

SIERRA  
CIRCUITS

# KiCad Design Guide

# Table of Contents

## Section 1. Creating a component symbol library in KiCad..... 4

1.1	Component library .....	4
1.2	Creating a schematic and symbol library.....	5
1.3	Creating a schematic .....	6
1.4	Creating a new library.....	6
1.5.	Creating a symbol.....	7
1.5.1.	Pin placement and numbering.....	9
1.6.	Using symbols in the schematic.....	10

## Section 2. Creating a schematic for high-speed design in KiCad... 12

2.1	Creating a schematic design .....	12
2.2	Placing components on a schematic.....	13
2.3	Creating a new symbol for high-speed design.....	14
2.4.	Adding pins to symbols.....	15
2.5.	Connecting symbols.....	18
2.6.	Annotating symbols.....	19
2.7.	Creating a high-speed schematic design.....	20
2.7.1.	Marking impedance and differential pairs.....	20
2.8.	Generating a BOM.....	22

## Section 3. Setting up the board parameters and rules in KiCad .... 23

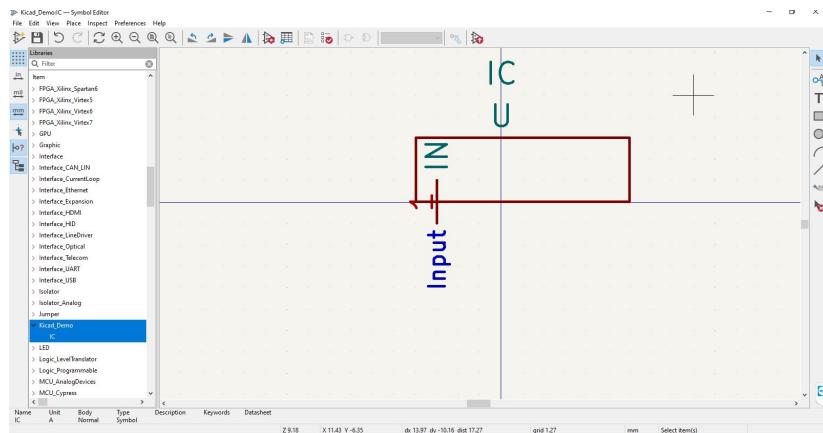
3.1	Board setup.....	26
3.1.1	Board stack-up.....	26
3.1.2	Text and graphics.....	29
3.1.3	Design rules.....	30
3.2	Modifying footprint references and grid selection.....	34
3.3	The differential pair design rules.....	34
3.3.1.	Differential pair length tuning.....	36

## Section 4: Component placement in KiCad..... 37

4.1	Drawing the board outline.....	40
4.2	Component placement .....	41
4.2.1	Mounting hole placement.....	41
4.2.2	Connector placement.....	43

4.2.2.1	USB connector.....	44
<b>4.2.3</b>	<b>Placement of the surge suppressor .....</b>	<b>45</b>
<b>4.2.4</b>	<b>Placing the auxiliary components as per signal flow.....</b>	<b>46</b>
4.2.4.1	Capacitor placement.....	46
4.2.4.2	Crystal placement.....	48
<b>Section 5. Routing a PCB in KiCad.....</b>	<b>49</b>	
<b>5.1</b>	<b>What is routing?.....</b>	<b>49</b>
<b>5.2</b>	<b>Arranging stack-up.....</b>	<b>49</b>
<b>5.3</b>	<b>Critical routing.....</b>	<b>50</b>
5.3.1.	How to route differential pairs.....	54
5.3.1.1.	Fiducials.....	54
5.3.1.2.	Stack-up.....	56
5.3.1.3.	How to pour ground.....	56
5.3.2.	Differential pair length tuning.....	60
<b>5.4.</b>	<b>Crystal routing.....</b>	<b>64</b>
<b>5.5.</b>	<b>Routing for decoupling capacitors.....</b>	<b>66</b>
<b>5.6.</b>	<b>Main pin routing.....</b>	<b>68</b>
<b>5.7.</b>	<b>Routing the layers.....</b>	<b>69</b>
<b>5.8.</b>	<b>Ref text placement.....</b>	<b>72</b>
<b>5.9.</b>	<b>Design rules checker (DRC) and clearing DRCs.....</b>	<b>75</b>
<b>Section 6. Exporting Gerber and production files in KiCad.....</b>	<b>80</b>	
<b>6.1.</b>	<b>Checklist for generating Gerber files.....</b>	<b>80</b>
6.1.1.	Page settings.....	81
6.1.2.	Add logo and text.....	82
6.1.3.	Add dimensions.....	84
6.1.4.	Stack-up.....	85
6.1.5.	Fab details.....	85
6.1.6.	Drill chart.....	85
6.1.7.	Stack-up detail.....	86
<b>6.2.</b>	<b>Files required for PCB manufacturing.....</b>	<b>86</b>
<b>6.3.</b>	<b>Exporting Gerber.....</b>	<b>87</b>
6.3.1.	Required settings in the Plot popup window.....	88
<b>6.4.</b>	<b>Exporting drill files.....</b>	<b>89</b>
6.4.1.	Required settings in the Generate Drill File popup window.	89
<b>6.5.</b>	<b>Generation of pick and place files .....</b>	<b>90</b>
<b>6.6.</b>	<b>Generation of netlist file.....</b>	<b>91</b>
<b>6.7.</b>	<b>Generation of BOM file.....</b>	<b>92</b>
<b>6.8.</b>	<b>Generation of PDF.....</b>	<b>92</b>
<b>6.9.</b>	<b>Generation of PDF for schematic.....</b>	<b>93</b>
<b>6.10.</b>	<b>Using GerbView.....</b>	<b>94</b>

# Section 1: Creating a component symbol library in KiCad



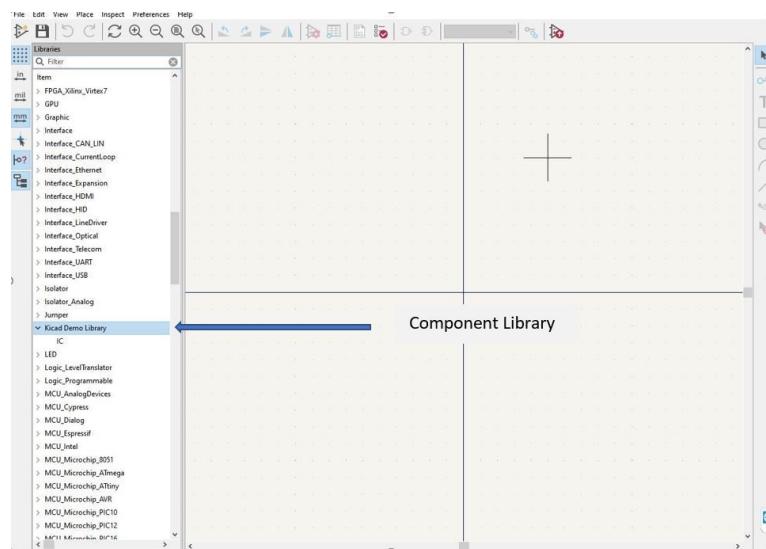
Creating component symbol

We will now explain how to create a schematic and a symbol library for components to be used in the schematic.

But before that let's have a brief look at the KiCad component library.

## 1.1 Component library

All the components used in a schematic are described in the component libraries. Several libraries are used to provide reasonably simple component management, through the grouping of components by its subject (by functions, manufacturers).



Component library

The component library contains a large variety of pre-installed libraries. Also, the Library Manager menu helps you to modify libraries like: build new libraries, add and delete components from libraries, and pass a component from one library to another.

Check out the KiCad library for symbols before creating your own:

<https://kicad.github.io/symbols/>

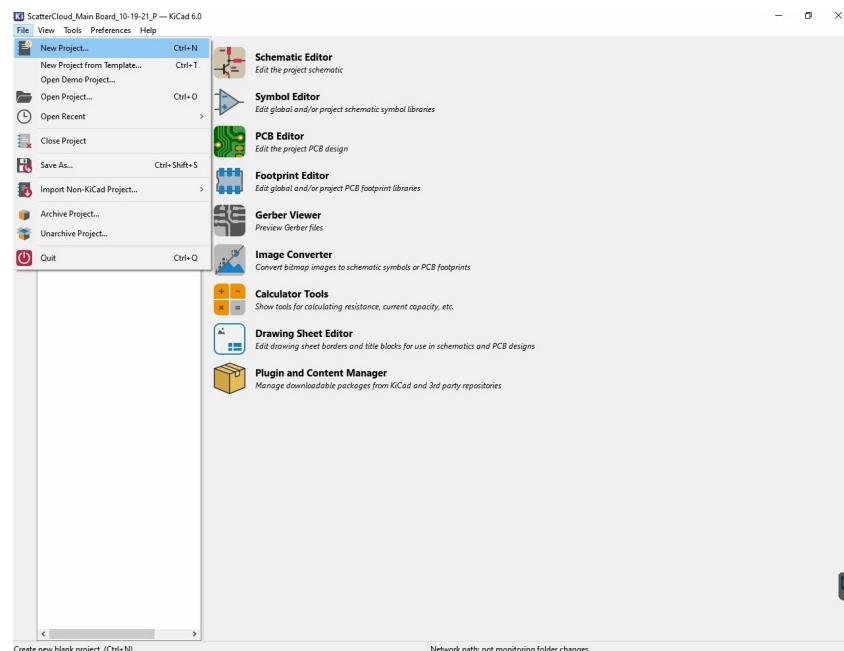
**DEMO VIDEO - [Creating a Component Symbol Library in KiCad](#)**

## 1.2 Creating a schematic and symbol library

If you just downloaded KiCad, you might have to install some **libraries**, like schematic and symbol libraries. Today, we are going to show you how to proceed. First, let's open KiCad on your computer.

To create a new project, follow these simple steps:

- Go to **File** on the left-hand side
- Click **New**
- Select **Project**



New project creation

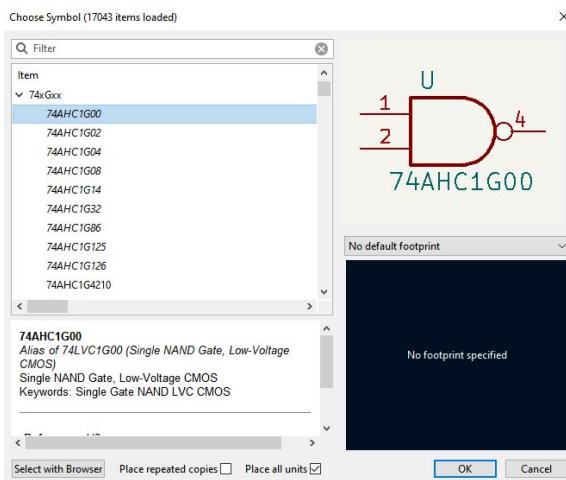
You can select the respective location where you want to save this new file and name it. For this demonstration, we will call it **New Design**. You can now click **Save**.

Once your new project is created, it opens automatically and you can see two files. One is the board file and the other one is the schematic file, which we created already.

## 1.3 Creating a schematic

To create a new KiCad schematic, double-click on Schematic. We need a schematic design which we will draw over the appearing frame. If the schematic design is available, you will find certain [components](#), such as capacitors, inductors, or maybe simple ICs, like regulators on the right-hand side. Click a symbol and drag it to the center of the frame.

**Double click on Schematic > select a pre-installed library/click create to add a new KiCad symbol**



Creating a new schematic

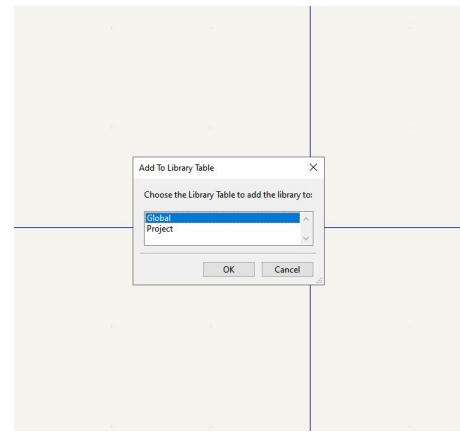
All the libraries will load and a window will pop up. These are pre-installed libraries and symbols which already exist in KiCad. For instance, we can click this schematic symbol and place it in the frame.

If a symbol is not available, you can create a new KiCad symbol by clicking Create, Delete, or Edit the symbol at the top.

## 1.4 Creating a new library

Before creating a symbol you need to create a new KiCad library. All the libraries you can see on the left-hand side are in-built so you won't be able to modify them.

To create a new library, on the left-hand side, under File, click New Library. You have to browse to the location where you want to save it and name it.



Add to Library Table window

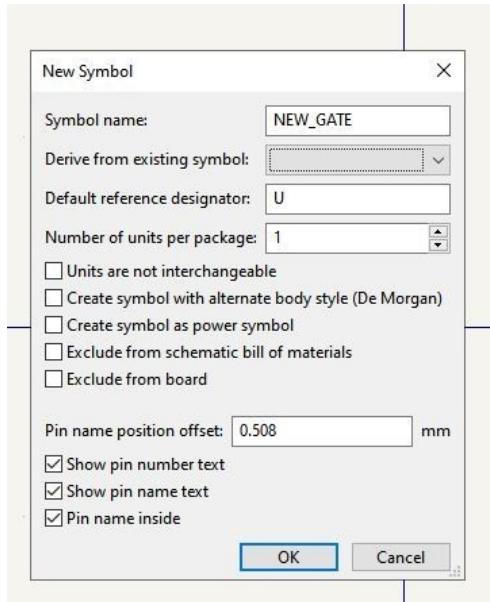
You now have two options: Global or Project.

If you are using this library specifically for your current project, you can select **Project**. If you select **Global**, this library will be added to the predefined libraries which already exist in KiCad. For this demo, we will select the global one. Now the library is created.

## 1.5 Creating a symbol

To create a new KiCad symbol, follow these simple steps:

- Go to **File**
- Click **New Symbol**
- Select the library in which you want to save the symbol



Configure component properties

This window will pop up where you can name your symbol, for instance, IC, and click Ok.

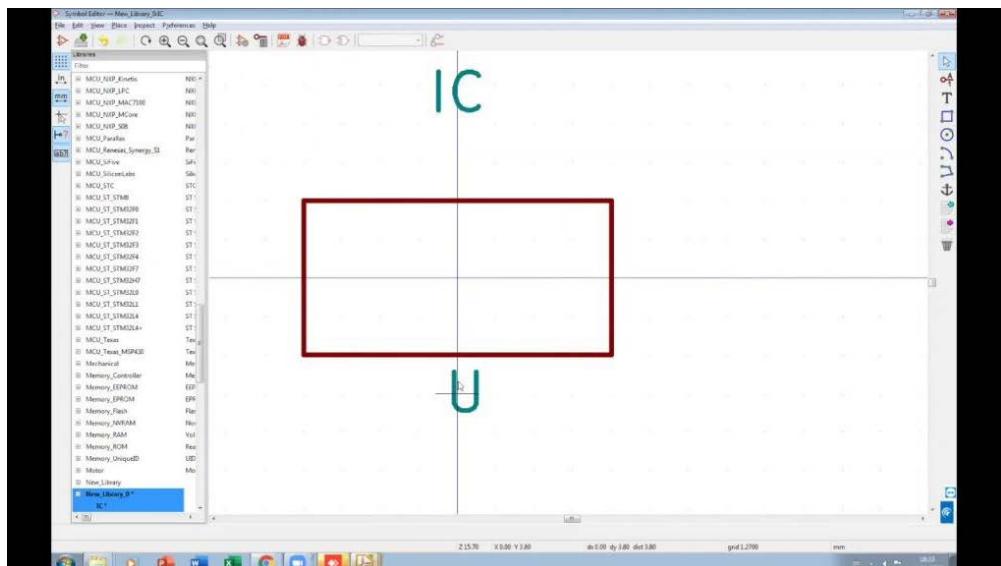
You can now see IC, which is the name of your symbol, and U, which is the reference.

Select the reference and move it from the center. Repeat the same action with IC.

Let's zoom out of it.

To set the grid, right-click on the screen and you will see an option called Grid. It's best to keep it to 50mils.

Let's say you want to create a symbol for IC. On the right-hand side, click the rectangle called **Add graphic rectangle to symbol body**. Select it and draw a box.



Adding graphics to symbol body

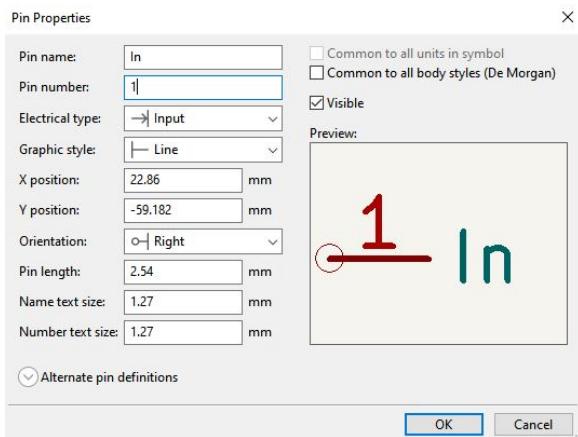
Before connecting the pins, which are nothing but the terminals of your IC, you need to fill the background of this box. There are two different ways to do this:

- Place your cursor on this line > right-click on it > select **Edit Rectangle Options** > select **Fill with body background color**. It will color the entire shape.
- Another way to complete this action is to keep the cursor on your line and click E on your keyboard. The **Rectangle Drawing Properties** window will appear and you can directly select Fill with body background.

### 1.5.1 Pin placement and numbering

You now need to place the pins. On the right-hand side, click the A1 symbol. Then, you can drag it and click the screen.

**Click on the A1 symbol from the right toolbar > Configure your pin properties**



Configure the pin properties

The **Pin Properties** window will pop up. You can now configure your pin properties.

Let's assume this as a regulator. A regulator has three pins which are input, output, and ground. Therefore, we will require three different pins.

Fill out the pin name and the pin number fields. Here, we will write **IN** and **1**. You can personalize the other properties.

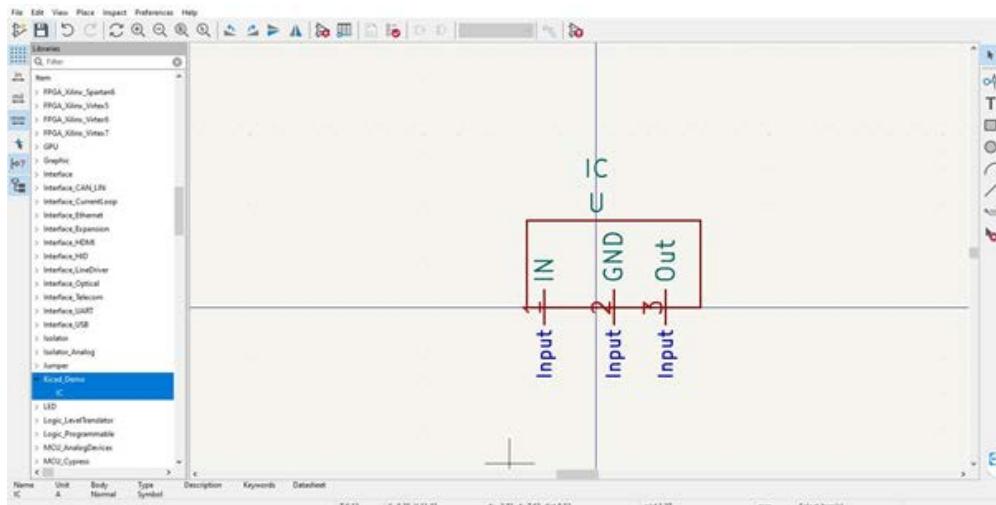
Let's say you want to rotate and place this pin on the left-hand side. Right-click on your screen and select **Rotate Clockwise**.

Repeat the same action for the two other pins.

The second pin name will be **GND** and the pin number **2**. As for the electrical type, you have the option to choose **Unspecified**. Click on the screen, select Rotate **Clockwise**, and place this pin over here.

The last one is output. The pin name will be **OUT** and the pin number **3**.

Here, we have used three pins with names **Input**, **Output**, and **Unspecified**, and connected to the bottom of the device. If you want to add more pins, you can add them on all sides of the rectangular box.



Component pin placement

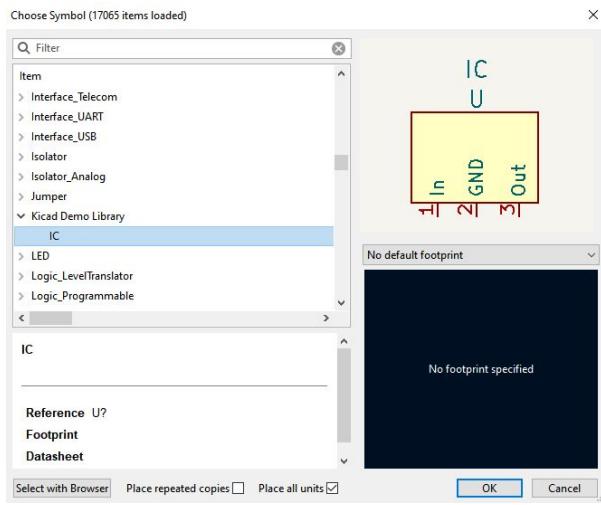
Once the pins are placed, **right-click on the screen > select End Tool**.

You can see the pins are placed and the names are given. The symbol is completed. You can save it. Remember this library name, which is New Library 0 here, and the symbol name, which is IC here, as you can see at the left-hand bottom corner.

## 1.6 Using symbols in the schematic

Once you save the symbol, you can minimize your window. To use the symbol in your schematic:

**Open your schematic page > select Place symbol on the right side from the toolbar > Drag it to the center of the screen > Choose Symbol window will pop up**

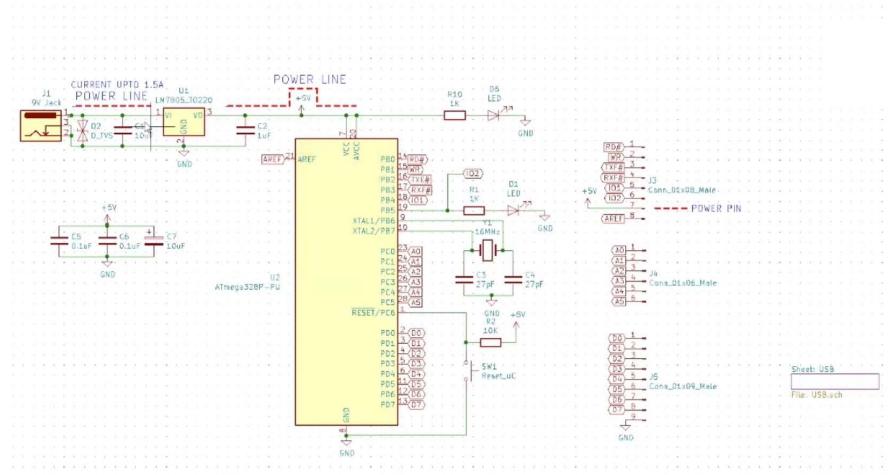


Choose a symbol from the library

From **Choose Symbol** window search for your library. Here, it's the **New Library 01**. You can see the preview of the symbol IC we just created. Place it on the schematic, right-click, select **End Tool** and **Save**.

This is how we create a schematic and symbol library in KiCad.

# Section 2: Creating a schematic for high-speed design in KiCad



An example of a PCB schematic

We will cover schematic design in KiCad. We will also discuss high-speed PCB design with the identification of differential pairs and the marking of high-current traces.

KiCAD is a widely used free Electronic Design Automation (EDA) software. The software is mainly used as a PCB layout tool and to create schematics. It also facilitates the creation of a Bill of Material (BOM). We will discuss such features below.

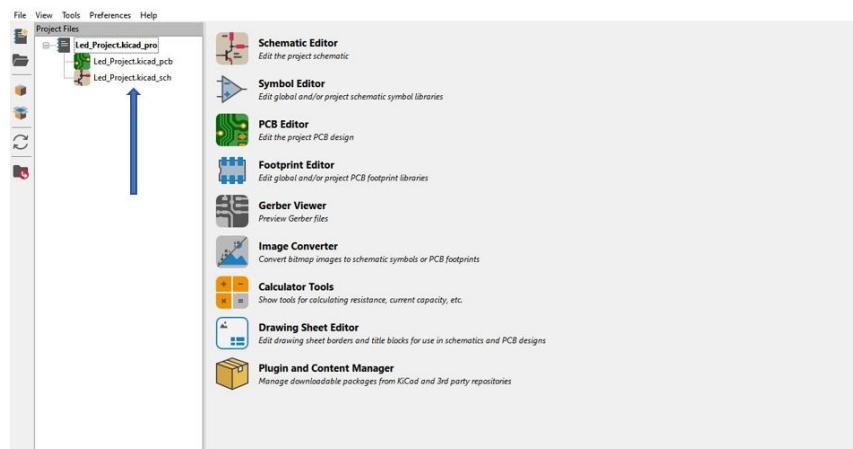
## [\*\*DEMO VIDEO - Creating a Schematic for High-Speed Design in KiCad\*\*](#)

### **2.1 Creating a schematic design**

We will demonstrate how to create a simple schematic design in KiCad. Also, we will talk about the factors we need to consider when doing a schematic design for a high-speed design. Let's start with the schematic part.

First, you need to create a new project. Go to File on the left-hand side, click New project. Choose the location where you want to save this new project and name it. In this demo, we will name it as an LED project. Click Save.

**Go to file > Click new project > Choose a location to save your project and name it > Click save (In this demo, we will name it LED project.)**



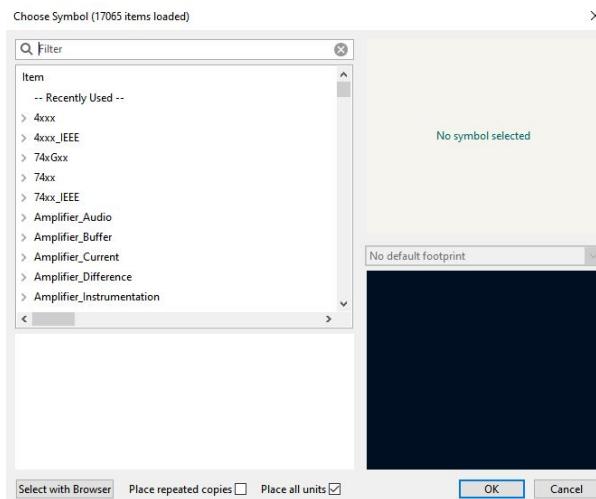
KiCad main menu

Two files are created in the name LED project at the top left side. From the top, the first one is a layout file and the second one is the schematic file. Let's open the schematic file first.

## 2.2 Placing components on a schematic

Here we are considering a simple schematic design for a battery, a regulator, and an LED. This schematic will be used to light up a simple LED. To place components in your schematic you need to choose components from the library. To do that follow these steps:

**Select Place symbol  icon from the toolbar on the right side > a window will pop up > choose your component**



Choose symbol window

You can choose your components from the library. Here we need a battery, resistor, capacitor, and an LED.

Search for a battery among the components. We will select a single-cell battery for this instance. Simply click the screen to place the battery on the schematic.

Let's repeat the same action to find a resistor. Select the one you want and place it on your schematic. Then choose an LED and two capacitors and regulators.

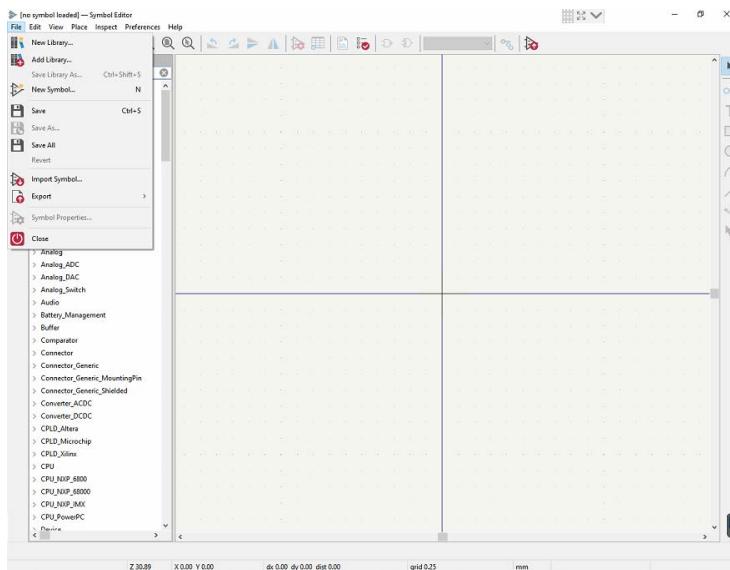
While choosing a capacitor, you will have two choices: an unpolarized and a polarized capacitor. The unpolarized capacitor is basically a ceramic capacitor, and a polarized capacitor is electrolytic. We will take the unpolarized one for now.

## 2.3 Creating a new symbol for high-speed design

We have selected components that are already available in the library. All the components we require may not be in the built-in library, for example, a regulator.

Since we require a regulator, we will create it here

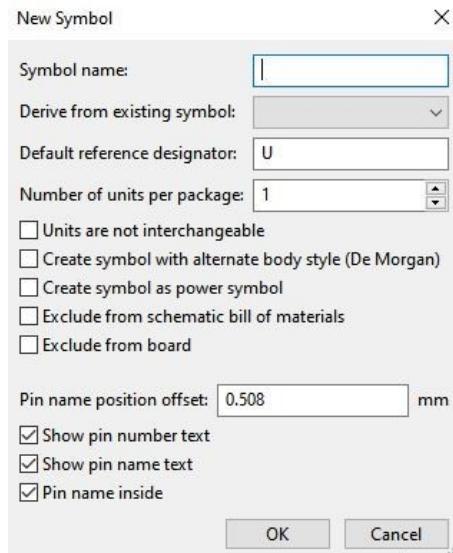
**Click Create, delete and edit symbols  from the toolbar on the top > select file from the new window > select new library > name your library > click save**



Creation of new symbol for high-speed designs

Here we will name the library as an LED project.

Now go to **File** again, select **New Symbol** and search for the library that you just created.



New symbol generation

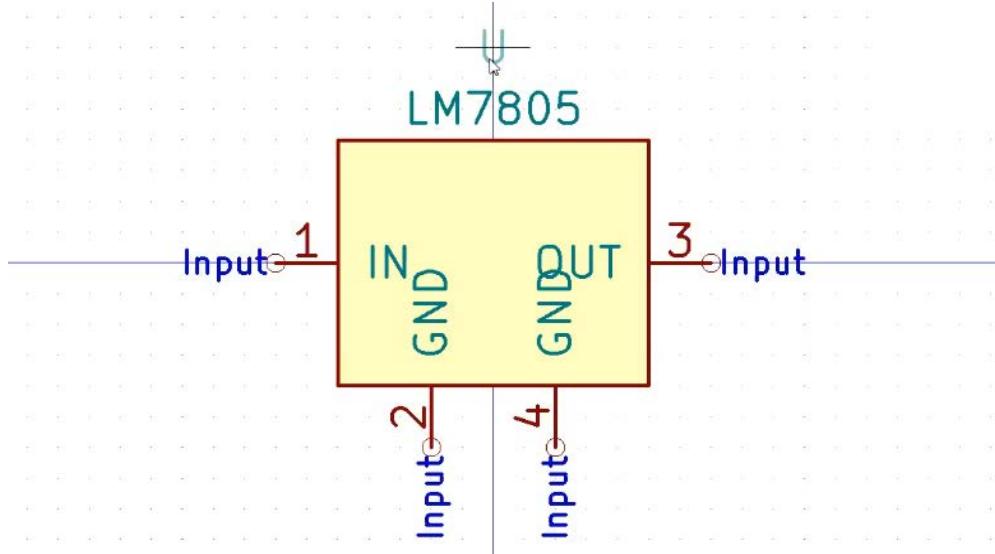
Name the new symbol, such as, in this case, LM7805, which is a five-volt regulator. You need to create a symbol for every component. Every component has its own datasheet and its serial port number.

You need to find the datasheet for the part number. Here we need the datasheet of the regulator LM7805. You need to see the pin configurations of your device from the registered datasheet by the manufacturer.

In this case, there are four different packages. These packages depend on the Manufacturer Part Number (MPN). So, you will have to decode the MPN which will give you the correct package. Here, we will require these three pins: input, ground, and output. You can see that there are more connectors, such as Tab. They all are ground.

## 2.4 Adding pins to symbols

We have created a schematic block, now we will have to create pins to have an additional pair on the board. Here in the case of a regulator, we need four pins. The ground will have two pins, which are Pin 2 and Pin 4. Pin 1 will be input and Pin 3 will be output.



Adding pins to symbols

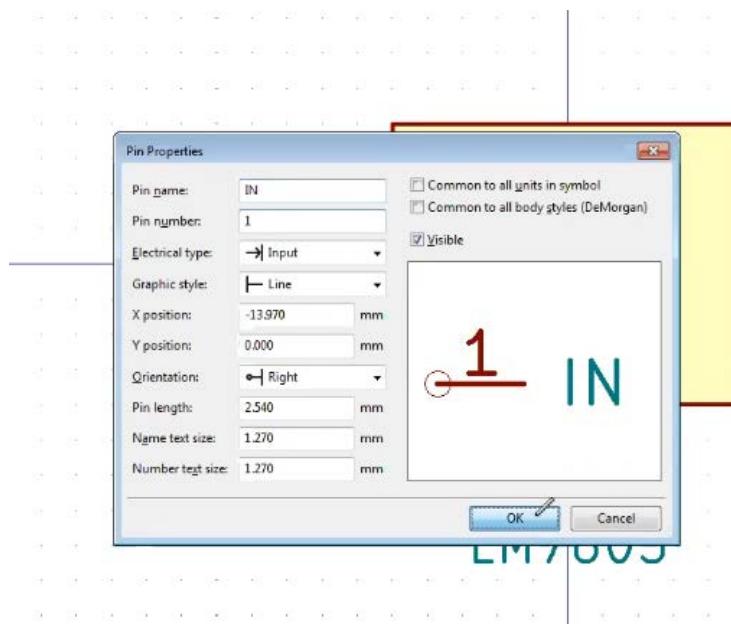
Let's start and move the components aside. We will use the **Add graphic rectangle to symbol body** option on the right side. But before that, right-click and select **End Tool**.

You have to select the grid first, so right-click again on the screen and make sure your grid is set to 50mils. Now, you can click this box and draw. **Right-click and select End Tool.**

To fill color in the box, **right-click the border of the rectangle > select Edit Rectangle Options > Fill with body background color > click Ok**. Now your regulator box is ready, now we have to add pins to it.

On the right-hand side, click the A1 symbol, and then drag it to the screen and click on the screen. A pop-up window will appear.

**Select A1  $\textcircled{1}^{\text{A}}$  symbol > drag it to the center of the screen > click ok > configure your pin in the appearing pop up window.**

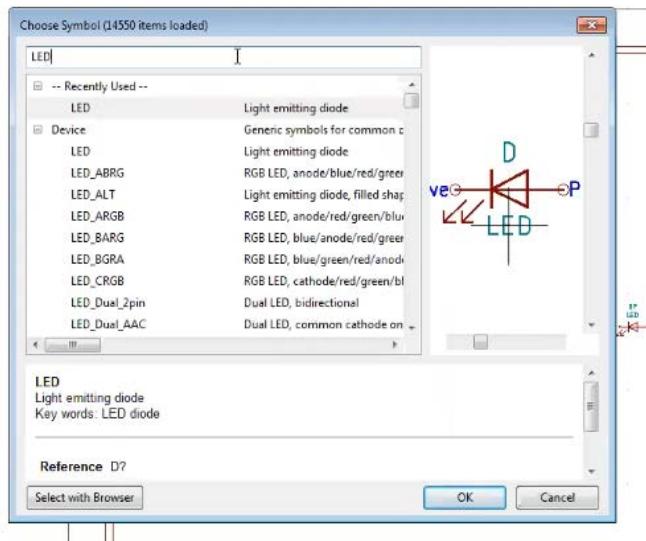


Pin Properties Window

Here the pin name will be input and the pin number one. In this way, you can create the required numbers of pins. If you have to rotate your pin, **right-click on the pin > select Rotate clockwise**.

Here, we have created four pins for the regulator. Once the pins are created, save them by clicking save all changes on the left-hand side.

Once the device is saved, you can import this in your schematic design file. To do this, **go to the main schematic file > select Place symbol**. A window will pop-up where you can find the LED project.



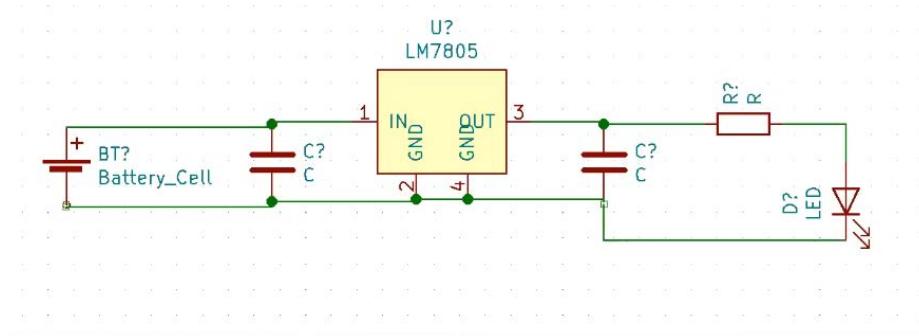
Choosing symbols

Now you can see the LM7805 device inside the LED project. Select it, click Ok and place it on your schematic. Import all the symbols required to proceed. You can then place these components properly and make the connections.

## 2.5 Connecting symbols

To wire the components first arrange these components properly, then

**Go to the toolbar on the right hand side > select place the wire  icon > draw a line between component leads where you need the connection.**

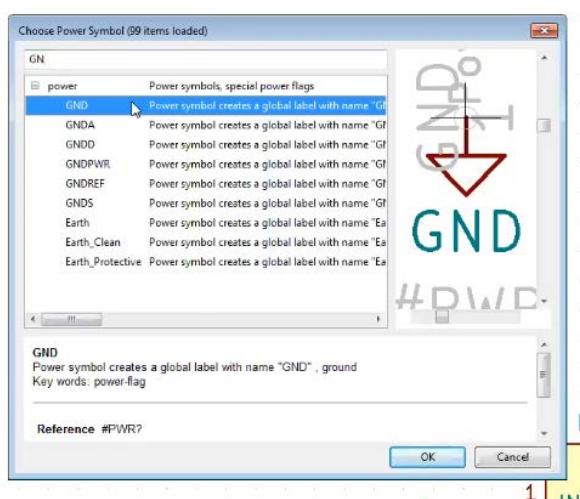


Connection of symbols

When you are done, right-click and choose the **End Tool**. You can see a small square on the connection line. If the wiring is not made properly, select that line and delete it. Again, select **Place Wire** and make the connection. Repeat everywhere it needs to be done.

You now need to connect a ground terminal.

**Click the Ground option  on the right-hand side toolbar > click on the screen.**  
A window will pop up.



Choose Power Symbol Window

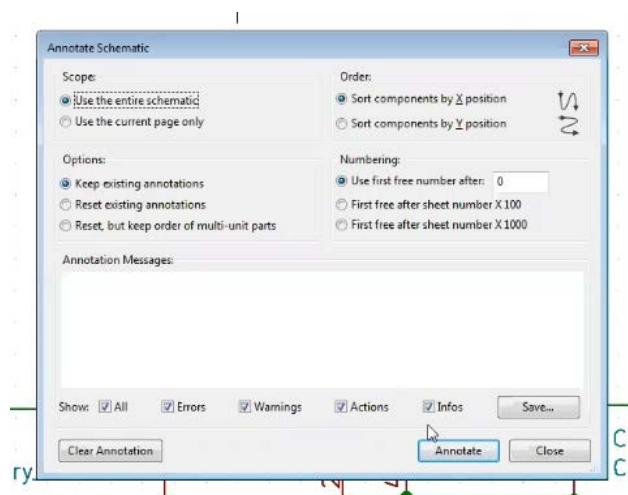
From the pop-up window search for **GND**, select the symbol, and place it on the schematic. Select **Place Wire** again and connect the negative to ground. **Right-click** and select **End Tool**.

The connections are done for the components of this schematic design in KiCad. Don't forget to save the file.

## 2.6 Annotating symbols

After the connections are done, if you see a question mark next to the **U** symbol, it means you have to annotate this symbol. On the top, there is an option called **Annotate Schematic Symbols**.

Select **Annotate schematic symbol** > choose the options – **Use the entire schematic** and **Keep existing annotations** > select **Annotate**



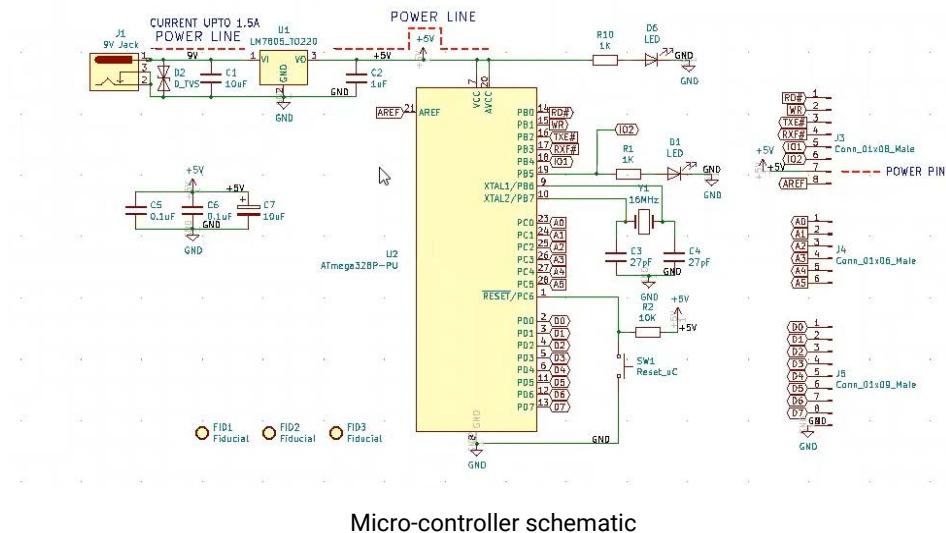
Annotate Schematic Window

Now you can see that the question mark has turned into a number. The tool automatically gives numbers to all the components. Again, don't forget to save.

The schematic design is complete but you need to assign footprints. The footprint information will be given in your datasheet. This is how you can create a simple schematic.

## 2.7 Creating a high-speed schematic design

Now we will explain how to create a high-speed schematic design. Here, we are using a circuit we previously created.



This is the schematic design of a microcontroller that is interfacing with an FTDI chip and a USB version 2.0. USB 2.0 is a high-speed signal, so it normally operates from 12MHz to 240MHz. For such cases, you need to be careful when designing the layout.

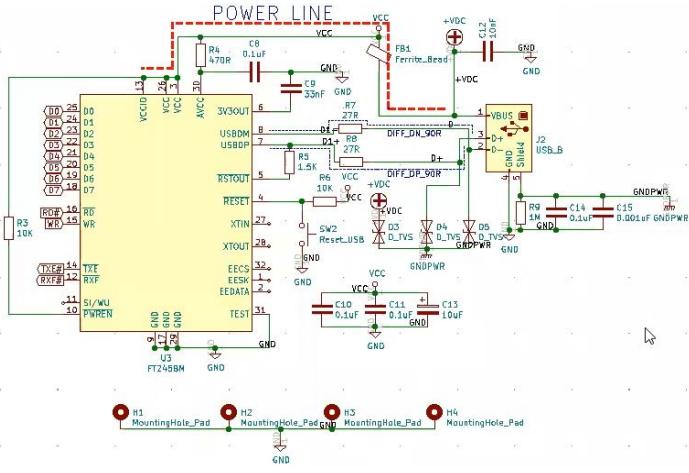
You need to design the specs to meet this requirement of 1.5 amps. You can see that there is a firewall over the top of the design as a red dotted line. This portion is our power line, designed to meet the spec 1.5 amps output.

There is a connector in the IC that is marked as a power pin because 5 volts are directly going to it and we don't know where this will be connected. It will depend on the person who will be using this. By marking the pin as a power pin, we indicate that the trace needs to be sufficient to carry 1.5 amps of current.

### 2.7.1 Marking impedance and differential pairs

Let's move onto another example.

This sheet has the FTDI chip which converts the input serial signal into a USB differential signal. The other yellow rectangle is the USB connector. Here, we are using a USB 2.0 which can operate from 12 MHz to around 240 MHz. You could also use a USB 3.0 which can operate up to 5 GHz.



Schematic of FTDI chip

Normally, we use differential pairs. Differential pairs are nothing but two lines carrying the same data in an inverted form with respect to each other at a particular point in time. You can see that we have **DIFF\_DN\_90R** and **DIFF\_DP\_90R**. These are the two pins that bring data into your IC from this connector. **DIFF\_DN\_90R** is a negative line and **DIFF\_DP\_90R** is a positive line.

Factors like the length of the trace, the width of the trace, the thickness of the trace, and the height of the trace from the ground plane influence the controlled impedance.

For instance, the main IC on this schematic has input impedance at pin8 of  $90\Omega$  as defined by the USB standard. The differential tracks should match this impedance. In order to do so, there are certain rules you need to follow, such as adjusting the width and thickness of the track.

There is also crosstalk, which is nothing but interference on one line due to another trace that is close by. You need to maintain a specific distance between these two tracks. The width of each track should be as per the requirements. The trace width and spacing for a certain impedance may be obtained using an impedance calculator.

**The length matching of these differential pairs is important because these two lines are time-dependent or delay-dependent. A signal comes from a connector at a particular time and reaches its destination at a particular time. The signals from both these tracks should reach at the same time. If the lengths are different, and since both signals on both differential pairs operate at the same speed, this can generate a false signal at the output.**

And there is a power track which operates at 500 millamps. 500 millamps is considered high for a USB. We normally put less power, less current. And this is how you get started on your high-speed schematic design in KiCad.

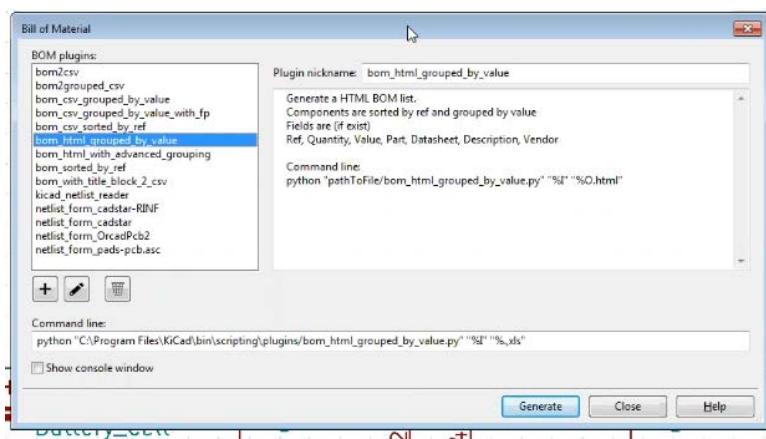
This is how we create a schematic design in KiCAD. We have covered every step in schematic designing including creating library and symbols, connecting and annotation components, and creating BOM for a design. Finally, we have gone through two high-speed schematic design examples.

## 2.8 Generating a BOM

You might also have to generate a Bill of Materials (BOM)., BOM is the complete list of all needed material in manufacturing the board, which includes the list of raw material, components, assemblies and subassemblies, and other items needed for product manufacturing.

To create BOM of your design,

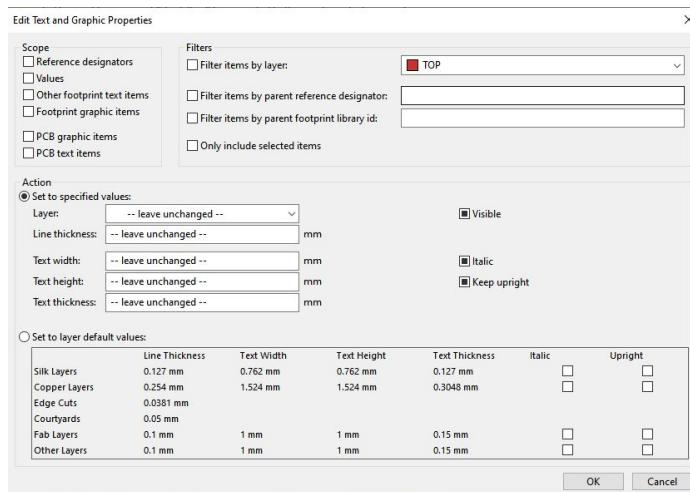
**On the top toolbar> Select BOM  > Select bom\_html\_grouped\_by\_value > make sure file name ends with .XLS extension > Generate**



Bill of Material Window

In the command line section, you can see **.xls**. Make sure that you have the **.xls extension** before you select **Generate**. This will generate a file in the folder of your project. The file contains the quantity, the value, and the part. This is your complete list, the complete data of your schematic design that indicates the components you will request for.

# Section 3. Setting up the board parameters and rules in KiCad



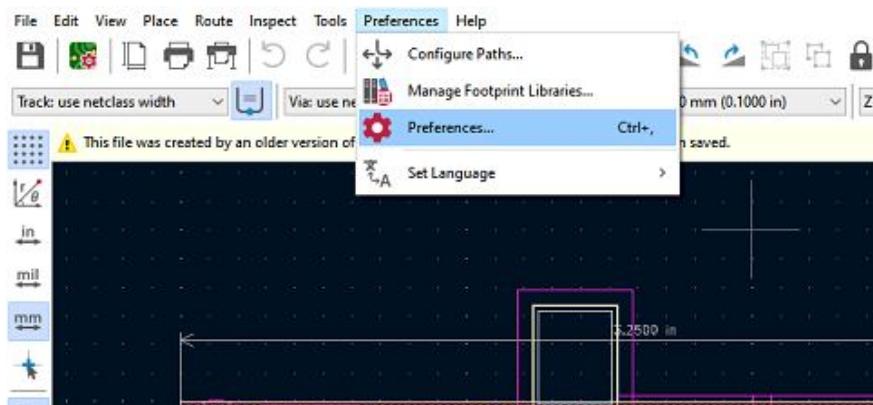
Defining the design rules

Before starting the layout, it is very important to determine a set of rules to be followed.

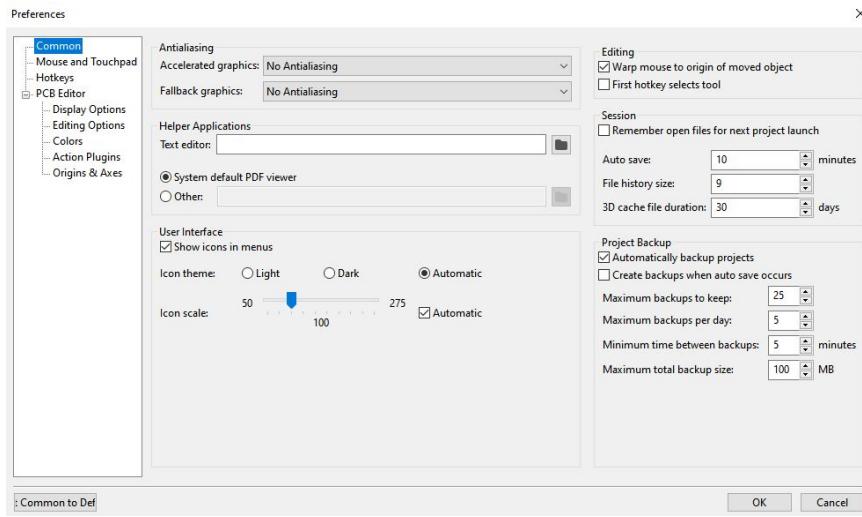
## DEMO VIDEO - [\(Design rules\) Setting up the board parameters and rules](#)

For selecting saving preferences follow the procedure below:

- Go to **Preferences**
- Click **Preferences**
- Fill Autosave information – **Autosave** and **File history size**

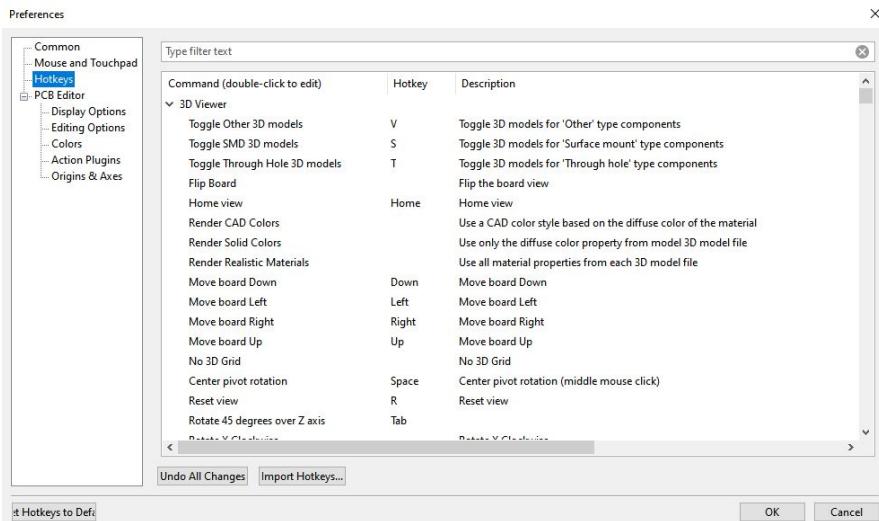


Procedure to select and save preferences



Selecting autosave frequency

In **Hotkeys**, you can choose shortcuts. This is an efficient way to do your layout. You have pre-saved commands but you can also add new ones.

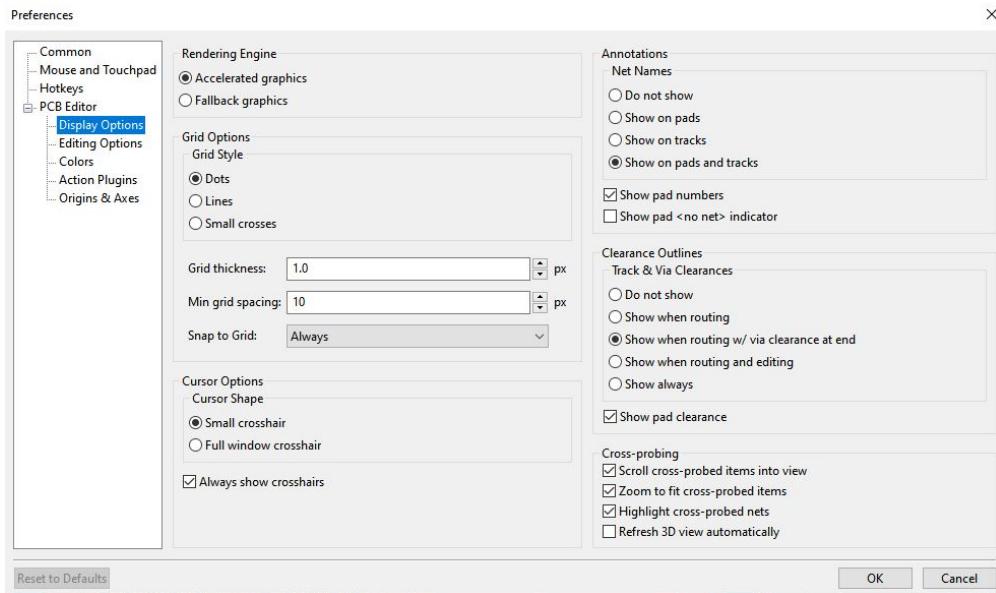


Hot keys options under preferences menu

Feel free to browse the display options as well. You can choose among many options, such as the grid style, thickness, spacing, etc.

To modify display options:

- Change **Grid style** and **Cursor Options** by selecting the required radio button.
- Enter values for **Grid thickness**, **Min grid spacing**, and **Dimensions** as required.

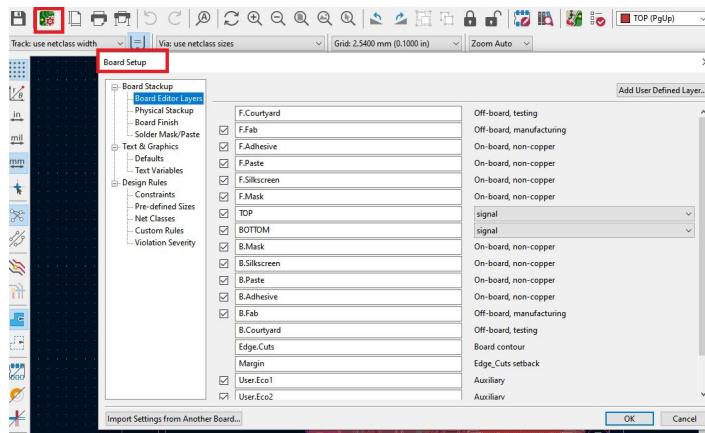


Display options in KiCad

## 3.1 Board set up

### 3.1.1 Board stack-up

To set up the board layers, click **Board Setup** at the top. In **Board Editor Layers**, you can customize your layer set.



KiCad board setup Menu

By default, you will have two copper layers. You can see the top copper layer and the bottom copper layer. These are signal layers.

Follow the below steps to set up your board stack-up:

A screenshot of the 'Physical Stackup' section of the KiCad board setup menu. It shows a table of layers with columns for 'Copper layers' (set to 4), 'Impedance controlled' (checkbox), 'Layer', 'Id', 'Type', 'Material', 'Thickness', 'Color', 'Epsilon R', and 'Loss Tan'. The table lists the following layers: F.Silkscreen (Top Silk Screen, Not specified, 0.01 mm, Not specified, 3.3, 0), F.Paste (Top Solder Paste, Not specified, 0.01 mm, Not specified, 3.3, 0), F.Mask (Top Solder Mask, Not specified, 0.01 mm, Not specified, 3.3, 0), TOP (Copper, FR4, 0.48 mm, Copper, 4.5, 0.02), Dielectric 1 (PrePreg, FR4, 0.48 mm, Not specified, 4.5, 0.02), GND (Copper, 0.035 mm, Not specified, 3.3, 0), Dielectric 2 (Core, FR4, 0.48 mm, Not specified, 4.5, 0.02), POWER (Copper, 0.035 mm, Not specified, 3.3, 0), Dielectric 3 (PrePreg, FR4, 0.48 mm, Not specified, 4.5, 0.02), BOTTOM (Copper, 0.035 mm, Not specified, 3.3, 0), B.Mask (Bottom Solder Mask, Not specified, 0.01 mm, Not specified, 3.3, 0), B.Paste (Bottom Solder Paste, Not specified, 0.01 mm, Not specified, 3.3, 0), and B.Silkscreen (Bottom Silk Screen, Not specified, 0.01 mm, Not specified, 3.3, 0). At the bottom of the table are buttons for 'Board thickness from stackup: 1.6 mm', 'Adjust Dielectric Thickness', 'Export to Clipboard', 'OK', and 'Cancel'.

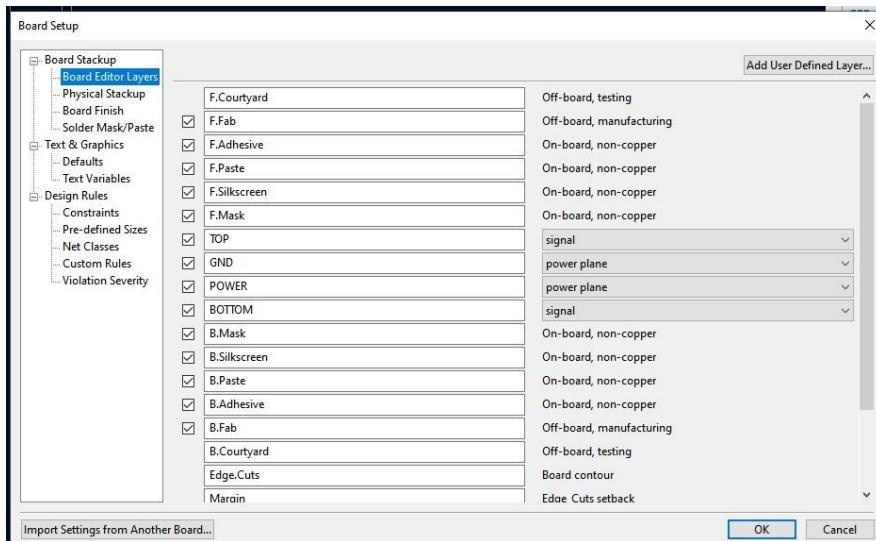
Physical stack-up section

- Select the **Physical Stackup** option
- Choose the number of copper layers using the dropdown
- If your board has any controlled impedance trace, then check the **Impedance Controlled** box.

In this section, you define the following board parameters:

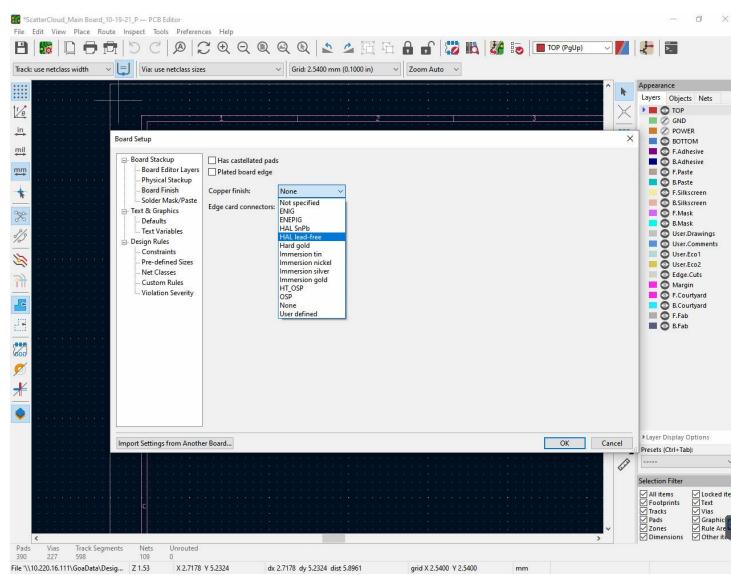
- Number of layers
- Type of dielectric layers
- Material of each layer
- Color of top and bottom silkscreen and solder mask
- Thickness of the copper layers
- Epsilon R and loss tan of the material

Now suppose you are using a differential track on the first layer, the reference layer to layer 1 should be a ground layer. This is because, on the differential pair, the ground layer should always be below the differential pairs without breaks. This is to ensure the impedance matching of the differential pair is constant over the entire length.



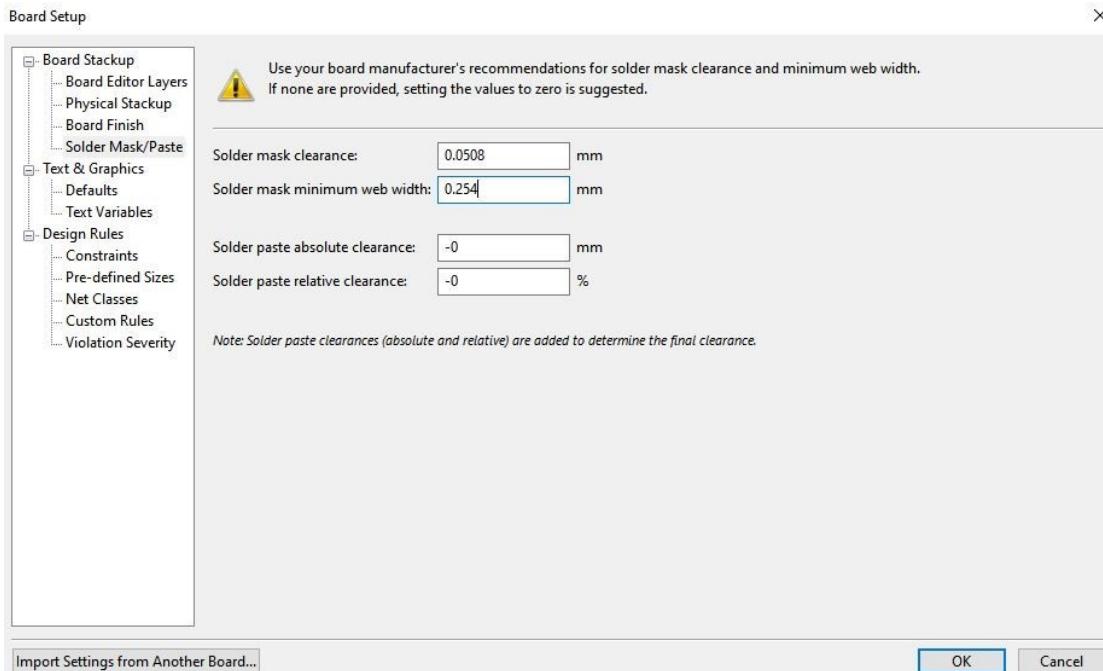
Board setup for a 4-layer stackup

The surface finish options can be viewed under the **Board Finish** section. The required copper finish can be chosen using the dropdown.



Board finish section in KiCad

Let's move on to the **Solder Mask/Paste** section. Here, you can define solder mask clearance and minimum web width.

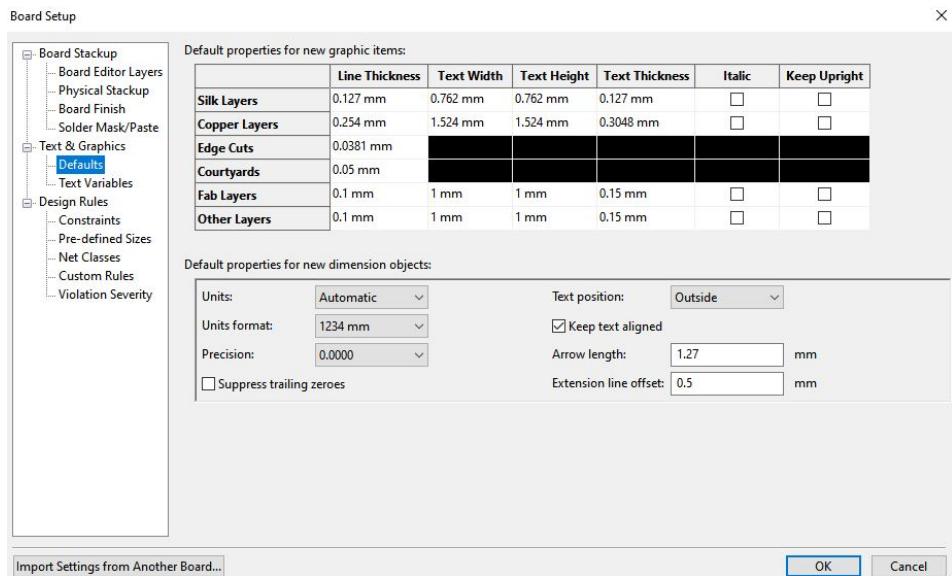


Solder mask/paste section in KiCad

The solder mask is used to cover the entire board except the pads defined in your footprint. If your pad size has a 10 mil diameter, your solder mask opening could be 14 mils (2 mils opening on each side). The solder mask clearance is the extra space that will not be covered by the solder mask.

### 3.1.2 Text & Graphics

The next section is **Text & Graphics**. This is where you can edit the text that will appear on your board. You can choose the required height, width, line thickness, etc.



Text and graphics option

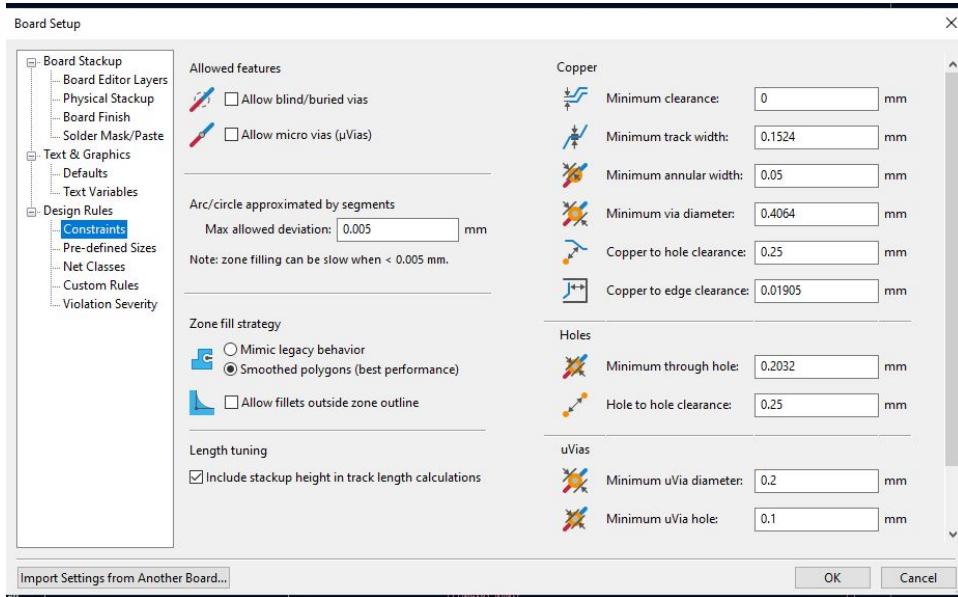
To edit the text that will appear on the board:

Select the **Default** option

Enter thickness, height, and width values

Select **Italic** and **Keep Upright** checkboxes as required

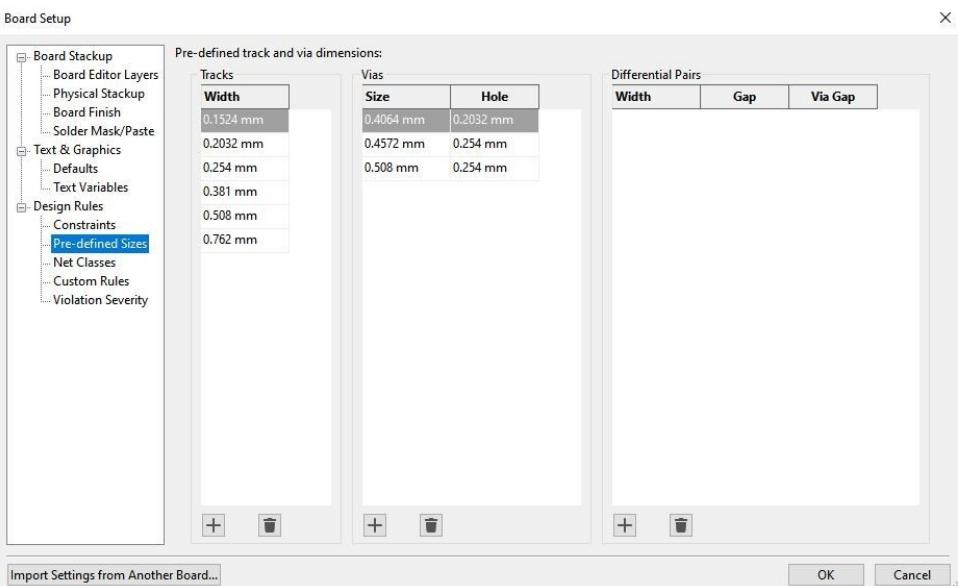
### 3.1.3 Design rules



Design rules constraints

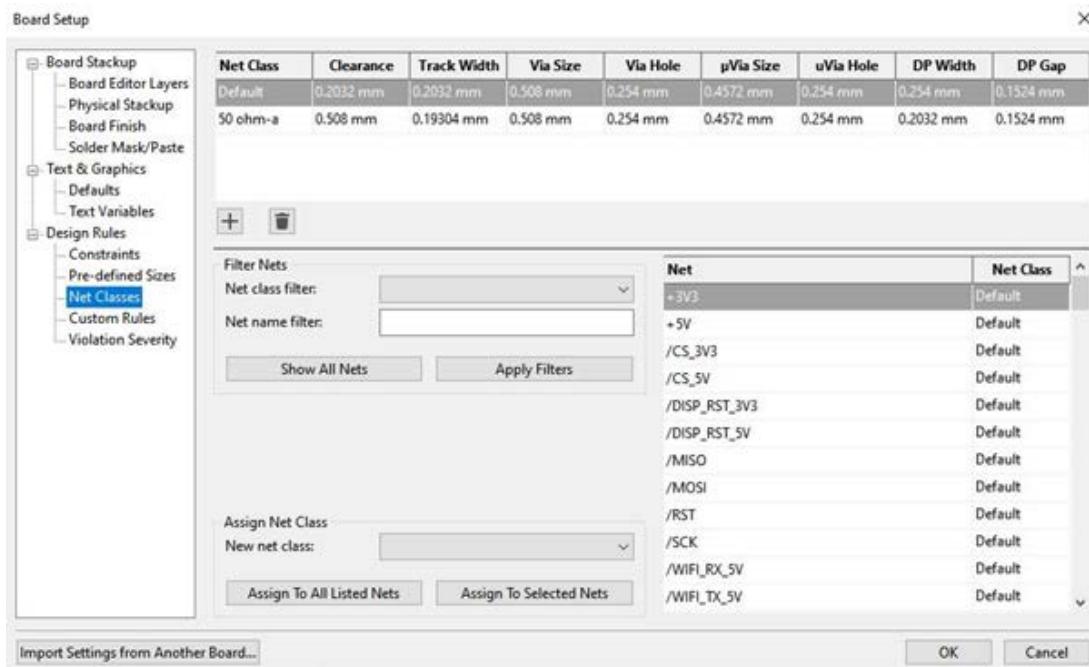
The first choice under the design rules section is **Constraints**. By selecting this option, you can view the details about the design rules set. If there are any changes to the rules you can edit them here.

The second option is **Pre-defined Sizes**.



Pre-defined track and via dimensions

Here you can view the predefined track and via dimensions.



Net classes in KiCad

Next option is **Net Classes**. You can see over here, there are different nets. And you can see there is a default class available. This defines a particular clearance. Clearance is the minimum distance that needs to be maintained between the two tracks. KiCad will handle that part once you define this minimum clearance.

You can also define the minimum track width for a particular net. A net class is a group of nets. So all these nets will be assigned to this net class. And this net class will follow these design rules in KiCad based on the minimum values you define for clearance, track width, via size, via drill, differential pair width, etc.

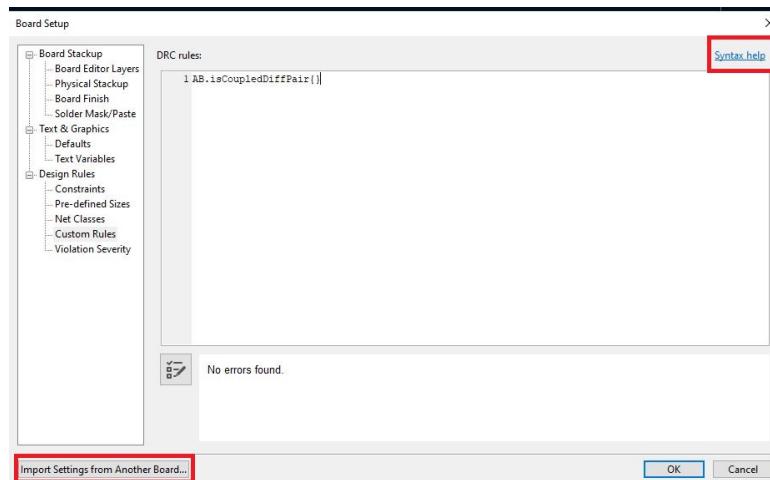
If you want to create a new net class, click the + sign. For this demo, let's create a net class for a  $90\Omega$  differential pair used for a USB.

We have created a new net class. The width of your differential pair will be defined in your **stack-up** along with the spacing. And normally, for differential pairs, the clearance is five times the track width.

These are the nets present in your circuit and this is the net class. Imagine you want to assign **Net-(BT1-Pad1)** and **Net-(C2-Pad1)** to some other class, click Default under Net Class, and select the class among the available ones.

If you want to assign **Net-(BT1-Pad1)** to the  $90\Omega$  differential pair net class, this net will now follow all the design rules in KiCad assigned for this class.

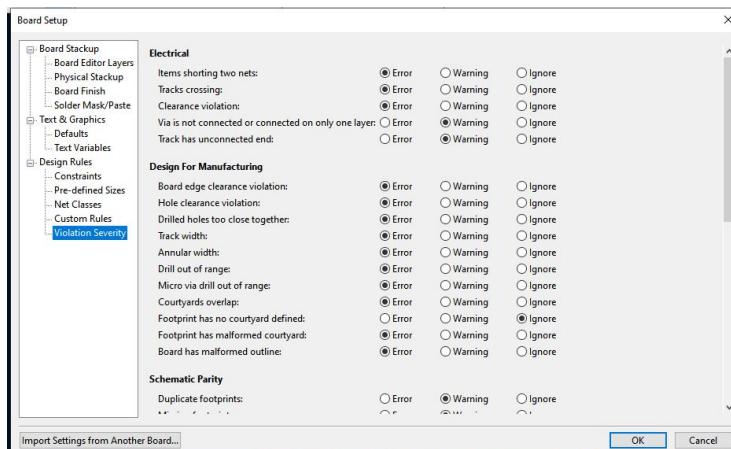
If you have multiple differential classes, each of the differential pairs should follow different track widths or different spacing. You can create multiple classes and define the clearances for each single-ended line and each differential pair.



Custom rules

If there are any customized rules applicable to your design, you can define them under **Custom Rules** section. Click on **Syntax help** if you require any assistance with regards to syntax.

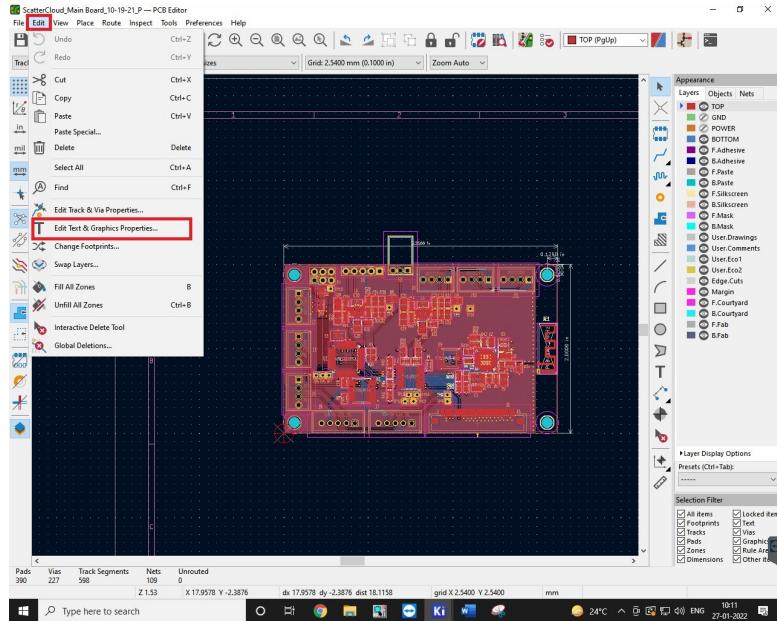
Under **Violation Severity**, you can define DRC and ERC rules for the schematic and the board. Each of these rules can be set to error/warning/ignore status. If a rule is set as error, any violation would result in error in ERC and DRC. Similarly, when we set a rule as a warning, the violations will display a warning message. A rule can be ignored by setting it to ignore status.



Violation severity in KiCad

Once the design rules are set, you can view them at the top of the screen.

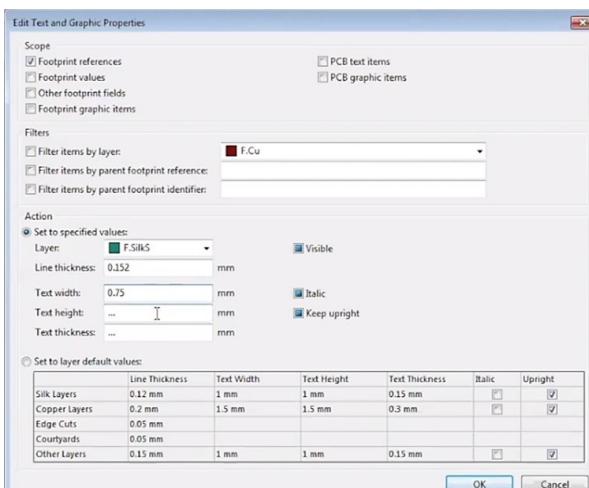
### 3.2 Modifying footprint references and grid selection



Modifying footprint references

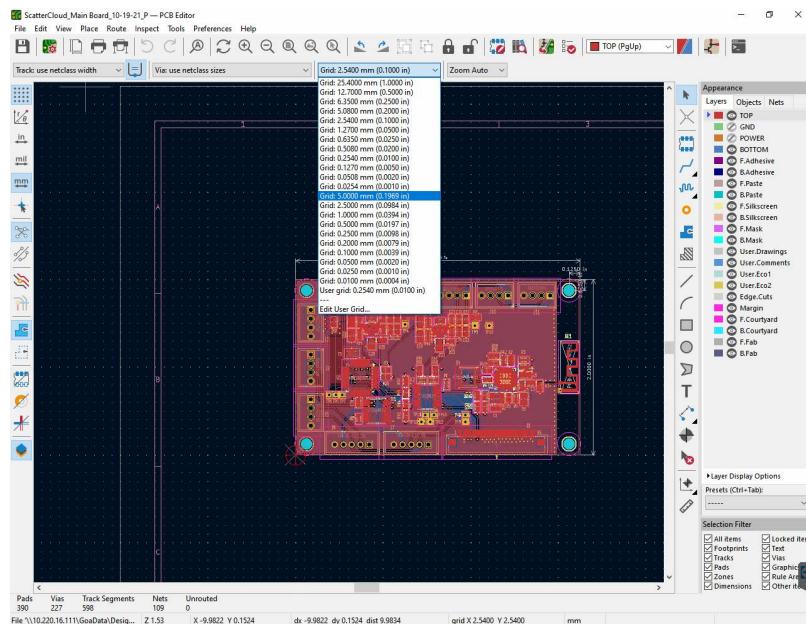
Follow the below steps to modify footprint references:

- Click **Edit Text and Graphic Properties under the Edit tab**
- Select layer
- Enter line thickness (0.152 mm = 6mil)
- Enter text width, height, and thickness at 0.75mm = 30mil
- Click **OK**



Edit text and properties menu

At the top, there are grid options. The grid will help you with the accuracy of your design. It is the distance between these two points. While designing, it's best practice to keep it at 5 mils. But if needed, you can click **Edit User Grid** and define your own values.



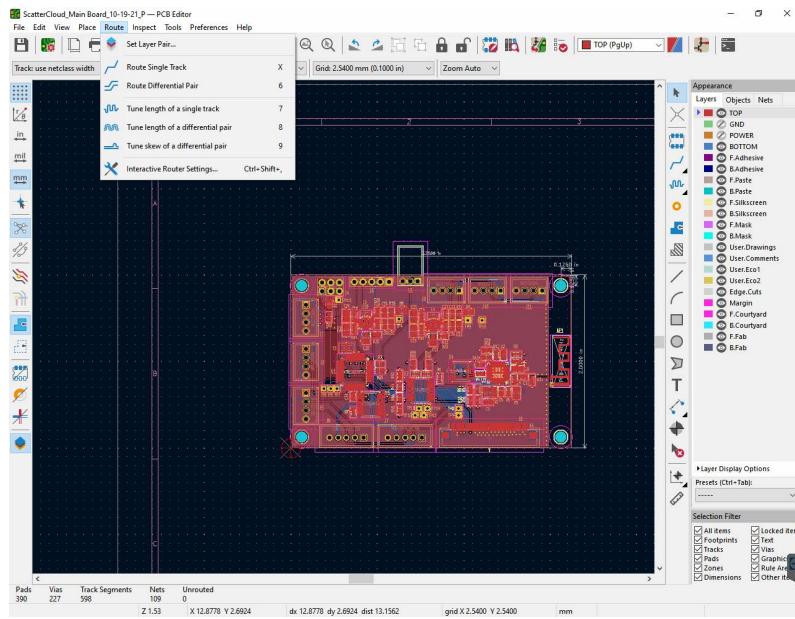
Grid selection in KiCad

### 3.3 The differential pair design rules

This circuit has a differential pair. There is a USB connector and two tracks are running from there, **D-** and **D+**. And there is another differential pair, **D1-** and **D1+**.

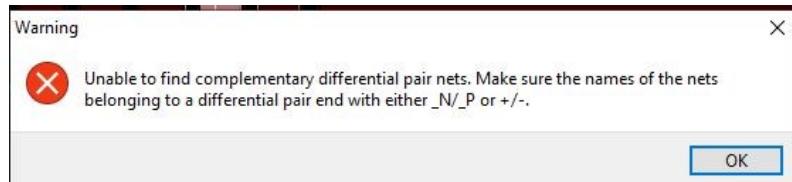
If you want to define a pair as a differential pair, select **Place Net Label** on the right-hand side and click the screen. Whatever name you choose, it has to end with + and – or \_P and \_N.

This is very important for differential pairs. When you start the routing, if KiCad notices a D-, it will search for the D+ and consider these two tracks as differential pairs.



Differential pair routing

Now if you want to route two differential tracks, click **Route** at the top. You can see the option for single tracks, which is already available on the right-hand side, and there is the differential pair option. When we click it, a window pops up and mentions it is unable to find complementary differential pair nets. This is why you need to make sure the labels end with – and + or \_N and \_P.



KiCad Error Message for Naming Differential Pairs

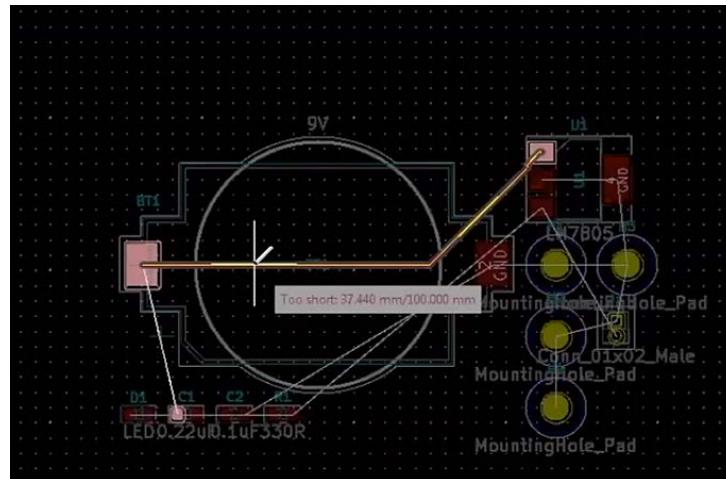
There are two different differential nets, and their length should be the same.

Under **Route**, there is an option called **Tune Differential Pair Length**. We can show you right now how it works for a single track. We select **Tune Track Length** and then this red track. A note appears and reads 'Too short: 37.440 mm/ 100.000 mm', which means that we have set the target length as 100 mm, but the actual length of our track is just 37.44 mm. This is very important in DDRs, for instance, where you use address lines and data lines. There are eight different address lines and eight data and their length should be equal.

### 3.3.1 Differential pair length tuning

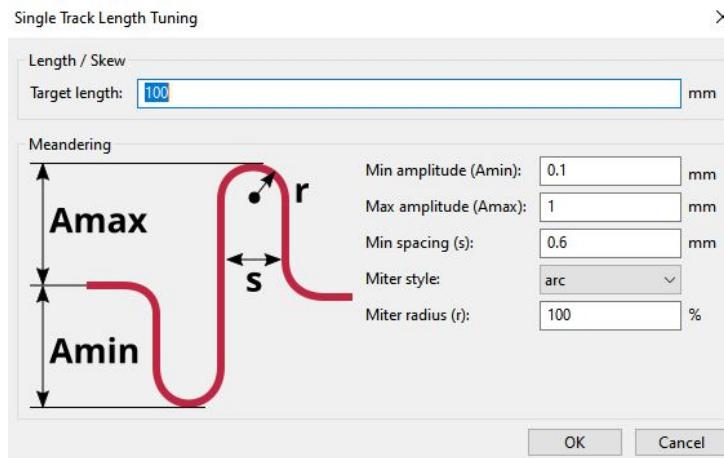
If you want to change your target length, right-click the track and select **Length Tuning Setting**. As we just saw, the target length is 100 mm. We will see what happens when we write 40 mm. Meanwhile, don't forget that all these parameters are very important in differential pairs. For now, we will just show you how the target length can be achieved.

So we set the target length to 40 mm. It is still indicating 'Too short: 37.44 mm/40 mm.' Now we want to make it 40 mm.



Setting track length

So place your cursor on the track and move it. You can see a small S-shaped track appearing. This is how you increase the size of your track.

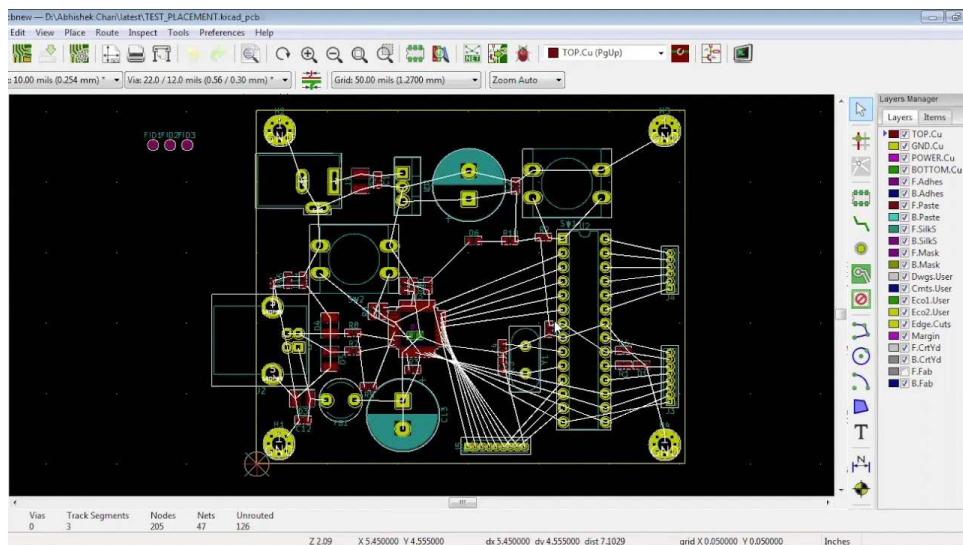


Single track length tuning

And the note now reads 'Tuned: 40 mm/40 mm', which means the length is exactly 40 mm.

KiCad is a versatile tool for electronic circuit design and corresponding PCB design requiring familiarity and expertise. This guide on setting up KiCad for your PCB design will help you design faster and better. Good luck with designing!

## Section 4. Component placement in KiCad

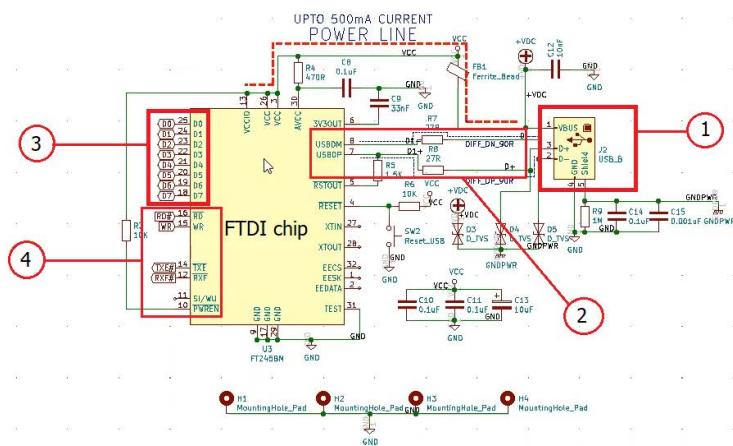


Component placement in KiCad

In this chapter, we will show you how to place components in KiCad and also cover how to define your PCB outline.

### DEMO VIDEO - How to Place Components in KiCad

Here we have created a circuit that converts and transfers data from USB to controller. FTDI chip is used to convert the differential pair output from the USB circuit to 8-bit data, which is fed to the controller. The process is controlled by the **Atmega328P controller**. Circuits are shown below:

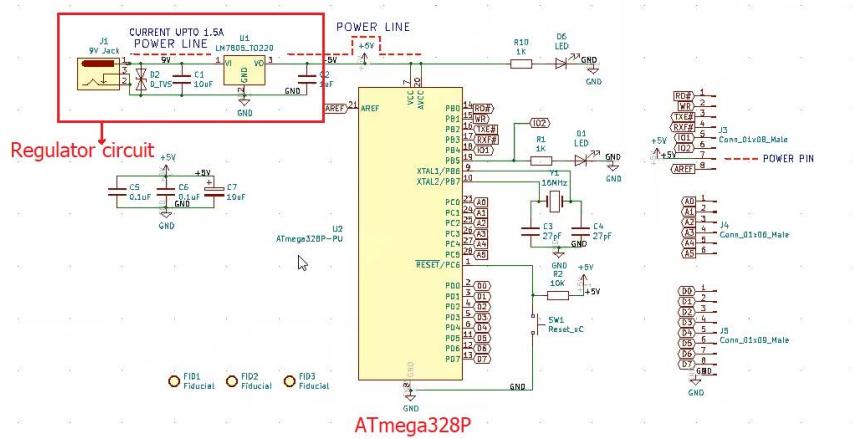


Circuit that converts and transfers data from USB to controller

The important sections are marked in the above picture,

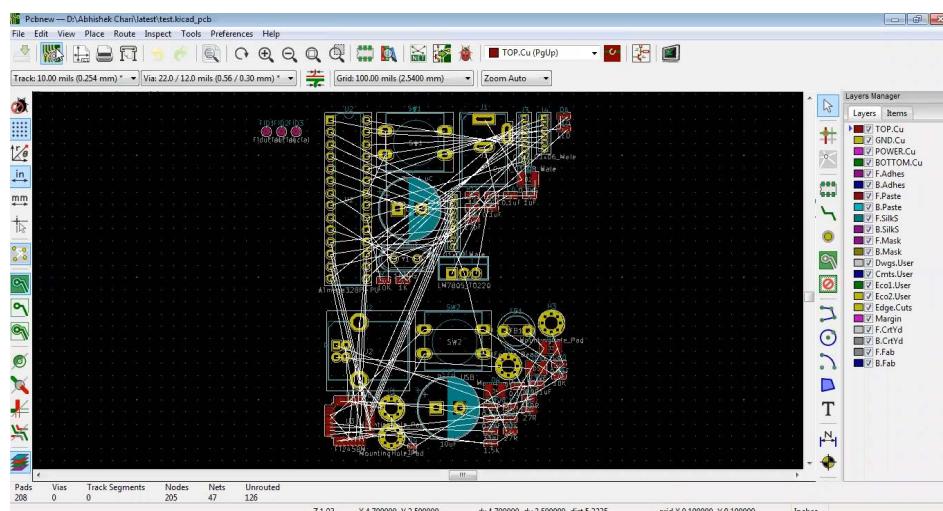
1. USB connector
2. Differential pair output connection between USB to FTDI chip
3. 8-bit output pins
4. Control pins

The controller circuit is shown below:



Controller circuit of the USB

Let's now assume that this is the circuit we are working on. Here, you can see a USB connector and a differential pair that is going to this chip. These eight data pins and four control lines here all go to the controller. The controller is powered using a 9V input that is converted to 5V using a regulator. We have already created a netlist using the Generate netlist option. From our netlist, we have imported all the components in the board file.

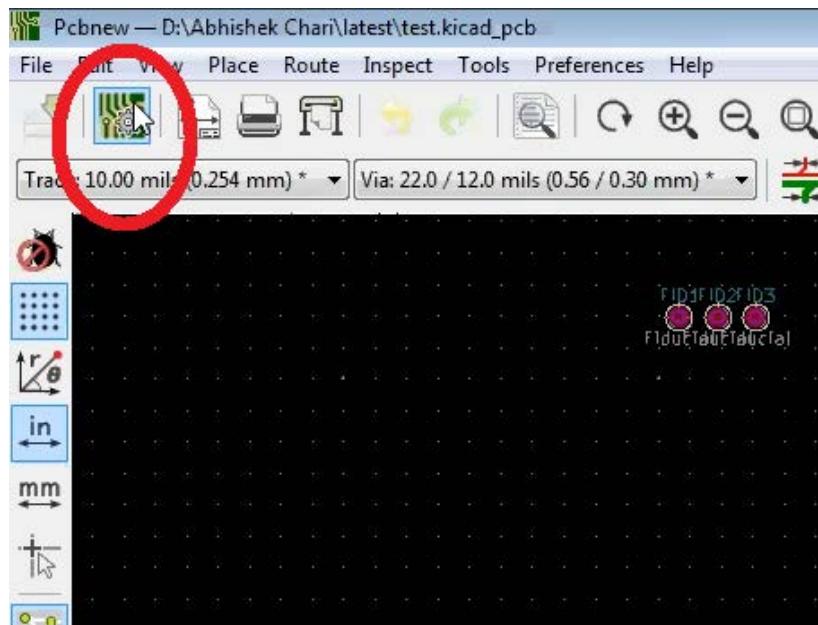


All components from the schematic

These are all the components from the schematic. We have set all the rules, such as the track widths and vias we will require. Right now, the grid is set to 50mils but it should be at 5mils, which is normally the grid setting when you need to place components in KiCad. Before doing the placement, we need the board outline, so let's define the edges of the board. Keep the grid at 100 mils so it is easy to draw the outline.

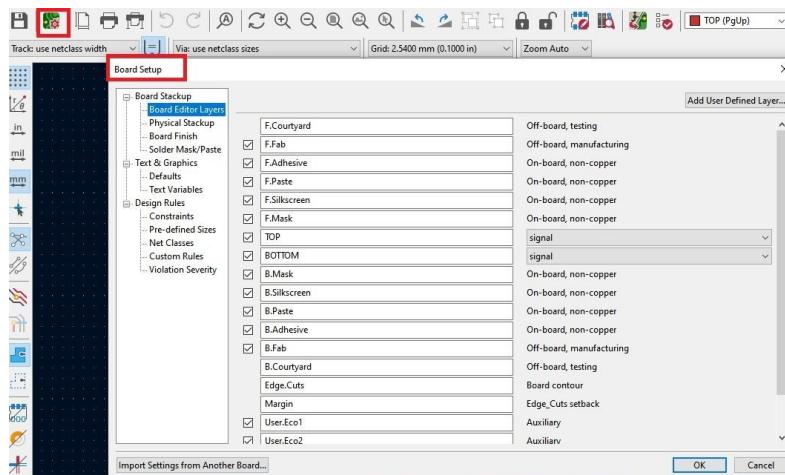
To set up your board, please follow the procedure:

- Click on the board setup icon to set your board

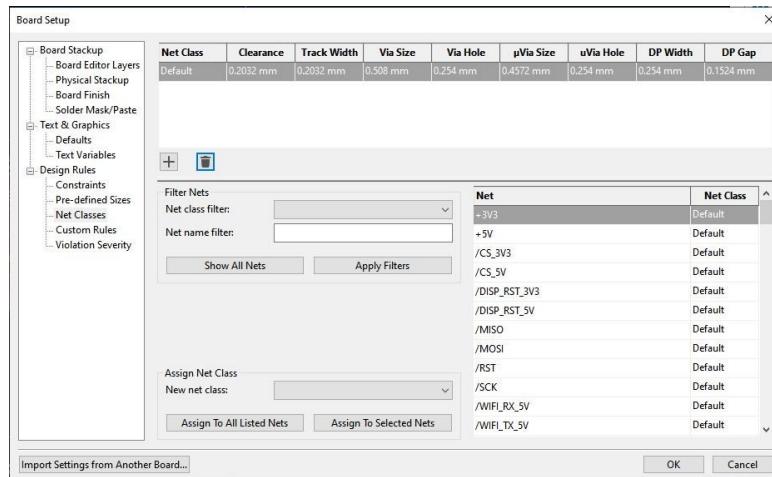


Board setup icon on the top toolbar

- Configure your text, net classes, tracks, vias, and solder mask here.



Board setup window in KiCad



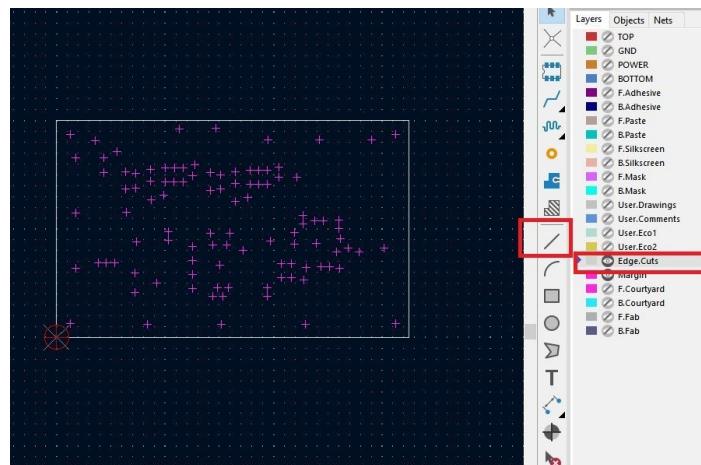
### Configuring text and design rules

We have selected a 62mil thick 4-layer board. Layer 2 is ground, Layer 3 is power, and the top and bottom are signal layers. In Net Classes, we have set the differential pair widths and gaps to 8mils. There are two types of power: USB power and regulator power. And we have set all the different net tracks over here. We have set the solder mask clearance to zero. We are now good to go with the placement.

## 4.1 Drawing the board outline

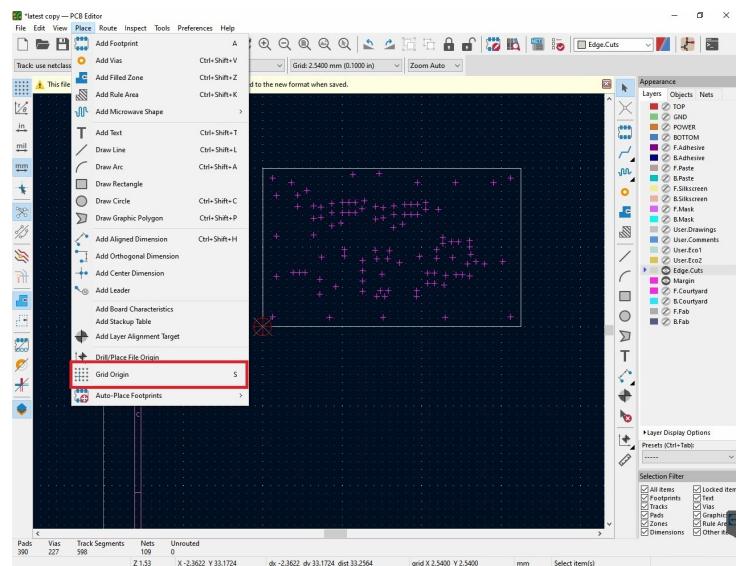
Before we show you how to place components in KiCad, it is necessary to draw the board edge. On the right-hand side, click **Add Graphic Lines**. Now you can see all the board layers. Select the edge cuts layer. You need to draw the exact board edge that will be cut during manufacturing. So, you have selected the edge cuts layer and the grid is at 100mils. Now click the screen and start drawing. Make sure your line is straight. You now have defined the borders. This will be your actual PCB.

On the right-hand side, click Add Graphic Lines > Select Edge Cuts from the layers > Draw outline



Graphic lines in KiCad

Now, you need to set the origin on the left bottom of the board. Go to **Place**, select **Grid Origin**, and place it at the left bottom corner. The first placement task is placing the mounting holes. There are four mounting holes in this schematic: H1, H2, H3, and H4. Let's go back to the board to begin the placement.



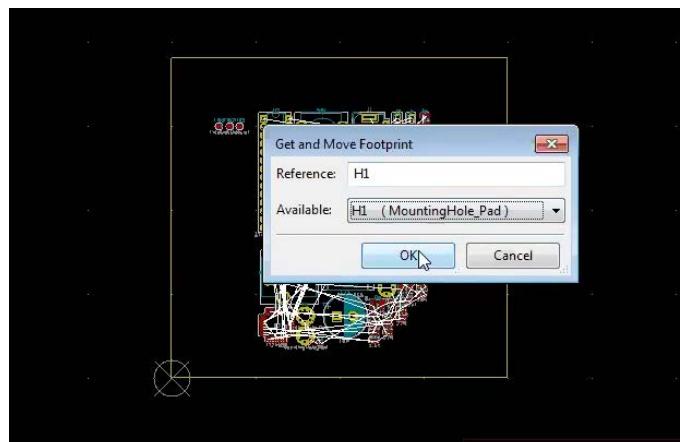
Grid origin

## 4.2 Component placement

### 4.2.1 Mounting hole placement

There are two ways to find components. You can press key T on your keyboard and a pop-up will appear. Type your component name or select it from the list. Another way to find your component is to press CTRL+S. Type the name and the component will be highlighted. If you want to select it, press key T. You can now put all the components outside of the board to place them one by one.

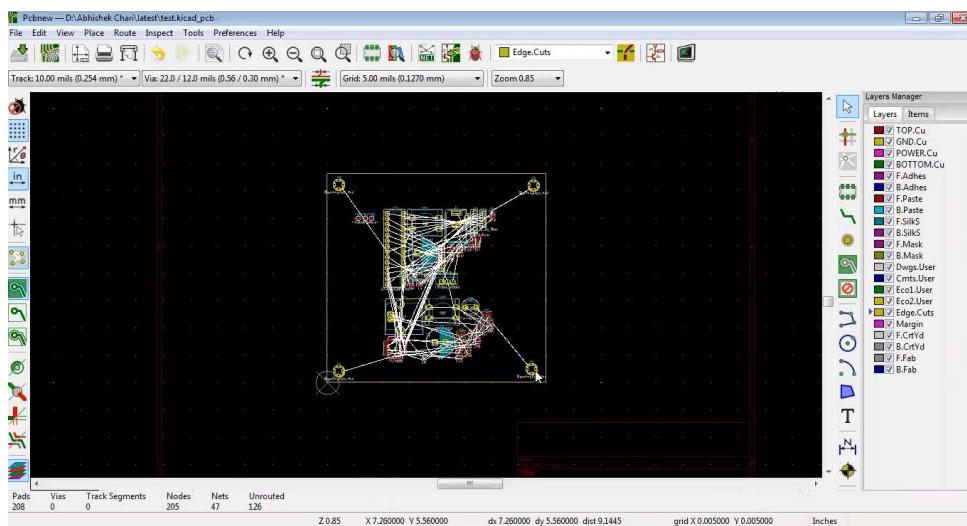
**Press T on your keyboard > Select your component**



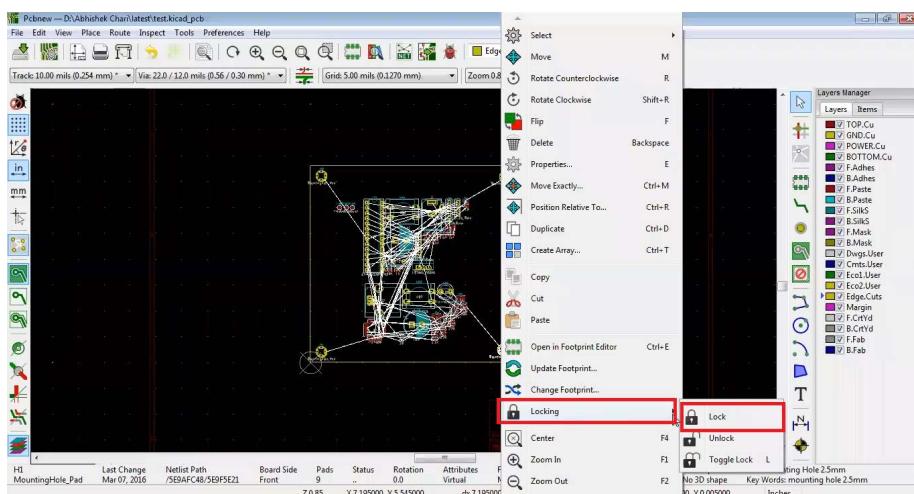
Finding components

There are four mounting holes in our schematic: H1, H2, H3, and H4. So these mounting holes will be present among components, we have to search and find them.

Change the grid to 5mils. Press key M and place the mounting hole in the corner. We are randomly placing this M1 but mounting holes should have fixed locations. And repeat with the three other mounting holes. Once placed, the mounting holes shouldn't move. Click them and select the footprint option. Now, right-click and select Locking and Lock. The mounting hole is fixed. If you try to move it, a pop-up will warn you about the locked item.



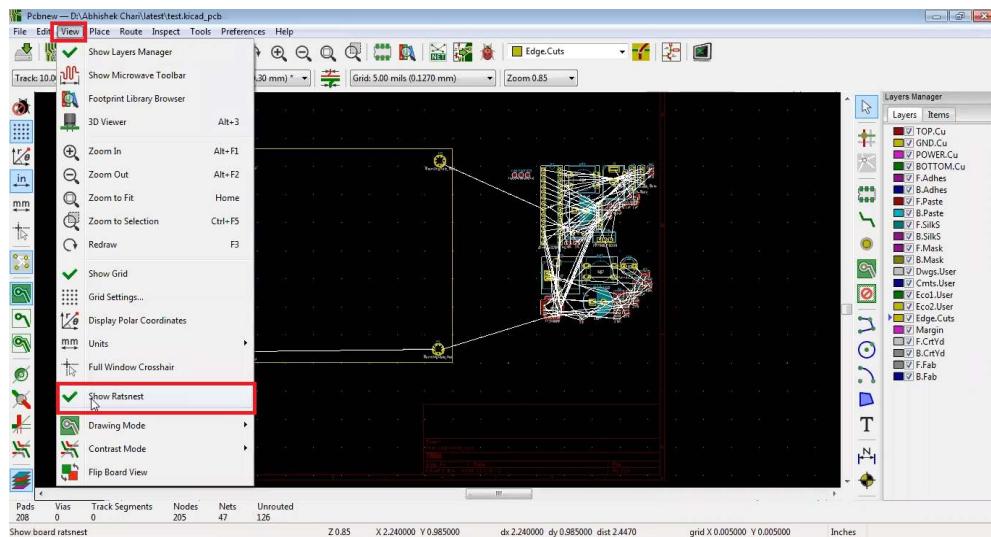
To lock your component, right-click on the mounting hole > select Locking > Lock



Steps to lock the components

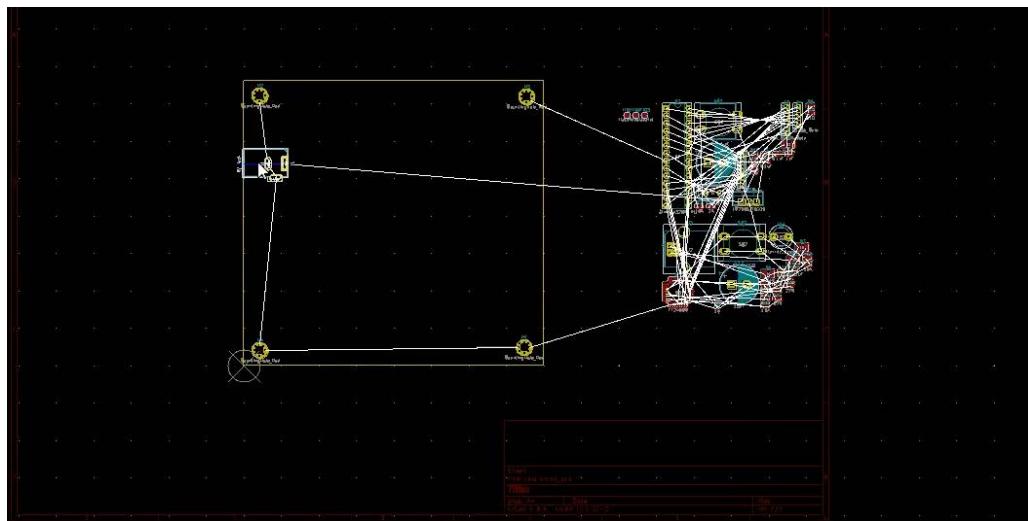
We will place the components one by one on the board. To do that, first, we must put all the components outside of the board. Select and drag all the components out of the board.

You can see there are many connections. If you find it confusing, you can turn off these wires. Go to **View**, click **Show Ratsnest** and the wires will disappear. We prefer to keep them as they indicate the connections.



#### 4.2.2 Connector placement

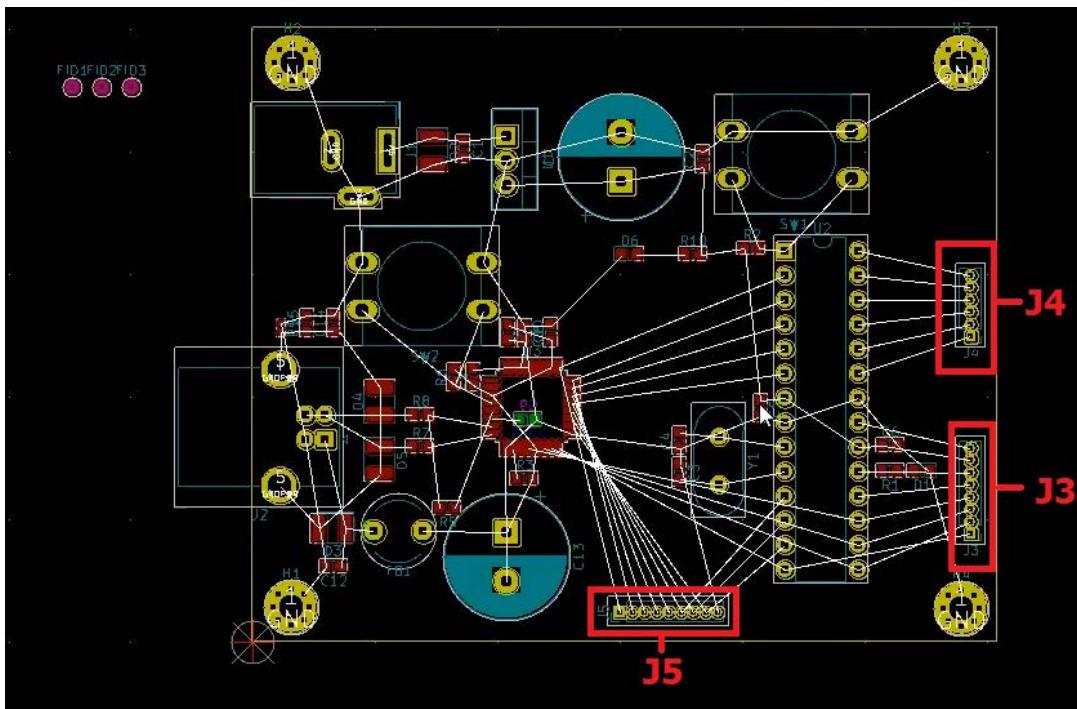
After mounting holes, the next step is to place the connectors. Because connectors should be in a particular direction as per the device design, which cannot be compromised, so we choose the connector next.



Connector placement

The next step is to place the connectors. Always refer to your schematic while you place components in KiCad. This is the 9V Jack connector, which is the **J1** connector, and we want to place it first. Let's go to the board again. Press key **T** and type **J1**. Once **J1** is selected, press key **R** if you want to rotate it, and place it. After placing the connectors, it is a best practice to lock them. Right-click, choose Locking and Lock.

#### 4.2.2.1 USB connector



USB section in the schematic

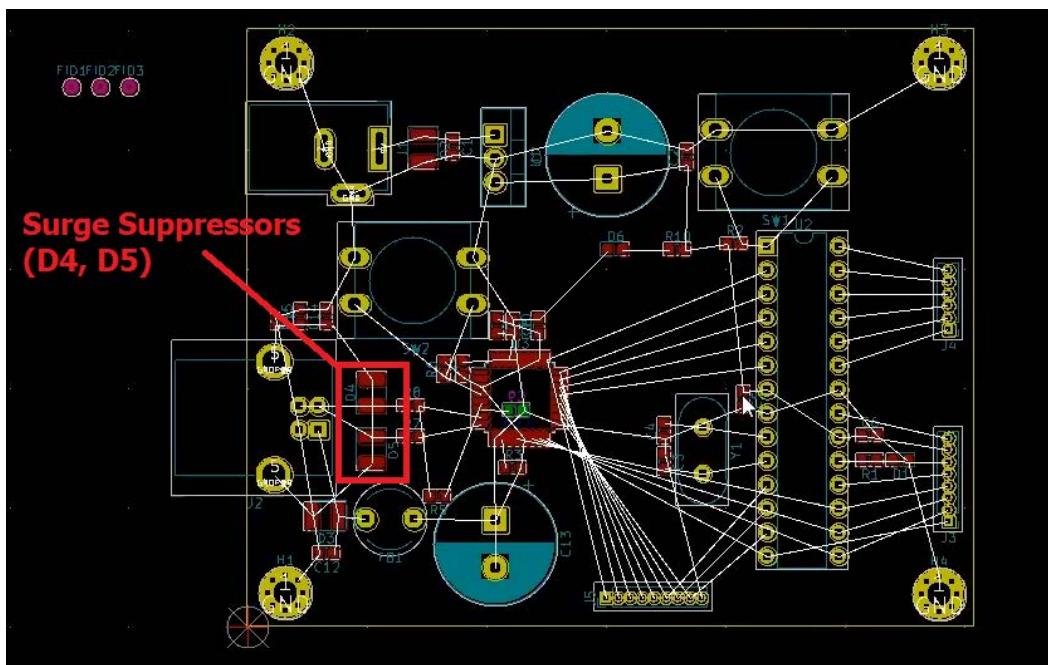
This is the USB section. These are the three connectors we were talking about: J3, J4, and J5. The USB connector consists of six pins. But pin number six is common for two pins of the shield of the connector. There are also voltage suppressors for the two differential pair inputs and one is synced to the main supply voltage, Vbus.

These three should be close to this connector so that it doesn't allow the unwanted voltage to enter into the circuit. And we have these two resistors. This particular connection is important: this is a differential pair working at high-frequency, at around 12MHz.

This differential pair should be routed properly, with proper length matching, meaning the skew should be minimum between the two lengths. This is an important parameter to consider while placing.

#### 4.2.3 Placement of the surge suppressor

This is the USB connector, and these two pins are the differential pair inputs. One pin is going over here. This is the surge suppressor, right after the connector. This will be followed by another trace. This is also a suppressor right after the input. And here, this pin is VCC which is also going to the surge suppressor. You need to place the surge suppressors first, and after that, the differential pair so it would be in a straight line. These two lines are going through the surge suppressor to the two resistors.



Placement of the surge suppressor

The next step is to follow the sequence of the supply voltage. Before coming into the circuit, the Vbus needs to go through this surge suppressor. Then, the capacitor should be close to the Vbus before going to the ferrite. After that, we want to place the capacitor C13 over there, and C11 and C10 over there by pins 3 and 26.

The capacitor, right after the surge suppressor will go to the ferrite bead. And after that, it should go to the capacitor.

This is a resistor and this is a capacitor. You can see these two points, this is R4 and this is C8. R4 is drawn from VCC and then is running to AVCC. But before going to AVCC, it should first pass through C8.

This is a capacitor and this is a resistor. This point is coming through VCC, through this capacitor, and into the input of the microcontroller. You need to follow the circuit. Once the placement is finalized and everything is properly maintained as per your requirements, you can start routing.

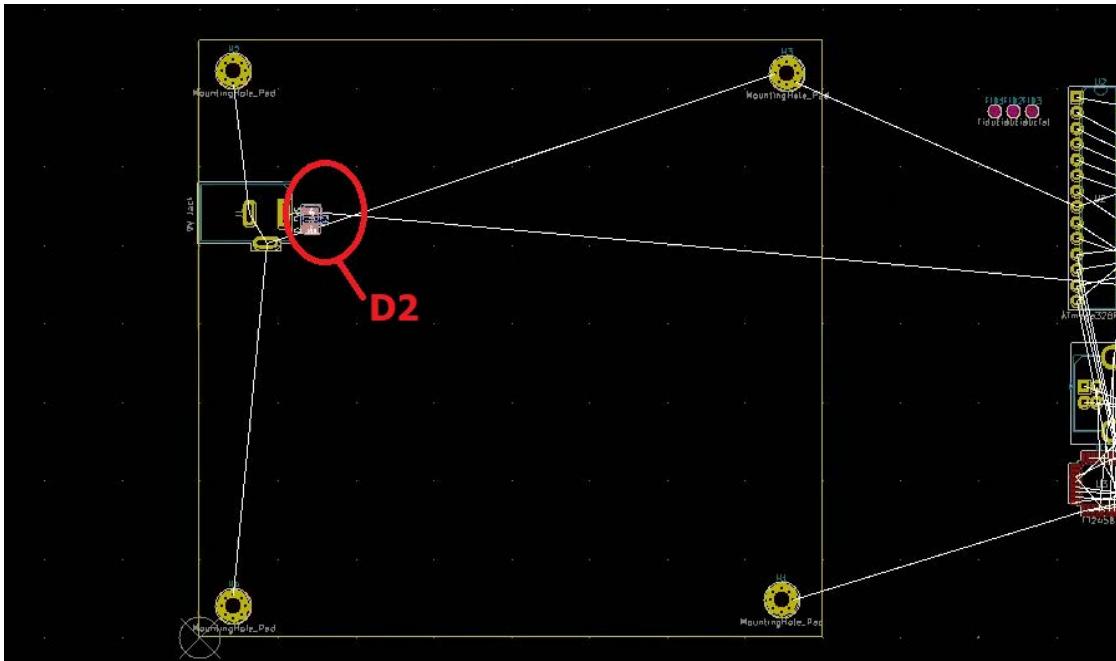
#### 4.2.4 Placing the auxiliary components as per signal flow

##### 4.2.4.1 Capacitor placement

Again, always refer to your schematic while you place components in KiCad. You can see that there are capacitors connected to five volts traces.

We have placed these capacitors on the side in the schematic so this doesn't get too congested. These capacitors will be placed at the output of this regulator. And these two capacitors are used as decoupling capacitors for pins, 7 and 20 of the microcontroller. This capacitor will be placed at the output pin of this regulator. And the surge suppressor needs to be placed near the connector because it is used to suppress any surge voltage coming into the circuit. As you can see, you just have to go with the flow of the schematic.

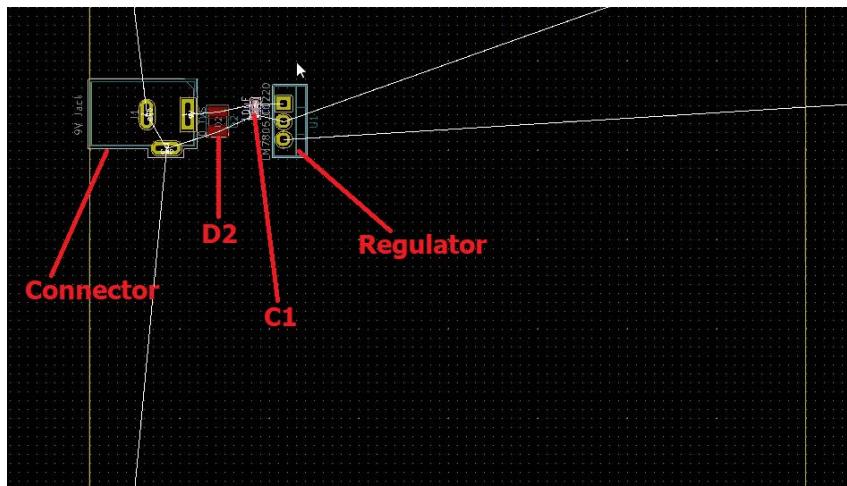
So, as per the schematic surge suppressor (**D2**) is the next. Now search for D2. Press **R** to rotate it and place it in the right place.



Placement of auxiliary components

Let's now place the capacitor **C1**, which will be at the input pin of the regulator. Find **C1** first, and place the regulator near to this capacitor, or place the regulator first and then place the capacitor near to this pin. Select the regulator first, rotate it, and place it here. Now search **C1** and place it close to this pin. You can see this is the input pin of your regulator and we have placed the capacitor close to this pin.

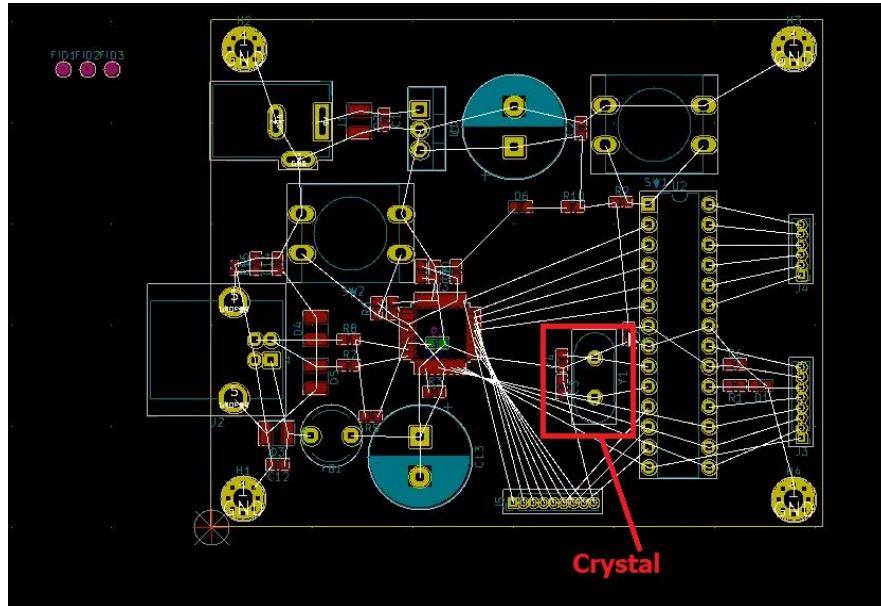
In this way, we have placed the connector-regulator circuit as per the schematic.



Connector regulator circuit

This is how you can place components in KiCad while following the schematic. Remember that you have to keep your power, analog, and digital sections separate so they don't interfere. As you can see, this is the **J1** connector, this is the suppressor, this is the capacitor close to this pin. **C2** is at the output of the regulator and it is **C7** here. This is the electrolytic capacitor and this is **C2** which is at the output of the regulator. There were two capacitors, one each near pins 7 and 20. Place the capacitor close to the input pins. And there is one more capacitor that is placed over here, this is pin 21. You can see this connection, both are the same nets. So these two capacitors are placed close to the pins.

#### 4.2.4.2 Crystal placement

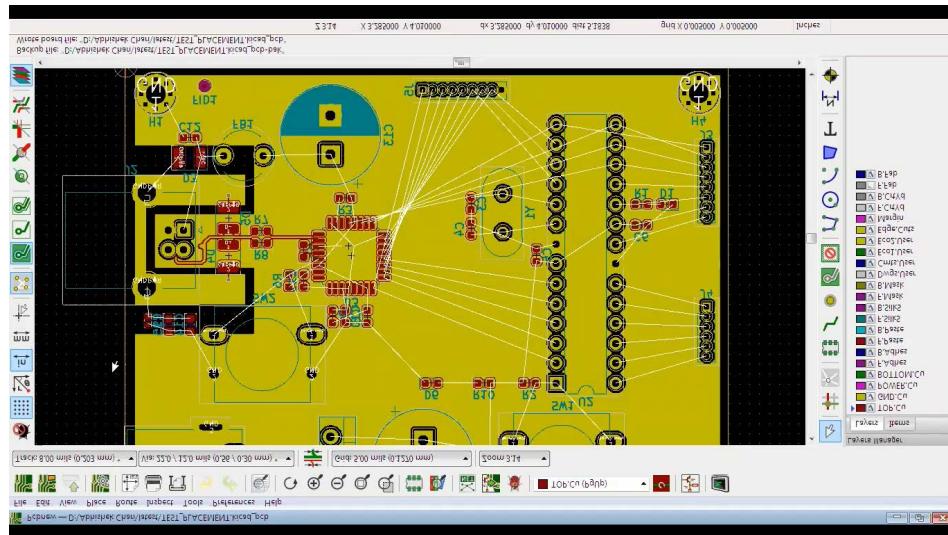


Crystal placement

The next step is the placement of the crystal. It is best to place the crystal as close as possible to the micro crystal input pins. These two pins are nothing but crystal input pins, this is the crystal and there are two capacitors. There are two capacitors connected to the 16-megahertz crystal. Those should be close to the crystal. And you can see, there are three more connectors in the circuit. Here you have a connector, a second connector, and a third connector.

All of these lines are input-output pins of the regulator. Ensure that no two connections overlap each other, or it will create problems while routing. To avoid this, it is recommended to pre-plan your connections so that when you place components in KiCad, it won't affect your routing.

# Section 5. Routing a PCB in KiCad



PCB routing

## 5.1 What is routing?

Routing is simply the process of wiring between the components placed in the PCB design while following all the design rules. The design rules vary depending on the type of signals traversing in traces and sometimes rules vary layer to layer.

Let us go through the PCB stack-up before we get to the PCB routing.

**DEMO VIDEO - [How to Route a PCB in KiCad](#)**

## 5.2 Arranging stack-up

We selected an 8mil track width and spacing for the differential pair. But this needs to be done correctly, according to the stack-up that contains certain requirements like **impedance**, number of layers, the thickness of the PCB, etc. Depending on the requirements you can generate a stack-up.

Try our [Stackup Designer tool](#) to easily create your PCB stack-ups.

You can see in this stack-up a  $90\Omega$  impedance on the top layer 1 and layer 4 (signal layers). The outer dark green layers in the stack-up are the top and bottom solder mask. The yellow layer represents the core and the light green layer the **prepreg**.

The 0.5 oz copper foil comes with a 0.5 oz plating, so in total it will be 1 oz. The outer layers are 1 oz and the required finished thickness is 62mils. The estimated thickness of the lamination is 59.79mils with a tolerance of +/- 6.2. The top and bottom layers have a 90Ω impedance and a differential line width of 5mils. You can see that the finished line width is 5mils for layer 1 and layer 4 and the finish spacing is 4mils.

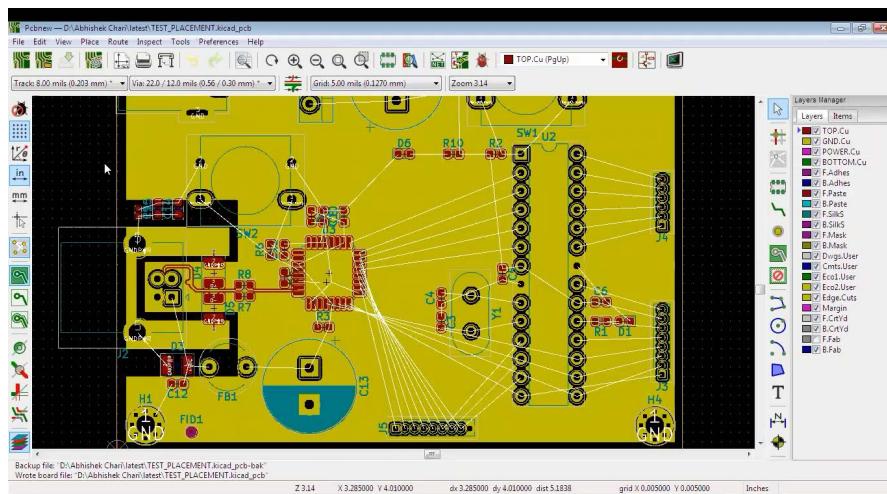


### Sierra Circuits stack-up and impedance report

The differential pair width should be 5mils and the spacing between the two tracks should be 4mils.

## 5.3 Critical routing

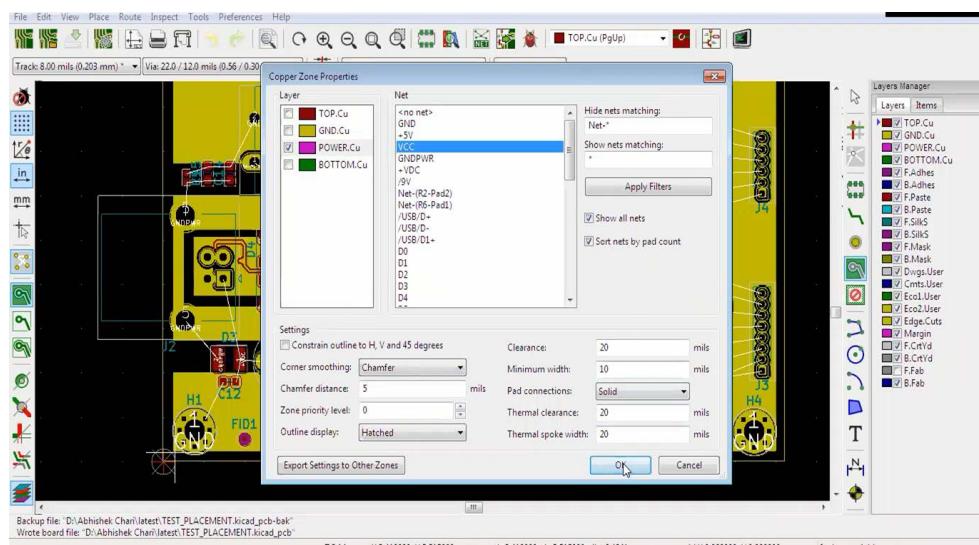
We will start what we could call the critical routing.



Critical routing

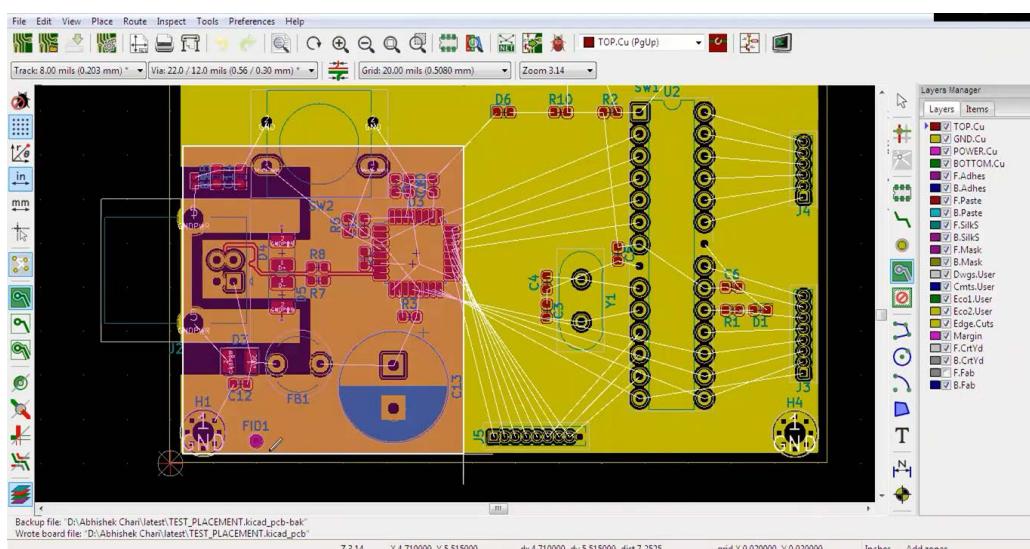
We want a plane on layer 3 which is a power plane and we want a plane of net VCC.  
To draw a plane:

- Select **Add Filled Zones** button on the vertical toolbar
- Click from where you want it to start.
- Select Power plane **"POWER.Cu"**
- Select a chamfer of 5mils
- Choose Pad connections as **"Solid"**
- Click **Ok**



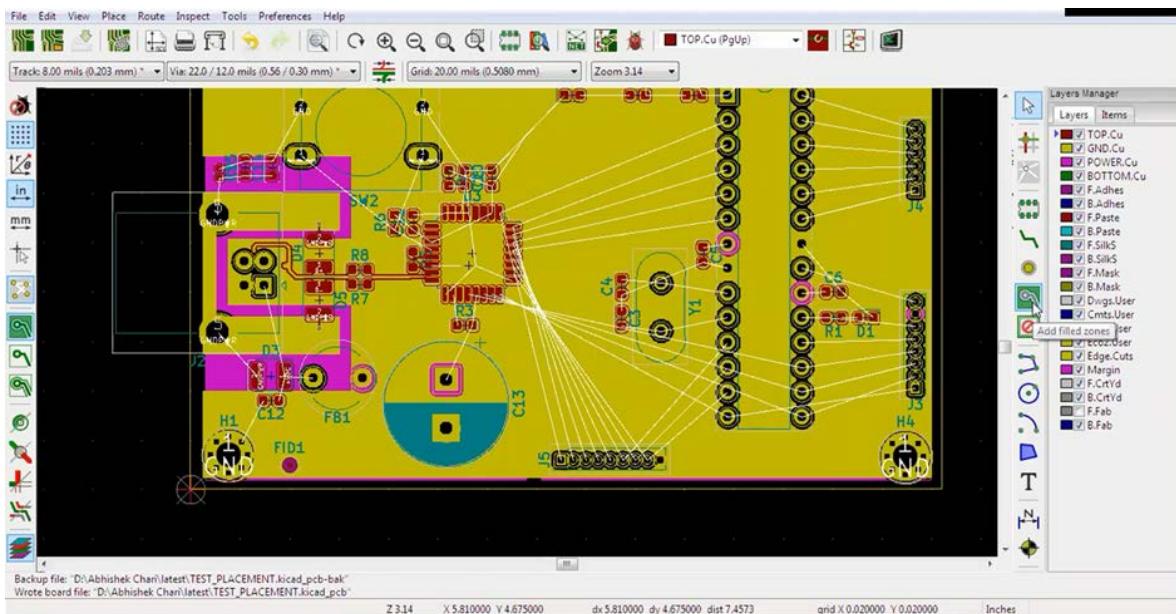
Specifying a plane

We are drawing a plane in the left bottom side because the VCC is present in this section. We have to define +5V for the remaining area on the third layer. The planes are now set up for VCC and +5V.



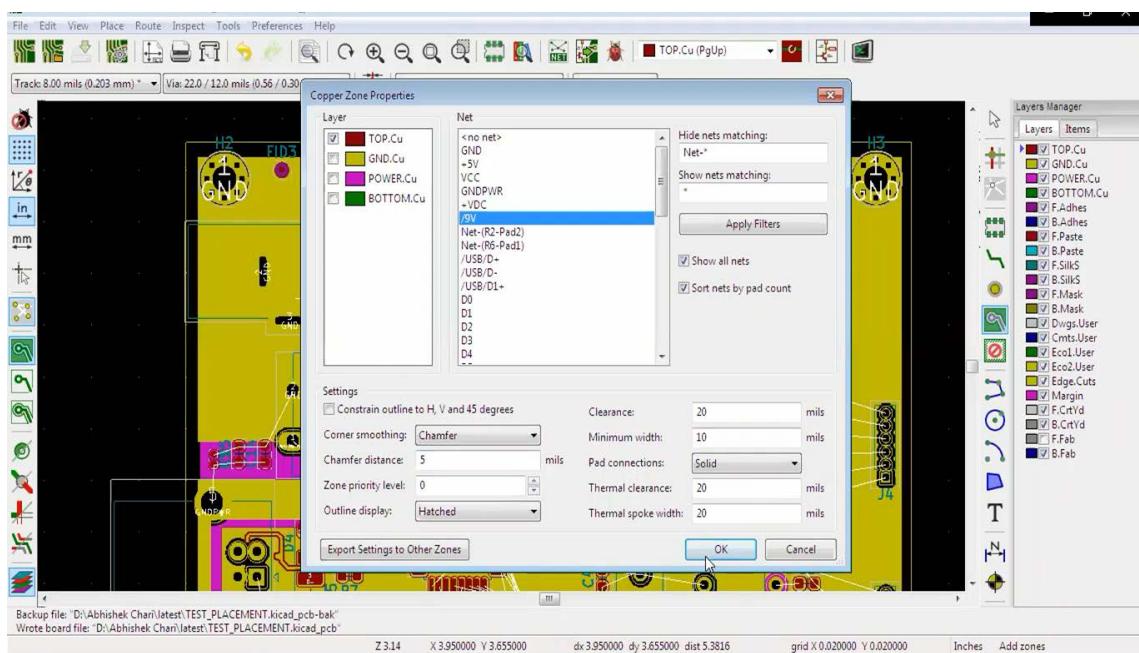
Drawing a plane on circuit layout

Let's route the main tracks, which are power tracks of 9V and +5V that run along this line to this point. This is the VCC point coming from the connector towards this point and this one is the VCC running into the circuit.



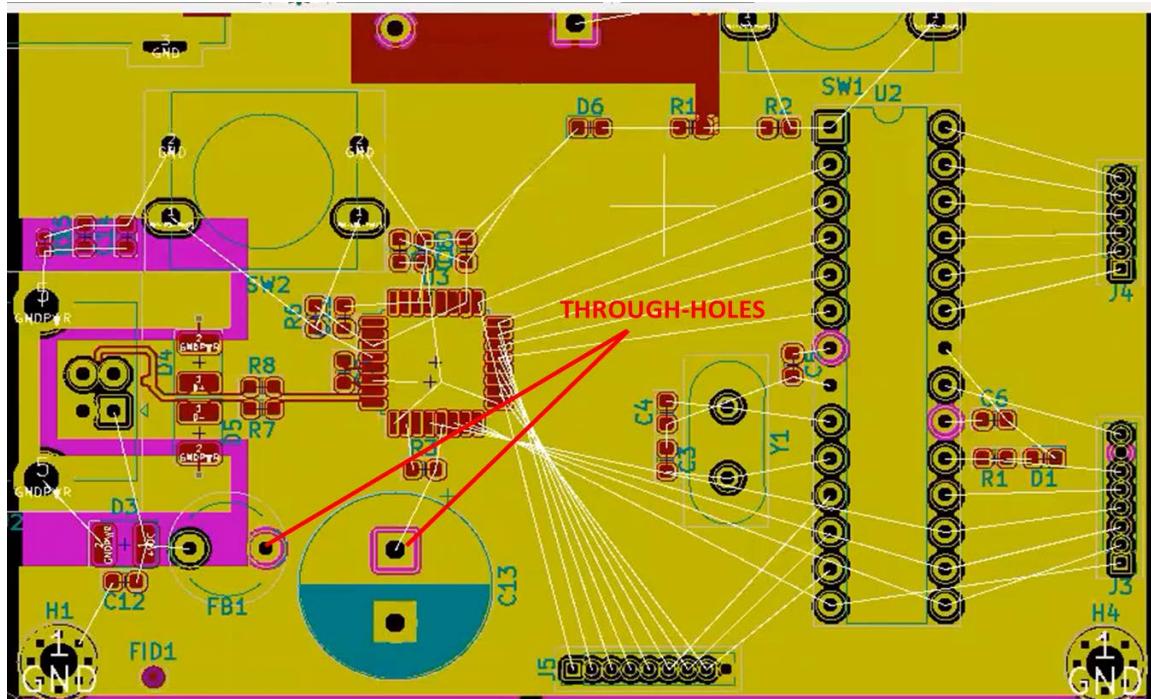
Adding filled zones

Select **Add Filled Zones again**, draw a complete plane on the layer because this is just a 9V and it will be running only in this region. You don't have to draw a special plane on layer 3, you can directly draw it on layer 1.



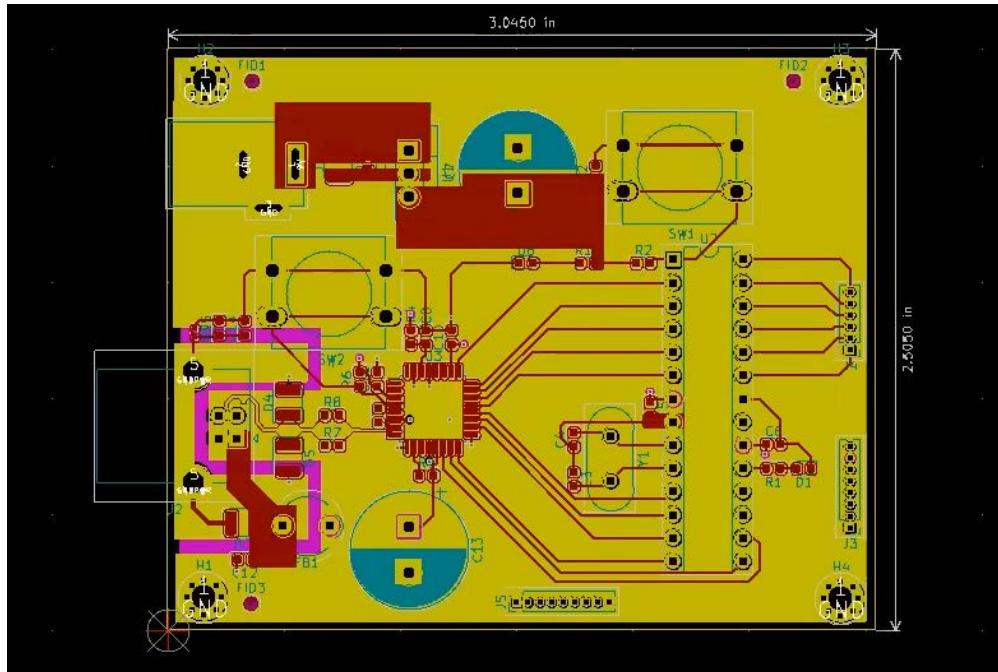
Drawing a plane for +9V

We are drawing a plane in this region because this is a power plane and it will carry high current. Similarly, we need to draw a plane for +5V.



FB1 and C13 connected with through-holes

As shown in the image above there is a connection between **FB1** (ferrite bead) and **C13** (capacitor) but you can't see any line over here as we have created a plane on layer 3 and this is a through-hole component. Through-hole components normally connect to all the layers. Since there is a plane on layer 3, **FB1** and **C13** are connected through layer 3. The C13 through-hole that is going through layer 3 is connecting to **FB1**. It is always a good idea to have a plane because it improves the flow of the current.



FB1 and C13 connected with through-holes

Now, we will go through how to route differential pairs. We assume that you know how to import components from your netlist and place them.

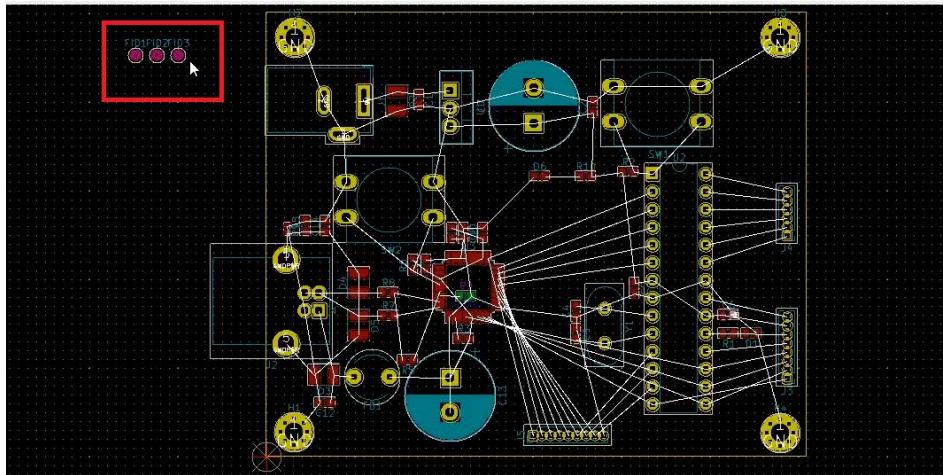
**DEMO VIDEO - [How to Route Differential Pairs in KiCad](#)**

### 5.3.1 How to route differential pairs

The most critical tracks in routing are the differential pair lines. Both wires in the differential pair should be identical in length and width. We need to make sure that they are routed properly and are separated by any other high-frequency signal. But if there are multiple differential pairs then there must be at least a **5W gap** between them. 5W is five times the width of the differential pair.

#### 5.3.1.1 Fiducials

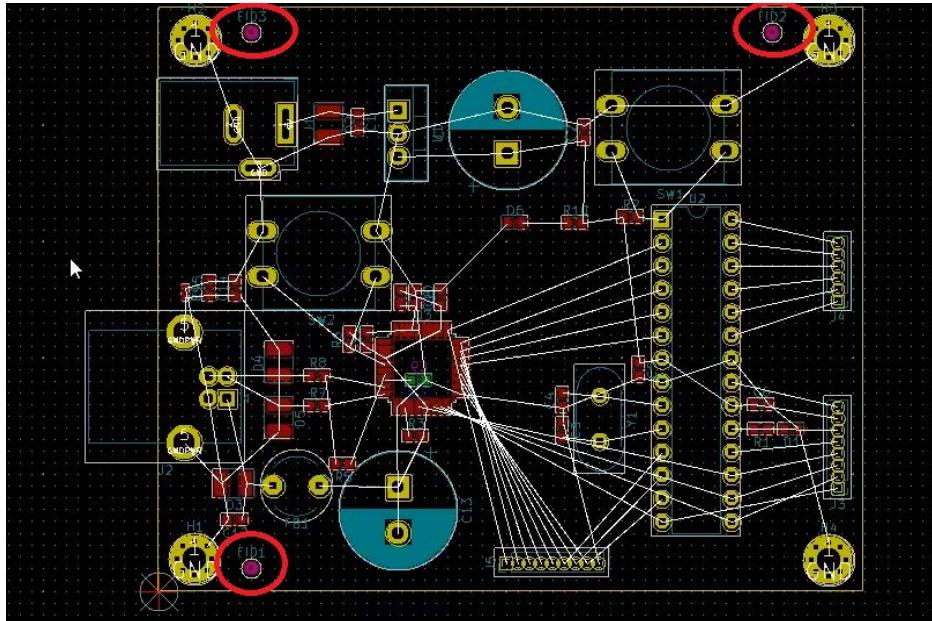
Before routing, let's take a look at these three circles. These are called fiducials.



Fiducials

Fiducials are normally not connected in the circuit, but they are required as a reference during manufacturing to indicate the proper orientation of the board. It will also help you when you place components in KiCad. Click them and place them in three corners.

**Select each fiducial > drag and place them in three corners.**



Selection of fiducials

### 5.3.1.2 Stack-up

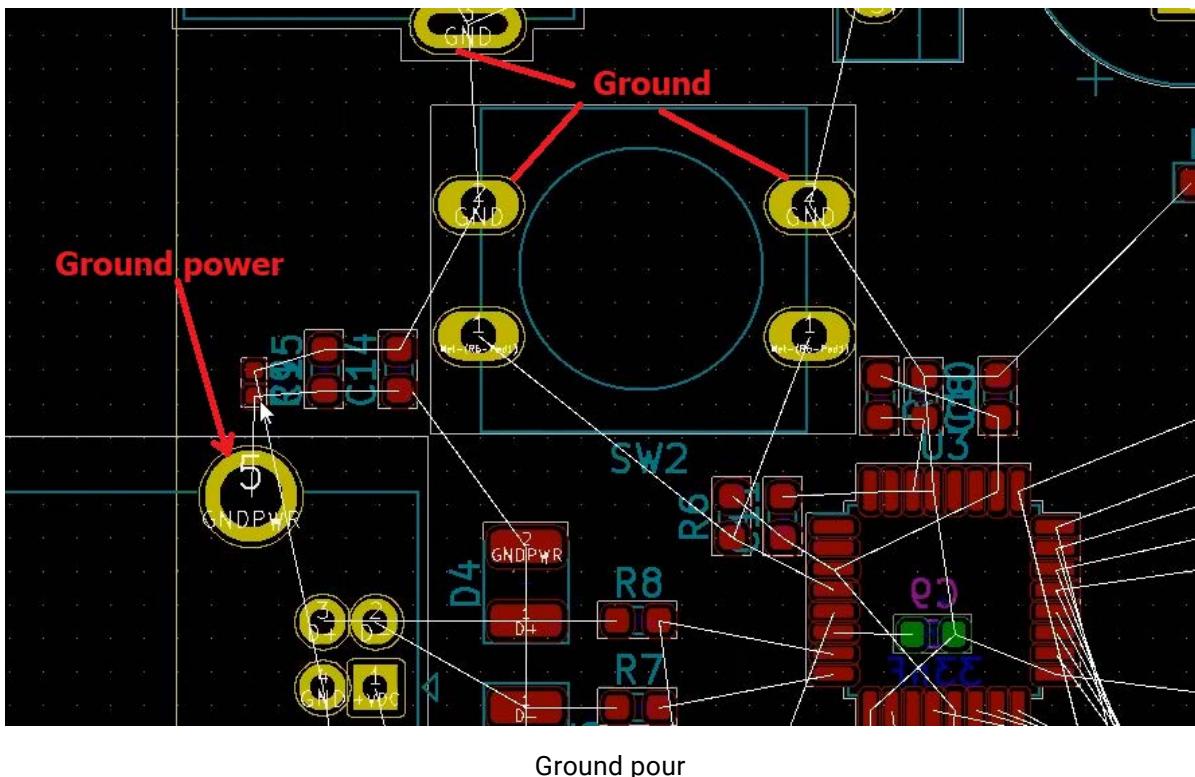
Let's start the routing process by planning the layers. In the PCB stack-up, we have used Layer 1 and Layer 4 as signal layers, which will be used for proper track routing. And Layer 2 and Layer 3 are power planes, so they won't have any tracks. We will fill the complete area with copper. Layer 2 will be ground so we can route the differential pair on the first layer. And even then, after the first layer, the reference for the differential pair should be ground.

We need to make sure that the differential tracks are covered by the reference ground over their entire length. There shouldn't be any cut in Layer 2. If there is a discontinuity, the overall impedance of the  $90\Omega$  differential pair will mismatch and it might affect the signals.

Now, let's place our plane layers.

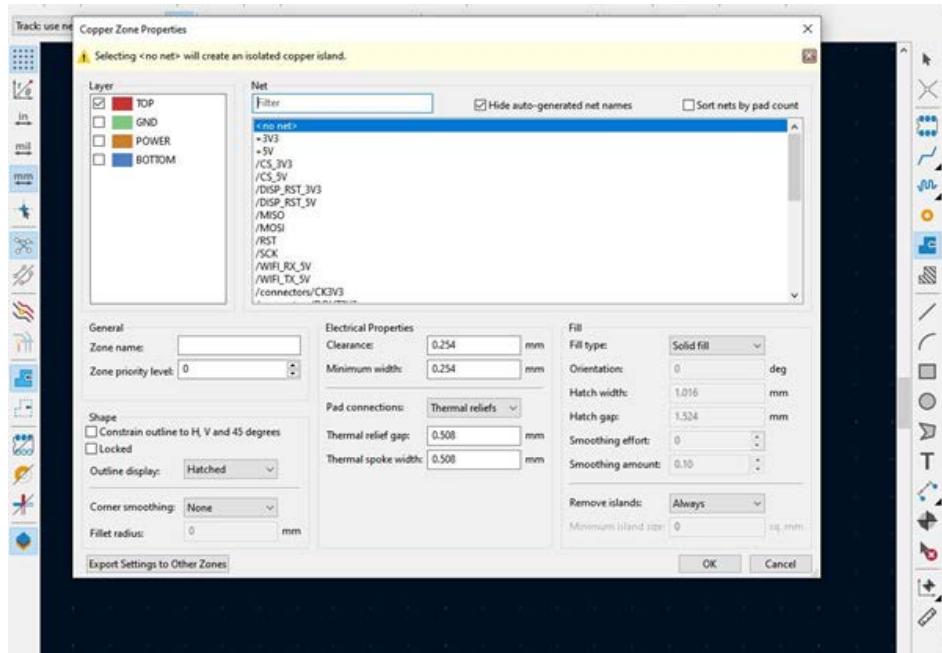
### 5.3.1.3 How to pour ground

There are two types of grounds here: there is **GND** for ground and there is GND PWR for ground power, which are connected to the shield of the USB connector.



So, we will require two types of planes, ground and ground power. But the ground power is not going anywhere in the circuit. In Layer 2, we will pour ground for only in the GND PWR portion and we will keep the remaining portion in the ground. For pouring ground, on the right-hand side, there is an option called **Add filled zones**.

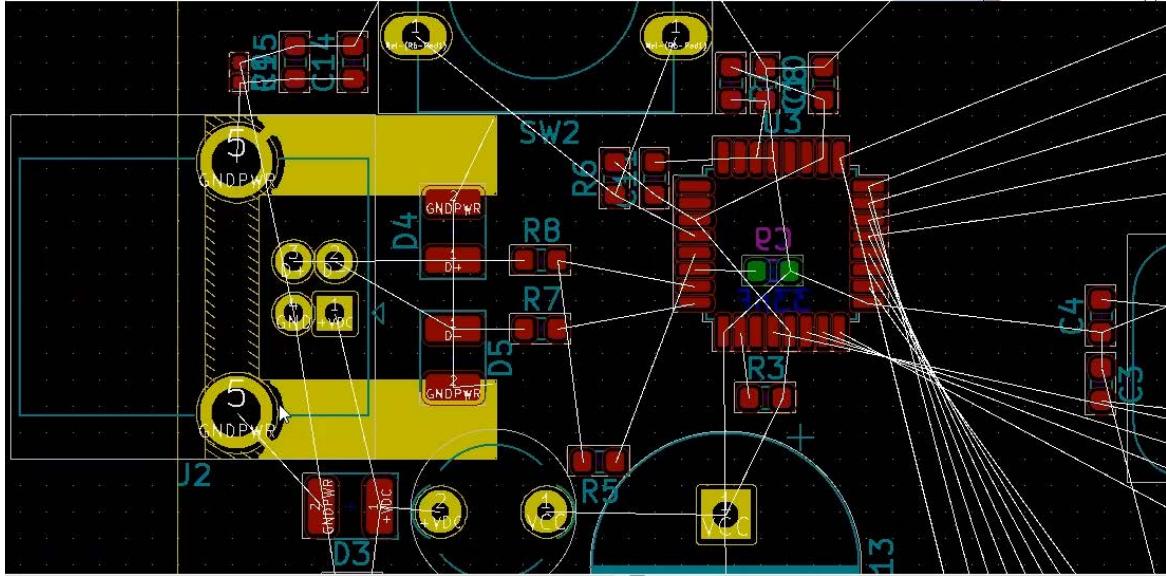
Go to toolbar on the right-hand side > click on Add filled zones icon  > click on where you want to start > a window will pop up.



Copper zone properties

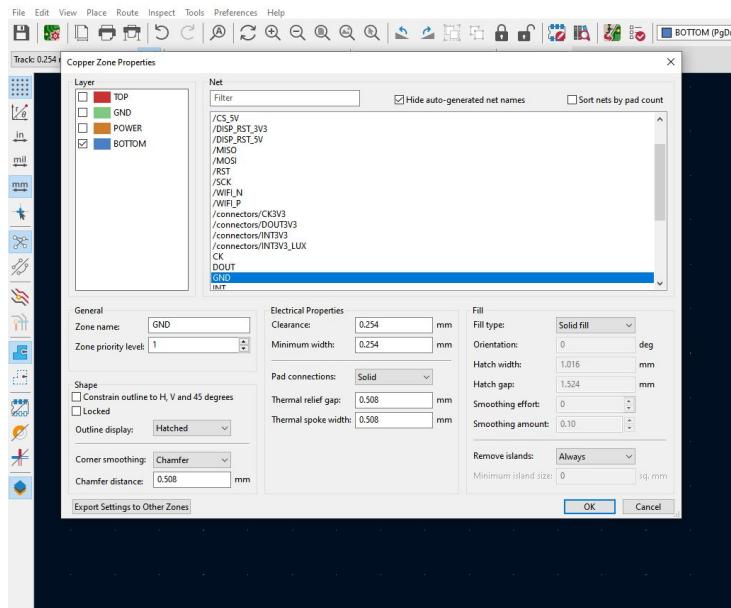
Click where you want to start. A window pops up with various nets. Select which net you want to pour. We will select the ground power and indicate on which layer we want to pour, which is Layer 2. You can see the clearance is 20mils. So, when you pour, there will be a **distance of 20 mils from any other component in the circuit**.

For corner smoothing, select Chamfer and mention 5mils so when you draw a square or a rectangle with 90 degree angles, it will automatically make them 45 degrees with a chamfer of 5mils. Click Ok and draw your shape while making sure to do a complete loop.



Now you can see the portion around GND PWR is connected over here but it is not completely connected. This is basically a thermal connection. To make it solid:

**Go to Copper Zone Properties > Pad Connections > choose solid.**



Copper zone properties

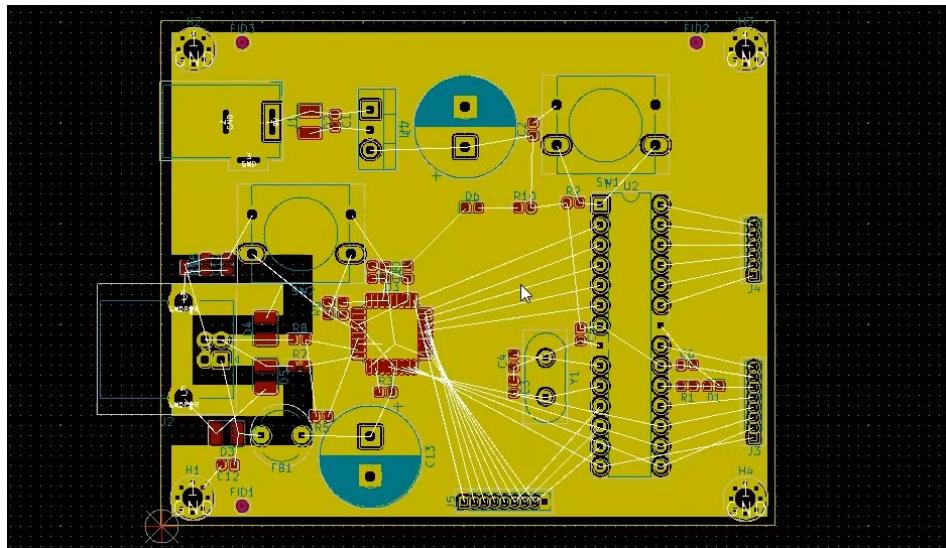


Poured ground

Now it is completely connected and we have poured this ground. If you select on the right-hand side GND or Layer 2, you can see the poured layer.

The next step is to pour the remaining portion with your component ground. Repeat the same pr but this time choose the ground (GND) net instead of GND PWR.

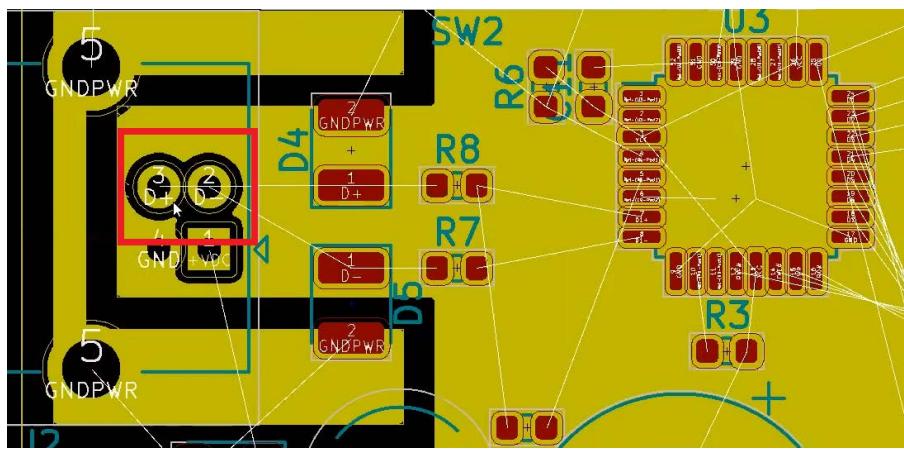
Draw on the required portion for the GND section and click icon to save the file. Now you have routed the ground layer. It looks like the one shown in the below picture.



Like this, we have to pour the power layer. There are two types of power here one coming from the regulator and from the USB. In the USB portion, we have used the VCC connection, and we can use the remaining portion for the +5V. That will be in Layer 2. Now, we will route the differential pair first.

### 5.3.2 Differential pair length tuning

Now, let us see the routing of the differential pair. Any net defined with a suffix of '+' and '-' are considered as differential pairs. Here it is already defined in the schematic that this is D+ and D-.



Differential pair length tuning

You need to set up certain rules. Here, we have set the differential tracks and spacing as 8 mils. Let's start to route differential pairs in KiCad.

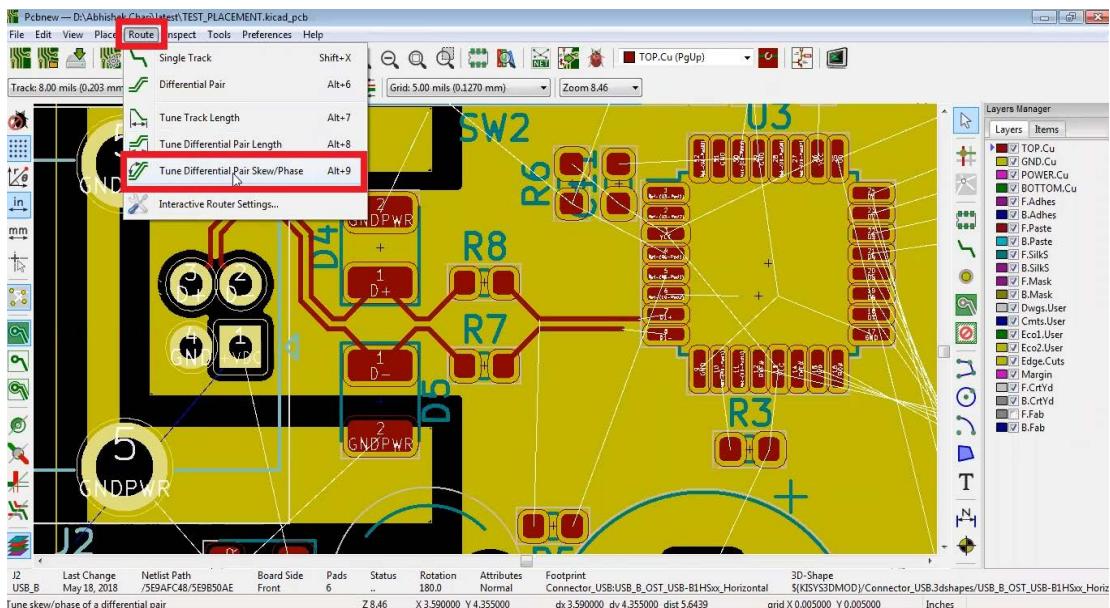
**Go to Route option > choose Differential Pair > click on any of the differential tracks > drag to draw the track.**



Differential pair routing

Go to **Route** at the top and choose **Differential Pair**. As you can see, the line on the top layer is not above the ground plane, so we will have to adjust that later. You can see the tracks are above the ground on Layer 2. Draw the complete track, make sure the tracks are above the ground layer 2. If the track lines go out of the ground plane, you have to adjust it.

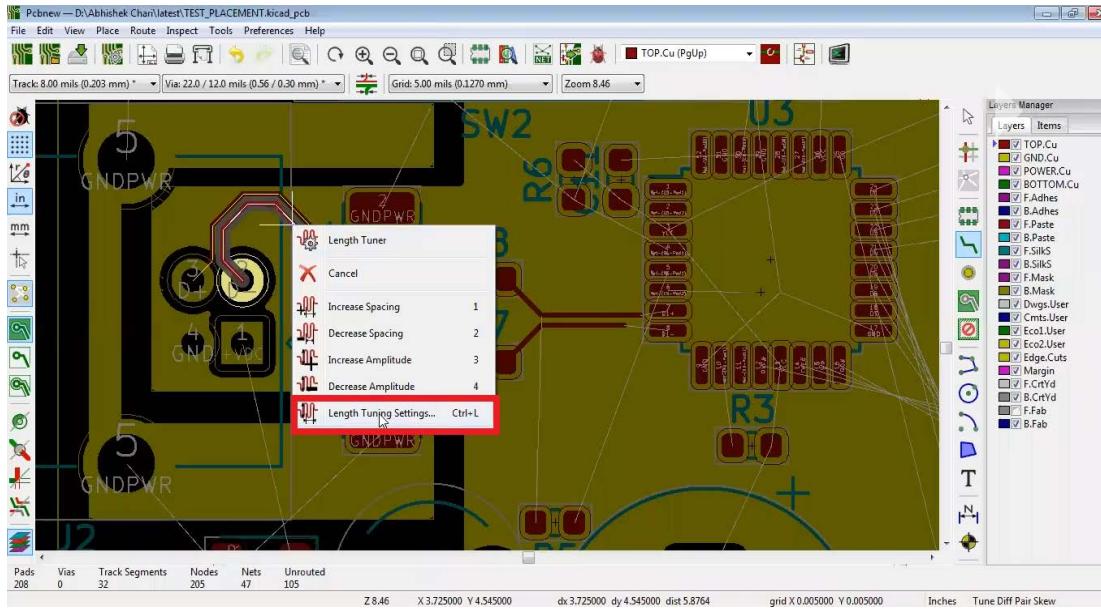
And one more thing you need to make sure is that all these tracks have an equal length. For that, Go to **Route** > click tune differential pair skew/phase > click on any of the tracks.



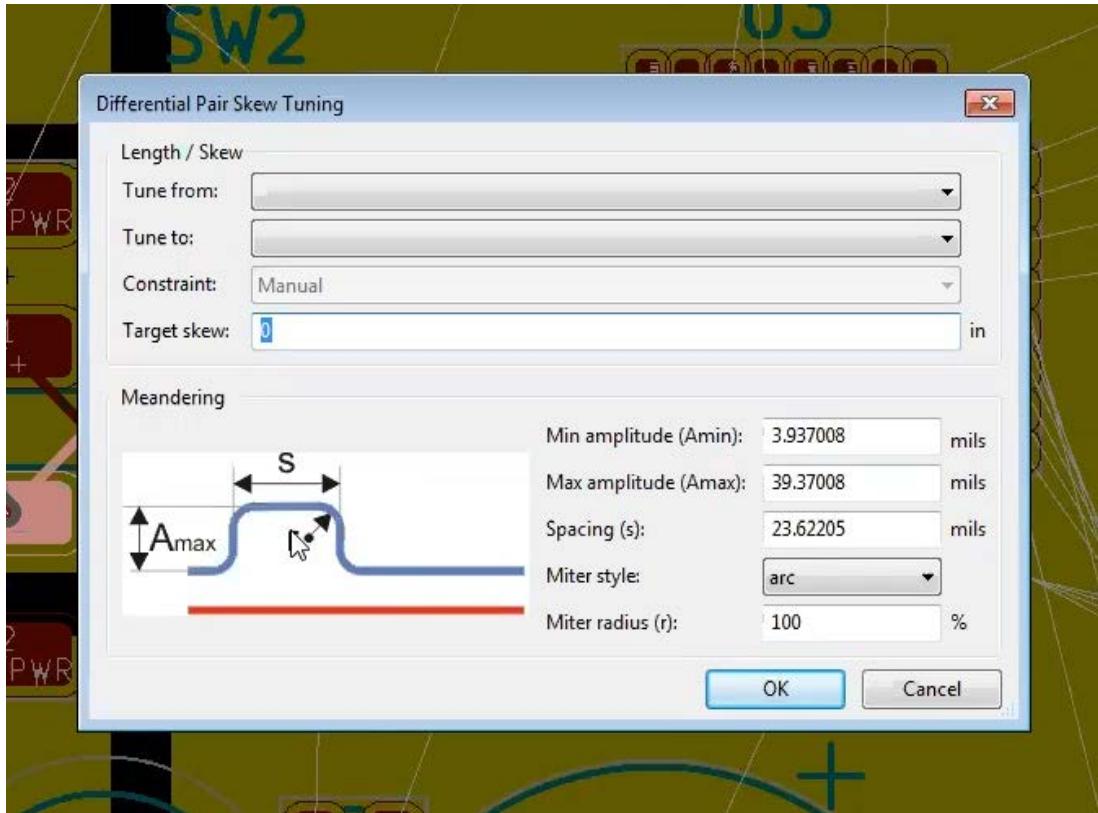
Tuning differential pair skew/phase

When you click on the track, you can see the length of the skew. Length of both tracks should be the same. If it is different, you need to increase the length of the shorter track to match with the longer track. To change the length of the track,

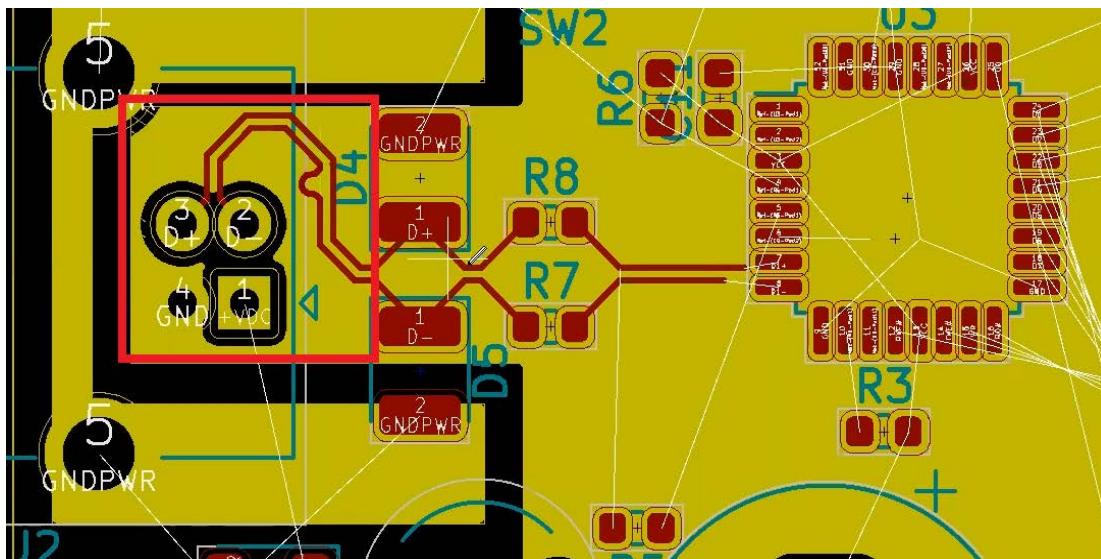
**Right-click on the track > select Length tuning setting > a window will pop up.**



There will be a bump in the shorter track in order to increase its length, so we have to set the dimensions of skew in this pop-up.

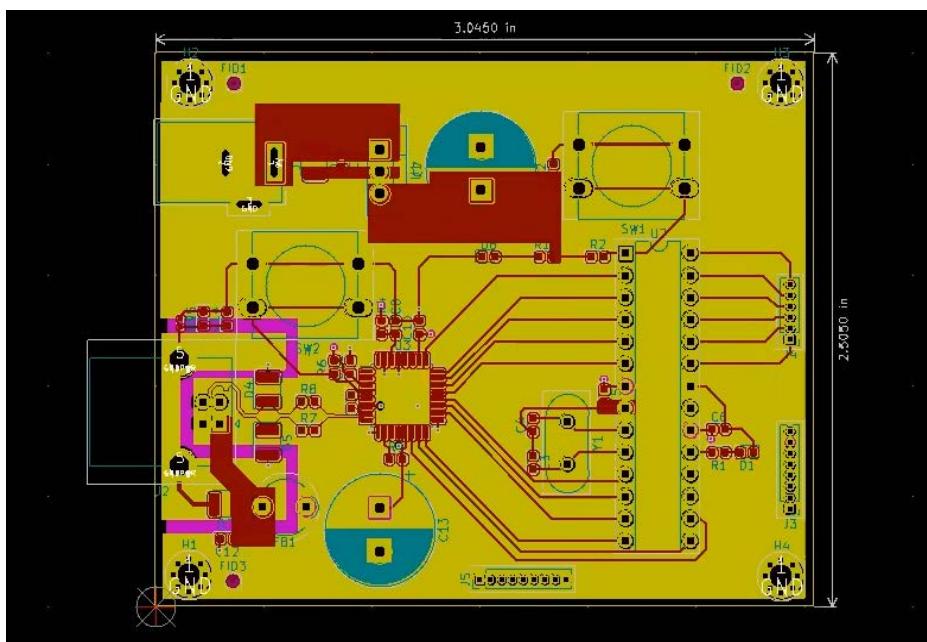


The spacing between the differential pair is set up to 8mils. Set the minimum space to 8mils as well. The spacing should be three times the width, hence, 24mils. Once you're done, drag your cursor over the differential pair and if the tuned skew is green, the tracks have an equal length. Repeat this action for all the pairs you have.



Tuned image of differential pair line

You can see over here, this differential pair we have already routed. When we click it, it says that it is tuned. The differential routing is done.  
This is how the completed design looks like.

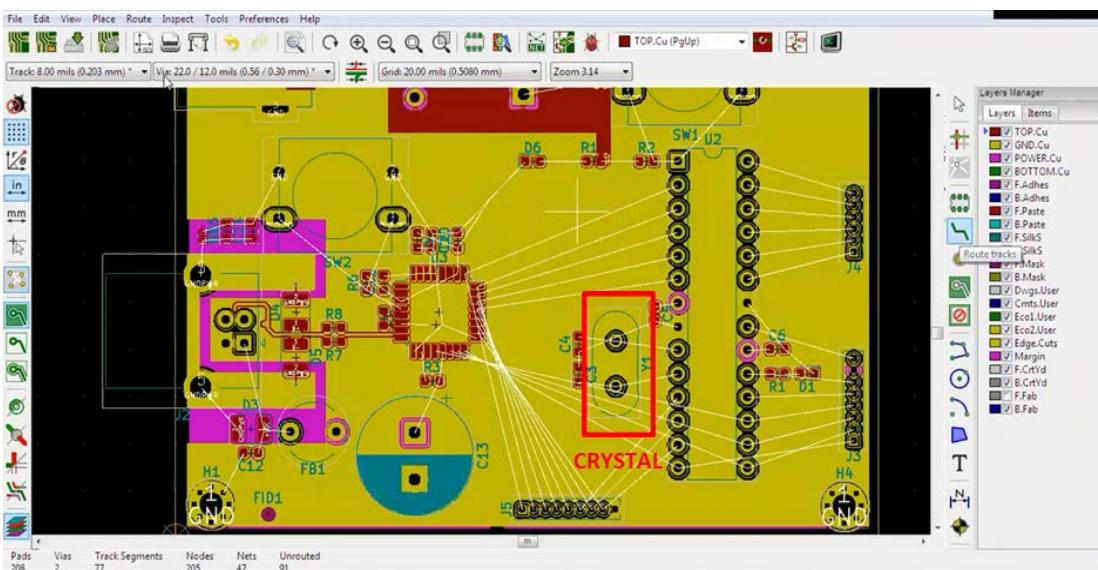


Completed design

This is how we route differential pairs in KiCad. You should do routing for one more layer if you are using a double-sided PCB. One thing we must tell you is to follow the schematic while you do the routing. It can help you to route differential pairs in KiCad.

## 5.4 Crystal routing

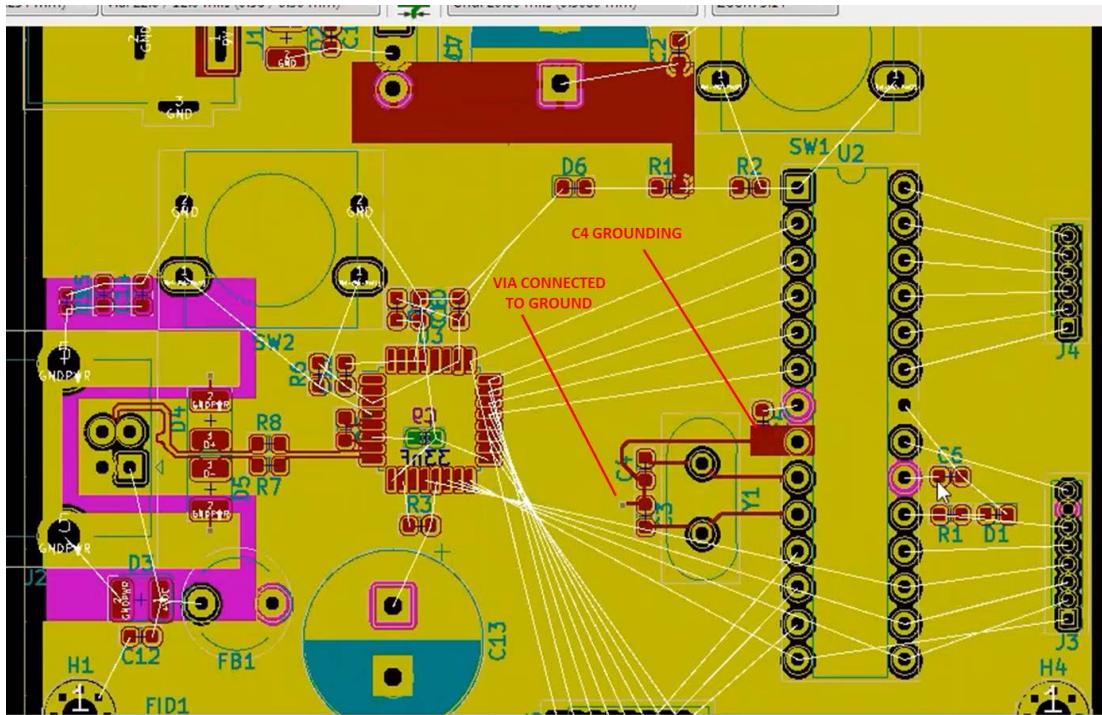
A crystal is basically a continuously varying signal which is required for any microcontroller to operate and used as a reference signal. **This should be very close to the microcontroller and its track should be as short as possible.**



Crystal location

We will route this line. Select on the right hand side **Route Tracks** and mention 10mils. You can go with 10mils when the board is not congested. If it is congested, then you can reduce it to 8mils. Depending on your requirements, you can even reduce it to 6mils.

First, let's do the routing of the critical nets. As mentioned earlier, the crystal routing has to be as short as possible.



Grounding capacitor through via

The ground of a crystal should always be connected to the nearest ground pin. C4 can be connected to a ground pin as shown in the image above. Now, the via is placed and will connect to the ground plane through the ground pins.

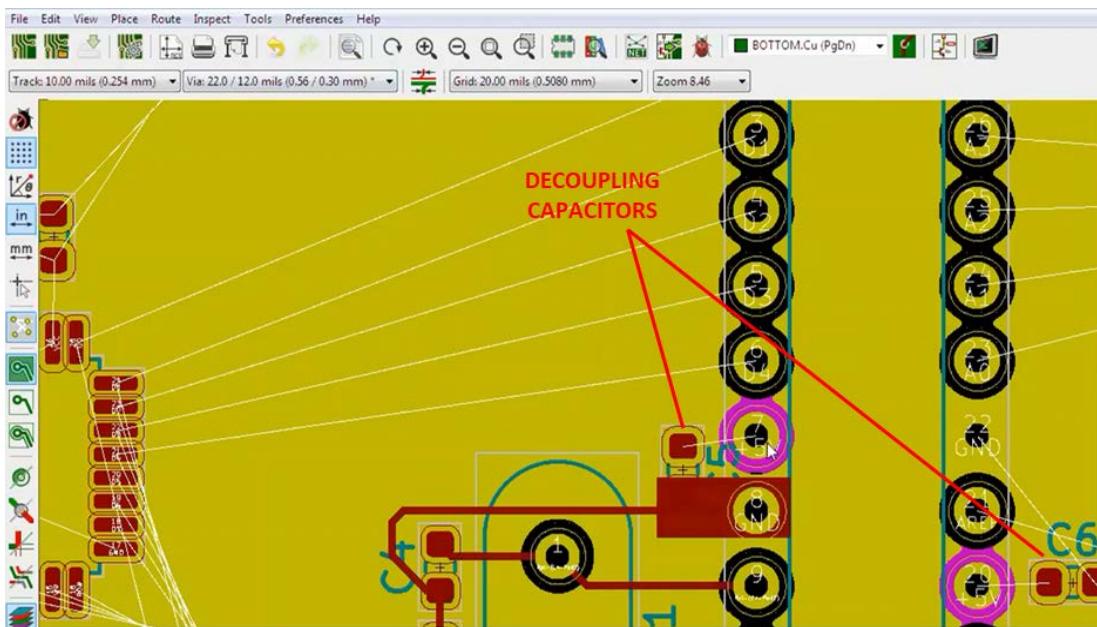
To connect C3 to the ground plane, you can place a via to connect to the ground as shown in the image above. In some high-speed designs, you can follow guard ring technique.

To place a via:

- Type **V** on your keyboard
- Click on the board to place the via
- Choose **Cancel**.

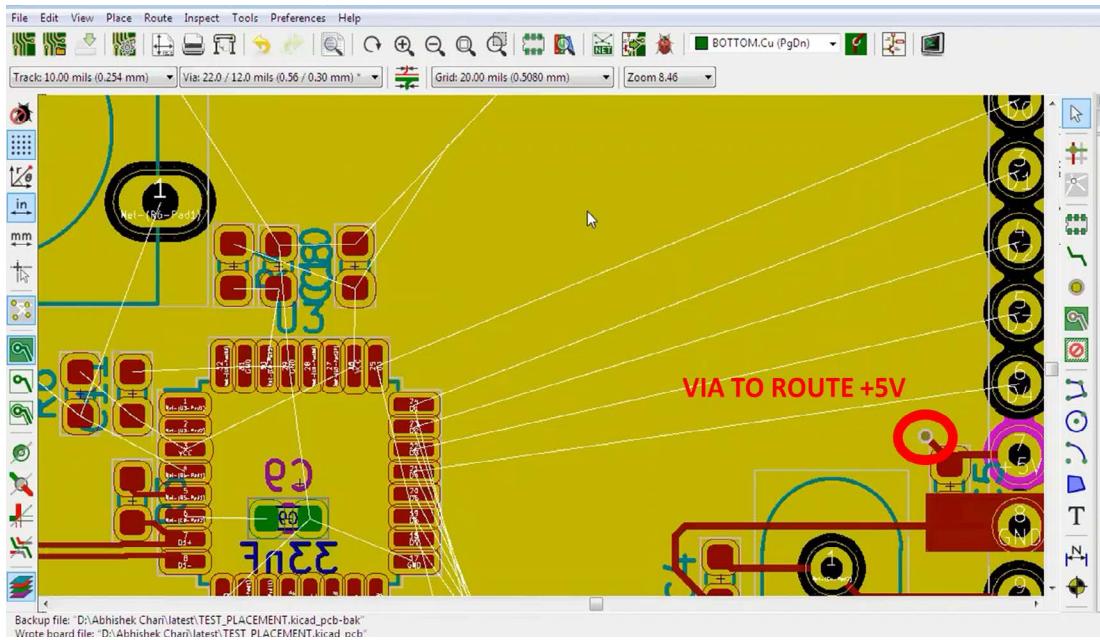
## 5.5 Routing for decoupling capacitors

First, we need to deal with the decoupling capacitors. You can see these are decoupling capacitors.



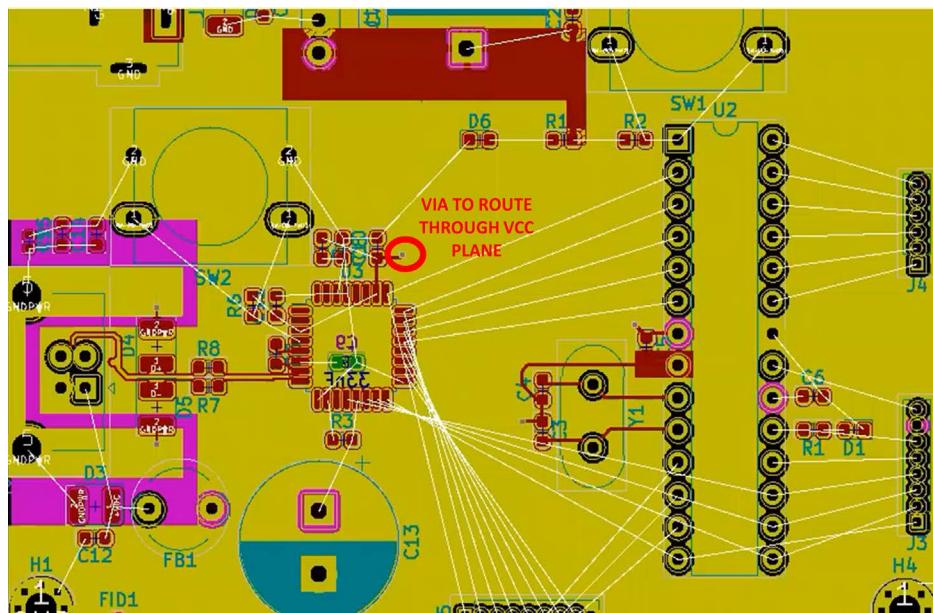
Decoupling capacitors

Here, the +5V comes from the regulator. And the track from the regulator should go towards C5 pins through the capacitor, not directly through the 5V. Let's connect this top layer and have the 5V distributed through it. We can place a via near C5. Now, the +5V will come through the plane towards the C5 capacitor through the via, and then it will go to the power plane.



Via to route +5V

As shown in the image below there is a power plane, VCC, and a capacitor. Select Route Tracks and connect this over here. The **plane** is connected but the VCC is still not connected and we have placed a plane on the layer. For that click V to place the same. Now click **Cancel** and **OK**.

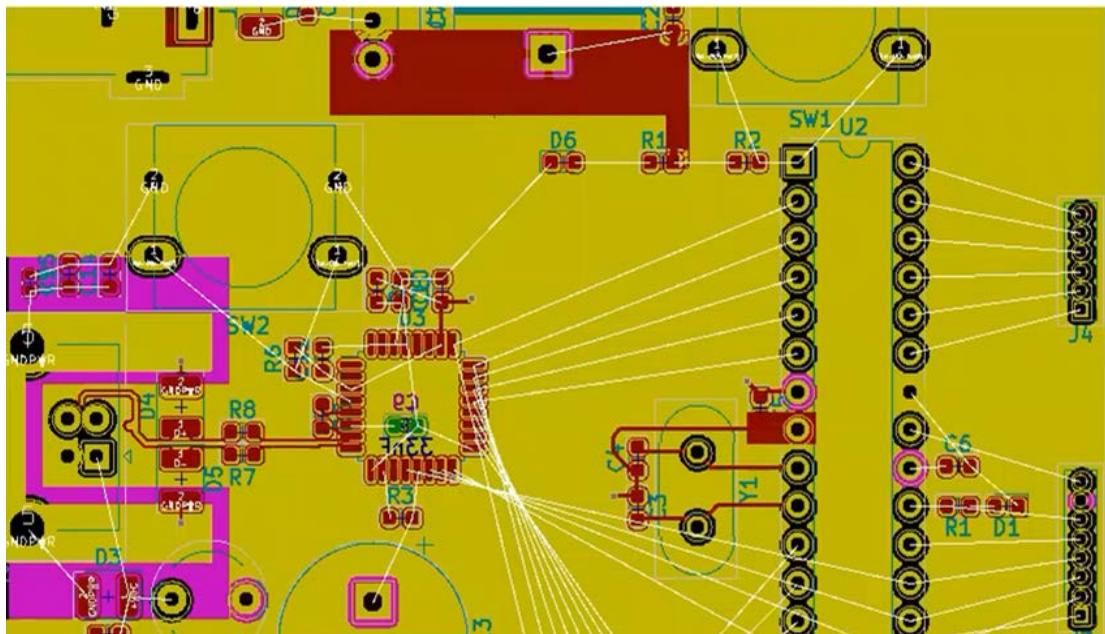


Via to route through VCC plane

Now, the connection will come through the VCC plane, go through the capacitor, and then to the VCC. Every decoupling capacitor should be routed this way.

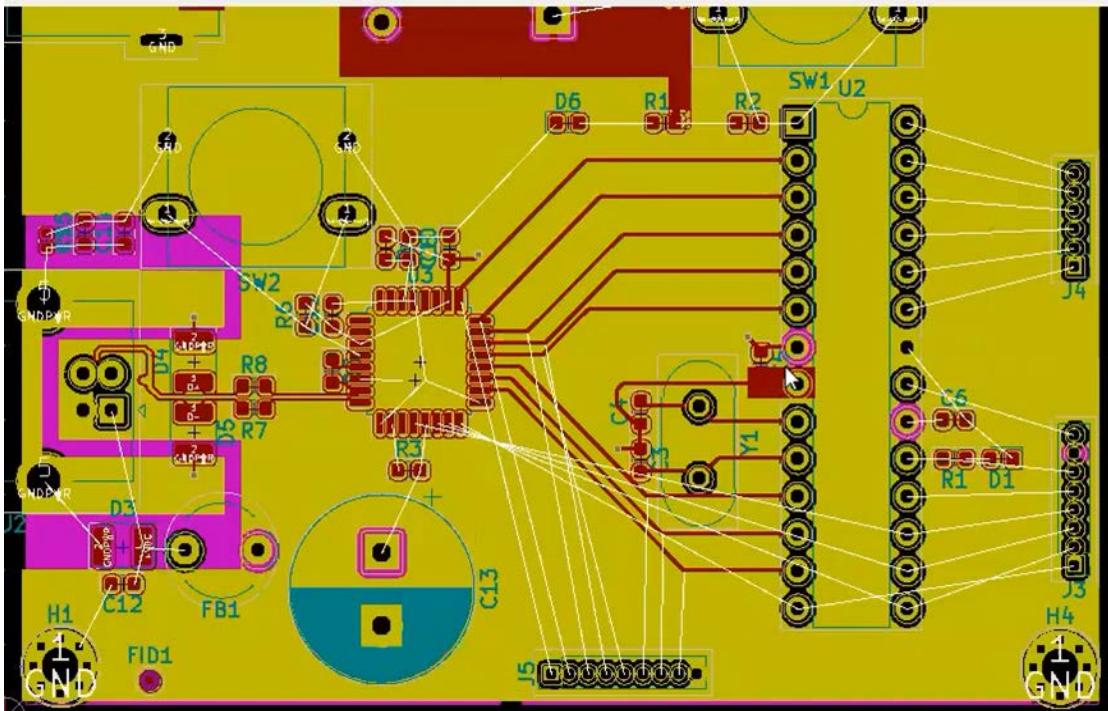
## 5.6 Main pin routing

The next step is to deal with the main pins. These are the control lines with which we **can control operations**. They will also give information about whether the data is available on the chip.



Main pins before routing connections

The pins are connected to the corresponding connectors. Normally, we follow an **orthogonal entry rule** while routing on top and bottom layers. An **orthogonal entry** means that the top lines are at 90 degrees with respect to the bottom lines.



Main pins after routing connections

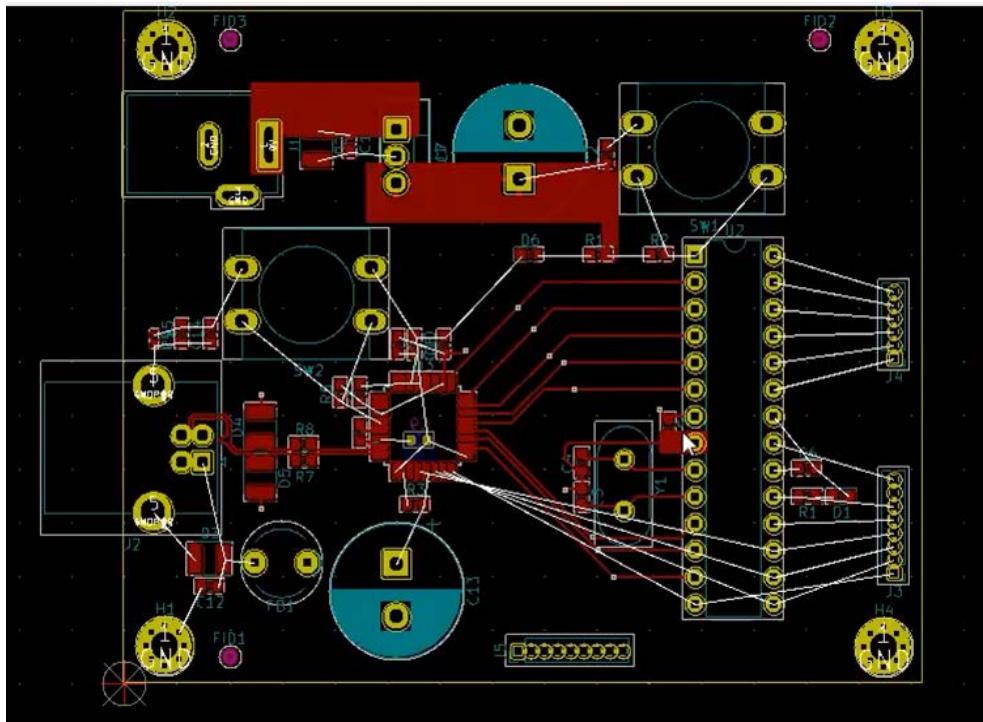
These are **8 data lines** that are going to the IC and the pins shown should be connected to the connector.

We actually need to move one of the tracks slightly away from the crystal zone. The data lines are connected and we need to connect these lines to this connector. To use the bottom layer, select the line and select bottom.

You can see these lines are running horizontally. But the **top to bottom line** is running vertically. This makes the PCB routing very easy if you're routing different lengths on the top and bottom. We can connect the via to these tracks on the top.

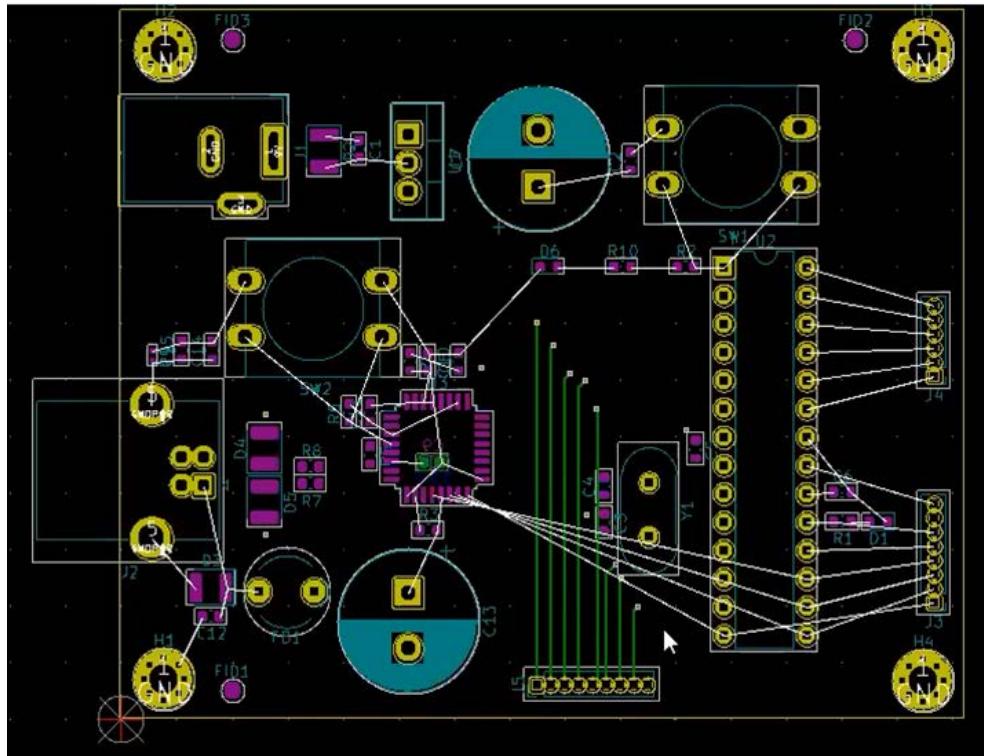
## 5.7 Routing the layers

We are demonstrating the principle of PCB routing horizontally and vertically on different layers. This is the top layer.



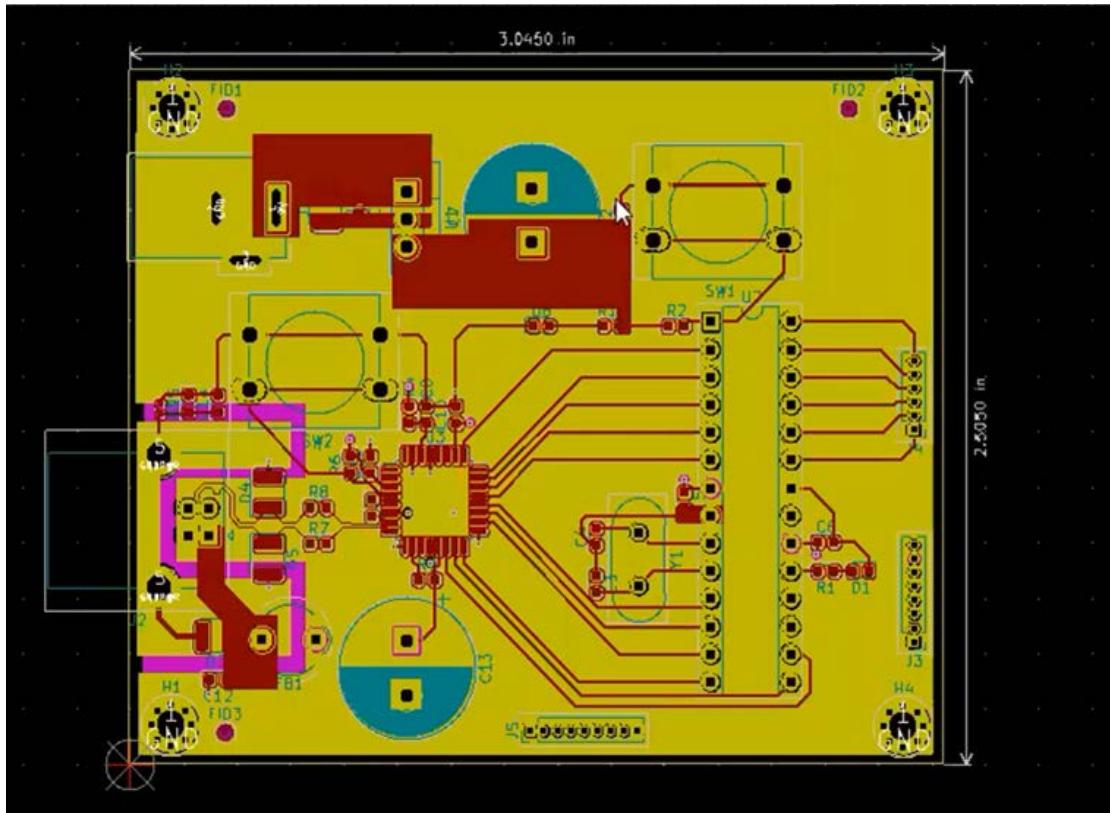
PCB top layer

And this is the bottom layer.



PCB bottom layer

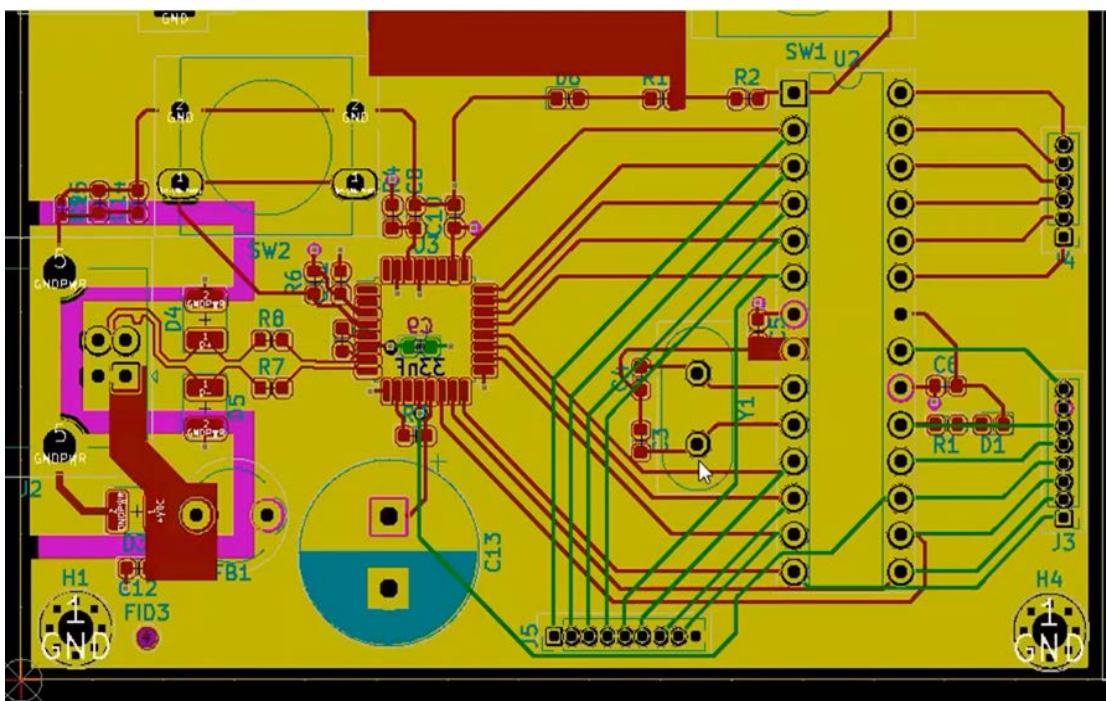
We will now show you how to do the entire PCB routing. Let's go ahead and save this part first and close it.



Saving the first part

As you can see, this is a completed PCB routing. You can see the planes. We have used a complete plane over here which is connected through layer 3. If you want you could also have a plane on top. You can see the via that is coming from the plane through the **capacitor and the pin**. Same scenario over here.

Now, let's have a look at the bottom lines. Here, they are in green.



**Bottom layers designated in green**

Once the PCB routing is completed, the next step is to make sure that the designators don't overlap with the **pairs**. It's fine if they overlap with the tracks but not the pairs.

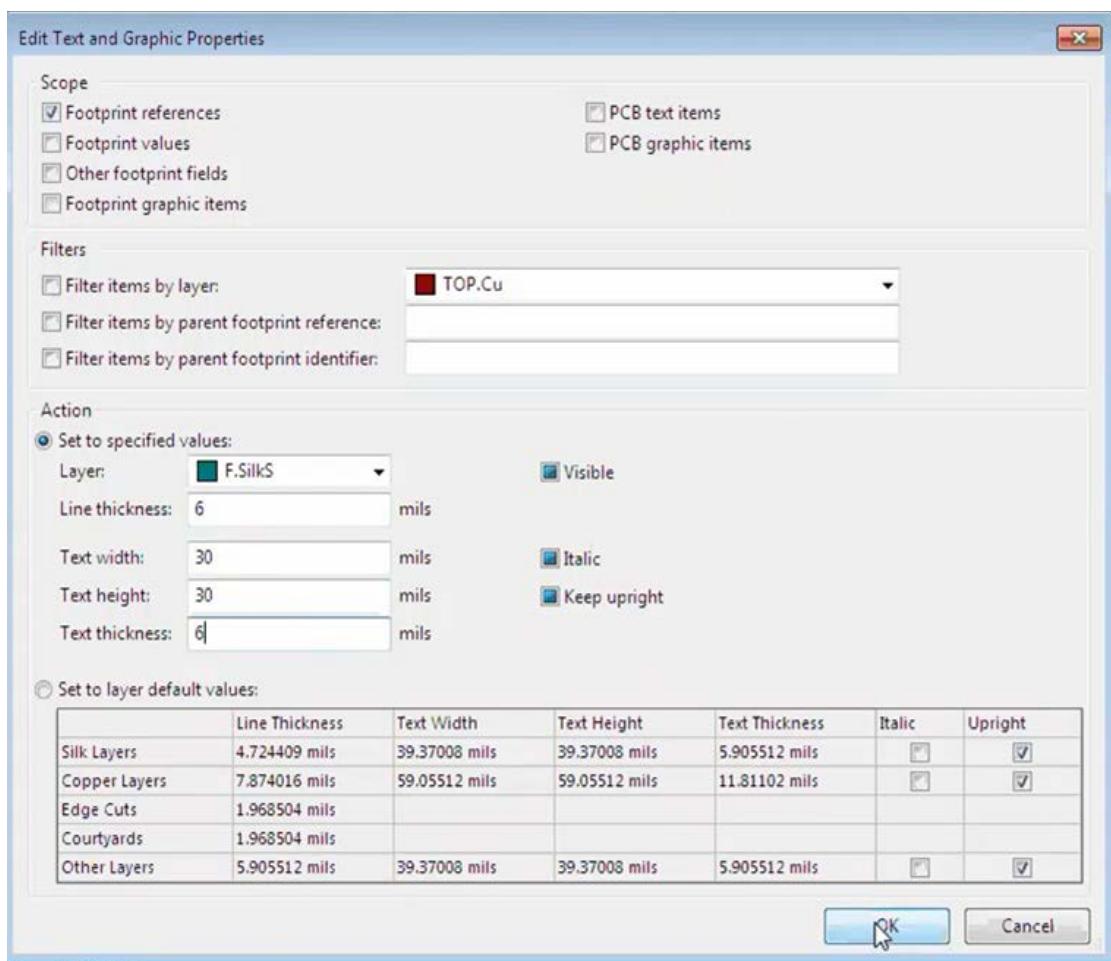
**You can see at the bottom, there are designators.**

## 5.8 Ref text placement

It is possible to place your **ref text** over your vias, if they are tented, which means they will be covered with a solder mask. But it is always a good practice to keep your **ref text** away from the pads as well as the vias. If it is not possible to keep it away from your vias, then tent your vias.

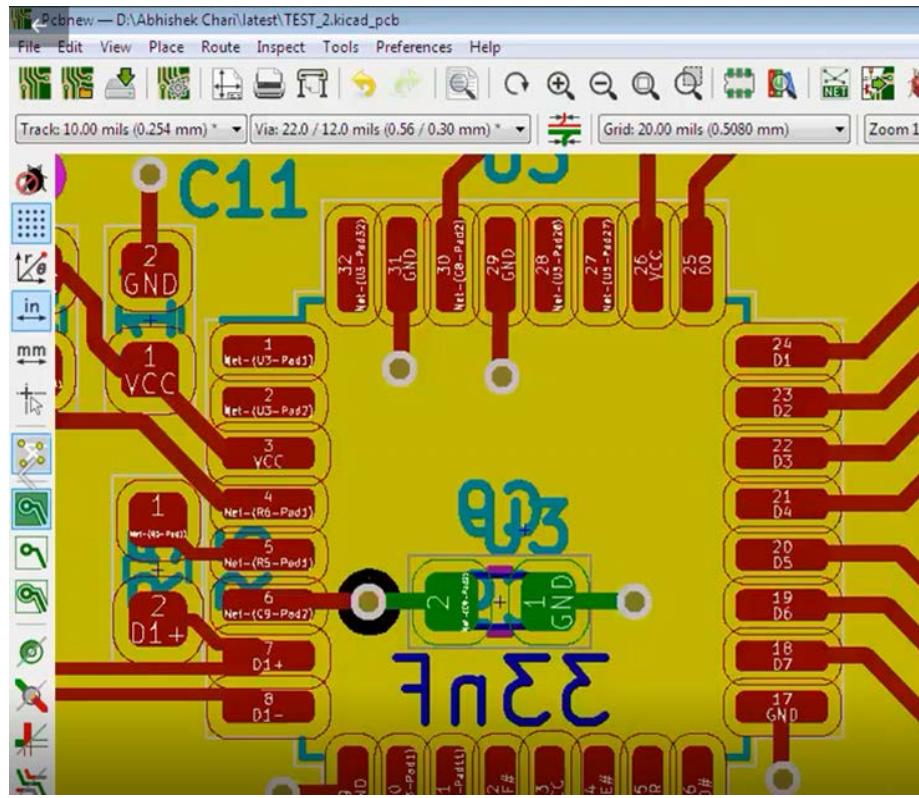
To move the ref text, place the cursor on it, select it and find a suitable place without overlapping. You need to select a proper location and make sure that the distance from the **ref text** is at least **6mils from the pads**.

Right now, the **ref text** sizes are quite big. To change the size, go to Edit and Edit Text & Graphic Properties. Select the footprint references and select the layer. Here, the line thickness is 6mils and the text width and height will be 30. The text thickness is also 6.



Edit text and graphic properties dialog box

Now that the size is reduced, it is easier to do the placement. If you zoom in a bit, you will see a white line on the footprints. If you can see this footprint on the right-hand side, this is nothing but a **place bound**. It is always a good idea to keep the ref text outside of this region bound so there is no possibility of interfering with the actual pads.



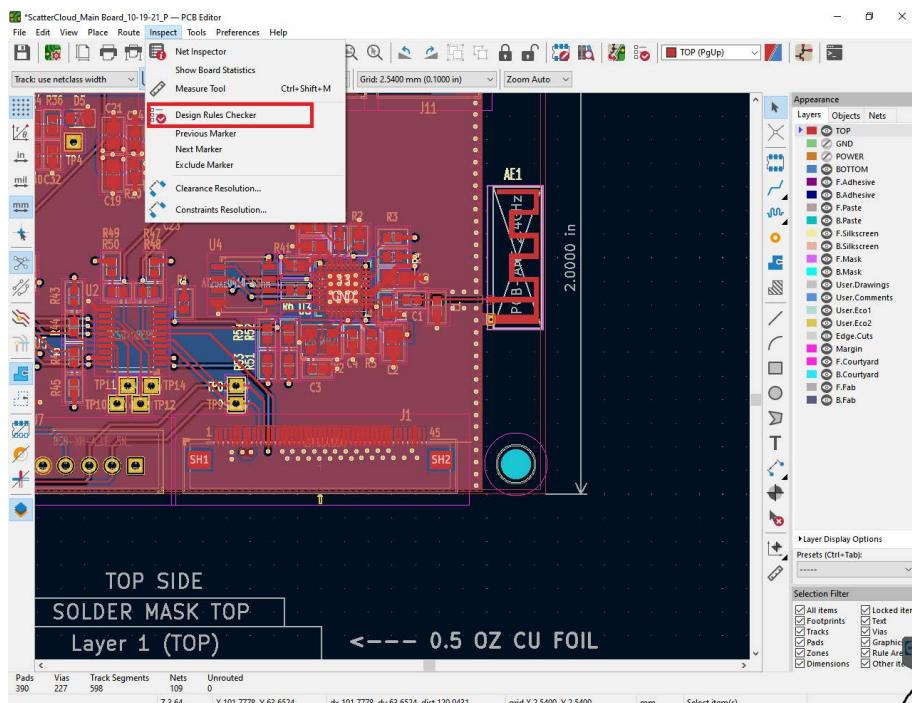
Ref text C11 placed cleanly outside bounds (white lines)

## 5.9 Design rules checker (DRC) and clearing DRCs

Let's complete the entire placing of the **ref text** before starting with the **DRC checks**. Make sure that all the **ref texts** are away from the pins. Once this is done, you need to start with the **DRC checks**, which are the rules that have been set up at the beginning. This will check whether those rules are being followed or not. So let's see how it works. Save the file progress and go to **Inspect** and **Design Rules Checker**.

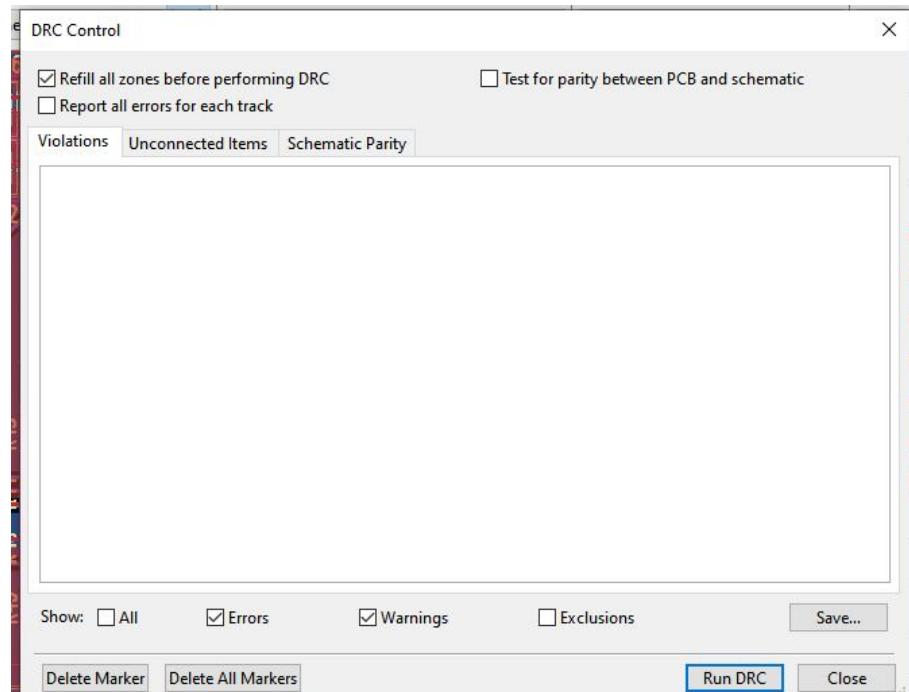
To create a report file:

- Click **Inspect** and select **Design Rules Checker**



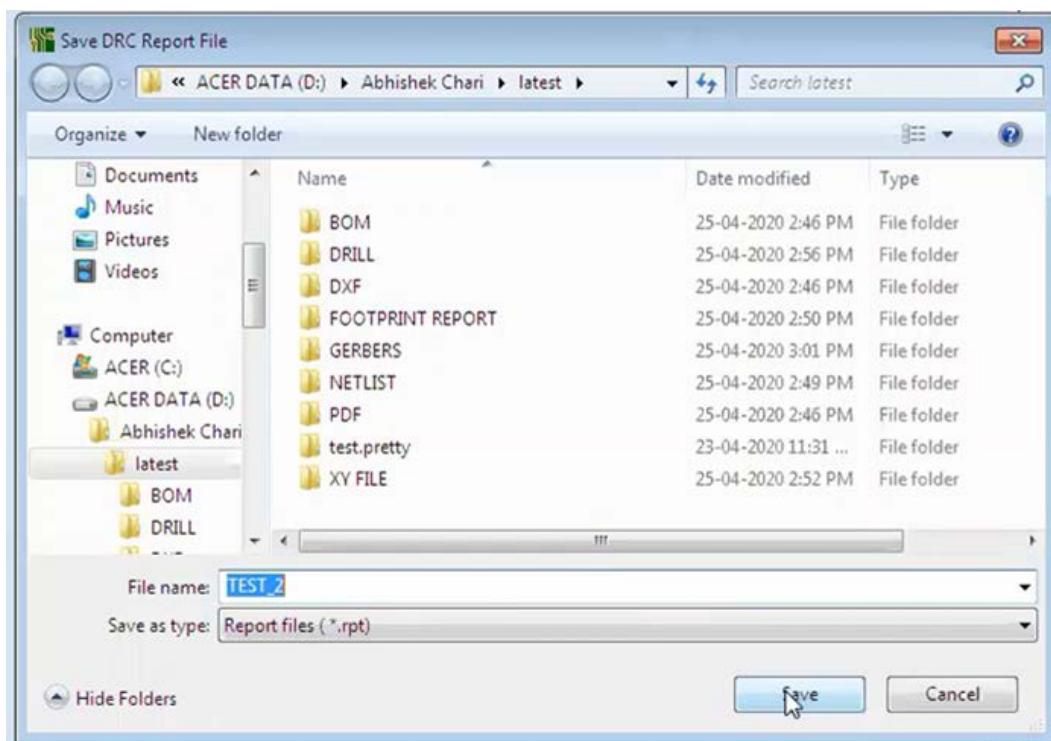
Inspect Menu > Design Rule Checker

- Select **Refill all zones before performing DRC**
- Click **Run DRC**



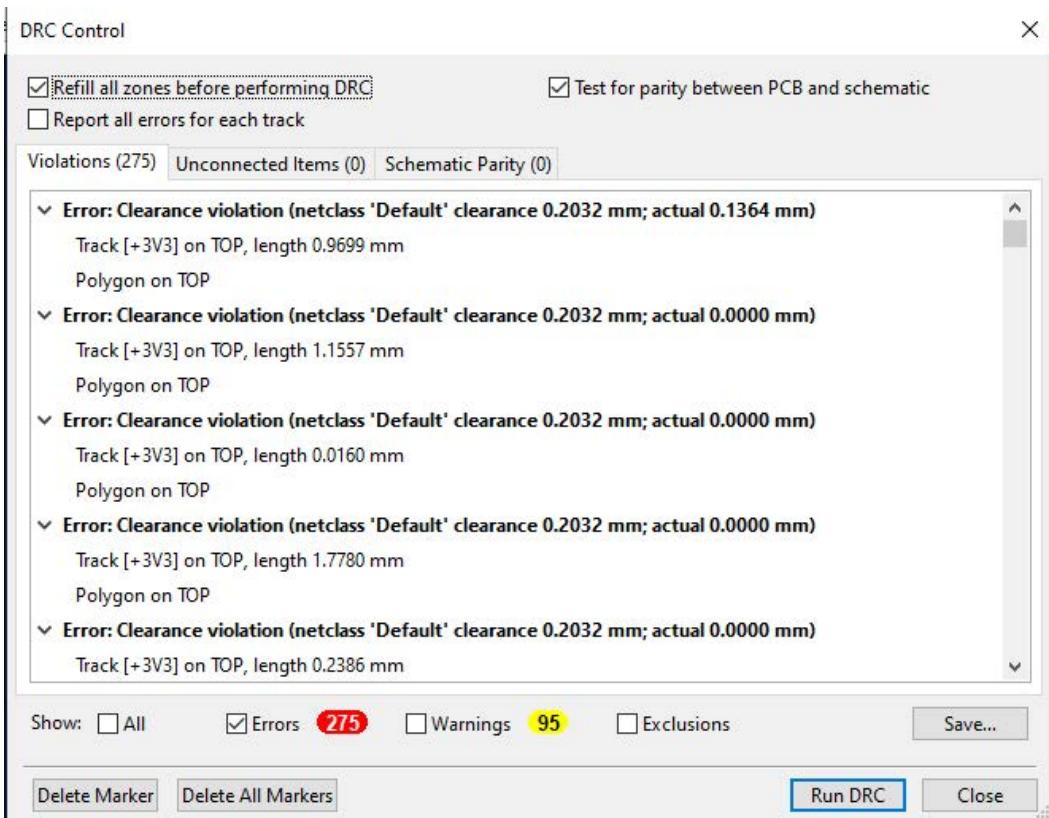
Creating report file

- Specify the report file location and click Save



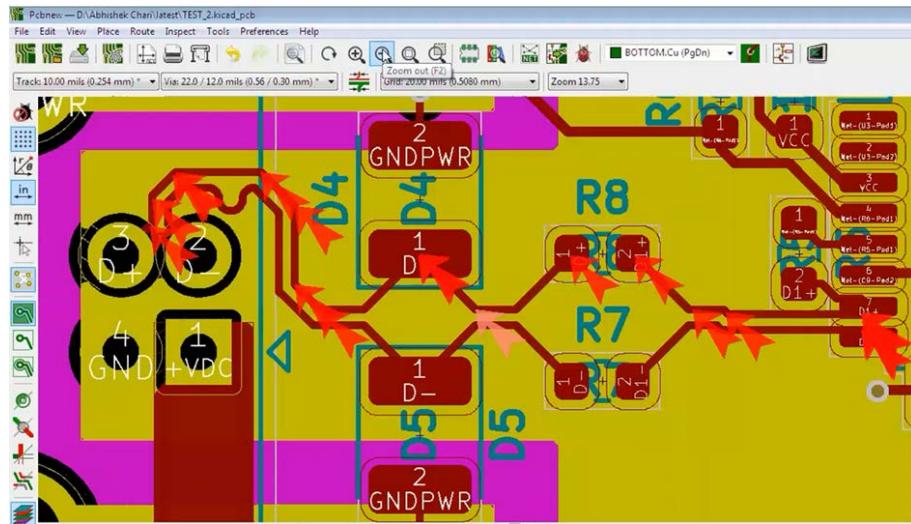
Saving report file

Now the DRC tells us there are a few errors. Most of these errors are concerning the differential lines. For instance, here it says 'two track ends too close.' Right now, we don't have to worry about this because we have set a particular rule for the minimum distance between two tracks.



DRC Error Report

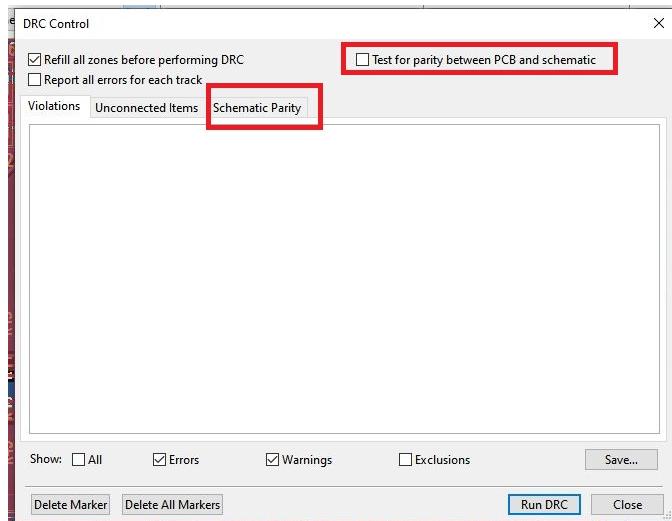
All the errors are in the section shown in the image below and are all the known errors. Even if we knew this would happen, it is possible to avoid it. We can change the clearance to 10mils and the spacing to 8mils. The DRC is considering each track separately and since the spacing is just 8mils, it is actually not following the rule of 10mils. That's why it is indicating there is an error, which can be ignored.



Track clearance error

If there are any other errors in the DRC, you need to solve them before going further.

The connectivity between the schematic and layout can be tested by checking the box beside **Test for parity between PCB and schematic**.

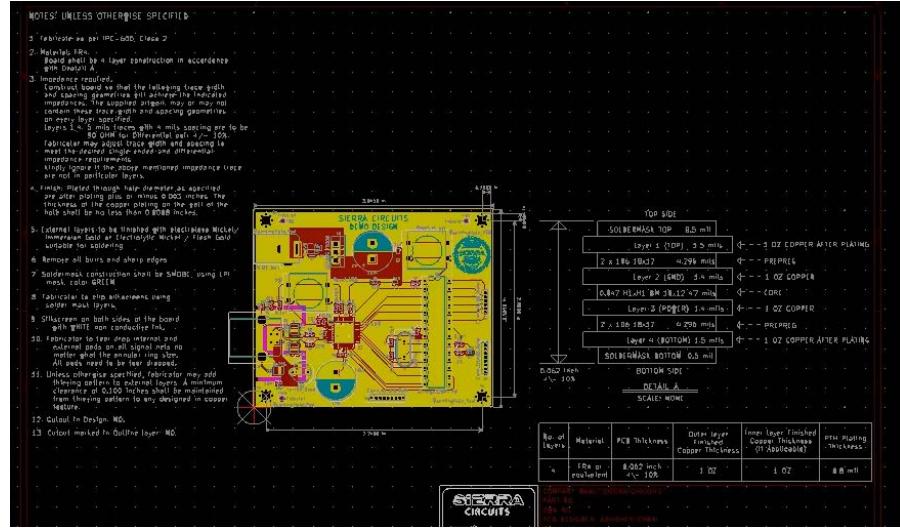


Test for parity between PCB and schematic

It also gives us details of unconnected traces. The data can be filtered by selecting Errors, Warnings or Exclusions.

This is how we carry out PCB routing in KiCad.

# Section 6. Exporting Gerber and production files in KiCad



An example of fab notes

To manufacture a PCB, we send these Gerber files and production files of the design to the manufacturers. Manufacturers send this data to plotters and Computer Numerical Controlled (CNC) machines to fabricate a PCB.

## DEMO VIDEO - [How to Export Gerber and Other Production Files in KiCad](#)

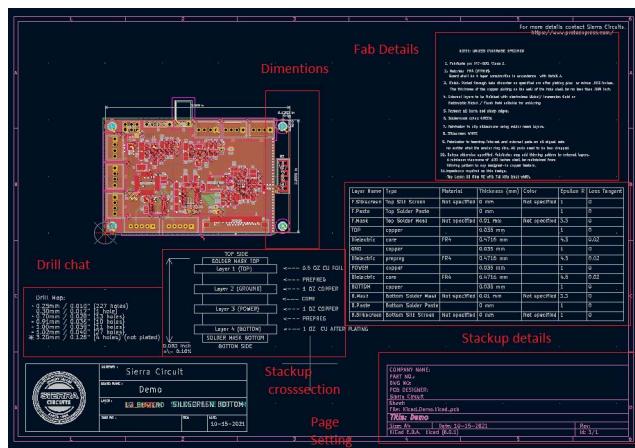
Once the routing is completed, we can export our Gerber files. But before generating the Gerber files, it is important to make sure our design is perfect with the schematics and the layout. So for that, we have created a checklist here.

### 6.1 Checklist for generating Gerber files

There are three types of checklists you need to go through before generating your production files – BOM, schematic, and layout.

You need to confirm whether the bill of materials (BOM) which you have created is as per your schematic. i.e. the footprint, the component values, etc., should match the manufacturer's part number.

Later, check the schematic values, whether the components are placed correctly and the quality markings are correct in the circuit. Once these are done the layout checks are complete. Now check whether you have placed all components required for manufacturing as per your design requirements. Also, there should be several other pieces of information that should be mentioned in the Fab drawing, which are marked in the image below:



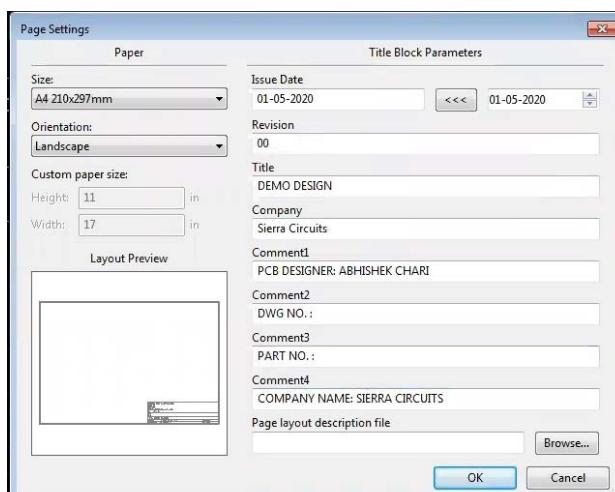
An example of fab drawing

Now let's check how to write the details in these sections before generating the Gerber.

### 6.1.1 Page settings

We have to add the details such as the name of the board, part number and required comments, etc. in the title block at the right bottom corner. To do that follow these steps:

- Go to Files > Select page settings**
- A window will popup**



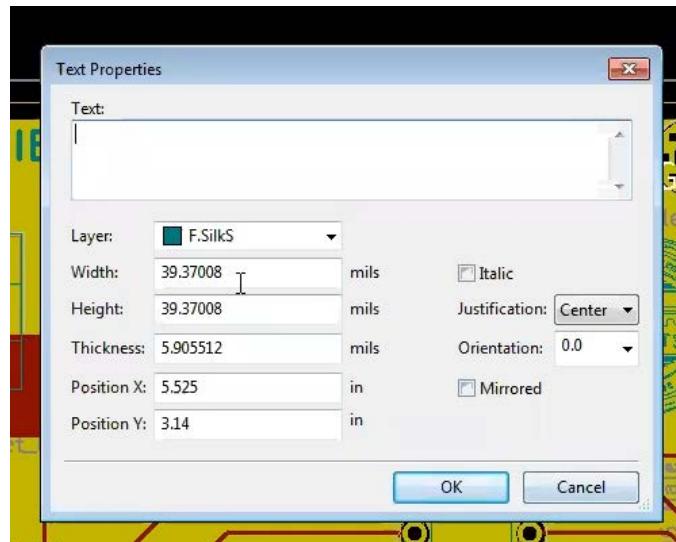
Page settings

Whatever you type in this window will appear on the right bottom corner. In the same way, you can do the page setting in your schematic as well.

### 6.1.2 Adding logo and text

You may need to add text on your board, such as the company name. To do that, follow these steps

- Select the silkscreen layer and select text icon **T** > click wherever you need to add the text.
- A window will popup.

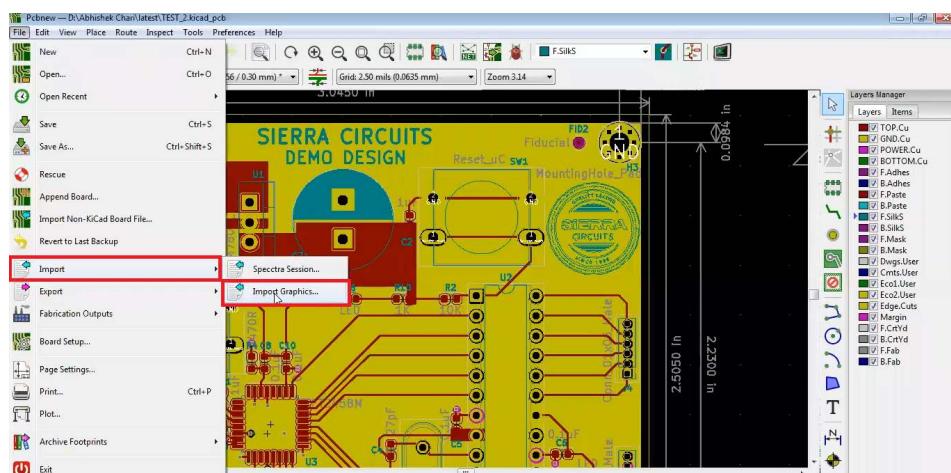


Adding text on the board

- Add the text in the popup window and set the dimensions of the text as per your requirements.

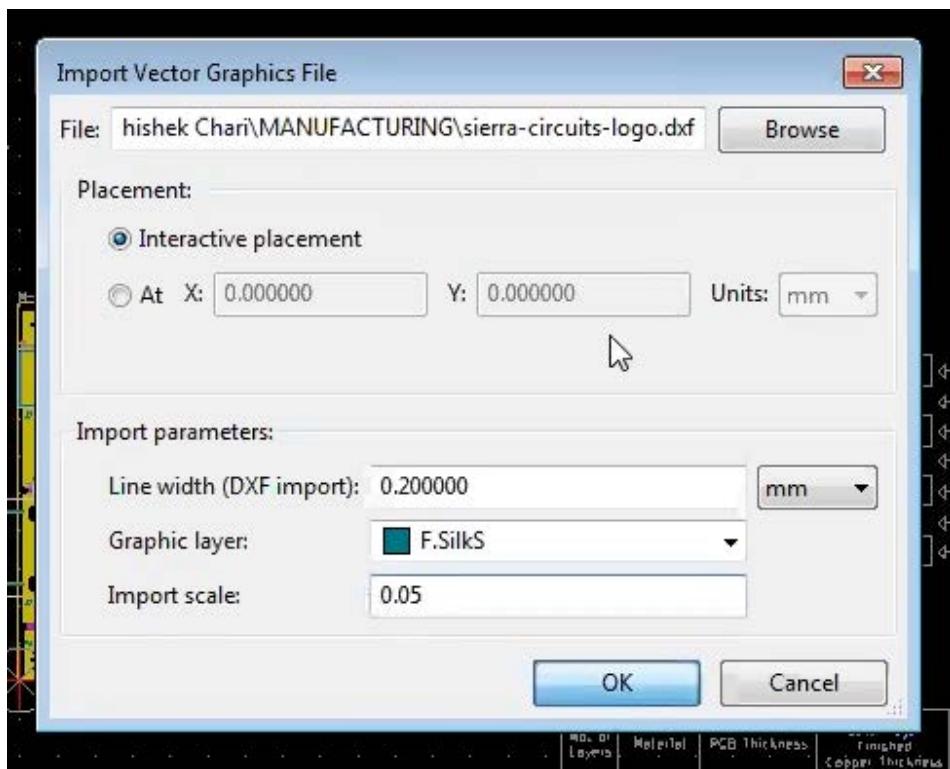
Now to place the logo of your company you will require a DXF file of the logo. If you don't have a DXF file then you can use an image of your logo, and you can find online software that converts your image into a DXF file. You can use that as the logo for your board. To import the file:

- Go to files > Import > Import Graphics.
- A window will popup.



Import graphics

Browse and select your DXF file from the popup window, and set the dimensions of the logo in the import scale option.

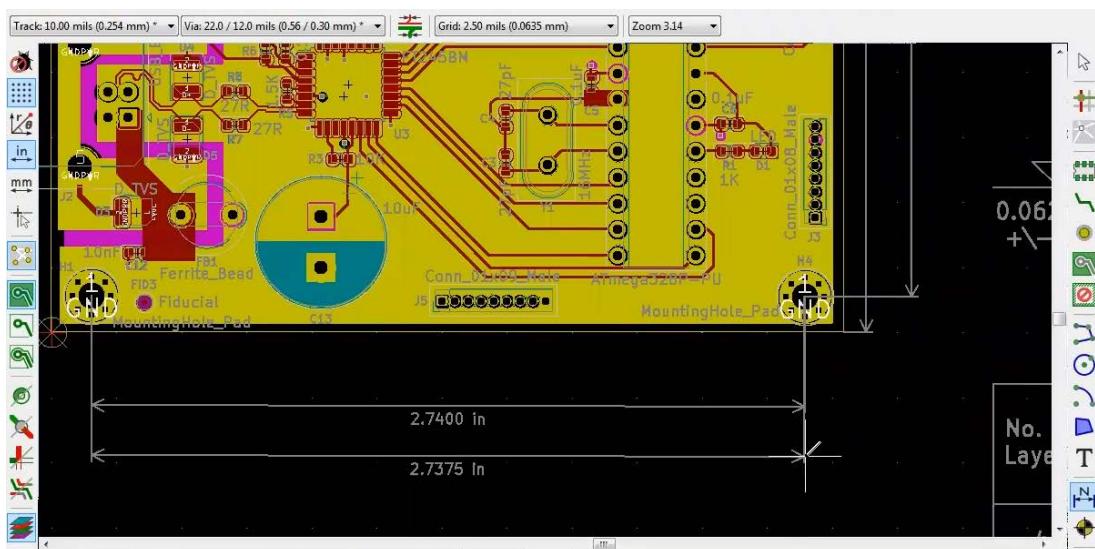


Import vector graphic file

### 6.1.3 Adding dimensions

We need to mark the dimensions of the board and then we need to mark the dimensions from one mounting hole to another. Also, the dimensions from the mounting hole to the board edge. Mark the dimensions in the fab layer. To mark, **select the add dimension icon ** and **select the fab layer**.

Move your cursor to the center of the mounting hole, a circle appears when it is at the center of the hole. Now click on that and then click on the mounting hole on the other side and pull the dimension down.



Adding dimensions

Once all dimensions have been marked, the layout of your board in the fab drawing is complete. Check if the orientation of your connectors and all components are as per the datasheet.

#### 6.1.4 Stack-up



An example of a stack-up

You need to draw the stack-up in your fab layer using the **Add graphics** tool mark all the details in the stack-up using the **Text tool**  .

The stack-up drawing should contain all the details related to the stack-up such as the thickness of each layer and the thickness of the top and bottom solder mask. The board details such as the total dimensions of the board, the thickness of the board, the tolerance of the board, etc., can also be mentioned. We get all this information from the stack-up report as we saw previously. We need to fill all this information over here in the fab drawing again

#### 6.1.5 Fab details

Fab details are written on the fab layer. In the fab details, you can point out all the required specifications and information about the board. Details include the IPC class for the board material, the number of layers in the board, solder mask color, etc. Next, there are pads on the board – plated through-hole pads and non-plated through-hole pads. The pad tolerances should be mentioned as well.

#### 6.1.6 Drill chart

There is also one more thing we need to mention in the fab drawing which is the drill chart. A drill chart describes the number of holes in the board, the different sizes of the holes, and whether they are plated or non-plated. It also mentions the number of vias being used and whether they are through-hole vias, blind vias, or buried vias along with the tolerances for all the holes.

### 6.1.7 Stack up details

To display the stack-up data on fab layers go to place> Add stackup table.

Layer Name	Type	Material	Thickness (mm)	Color	Epsilon R	Loss Tangent
F.Silkscreen	Top Silk Screen	Not specified	0 mm	Not specified	1	0
F.Paste	Top Solder Paste		0 mm		1	0
F.Mask	Top Solder Mask	Not specified	0.01 mm	Not specified	3.3	0
TOP	copper		0.035 mm		1	0
Dielectric	core	FR4	0.4716 mm		4.5	0.02
GND	copper		0.035 mm		1	0
Dielectric	prepreg	FR4	0.4716 mm		4.5	0.02
POWER	copper		0.035 mm		1	0
Dielectric	core	FR4	0.4716 mm		4.5	0.02
BOTTOM	copper		0.035 mm		1	0
B.Mask	Bottom Solder Mask	Not specified	0.01 mm	Not specified	3.3	0
B.Paste	Bottom Solder Paste		0 mm		1	0
B.Silkscreen	Bottom Silk Screen	Not specified	0 mm	Not specified	1	0

Stack-up table

## 6.2 Files required for PCB manufacturing

Once we are done with the design part, the Gerber files should be generated for the manufacturer. Gerber is an open and universally accepted file format used to export PCB designs. In this demo, we are exporting the design in the Gerber format.

The set of files that are required for the production of PCBs are as mentioned below:

- 0- Gerbers:
  - Top etch
  - a- Bottom etch
  - c. Middle layer etch
  - d. Top and bottom silkscreen
  - e. Top and bottom paste mask
  - f. Top and bottom solder mask
  - g. Top and bottom assembly
  - h. Fab details
  - i. Board outline
2. NC drill
3. Pick and place(XY)
4. IPC 356 netlist
5. PDF:
  - a. Top and bottom assembly
  - b. Fab details
  - c. All layers
  - d. Top and bottom silkscreen
  - e. Board outline
  - f. Schematic
  - g. Stack-up report
  - h. Top and bottom solder mask (optional)
  - i. Top and bottom paste mask (optional)
6. BOM

**Gerber** files are a set of files that contains production information of each layer of a PCB. Here we are using a four-layer board, so we need to create Gerber for all four layers. The top and bottom silkscreen, top and bottom paste mask, top, and bottom solder mask, top, and bottom assembly layer should be mentioned in the fab details. You have to create Gerber files for each layer used in your design.

Next, we need an NC-drill file. An **NC drill file** gives information about all the holes required in the board. It will serve as an input to the drill machine to drill the required holes on the board.

**Pick and place** files are used by the machine to identify the location of the various components on the board using coordinates.

**IPC-356 netlist** files have information about the connections between various components. Once you create your netlist make sure that it matches with your schematic netlist.

Next, we will need **PDFs**. Generate PDFs for top and bottom assembly, fab details, all the layers of your board, the schematic, and a stack-up report.

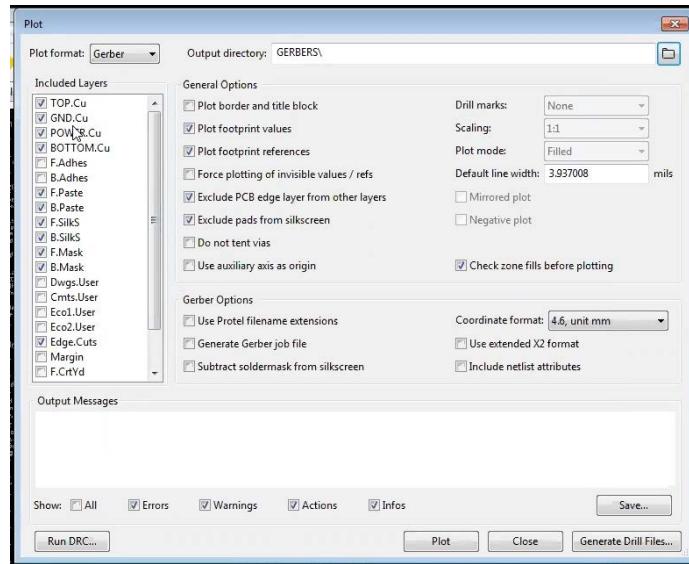
Finally, we need a **BOM**. The BOM consists of all of the components which will be used in our board and the manufacturer part numbers that will help the PCB manufacturer in identifying the components.

Now you know the files that you need to generate to manufacture your design. Let's check how these files are generated in KiCad.

### 6.3 Exporting Gerber

Now let's generate the Gerber for our design. To do that,

- Select the Plot icon  from the toolbar on the top.
- A window will popup.



Exporting Gerber

### 6.3.1 Required settings in the Plot popup window

- Select the **Gerber** from **Plot format**.
- Select the **Output directory** where you want to save the file. – First, make a separate folder for Gerber in your files.
- Select all the layers for which you need Gerber files, from **Included layers**. If you are creating a 4 layer PCB select all four layers. Apart from that, select “forward paste”, “stop paste”, “bottom paste”, “forward silk”, “bottom silk”, “forward solder mask”, and “bottom solder mask”. Here we want an “Edge cut layer” which is the actual boundary of the PCB and fab layers, which are “front fab” and “bottom fab”. These are the minimum number of layers we require for generating Gerber.
- Select the **General options** that are required.
- Deselect **Plot footprint values** because normally we don't plot values of the components of the board. But we plot the references so select **Plot footprint references**.
- Select **Exclude PCB edge layer from other layers** and **Exclude pads from your silkscreen**.
- Choose **Select Use Protel filename extensions** in **Gerber options**.

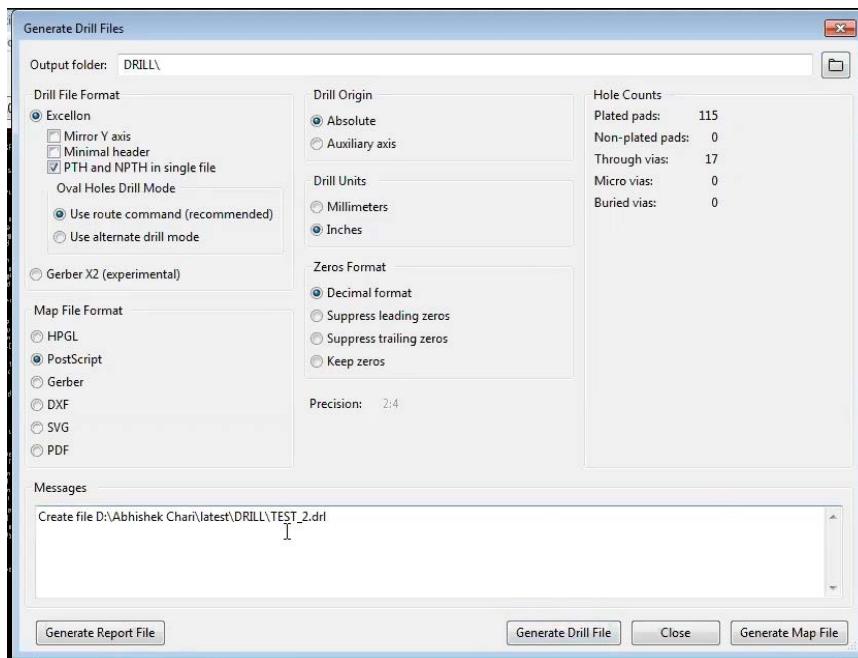
Once these settings are done, we are ready to plot our Gerber. Now **Click Plot**.

Your files will be saved in your selected folder. It will be displayed in the “**Output message**” section in the above popup. Don't close the popup!

## 6.4 Exporting drill files

Now we need to create the drill file. Once you complete generating the Gerber click on **Generate Drill File**

A new window will popup.



Choose all required settings from the window.

### 6.4.1 Required settings in the Generate Drill File popup window

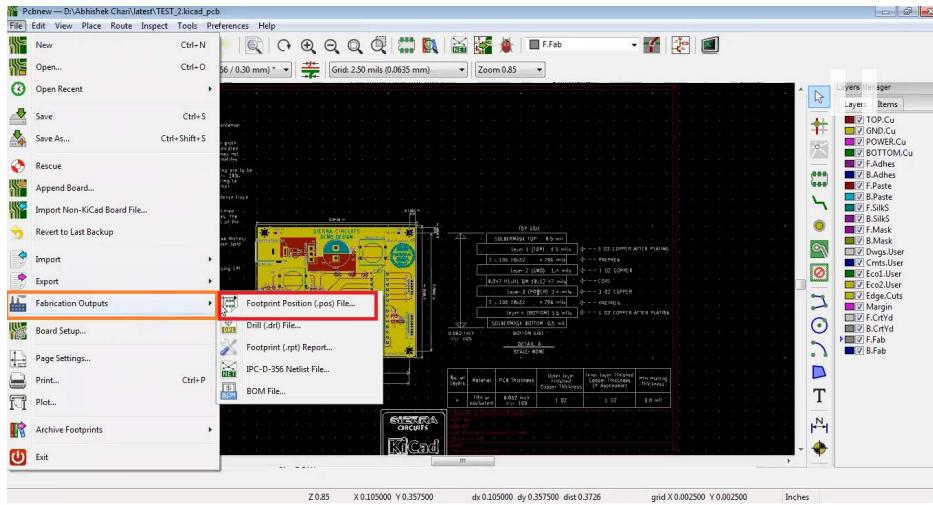
- Select the **Output folder** where you want to save the file. Save the drill file in a separate drill folder.
- Make sure you have selected "**PTH and NPTH in single file**" (recommended).
- Select the Drill units as "Inches".

Once the settings are done, **click Generate Drill File**.

We need to create a map file as well, which gives you the dimensions of holes in the drill chart. Click on **Generate Map File** for that. Save the map file in the Gerber location. After the files are saved, you will receive a notification in the **Messages** box about the saved file.

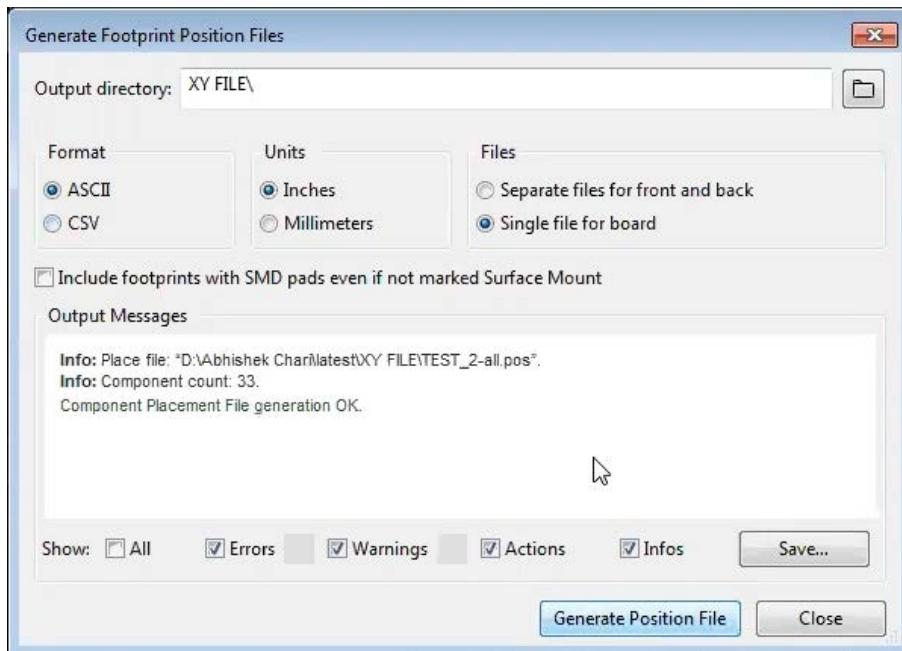
## 6.5 Generation of pick and place files

- Go to File > Fabrication Outputs > Footprint Position (.pos) file.
- A window will popup:



Generation of pick and place file

The popup window:



Generation of footprint position files

## Required settings in the popup window:

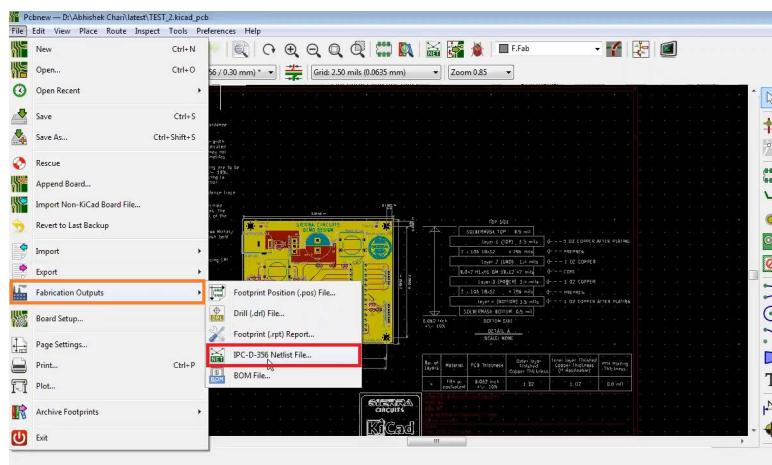
1. Select the **Output directory** where you want to save the file.
2. Under Format select ASCII. which will give you the drill locations and selecting CSV will give you drill locations in Excel format. If you want, you can generate both files.
3. Select the **Units** in inches.
4. Select **Files**. If you select **Single file for the board**, all the top and bottom components will be in a single file. To generate two different files for top and bottom layers, select **Separate files for front and back**.

Once the settings are done, to generate the position file, **click on the Generate Position File**.

After the files are saved, you will receive a notification in the **Messages** box about the saved file.

## 6.6 Generation of netlist file

Go to **file > Fabrication Output > IPC – B – 356 netlist file** 

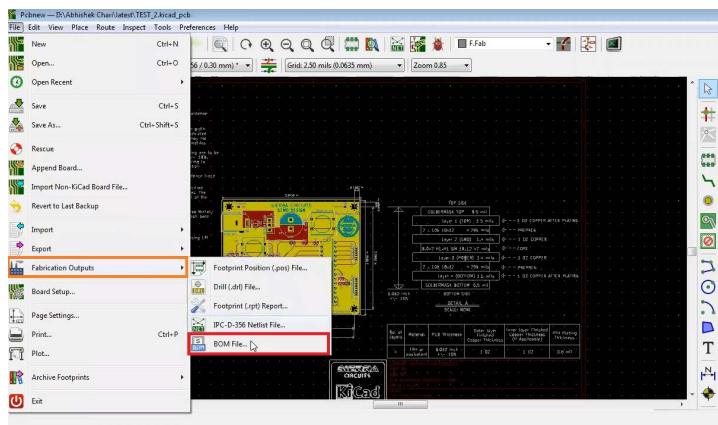


Generation of netlist file

Once you finish this step, a window will pop up in which you can choose the location to save the file. Select the location and the file will be exported to that location.

## 6.7 Generation of BOM file

Go to **file > Fabrication Output > BOM**



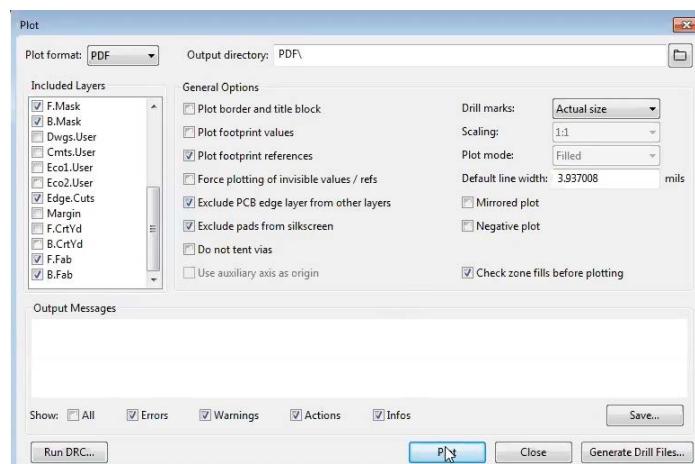
Generation of BOM

Here we need to follow the steps similar to the ones we used for netlist generation.

## 6.8 Generation of PDF

Steps to generate PDFs are similar to the ones we used while generating Gerber files with some changes needed in plot settings.

- Select Plot icon from the toolbar on the top menu.
- A window will popup:



Generation of PDF

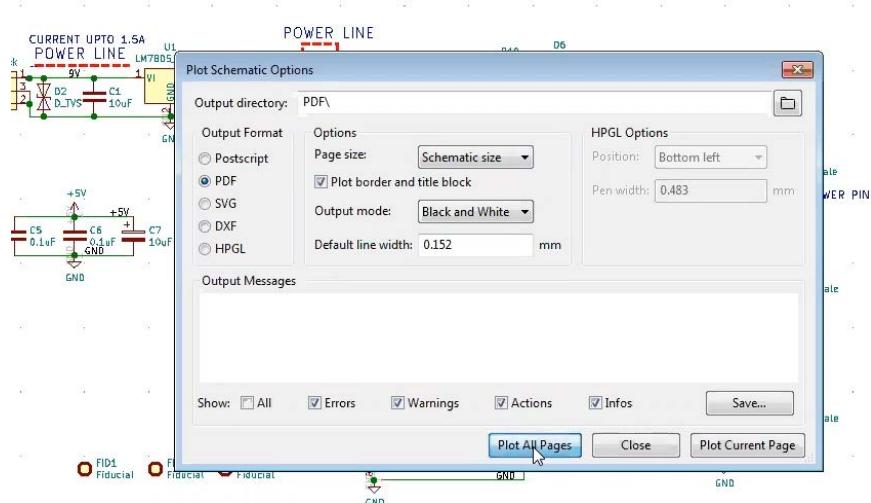
### Required settings in the popup window:

- In **Plot format**, select **PDF**.
- Select the **Output directory** where you want to save the file.
- In **Drill marks**, select the actual size.
- You can select all the layers you selected while generating Gerber files.
- **Click Plot**

## 6.9 Generation of PDF for schematic

Now go to your schematic window.

- **Select Plot** 
- **A window will popup:**



Generation of schematic in PDF

### Required settings in the popup window:

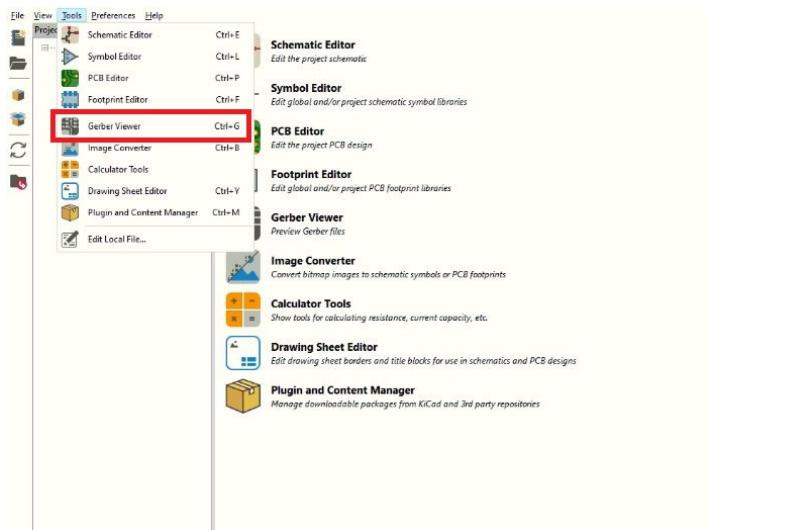
- Select the **Output directory** where you want to save the file.
- Select the **PDF** option from the **output format**.
- Choose **Schematic size** in **Page Size**.
- Select the **Output mode** either color or black & white.
- Now **click Plot All Pages**.

Your PDF will be created in the selected folder. We'll require a stack-up as well. Save the PDF of the required stack-up along with the schematic PDFs.

## 6.10 Using GerbView

Now as per this document, all of the files are created. We now need to check whether the BOM which was generated by us, was properly generated before it goes to manufacturing. To view all your Gerber files:

Go to the Project manager of the KiCad > Tools > View Gerber Files



Gerber viewer

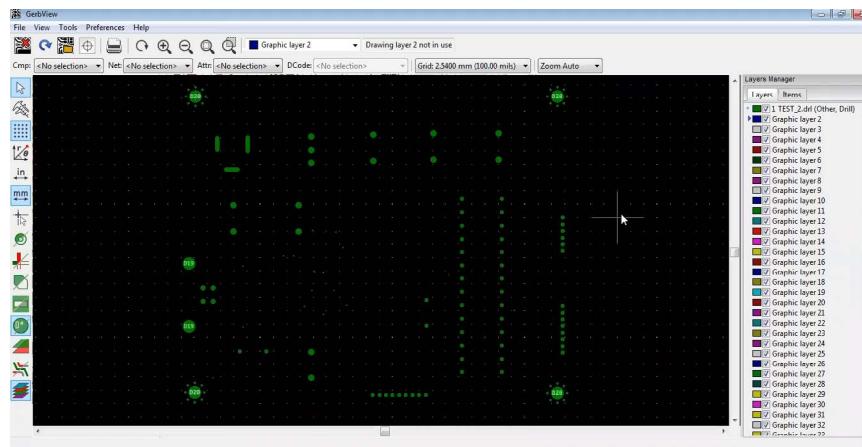
Now, the Gerbview screen will appear. You would need to see whether the Gerber and the drill file generated was properly created. For that:

Go to file > Open Excellon Drill File



Validation of Gerber and drill files

Then browse and select your drill file from the location where the drill file was saved and **click Open**.

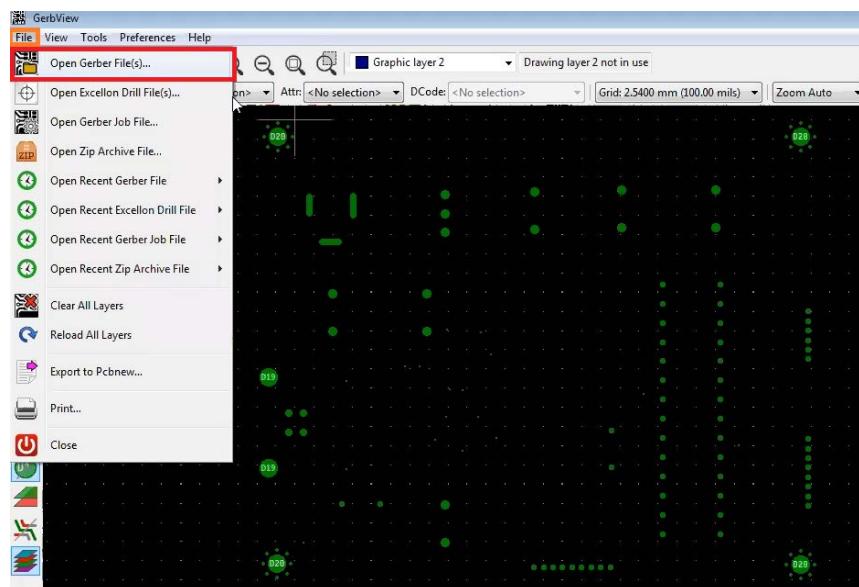


GerbView window

You will be able to see many green dots on the screen. This tells you the locations of all the holes and their dimensions. Let us see whether all the holes fit within our board. For that, we need to open the edge cut layer from the Gerber.

- **Go to file > Open Gerber file.**
- **Select the Gerber of the layer which you need to view.**

Here we need the **edge cut layer**.



Open Gerber file

Also, open the **map file** from the gerber files, so that you can view the dimensions of the holes in your Gerbview.



Map files in Gerber files

You can see the green dots which are the map of drill holes. There is a blue rectangle that covers the entire board set up which is the Edge cut layer. The drill map is displayed below the board.

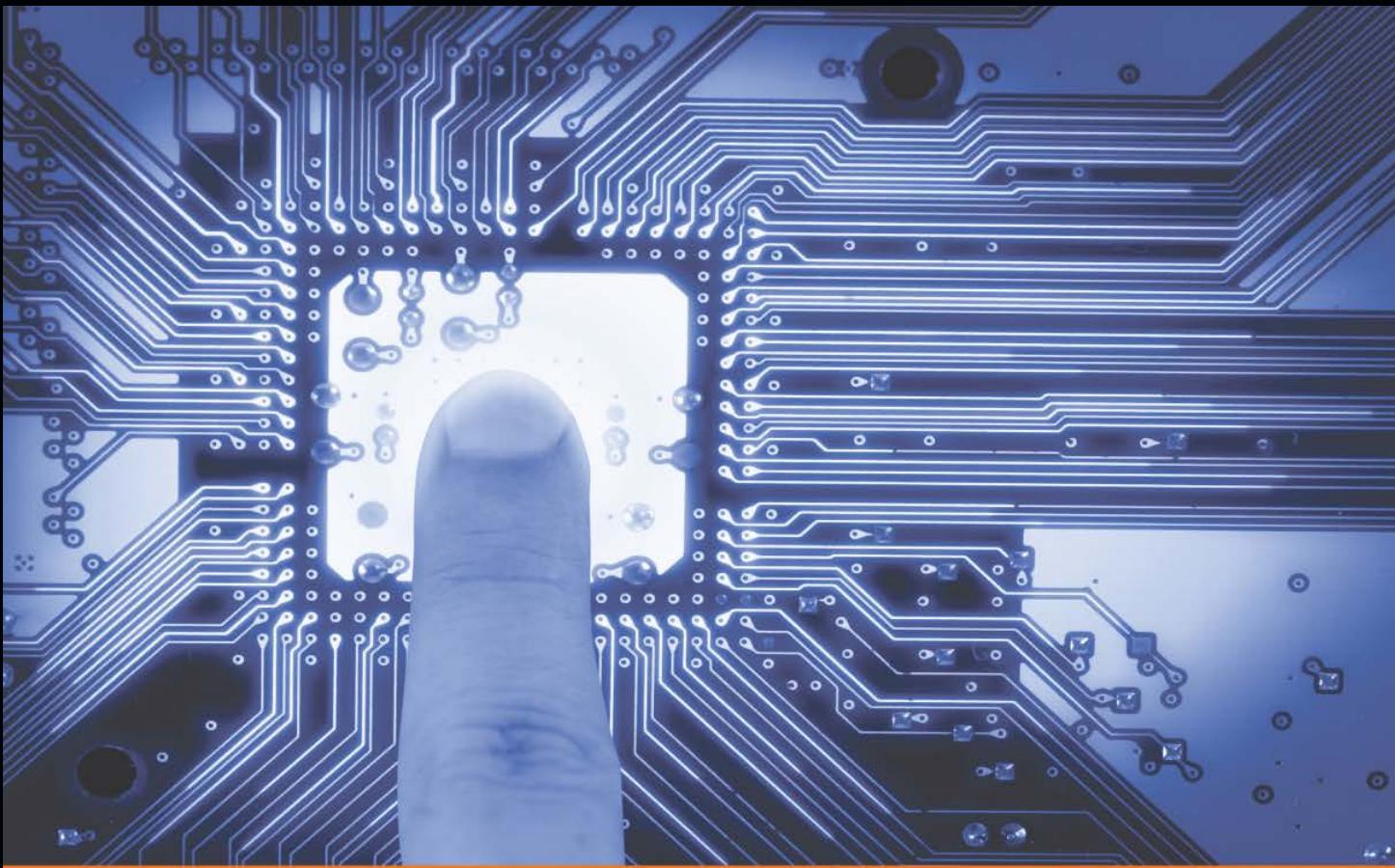
Similarly, you can select each layer from the Gerber and can check whether all the files for the board are following the layout design. For an overlapping view, you can also select several layers at a time. For example, we selected the top layer of our board as depicted in the image below:



Top layer of the board

Once your files are verified, you can send the files for manufacturing. Send the files in a single zipped compressed file format. This will ensure that there are no errors during the transmission of the files.

To summarize, we have detailed the deliverables required for board manufacturing. You will require the Gerber files, the drill chart of the map, the drill file, the placement file, along with the IPC Netlist. You need to check and verify each of your layers and the edge cuts carefully. If you find errors in Gerbview, you'll need to go back to PCB new, fix the errors in your layout, and then re-generate Gerber files using the plot tool. It is easier to fix them at this point as it won't be easy later on. Hope you found this Design Guide to be helpful.



**SIERRA  
CIRCUITS**

**Sierra Circuits**  
**1108 West Evelyn Avenue**  
**Sunnyvale, CA 94086**  
**+1 (408) 735-7137**

[www.protoexpress.com](http://www.protoexpress.com)

# Foreword



Why would we write on a topic that has been written about numerous times? What are we trying to say that is different from what has already been said? Through this design guide, we make an attempt to convey the design aspects from a manufacturing perspective and the fine details that make a big impact - not just

the theoretical, but also the practical elements. We started offering design layout services because, for our customers, it comes down to completing the project on schedule and knowing that the design will work in thousands as well as in ten-piece prototypes.

The impractical design decisions made at the initial design stage will eventually get baked into the prototype, stay through the lifecycle of the product, and inadvertently cause quality defects and/or increase costs over the long term. Yes, a design review can help! But how often do you see a PCB manufacturer at the table in a design review? And it's not a one-meeting solution.

Designing an optimum PCB that is manufacturable requires immense practical experience. This practical experience is gained from processing thousands of designs and understanding the ramifications of placing a via too close to a trace, selecting a suitable component instead of an obsolete one, designing a stack-up to reduce the EMI issues from the outset and much more.

We have tried to distill the basics of what we believe every electrical engineer and designer should perceive when it comes to high-speed PCB designing. We will address most of the important design techniques and hope that we will be able to impart something that you didn't know before.

**Amit Bahl**  
Director of Sales & Marketing at Sierra Circuits  
email: amit@protoexpress.com