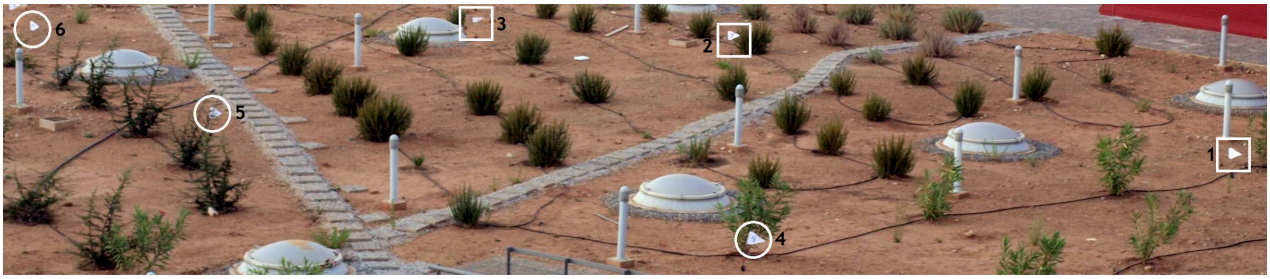
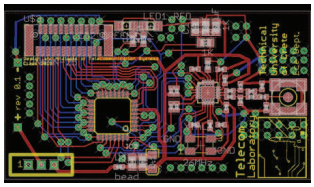
Technical Universtiy of Crete
E.C.E Department
Fall Semester 2010-2011

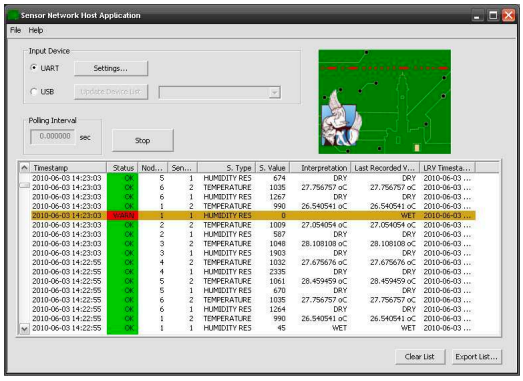
TEL 412 Analysis and Design (Synthesis) of Telecom Modules

7th Semester

Authors: Kampianakis Lefteris, Konstantinos Tountas







PCB Tutorial

Intructor: Aggelos Bletsas
aggelos@telecom.tuc.gr

version 0.1.0

Contents

1 What to expect from this manual

Before reading this manual please read the TI design rules for RF PCB design.

This tutorial provides a basic introduction to the EAGLE PCB-Design Package. It covers the use of the EAGLE Schematic Editor, Layout Editor, and Autorouter. This guide will lead you through the program in the following order, starting with the Schematic Editor module and working through to board design and autorouting.

Having completed this tutorial you should be able to start working on a serious project. While creating your initial designs, however, you should frequently use the help function and the EAGLE Reference Manual to learn more about specific details.

2 Introduction

2.1 Before Getting Started

In this tutorial we will use the free edition of EAGLE. There are some restrictions but for learning purposes it's ideal. Before we start designing we need to setup EAGLE and import the component libraries, useful keyboard shortcuts.

Things to do:

- Download Eagle itself. Versions are available for Windows, Linux, and Mac.
- Download the SparkFun Eagle Library. This is the collection of all the components SparkFun designs with and therefore components and footprints have already been tested. Unzip and put the SparkFun.lbr file into the Eagle\lbr directory.
- Download the SparkFun Eagle keyboard shortcuts. Place this file in the Eagle\scr directory. (Optional)
- Download the sfe-gerb274x.cam file needed for exporting the Eagle\cam directory.

When we are done with installing, we can start designing boards.

Note 1: For linux users, regular copy-paste won't work. Use the mv *command from the terminal.*

Note 2: The SparkFun Eagle shortcut key script file has an .scr extension. This is a common virus infiltration method. If you choose to download the keyboard shortcuts, and you don't trust us, rename the file to a .txt extension and view it in a text viewer. There's nothing there but text and Eagle commands. Just be sure to rename the file to the .scr extension so that Eagle can use it.

3 Learning to design.

3.1 Design Process

A. Design the schematic on paper.

B. Simulate the circuit or prototype on a bred-board.

C. Test.

D. If the test is successful, we proceed to design using CAD software.

Note 1: Using CAD software is the **last stage** *of of prototyping.*

Note 2: Mistakes in the first stages of the design can be very expensive!

3.2 Cadsoft EAGLE

In this class we will use the EAGLE software. It's free, easy to learn and use and is also used in the industry.

EAGLE software has 3 main functionalities:

- **Schematic Design:**
  Using this functionality, we implement the circuit design we created in the earlier stages.
- **Layout Design:**
  Using this functionality, we design the layout. (how the PCB looks)
- **CAM files exporting:**
  Using this functionality, we create the files that we send to our fab house.

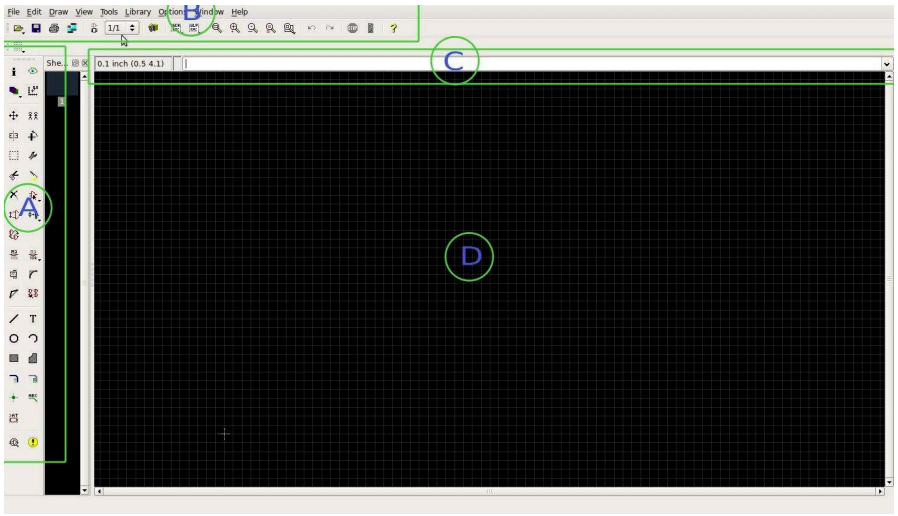
3.2.1 The Panel

A. Functions Toolbar.

B. Menu and useful functions.

C. Command Line.

D. Design panel.



Σχήμα 1: The EAGLE schematic design window.

3.2.2 Starting to design

After the first stages of the designing process are completed and we have an idea of the circuit we want to design, we can start adding components to our schematic.

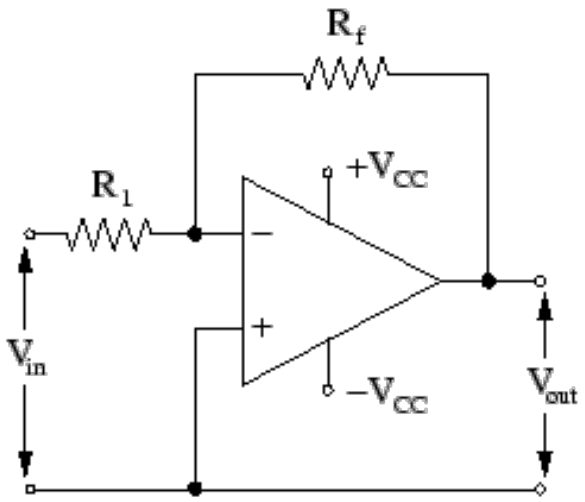
But before we start some things we need to know about the components:

- Components may exist (e.g. resistor, micro controller) or not (e.g GND, VCC)
- They have a different representation in the schematic and the layout
- Each component has package information, which describes the pads on the layout design.
- We can create our own components.

3.2.3 The Operational Amplifier

The circuit we are going to implement is an inverting operational amplifier. The gain of this amplifier is

$$A = -\frac{R_f}{R_1} \tag{1}$$



Σχήμα 2: Operational amplifier circuit.

And the circuit we are going to implement is the following:
For this implementation we want the gain to be $A = -10$, so the $R_f = 1K$ and the $R_1 = 100$.

3.2.4 Components List

All the components can be found in the Sparkfan library.

A. 2 x Resistor (smd package 1206)

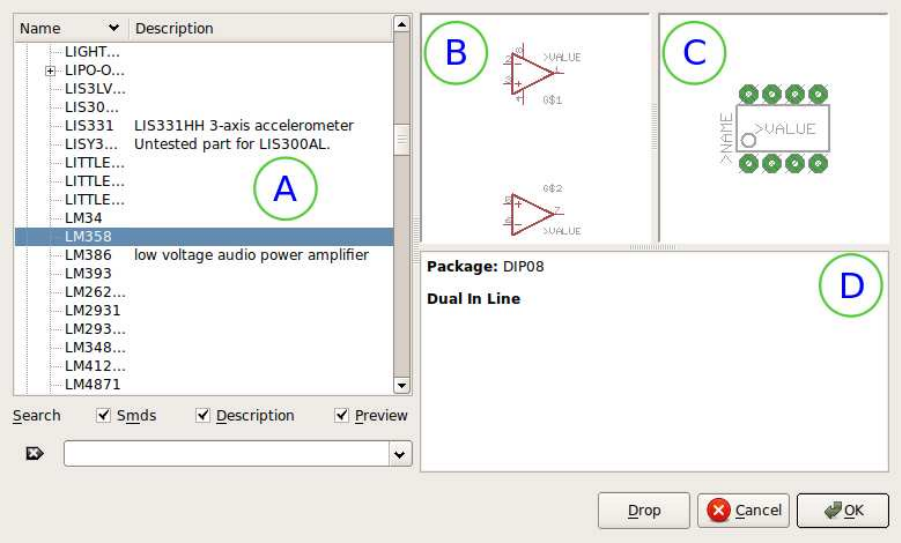
B. 1 x 3 x 1 header (M03LOCK)

C. 2 x 1 x 1 header (M01PTH)

D. GND

3.2.5 Adding components to the design

In order to insert new components to the design we use the command **ADD** in the command line. A new window will open, where we search the component we need.



Σχήμα 3: The EAGLE ADD window.

After we find the component we need, let's take a closer look on the ADD window.

A. Component List.

B. Preview of the component in the schematic design.

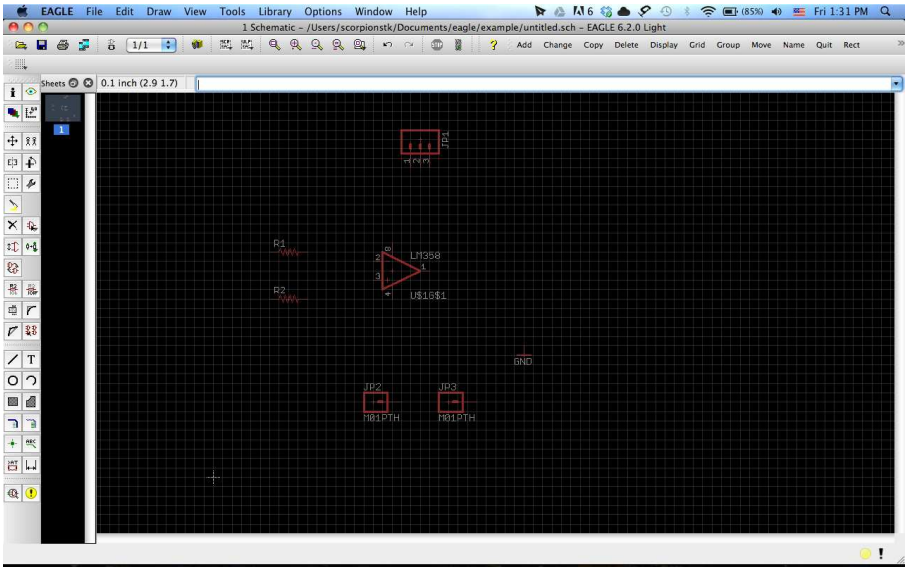
C. Preview of the component in the layout design.

D. Component description: usually contains package and component information.

We choose the component and we place it in the right position on the schematic design. In order to move the component around the schematic we use the command **move** and if we need to move many items we use the command **group**, then right click and **move group**. While moving a component we can right click in order to rotate it. Afterwards we do the same thing for all the other components.

Tip: We can either use the command line or the toolbar. The first one is faster, since we don't need to type the whole command, just 2-3 letters are enough.

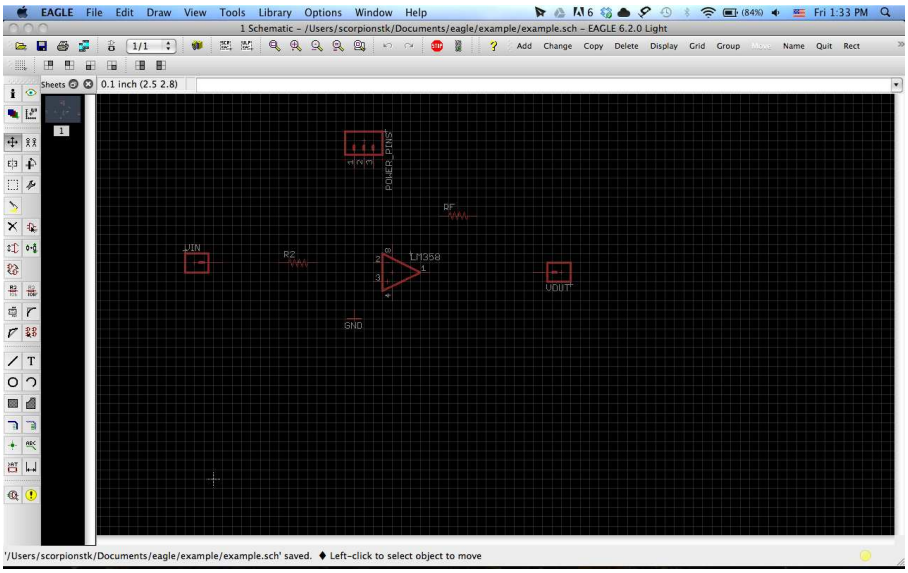
After we have all the components on the design, it should look like this:



Σχήμα 4: The schematic with the unconnected components.

3.2.6 Renaming the components and changing values

Now we can rename the components so that we can understand the schematic better. Therefore we use the command **name** in order to change the name of the component and the command **value** in order to change the value of a component. e.g $R_2 \rightarrow R_f$, $J1 \rightarrow Vin$.



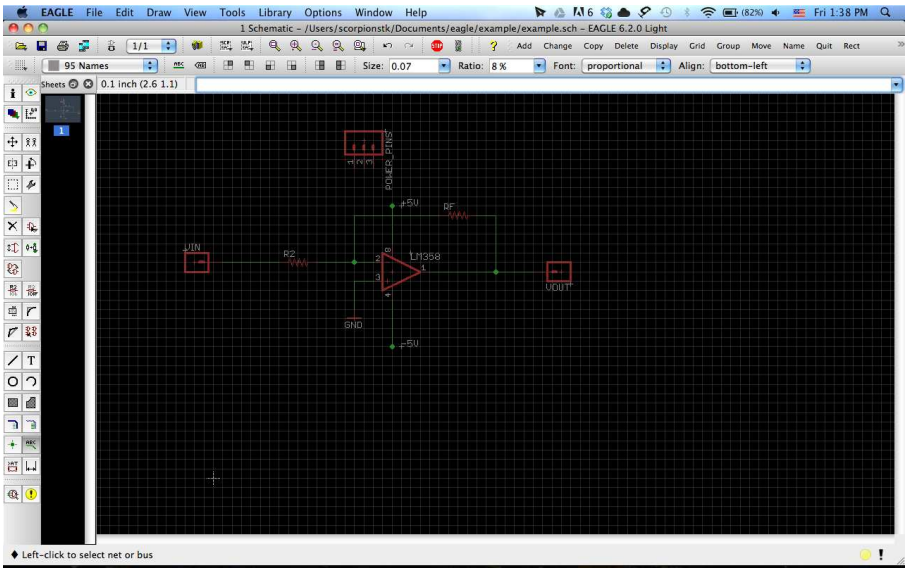
Σχήμα 5: The schematic after renaming the components.

3.2.7 Connecting the components

Now all we have to do is connect the components according to our circuit schematic. We do that using the **wire** command.

Tip: It's good practice to use a junction when wires meet if we want them connected.

Tip: In order to check if a component is connect to a wire we use the **move** *command.* On the following photo we have connected everything but the 3 x 1 connector.

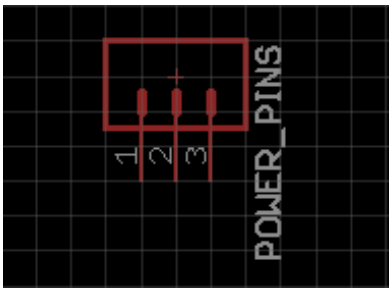


Σχήμα 6: The schematic with connected components.

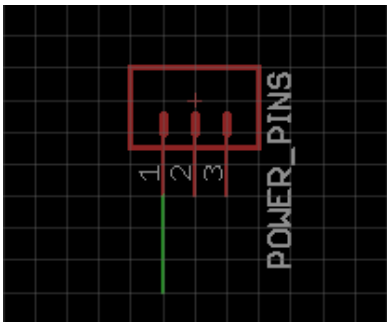
Sometimes we need to connect components that are away from each other or in a position that is not easy for the wire to reach. In order to connect them we create an imaginary connection.

To create the imaginary connection:

A. We use the wire command to create a small wire from the pin we want to connect.

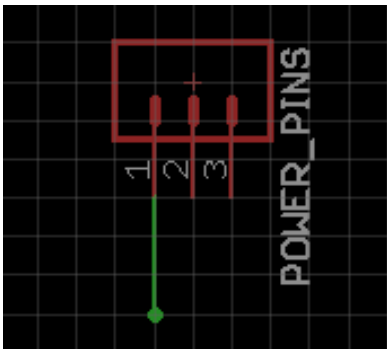


Σχήμα 7: Component before connecting the wire.



Σχήμα 8: Component after connecting the wire.

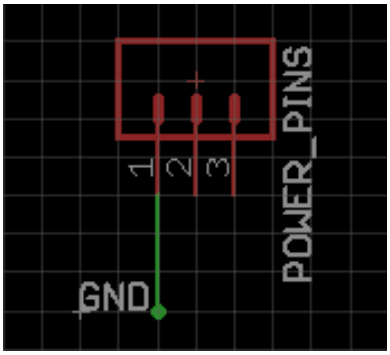
B. We put the junction at the end of the wire.



Σχήμα 9: Wire with junction.

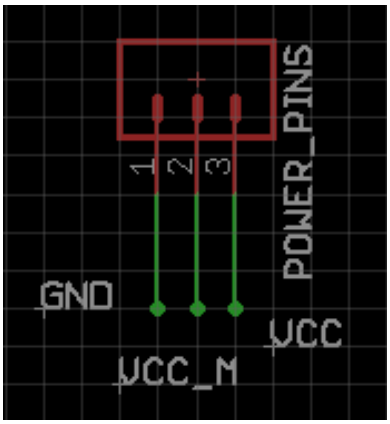
C. We use the name command on the junction (or the wire) in order to name the net. e.g if we name it GND it will connect to the Ground.

D. We use the label command to label the junction in order to remember, where it's connected.



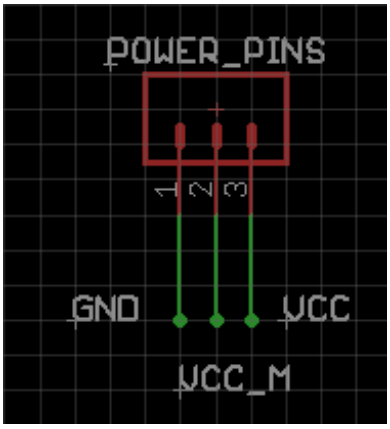
Σχήμα 10: Pin connected to GND.

We follow the aforementioned steps for each of the pins of the connector. The connector should look like this:



Σχήμα 11: All pins connected.

Tip: If the position of the name or the label of a component is not convenient, we can use the command **smash** *in order to change their position.*



Σχήμα 12: More beautiful and convenient way of design.

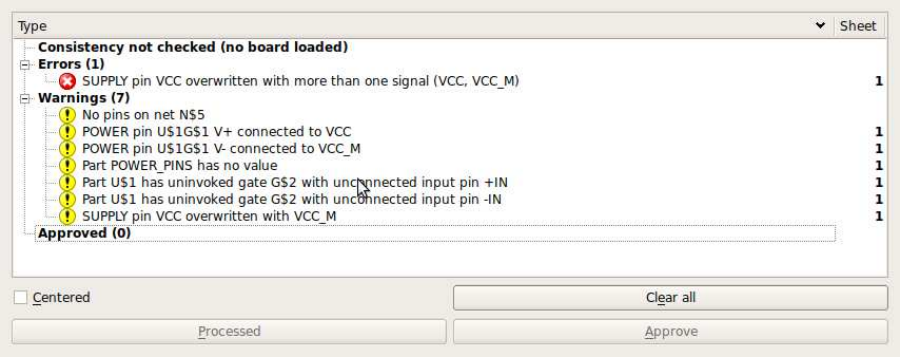
3.3 Connection checking

Next we must check if the components are connected correctly. In order to do it we use the command **show** on the wire we want to check. All the wires and components connected to the wire are highlighted. This way we can check for short circuits or unwanted connections.

3.3.1 Electrical Rules Check (ERC)

This is the last and most important step of the schematic design. We perform an ERC check to find all the errors and warnings. We need to check if we either need to change our design or approve them and continue to the layout design.

As you can see bellow we get an error about our negative power source. We can approve it because it's not a problem in the design, we just don't have a component for a negative power source so the ERC check thinks we are mistaken.



Σχήμα 13: ERC Check window.

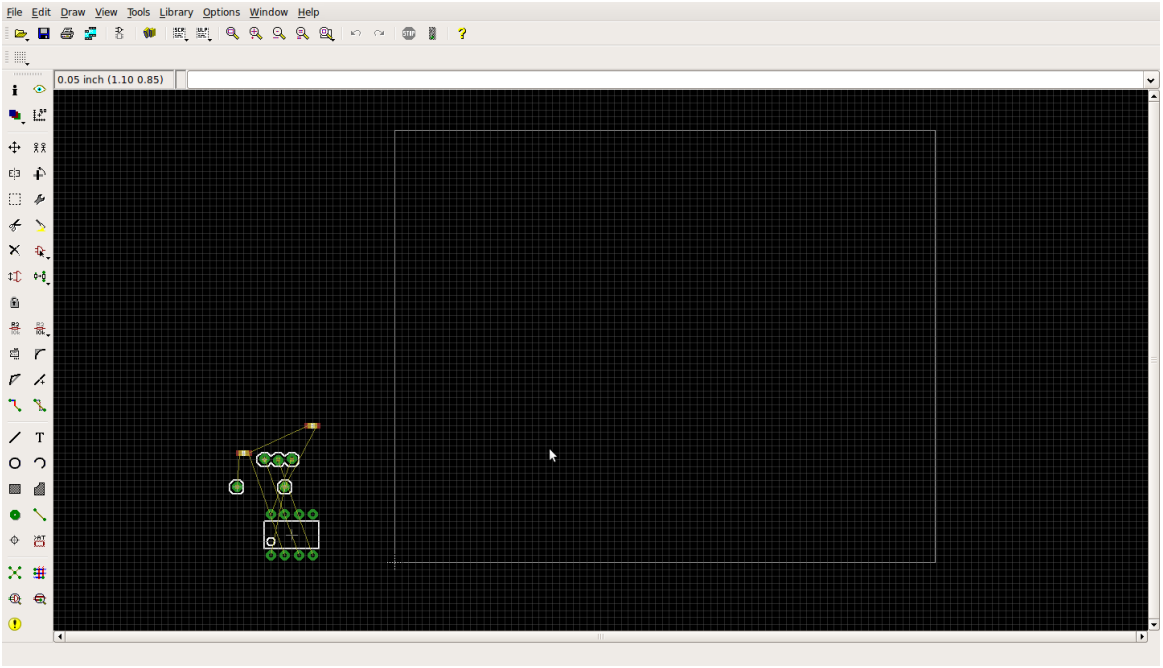
3.4 Layout Design

Now we start the Layout Design, by clicking the **board** button on the Menu.

Σχήμα 14: Board button.

We click yes on the window that opens and a new window opens. This is the layout design window. Here the components are represented by their footprints, which are like the pads that are going to be on the actual PCB (on a different scale). The yellow lines are the connections between the pads of the components. Now we need to move the components on the board and figure out the dimensions of the white lines. The white lines are the shape that the final board will have.

Tip: We can create shapes other than rectangles, but the other shapes are more expensive.



Σχήμα 15: The layout design window.

The layout design has many layers. In this free edition of EAGLE we can use only 2 layers, top and bottom, where we can place components.

3.4.1 Basic items used in layout design

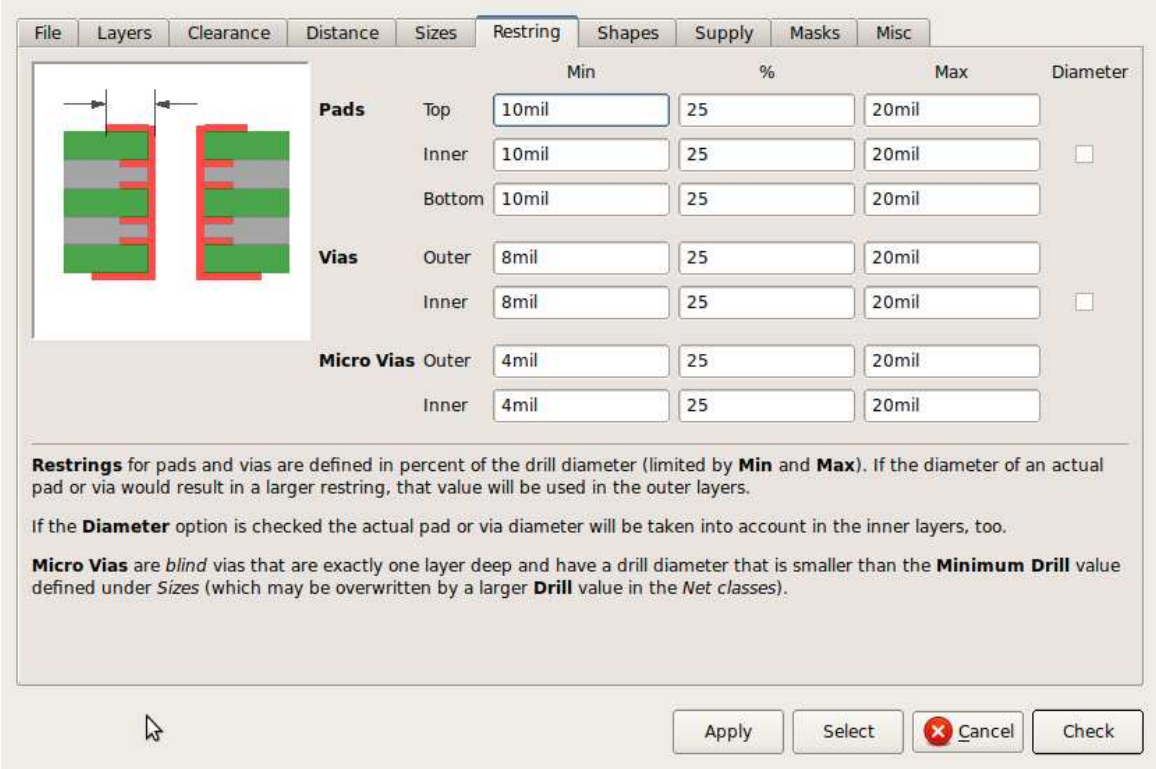
- **Pad:** pieces of copper, where the components are placed (found in SMD components.
- **Via:** copper holes on the board, used to connect the top and the bottom layer.

- **Drill Holes:** holes on the board, used for supporting the components.
- **Wires:** wires that connect the pads and the vias. They are the physical connection between the components.

3.4.2 Rules Check (DRC)

Before designing the layout, we need to configure the DRC. We use the command **DRC** on the command window. The rules are given by the fab house and we can't change them.

The DRC rules are essential for checking the design and for using the **Auto-route** tool.

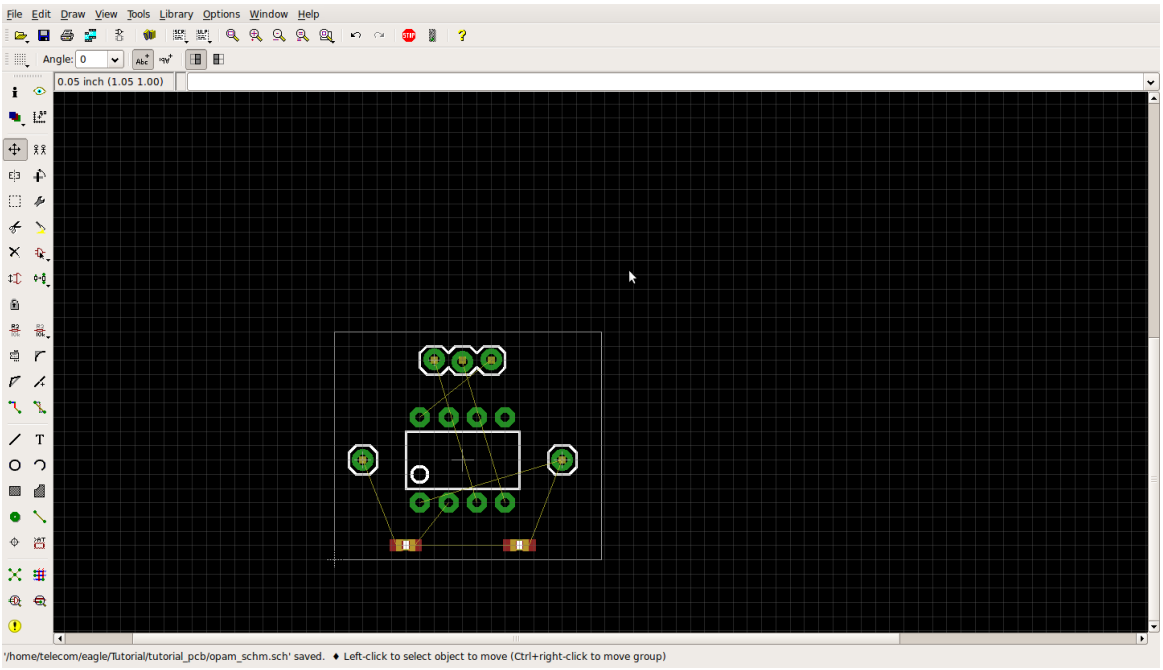


Σχήμα 16: DRC window.

3.4.3 Designing the layout

After the DRC is configured we can start placing the components on the board.

Tip: It's good practice to design the board on paper first, in order to get a better idea of where to place the components before you design it with EAGLE.



Σχήμα 17: Layout with all the footprints placed.

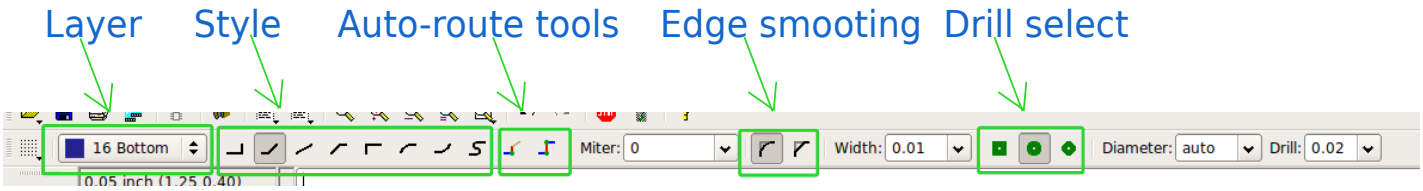
3.4.4 Routing: Connecting the pads

When the components are placed we need to connect the pads, as the yellow lines are connected. We have two choices:

- Auto-routing
- Manual-routing

Auto Routing: For Auto routing, we use the Auto Route button from the toolbar.

Manual Routing: For manual routing, we use the command **route**. The following toolbar appears.



Σχήμα 18: The Rsoute toolbar.

From this toolbar we choose the layer on which we want to design and the style of the wire.

A very useful tool is the **follow me** router, with which EAGLE tries to follow the mouse while routing and find an optimal path to it, given the user constrains on the DRC.

Routing and layout design is a very difficult process that takes time to master it, so don't get discouraged if you fail a couple of times.

Other useful tools used in layout design:

- **rip:** Deletes the routes from the layout. Be careful and don't use **delete** for that.
- **ratsnest:** Calculates the shortest possible wires and polygons.
- **group:** groups components and wires.
- **display:** Choose layers. Very useful for checking the top-bottom layer.
- **mark:** Places a point reference on the layout in order to count distances.
- **grid:** changes the grid of the layout and the units of distance.

3.4.5 Designing the ground plane

The ground plane is a copper layer, placed under the top and bottom layer in order to minimize the noise generated by the circuit.

In order to create the ground plane we do the following:

A. Use the polygon command.

B. Design a polygon on both layers (top, bottom).

C. name each polygon GND.

D. use the command rat.

3.4.6 DRC again

After the design is over we have to do another DRC check. If everything is placed and connected correctly there should be no problems and we could continue to the **gerber** files export.

3.5 CAM files export.

Creating the gerber files:

- Download Cam Script from SparkFun.
- Open Cam processor.
- File → Open → Job.
- Select "sfe-gerb274x.cam".

- Go to Drill File tab.
- Change File from ".TXT" to ".DRD".
- Process Job.
- Files created:

  A. .GTS - Top Solder Mask
  B. .GTP - Top Solder Paste
  C. .GTO - Top Silkscreen
  D. .GTL - Top Copper
  E. .gpi - Apperture File
  F. .GBS - Bottom Solder Mask
  G. .GBL - Bottom Copper
  H. .GBO - Bottom SilkScreen
  I. .DRD - Drill File (2:4 Leading)
  J. .dri - Drill Station Info File

- Copy the above only to a folder other than root (e.g Gerbers)
- Zip and and send the file to the fabhouse including a readme file with a list of corresponding files and explanation(see above)

Always check the CAM files:
View Exported Files:

- Download viewplot setup
- Install and run viewplot.
- File $\rightarrow$ Load files.
- Select all the output files except .DRD and .DRI.
- Click OK.
- Click Read apertures.
- Click OK.
- Load .DRD file.
- Click OK.
- Select 2 4 Number format and Leading zero suppression.

- Click OK.
- You should be seeing your pcb, check for errors

Always print a 1:1 version of your PCB and check if the components can be placed on the footprints.

4 Creating new components

Sometimes you can't find the part you need in an existing library though, and it's time to make your own. Making a new components in Cadsoft Eagle can be intimidating for new users, but this guide shows how we make our own Eagle components step by step. To start creating your own components, from the EAGLE control panel go to *File* → *New* → *Library*. This will open the component design window.

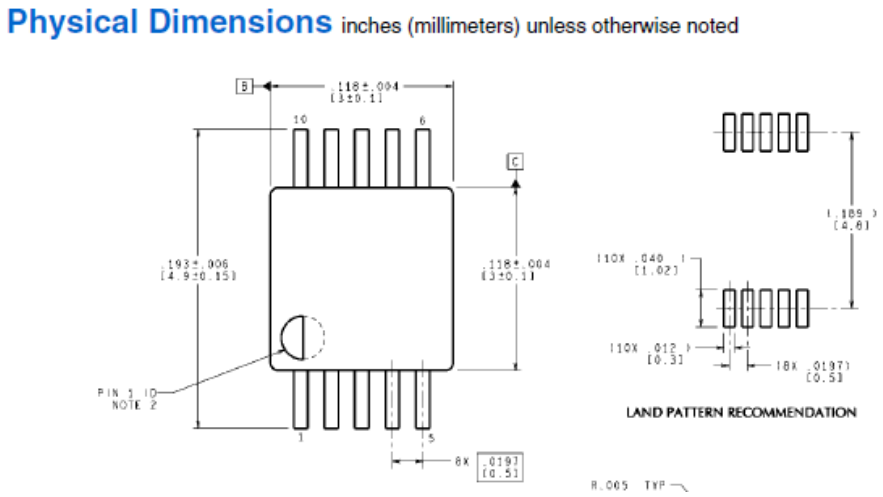
4.1 The part design process

A component is composed of 3 sections:

- Symbol - the thing you connect to other parts on the schematic
- Package - the footprint on the PCB that you attach the part to
- Device - a symbol and a package together. Symbol connections are assigned to package pins

4.2 Example: LM5022

Let's take the LM5022 as an example, a simple 10 pin chip. The datasheet has the packaging dimensions we need to build the footprint.

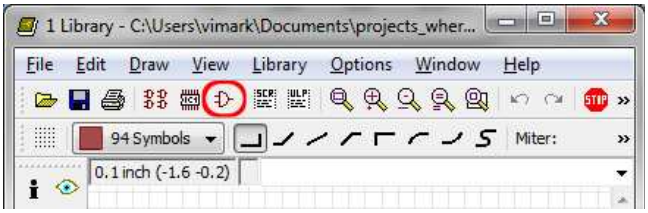


Σχήμα 19: LM5022 diagram.

It's very important to have the landing pattern and not just the pin dimensions. Some datasheets will only give the pin size, the landing pattern is larger and optimized to help the chip solder easily.

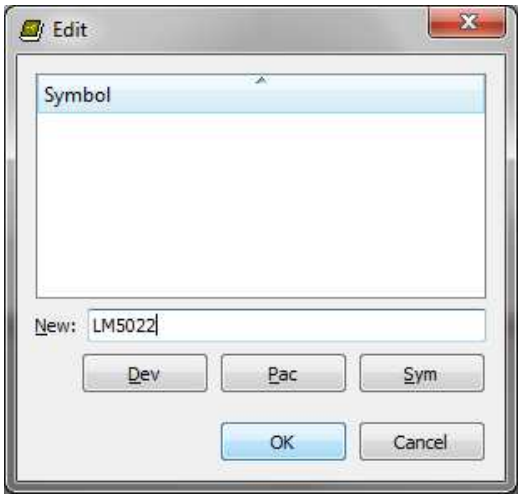
4.3 Making the symbol

Let's start with the symbol. Symbols are what you connect to other parts in the schematic diagram. We will design this one as simple as a block with pins.



Σχήμα 20: The component Design toolbar.

Create a new symbol by clicking the symbol icon in the library editor.



Σχήμα 21: Edit symbol window.

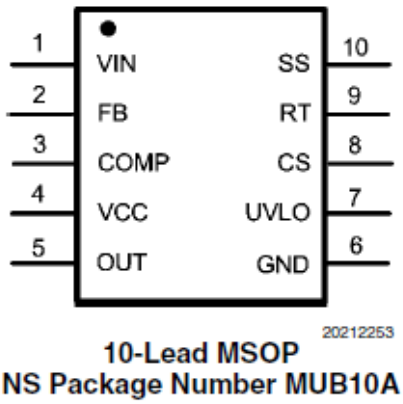
A window will open and ask for a symbol name. Enter the chip name or part number and click OK, then click Yes when asked to create a new symbol.

In the chip's datasheet we can find the pinout or connection diagram, this will be the basis for our symbol.

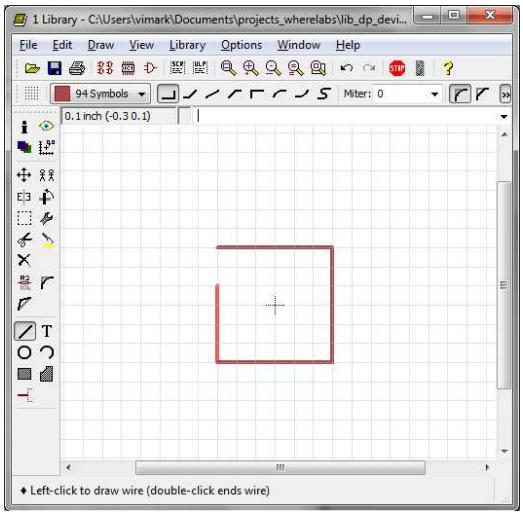
Start making the symbol by using the **wire** command in the 94Symbol layer as an outline, ICs are rectangular so we just make a simple box to represent the chip. Always put the symbol on its center (0 0) as this will be your anchor when moving in the schematic.

Place pins on the chip. Click the pin button, then click on the symbol to drop pins.

Hint: When placing the pins just leave the current settings for the grid (0.1inch), this makes the connection easier to snap when working with the device later.



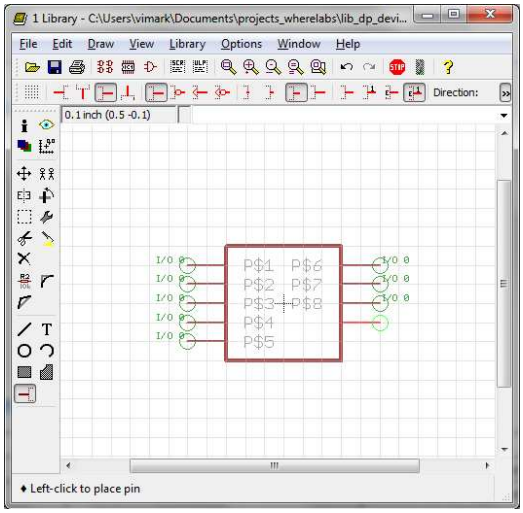
Σχήμα 22: LM5022 connection diagram.



Σχήμα 23: Making the symbol step 1.

Name each pin by using the **info** command, and clicking on the pin. Name the pins according to the datasheet to make your life easier later. You can tweak other pin attributes in the info window too.

Add a space for the part designation on the 95Names layer. This is where the part number such as IC1, C3, or R2 will appear on the schematic.



Σχήμα 24: Making the symbol step 2.

Σχήμα 25: Making the symbol step 3.

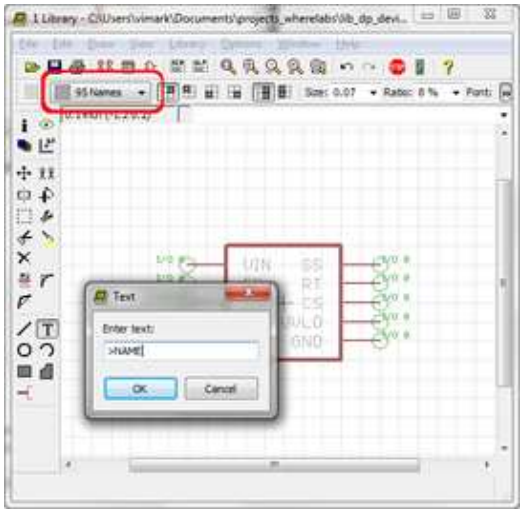
You can also place the **value** label for the part on the 96Values layer. We don't usually assign a value because we usually change it in the schematic design phase.

4.4 Making the footprint

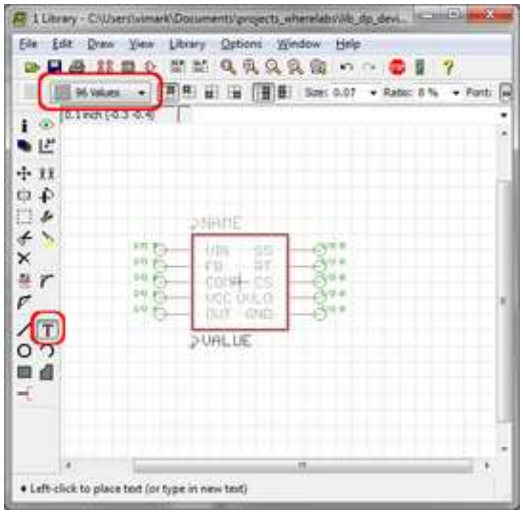
Footprints, sometimes called landing patterns, define where a chip attaches to a circuit board. The footprints will be the copper or silver shiny pads that you will solder parts on later. Eagle calls this a **package**.

To access the package editor click the **package** icon or the **symbol** icon and then choose Pac.

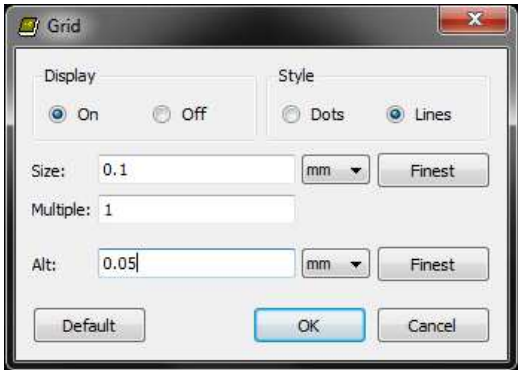
Name the package, this time by the package type which is MSOP-10 for this chip.



Σχήμα 26: Making the symbol step 4.



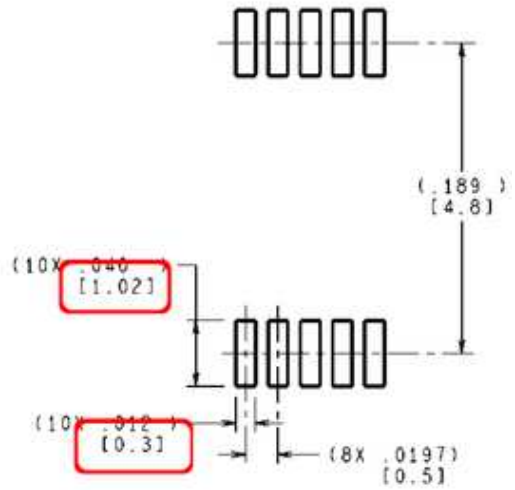
Σχήμα 27: Making the symbol step 5.



Σχήμα 28: Configuring the grid.

Time to configure the grid, by using the **grid** command. The datasheet gives the units in

inch and mm. We'll go for the metric unit, but you could also use inches.

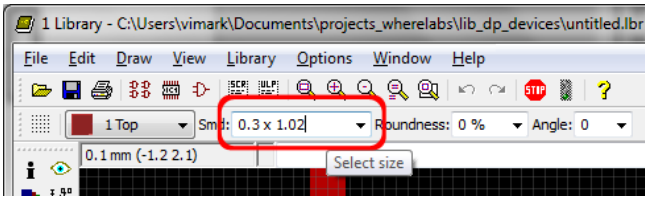


Σχήμα 29: Pads on the datasheet.

Check the datasheet for the dimensions of the recommended pin pads. We'll use this to make the optimal pad in our footprint.

Here, the manufacturer recommends 0.3 x 1.02. This area will be slightly larger than the pin itself, and is optimized for the package type.

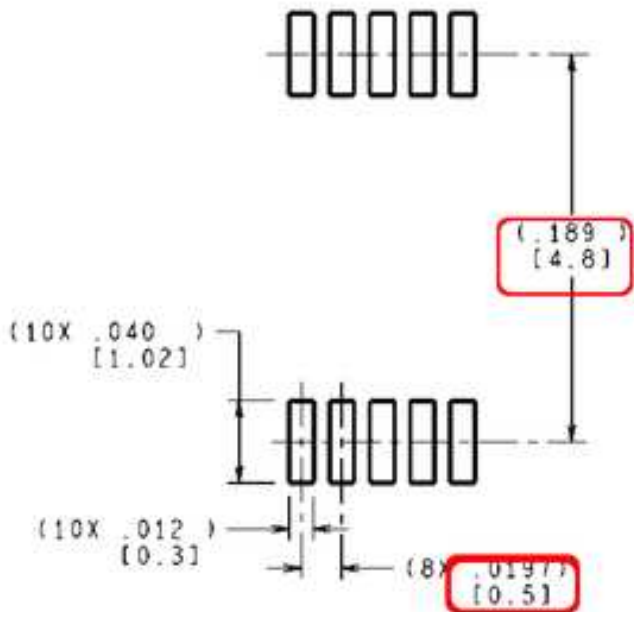
Important: It's very important to have the landing pattern and not just the pin dimensions. Some datasheets will only give the pin size, the landing pattern is larger and optimized to help the chip solder easily.



Σχήμα 30: SMD toolbar.

This is a surface mount part, so we click **SMD** or use the **SMD** command to get the right pad. For through-hole parts click the round green via next to the SMD button.

Set the dimension of the pad to the values we pulled from the datasheet.



Σχήμα 31: SMD toolbar.

There are many ways to place the pad in the correct position. You can place it manually if the grid you set (View ¿ Grid) fits the pad pitch. Alternately you can calculate the correct pad coordinates, we'll use this method.

What we're going to do is calculate the location of the center of each pad using the values from the datasheet. This seems a little tedious, but we think it's easier than messing with the grid values constantly.

The formula for the X axis is $Px = (n-1)Dpx$ where:

- n is the order of the pad from the center
- Dpx is the major pitch distance of each pad

For Pad 1 X (3rd pad from X center):

Pad1X=(3-1)0.5 will be 1. Since pad 1 is located on the 3rd quadrant its X value will be negative hence -1.

The formula for the Y axis is $Py = Dpy/2$ where:

- Dpy is the minor pitch distance of each pads

For Pad 1 Y

Pad1Y=4.8/2 will be 2.4. The lower pads are below the X axis, and should be negative. The Y location of Pad1 is Pad1Y=-2.4. On this chip pins 1-5 will have negative Y axis value (-2.4), and pins 6-10 will have a positive Y axis value (2.4).

- Pad2X=(2-1)0.5, and it's still on the 3rd quadrant so the value is -0.5
- Pad2Y=(4.8)/2, still below X, -2.4
- Pad3X=(1-1)0.5, yields 0
- Pad3Y=(4.8/2), still below X, -2.4
- Pad4X=(2-1)0.5, now we're on the fourth quadrant, which is on the positive side of X axis. This will be positive 0.5
- Pad4Y=(4.8/2). still at -2.4

Now we have: Pad1 at (-1 -2.4), Pad2 at (-0.5 -2.4), Pad3 at(0 -2.4), and Pad4(0.5 -2.4). Repeat for each pad. Remember that the Y values only need to be calculated once.

Coordinates: (-1 -2.4)(-.5 -2.4)(0 -2.4)(.5 -2.4)(1 -2.4)(1 2.4)(.5 2.4)(0 2.4)(-.5 2.4)(-1 2.4);

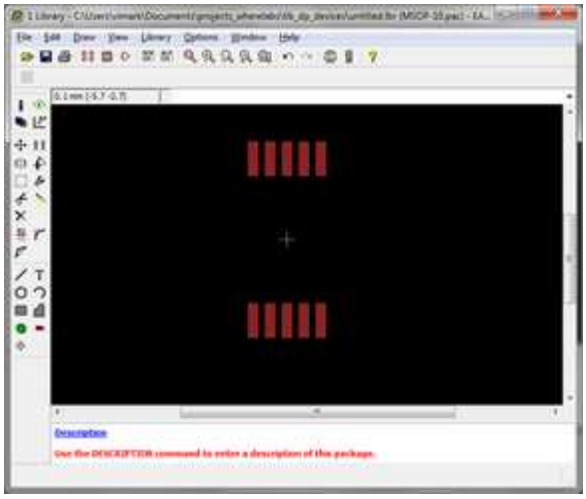
With these coordinates we can place the pads easily.

A. Click on the SMD pad button

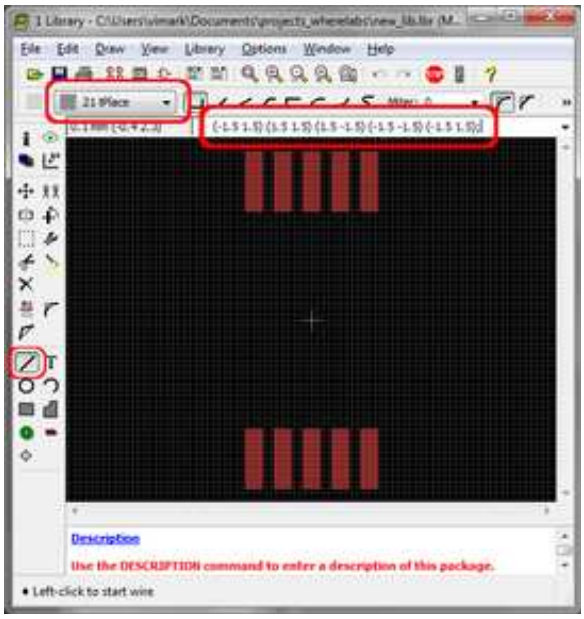
B. Enter the coordinates into the command box at the top of the library editor as shown

C. Press enter. All pads are now in place

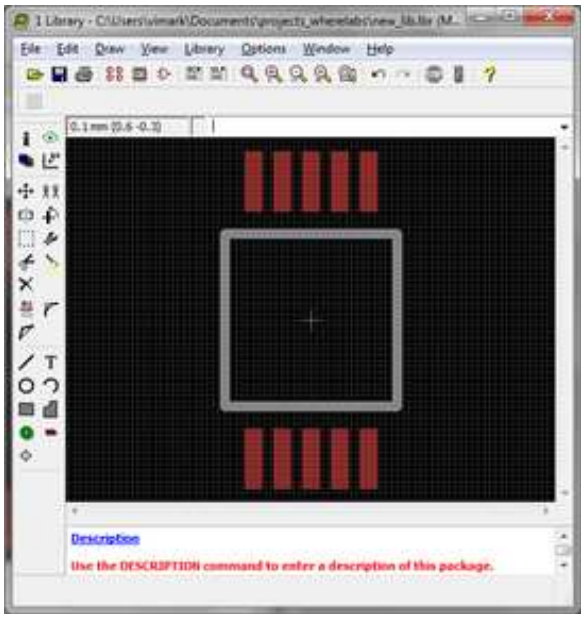
Σχήμα 32: Placing the pads.



Σχήμα 33: All pads placed.



Σχήμα 34: Creating the outline.



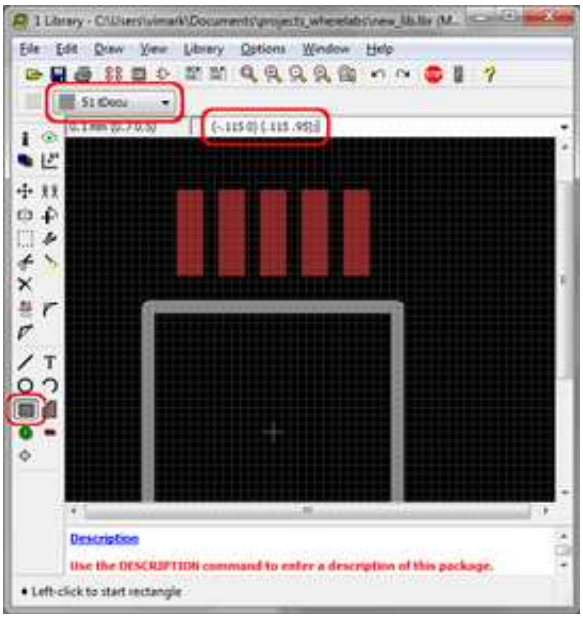
Σχήμα 35: Outline Done.

The order of the coordinates in the command box assigns numbering to the pads. Its not necessary to do it like that. You can always use the 'info' command and edit the pad name or fix mistakes.

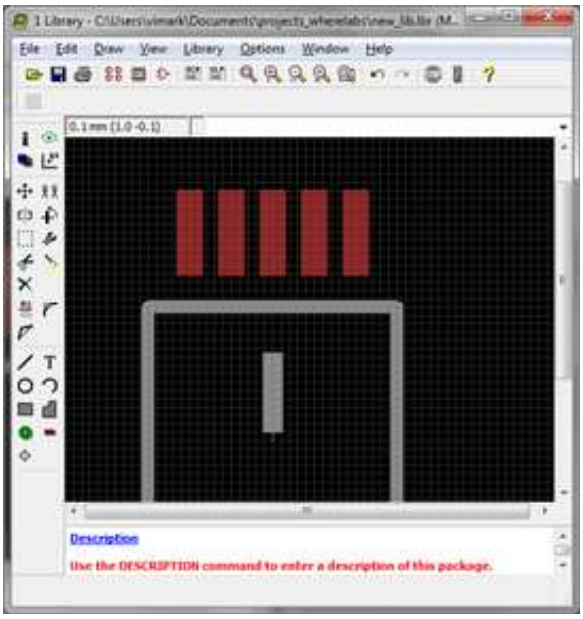
Now we'll make the chip package and pin outline, again using the values from the datasheet. This is a 3x3 mm chip with 0.23x.95 sized pins.

Use the **wire** command to draw the outline on the top silk **21tPlace** layer.

We used the package dimensions in the datasheet to calculate the coordinates of the chip's 4 edges.



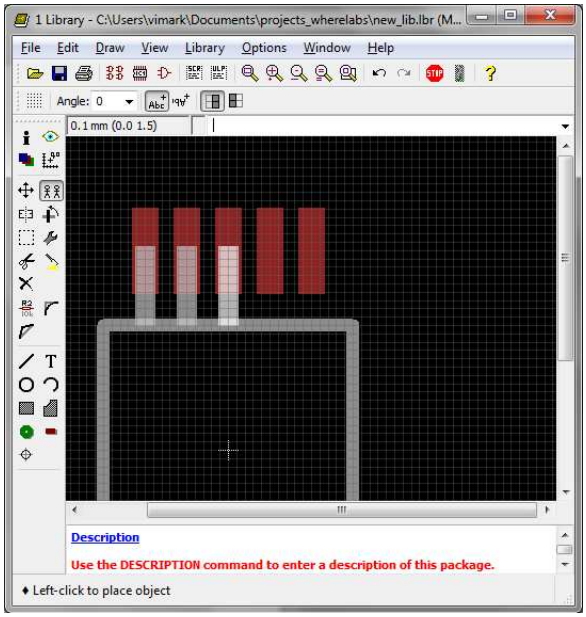
Σχήμα 36: Creating the outline.



Σχήμα 37: Outline Done.

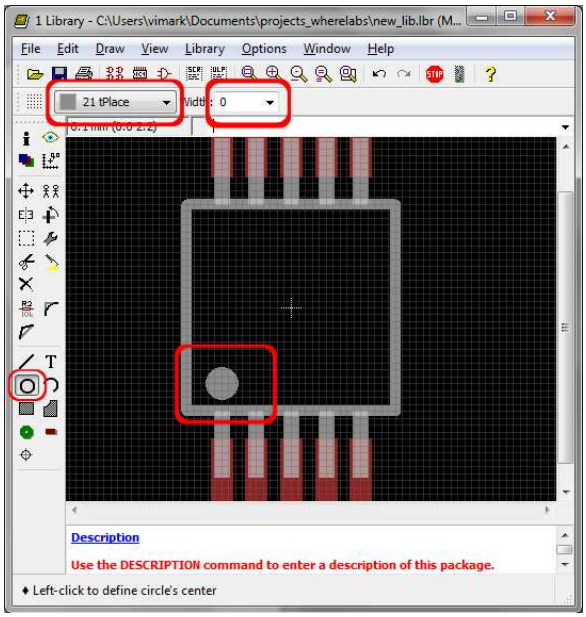
It's nice to see where the actual pins will be when you route the PCB. We'll create a pin reference on the Document layer. We use the Document layer so the reference will not be included on the top silk or overlap the pin pads.

- Select the **Rectangle** command and set the layer to top Document **51tDocu** layer
- Create a single pin centered on the part origin, this makes it easy to copy and snap onto our pads. We use a rectangle that is half of the width of the pad (0.23/2= 0.115).



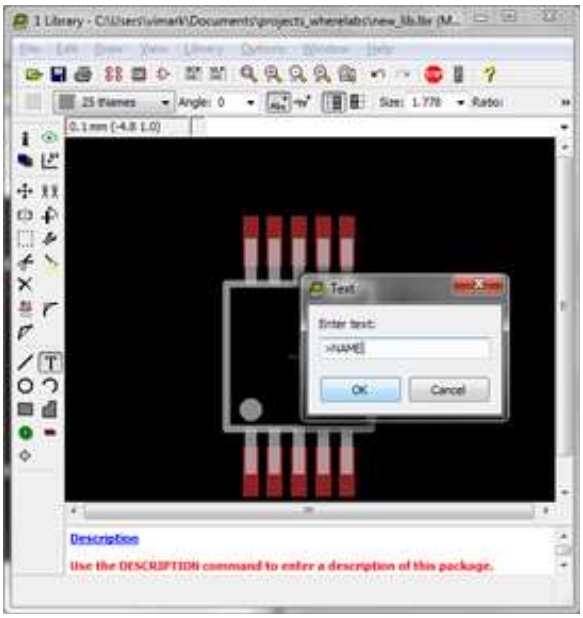
Σχήμα 38: Placing the pins.

Use the **Copy** command to copy the first pin. Place a copy on each pad with its base touching the chip outline.

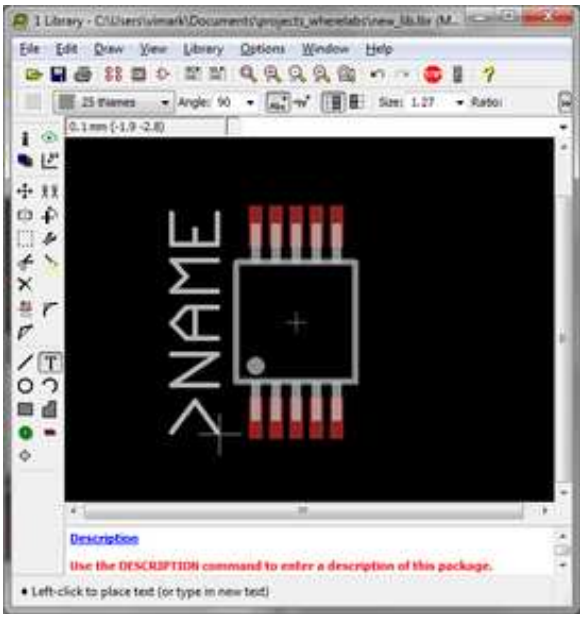


Σχήμα 39: Placing a circle.

When we're building the board we need to know how to position parts. Mark pin 1 with a zero width **circle** on the top silk 21 **tPlace** layer. The zero width circle will fill up.



Σχήμα 40: Naming the chip.

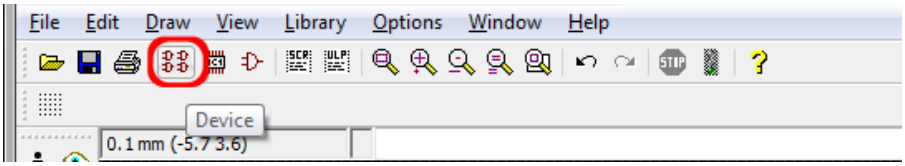


Σχήμα 41: Naming the chip Done.

Finally put a 'name' text near the chip, on the top silk name layer 25 tNames, and we're done.

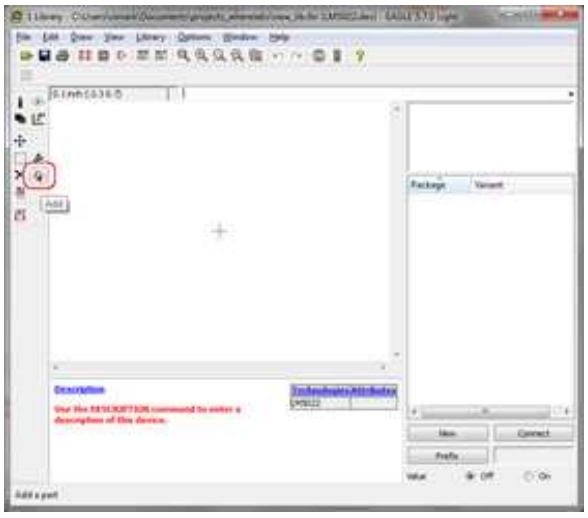
4.5 Making the Device

A device links the symbol and package and makes a complete part you can use in Eagle.

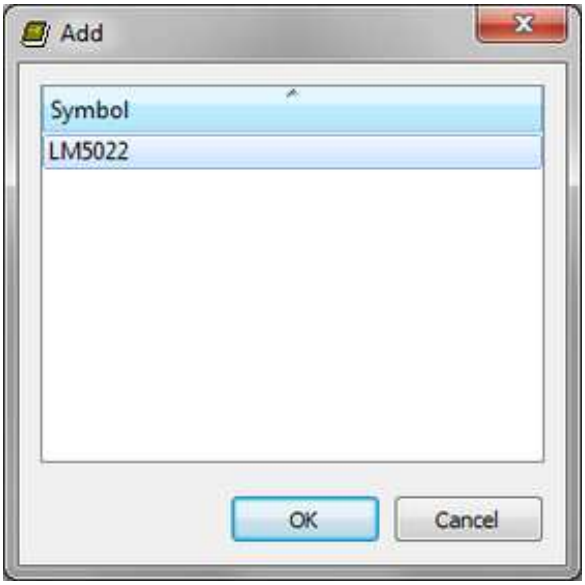


Σχήμα 42: The device button.

Create a new device by clicking the **Device** symbol.



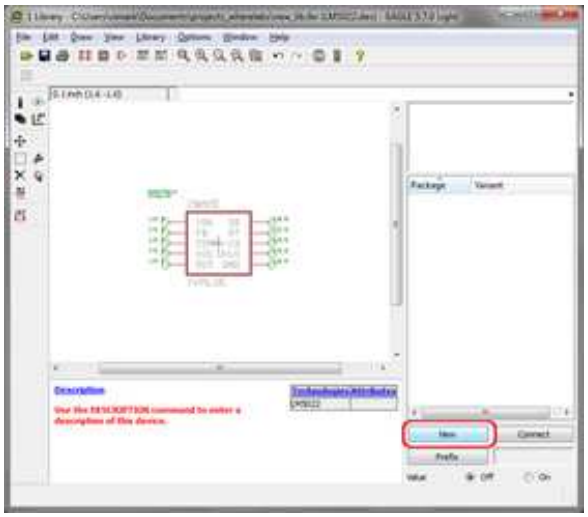
Σχήμα 43: Importing the symbol.



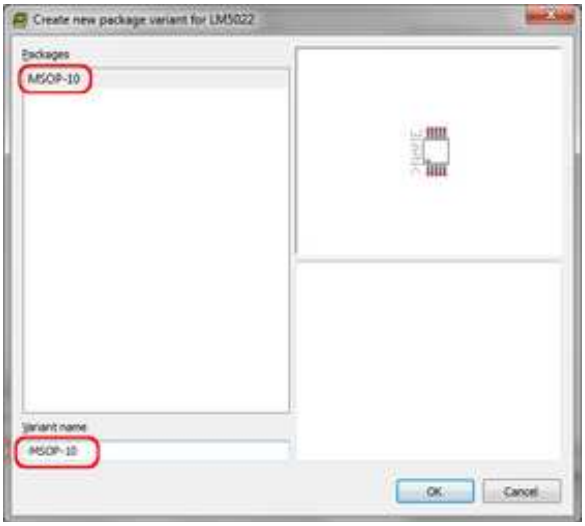
Σχήμα 44: List of symbols.

Import the symbol by clicking the 'Add' icon.

A window will pop-up with a list of symbols in the library. Pick the symbol that we made before and click OK.



Σχήμα 45: Importing the package.

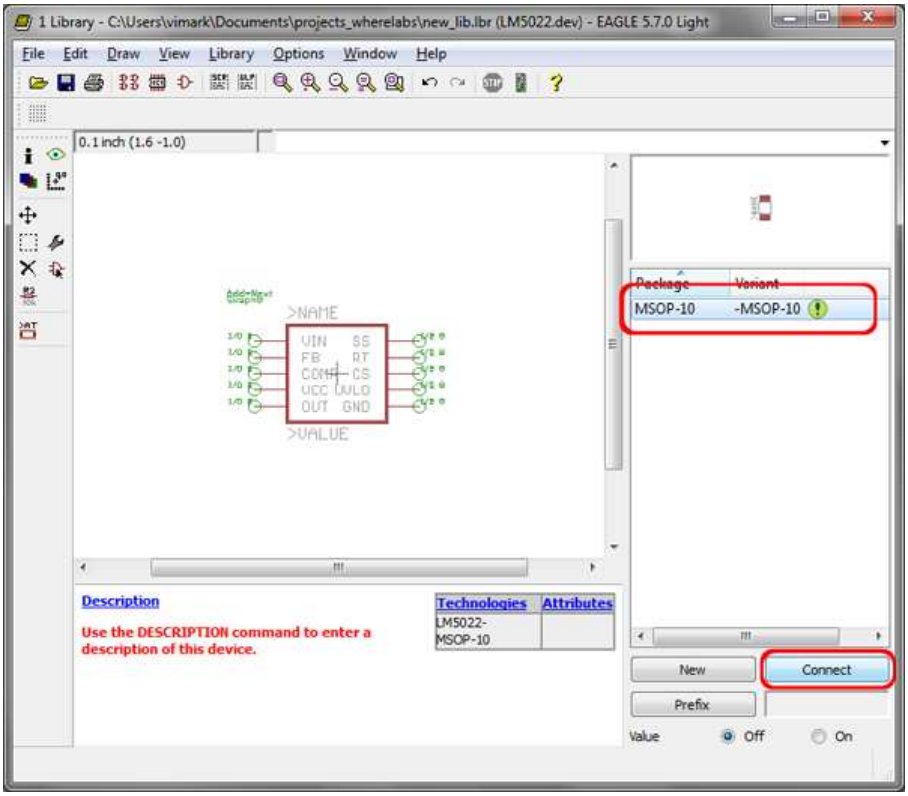


Σχήμα 46: Package list.

Now import the package by clicking the 'New' button.

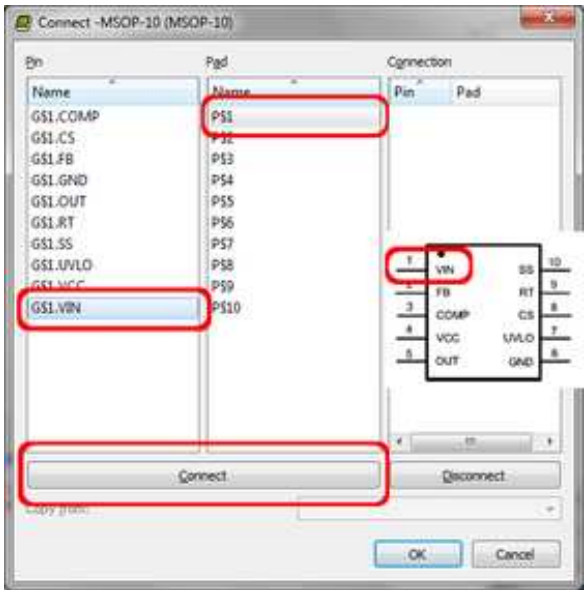
Eagle highlights packages that will fit the symbol. Choose the package/footprint we created before.

In the variant name box enter the same name as the package. This can be used to make different sizes of the same device. Click OK.

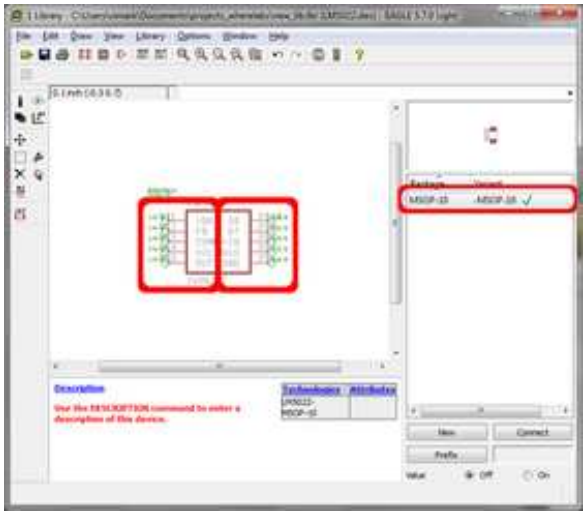


Σχήμα 47: Package imported.

The footprint appears on the package list with an '!' icon showing that the symbol and the package are not linked or connected yet. Click the 'connect' button to link both.



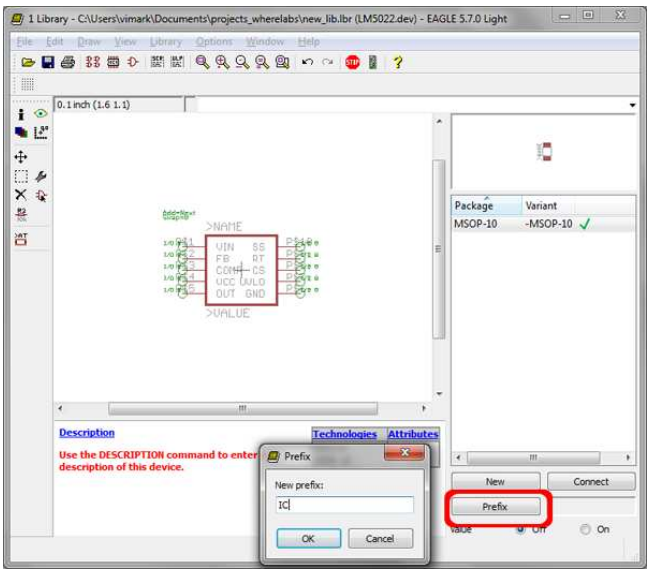
Σχήμα 48: Unpaired pins.



Σχήμα 49: Each pin is connected to only one pad.

After clicking the connect button, another window will appear showing unpaired pins.

Match each symbol pin with the corresponding pad and click connect. Each pin should only connect to one pad.



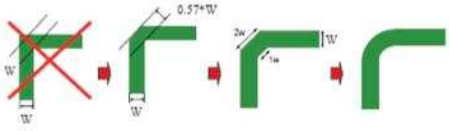
Σχήμα 50: Using the Prefix button.

Finally, click the **Prefix** button and specify a prefix for this chip. Eagle will automatically increment this when you make your schematic, e.g IC1, IC2, IC3. It will also identify this part in your bill of materials.

Save the library and you can try your newly created part.

5 Layout Design Rules for RF Layout

- Keep via inductance as low as possible. Usually means larger holes or multiple parallel holes)
- Keep top ground continuous as possible. Similarly for bottom ground.
- Make the number of return paths equal for both digital and RF
  - Current flow is always through the path with least impedance. Therefore digital signals should not find a lower impedance path through the RF sections.
- Compact RF paths are better, but observe good RF isolation between pads and or traces.
- Keep copper layer continuous for grounds. Keep connections to supply layers short
- Use vias to move the PCB self resonance higher than signal frequencies
- Keep trace and components spacing no less than 12 mils.
- Keep via holes large at least 14.5 mils
- Separate high speed signals (e.g. clock signals) from low speed signals, digital from analog. Placement is critical to keep return paths free of mixed signals.
- Route digital signals traces so antenna field lines are not in parallel to magnetic fields lines.
- Keep traces length runs under a 1/4 wavelength when possible.
- Avoid discontinuities in ground layers
- Don't use sharp right angle bends



- Do not have vias between bypass caps