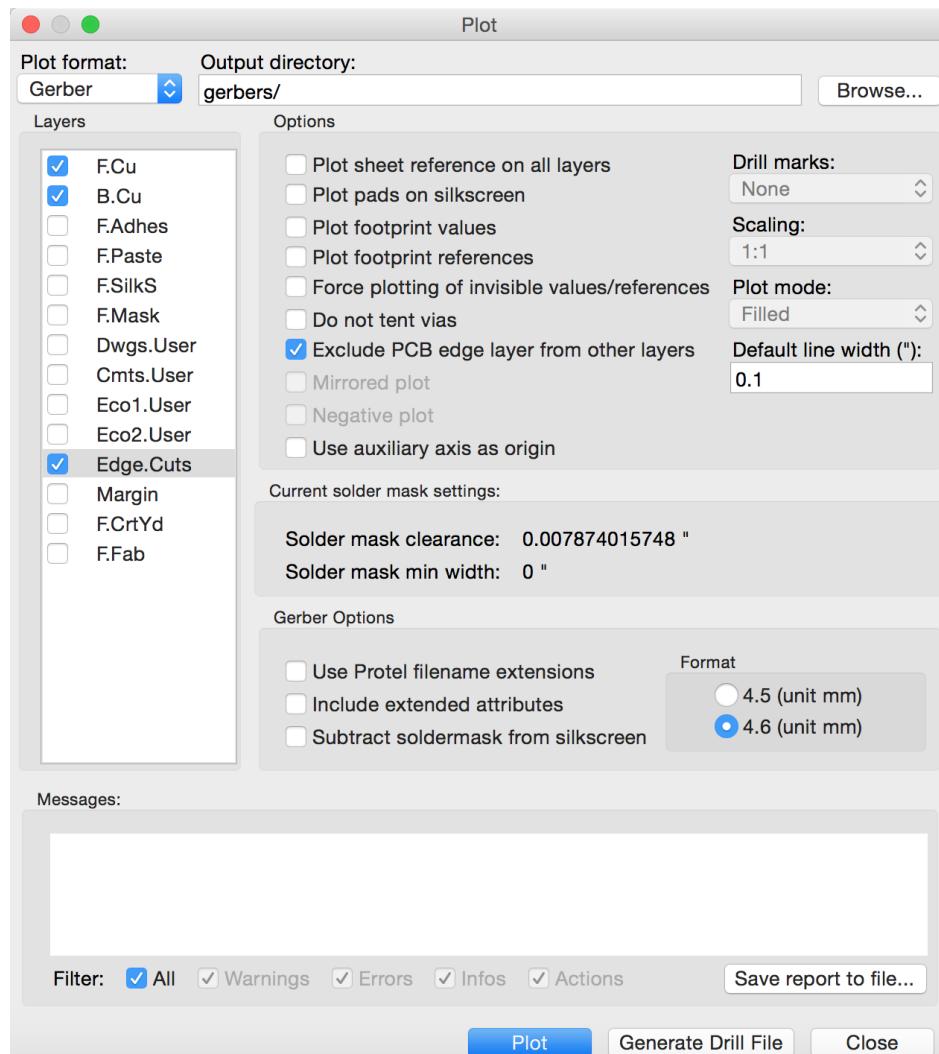


Figure 50: Plot Dialog

Before you click close, click on the **Generate Drill File** option. The drill file is a text file that tells the machine where to place all the holes in the PCB and what size they need to be. Set the options as shown below, then click the **Drill File** button to generate the drill file.

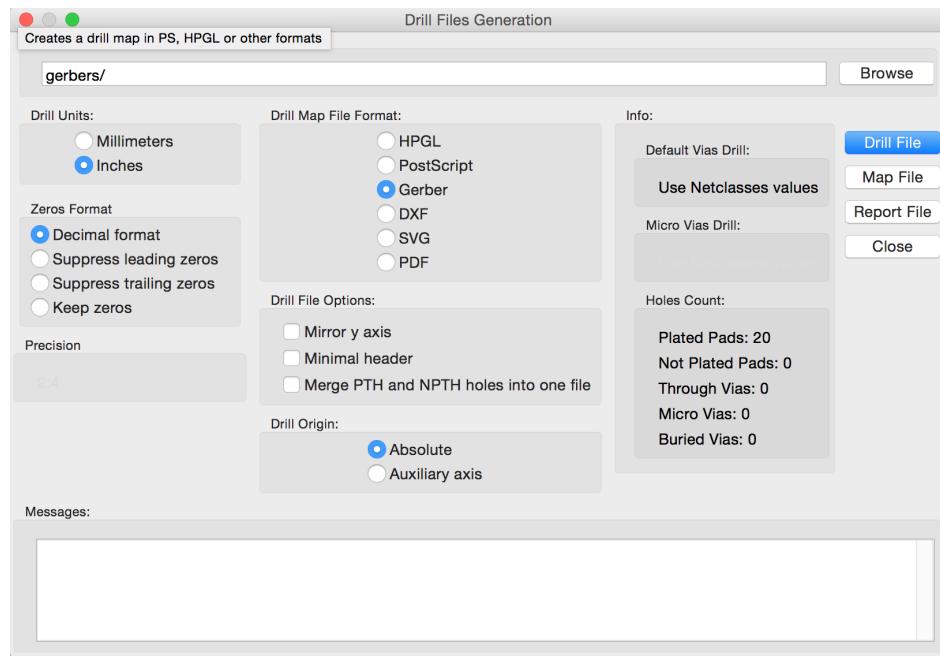


Figure 51: Drill File Dialog

These files can then be uploaded into the Otherplan mill control software. Don't forget to save before you exit!