FreePCB User Guide
Version 1.4

Allan Wright

April 14, 2007

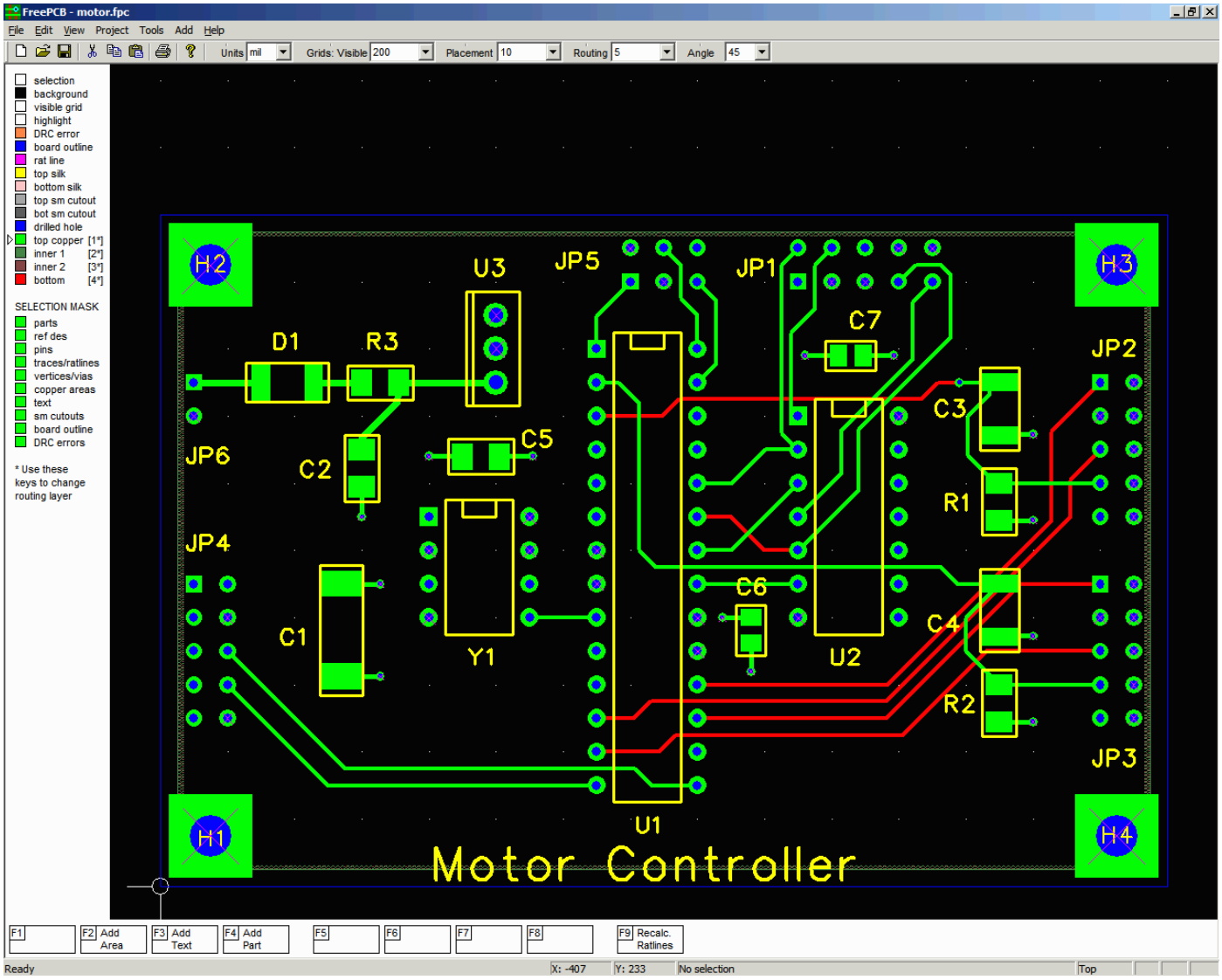


Table of Contents

1. Introduction

FreePCB is a free, open-source PCB layout editor for Windows, that I starting writing because I was unhappy with currently available free or low-cost PCB editors. The ones that I tried were either crippled by pin and layer limits, or buggy and difficult to use. I finally decided that any idiot could write a better one, and I was just the idiot to do it!

This user guide contains chapters on the PCB design process and the FreePCB user interface, and a tutorial which takes the user through the process of creating a PCB from a schematic and a netlist file. In the process it describes most of the main features of FreePCB.

FreePCB was written for Microsoft Windows, but it can run under Linux by using Wine, or on Macintosh computers with VirtualPC. The source code is copyrighted but released under the GNU General Public License (version 2 or later). Here are the terms of the License.

If you have questions about FreePCB, please start by checking the User Forum at www.freepcb.com. If you want to contact me directly, my email address is:

allan@freepcb.com

Thanks for trying FreePCB.

2. User Guide History

2.1 What's new in version 1.4

Here are the new features that have been added since version 1.2:

- Functions added for importing and exporting files to/from the FreeRoute web-based autorouter.
- **Project > Options** now allows renaming the project, changing the default library folder, and changing the number of layers.
- Autosaving improved, so that autosaved files are stored in a subfolder of the project folder, with incremental file names generated automatically.
- Gerber format changed from 2.6 to 2.4.
- Group copy/paste functions added. Groups can be saved as project files, and a project file can be pasted as a group into another project.
- Traces can now have branches.
- A new function **Tools > Check Traces** has been added, which cleans up traces by combining colinear segments and eliminating zero-length segments.
- Routing from off-grid points has been improved, so that "dogleg" trace segments are created to get back on the grid in one step.
- The widths of traces and vias can now be changed independently in **Project > Nets**.
- You can change the reference designator of a part in **Edit > Part**.
- There is a **Change Layer** function for copper areas..
- Editing of copper areas has been improved. Copper area cutouts can now contain other copper areas.
- In the Footprint Editor, when you change the settings for a pin you can apply the changes to other pins.
- Gerber output now include paste masks.
- Gerber and drill outputs can now generate panelized data.
- Boards may contain multiple outlines.

2.2 What's new in version 1.2

Here are the new features that have been added since version 1.0:

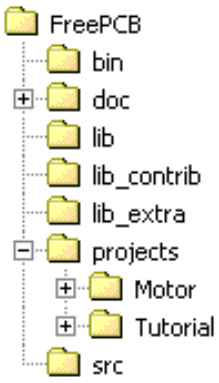
- When routing with a ratline, you can terminate the trace on any pin in the net by routing to the pin. You could already do this with stub traces.
- **View > Show part...** lets you select a part in the project by reference designator, then centers the window on it.
- Footprints can now contain silkscreen text
- Oval, octagonal and rounded-rectangle pads are supported.
- The function-key and layer-list menus are now "clickable"
- Some of the right-click menus have been expanded.

- Footprints can now contain SMT pads on both sides of the board. This feature was added mainly for edge connectors.
- Options have been added to **File > Import Netlist**, to allow choices in how to handle parts, footprints and nets that already exist in the project.
- The "recent files" list now works properly.
- You can swap connections to pins by selecting one pin, holding down the "s" key, and then selecting the other pin.
- You can create cutouts in the solder mask layers (for example, to eliminate "dams" of solder mask between the fingers of an edge connector).
- A **Check connectivity** tool has been added to the **Tools** menu. Also, the Design Rule Checker can flag connectivity errors.
- Cutouts can be added to copper fill areas.
- A **Move Origin** tool has been added to the **Tools** menu.
- An entire trace can be highlighted by holding down the "t" key and clicking on a segment or vertex. An entire net can be highlighted by holding down the "n" key and clicking on a segment, vertex or pin.
- A "Change layer" function has been added when segments or traces are selected.
- When you change the trace width of nets in the **Project > Nets...** dialog, you can apply the new width to existing traces.
- Through-hole pads are now drawn on each copper layer. The "through-hole" layer has been replaced with a "drilled hole" layer.
- Most items can be moved with the arrow keys.
- You can select a group of items (such as parts, traces, copper areas, etc.) by drawing a rectangle with the mouse. Then the entire group can be moved with the mouse or arrow keys.
- Options have been added to the **File > Generate CAM files...** dialog, so you can choose whether to use thermal reliefs on pads and vias, and whether to create openings in the solder mask over via pads.
- A "Selection mask" menu has been added, so you can control the types of items that can be selected with the mouse.
- When moving items, the relative distance is shown in the status bar.
- In the **Help** menu, there is an item that shows a list of keyboard shortcuts.

3. Installing FreePCB

Eventually, FreePCB will come with a nice installer, but for now you have to do it manually. Here are the instructions

- Download the latest version of the software from www.freepcb.com. It may be in the form of a zipped archive file, with a name like **freepcb_1000_bin.zip**, a self-installing archive such as **freepcb_1000_src.exe**, or a full installer application such as **freepcb_1000_setup.exe.**
- Install the software into a folder on your hard drive such as **C:\FreePCB** or **C:\Program Files\FreePCB.** This folder should now contain subfolders as shown:



  - The subfolder **bin** contains the **FreePCB.exe** executable, the default configuration file **default.cfg**, and a few other files.
  - The subfolder **doc** contains documentation (mainly this User Guide).
  - The subfolder **lib** contains the core footprint libraries.
  - The subfolder **lib_contrib** contains footprint libraries contributed by other users.
  - The subfolder **lib_extra** contains less commonly used footprint libraries.
  - The subfolder **projects** contains folders for each PCB project.
  - The subfolder **src** contains the source code for FreePCB.
- If you used the installer application, you were prompted to create a desktop shortcut to start FreePCB. If you used one of the archives, you can make a shortcut manually. Open the **.\FreePCB\bin** folder in Windows Explorer, and drag the file **FreePCB.exe** onto the desktop with the RIGHT mouse button. You should be prompted with a list of options including creating a shortcut.

4. Overview of the PCB Design Process

This is a brief overview of the PCB design process, with explanations of some terms which have special importance to FreePCB. These are shown in **bold** type as they are introduced.

4.1 Schematic Diagram

Most PCB designs start out as a **Schematic Diagram**, which shows the **Devices** used in the design and the **Connections** between them. Each device in the schematic has a **Reference Designator** such as "U1" or "R3". The connection points on the devices are called **Pins**, even though they may actually be non-pin terminals such as tabs, wire leads, screw terminals, etc. In general, the PCB should contain devices and connections exactly as shown on the schematic. If the designer makes changes to the PCB layout which create discrepancies between the PCB and the schematic, the schematic should be **Back-Annotated** to reflect the changes.

4.2 Specifying Parts, Packages and Pin Names

The first step in designing a PCB from a schematic is to select a physical **Part** for each device in the schematic. At the very least, the part specification should include the **Package** and **Pinout** for the part. Once parts and packages have been chosen, it may be necessary to go back to the schematic and assign **Pin Names**. A name must be assigned to every pin in the design, even on parts which have interchangeable pins, such as resistors.

Pin names are usually numbers, but may also contain letters. If letters are used, FreePCB requires that they precede any numbers. For example, "12", "A", "SOURCE" and "A23" are legal pin names, while "1A" and "A1A" are not. Characters other than letters and numbers are not allowed.

The **Package Identifier** is a string such as "DIP16" which identifies the package to FreePCB, and determines which **Footprint** to use for the part. The footprint includes the copper **Pads** to which pins will be soldered, as well as graphical elements such as a **Part Outline** and a **Text String** for the reference designator.

4.3 The Partlist

The Partlist is a list of all of the parts in a design, with their reference designators and package identifiers. A sample partlist is shown below:

Reference Designator	Package Identifier
U1	DIP14
R1	RES_1/4W_AXIAL
C1	CHIP_0805

4.4 The Netlist

A **Net** is a set of pins which are connected together by lines on the schematic. Each pin in the net is identified by a text string consisting of the reference designator of the part containing the pin, the character ".", and the pin name. For example, pin 8 on part U5 would be "U5.8". Each net must have a unique name, which may be meaningful and descriptive, or merely distinctive. The set of pins in a net is called the **Pin List**.

A **Netlist** is a list of all of the nets in the design, with their pin lists. A sample netlist is shown below:

Net Name	Pin List
VCC	U1.14, R1.1, C1.1, Q1.C
GND	U1.7, C1.2, Q1.E
$$1234	U1.1, R1.2, U1.2, Q1.B

4.5 Making Netlist Files

If the schematic was created with a schematic editor, the partlist and netlist can be automatically written to a **Netlist File**. Most editors will produce netlist files in a variety of formats. You should choose a format which is recognized by FreePCB, such as "PADS-PCB". A netlist file in this format is shown below. The listing is pretty much self-explanatory.

```
*PADS-PCB*
*PARTS*
U1 DIP14
R1 RES_1/4W_AXIAL
C1 C0805

*NETS*
*SIGNAL* GND
U1.7 C1.2

*SIGNAL* $1234
U1.1 R1.2 U1.2
```

If you are working from a printed or hand-drawn schematic, you will have the make the netlist file manually. This is actually pretty easy, although it can get tedious on a large design. I usually make a photocopy of the schematic that I can draw on. Then I start making the netlist file using a text editor. For each part, I highlight the reference designator on the schematic with a yellow felt-tip pen as I add it the partlist. Then I assign net names and pin names (if they are not already on the schematic) and make the netlist. For each net, I use the yellow highlighter and draw over each connection as I add it to the pin list. When I am finished, every part and every connection on the schematic should be highlighted in yellow. Then, I print out the netlist file and check it line-by-line. As I go, I again highlight the parts and connections in the schematic, this time with a red pen, making sure that every part and every connection is included. I have made quite large netlist files this way without errors.

4.6 Importing Netlist Files into FreePCB

Once your netlist file is complete, you can import it into FreePCB to start your design. FreePCB will try to find a footprint to match the package identifier for each part. If this is unsuccessful, you can assign footprints manually, or go back and edit the netlist. All of the footprints will be placed near the lower left corner of the PCB design area. FreePCB then loads the nets from the netlist, breaking each net into a set of connections between the pins in the net. These will appear as "rubber-band" lines between the pins, called **Ratlines**. As you move the parts around, the ratlines will move with them.

4.7 Adding Parts and Nets "On-the-Fly"

Alternatively, you can create a design in FreePCB without a netlist file by using the "**on-the-fly**" editing technique. Starting with an empty design, add each part, net and connection using the **Modify** and **Add** menus. You will be prompted for the reference designators and net names as necessary. This works reasonably well for small designs, but I prefer to create a netlist file as described above.

4.8 Placing Parts

The next step is to **Place** the parts on the PCB by moving and rotating them as necessary. After you move a part, FreePCB will automatically reassign the ratlines of every net connected to the part to minimize their total length. Alignment of parts is made easier by using a **Snap Grid**.

4.9 Routing Traces

Once the parts are placed, **Routing** begins. This is usually the most time-consuming part of the PCB design process. Each connection must be converted to a copper **Trace**, consisting of multiple straight-line **Segments**. The points where trace segments join are called **Vertices**. The segments may all be on the same copper **Layer**, or may be on different layers, in which case they will be connected by **Vias**. At this point, FreePCB only supports full-thickness vias, not blind or hidden vias. Traces may contain **Branches** to additional pins, besides the original end-points of the trace. **Copper Areas** can also be created, which are useful for ground and power planes. **Stub Traces** can be used to connect SMT pads to the copper areas. The snap grid is very useful for aligning traces. You can also set a **Snap Angle**, which forces trace segments to be oriented at multiples of the snap angle (usually 45 degrees).

4.10 Adding Text

Text Strings may be added for labels, revision numbers, copyright notices, etc. These will usually be placed on the silk-screen layers.

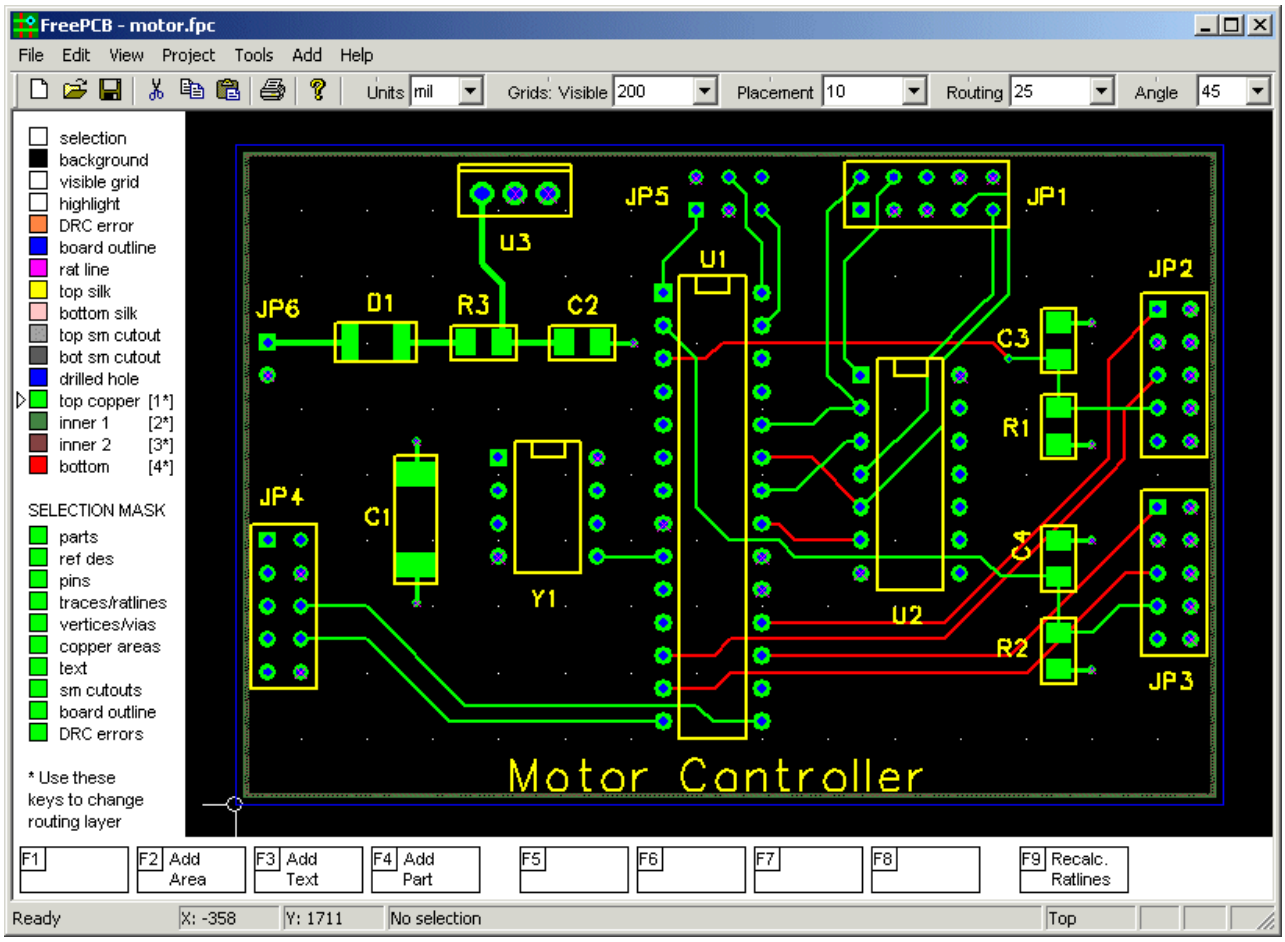
4.11 Plot and Drill Files

Plots of each layer can be exported as **Gerber Files**, which are used to make **Photoplots** for PCB fabrication. FreePCB can also produce **Drill Files**, which specify all of the holes to be drilled in the board. You can use a free **Gerber Viewer** such as ViewMate by PentaLogix to check your files and make **Check Plots** on a Windows printer. Then send your Gerber and drill files to a **PCB Fabrication Company** (along with some **Money**), and they will send back finished boards, usually in a couple of days. How **Cool** is that!

5. PCB Layout

5.1 Screenshot

A screenshot of the FreePCB window is shown below.



The main elements of the window are:

1. The menu bar, with **File**, **Edit**, **View**, **Project**, **Tools**, **Add** and **Help** menus
2. The taskbar (immediately below the menu bar), with 8 pictorial buttons and 5 list-boxes for setting the **Units** (mil or mm) and the spacing of the **Visible**, **Placement** and **Routing** grids and the snap **Angle** (in degrees).
3. The layer list on the left side of the client area, with boxes next to each layer showing its color and visibility.
4. The selection mask, below the layer list, with boxes to enable/disable selection of various items.
5. The layout window, showing the PCB.
6. The function-key menu below the layer list and layout window, showing the available function key commands.
7. The status bar at the bottom.

These elements are described in more detail in the following sections.

5.2 Menus

The following menus are available from the menu bar:

- File
  - New - create a new PCB project (shortcut = Ctrl-N)
  - Open... - open a PCB project file (shortcut = Ctrl-O)
  - Save - save a PCB project file (shortcut = Ctrl-S)
  - Save As... - save a PCB project file under a new name
  - Close - close the project
  - Import netlist... - import a netlist file
  - Export netlist... - export a netlist file
  - Convert library... - convert footprint libraries from Ivex to FreePCB format
  - Generate CAM files... - convert footprint libraries from Ivex to FreePCB format
  - Open Footprint Editor... - switch from the PCB Editor to the Footprint Editor
  - Export .dsn file... - export design file for autorouter
  - Import .ses file... - import session file from autorouter
  - Print... - (not yet implemented)
  - Print Preview - (not yet implemented)
  - Print Setup... - (not yet implemented)
  - Exit - close the application

- Edit
  - Undo - undo the last operation (shortcut = Ctrl-Z)
  - Cut - save to clipboard and delete (only available for groups)
  - Copy - save to clipboard (only available for groups)
  - Paste - paste from clipboard (only available for groups)
  - Save group to file... - save group as a project file
  - Paste group from file... - paste a group from a project file

- View
  - Show board outline - zoom to display the entire board outline
  - Show all - zoom to display all items in the layout (shortcut=Home)
  - Show part... - open a dialog to select a part, then select it and center on it
  - Layers... - open the View/Edit Layers dialog
  - Show log... - show log window

- Project
  - Options - open the Project Options dialog to edit the project options
  - Parts... - open the View/Edit Parts dialog to edit the partlist
  - Nets... - open the View/Edit Nets dialog to edit the netlist

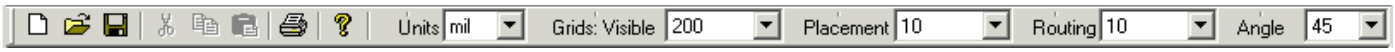
- Tools
  - Move origin - move the origin of the coordinate system to a new position
  - Footprint Wizard - open Footprint Wizard to create a new footprint
  - Footprint Editor - open the Footprint Editor to create or edit a footprint
  - Check parts and nets - check project for internal database errors
  - Check traces - check and clean up traces
  - Check connectivity - check project for incomplete netlist connections
  - Check Copper Areas - check copper areas, combine if necessary
  - Design Rule Check - check for compliance with design rules, and show errors
  - Clear DRC Errors - clear the symbols for all DRC errors
  - FreeRoute autorouter... - launch web-based autorouter

- Add
  - Board Outline - add PCB outline
  - Part - add new part
  - Net - (not yet implemented)
  - Copper Area - add copper area
  - Text - add text string
  - Solder Mask Cutout - add cutout in a solder mask layer

- Help
  - User Guide - show this User Guide
  - FreePCB Website - go to the FreePCB website
  - FreeRouting Website - go to the website for the FreeRoute autorouter
  - Keyboard shortcuts - show a list of the keyboard shortcuts
  - About FreePCB... - show About box

5.3 Taskbar

The taskbar is shown below:



It contains 8 pictorial buttons which are shortcuts for the following menu items:

- **New**
- **Open**
- **Save**
- **Cut**
- **Copy**
- **Paste**
- **Print**
- **About**

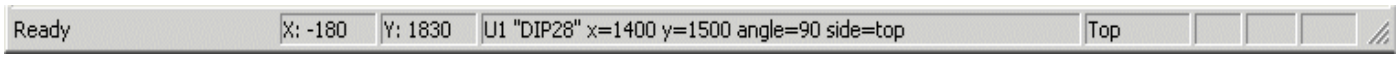
These are followed by 5 drop-down menus for setting the units and grids used by FreePCB. These are:

- **Units** - select mils or mm for dimensional units
- **Visible** - sets the spacing of a rectangular array of dots in the layout window, as a visual reference
- **Placement** - sets a snap grid for placing parts, text, etc.
- **Routing** - sets a snap grid for routing traces and drawing copper areas
- **Angle** - sets a snap angle (in degrees) for routing and drawing polylines such as the board outline and copper areas

The default entries for the drop-down menus (except the **Angle** menu) are in the **default.cfg** file in the folder which contains **FreePCB.exe**. When a project is created, they are copied into the options section of the project file. They may be modified by editing either file with a text editor such as Notepad.

5.4 Status Bar

The status bar (at the bottom of the FreePCB window) is shown below:



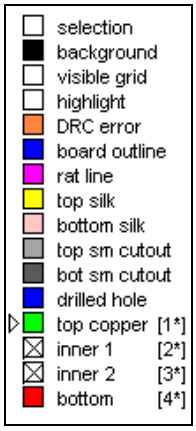
It contains the following items:

- A tooltip ("Ready"), which will change as the cursor is placed over various user interface items
- The X and Y coordinates of the cursor in mils or mm
- Information about the item which is selected (if any). In the example above, part U1 has been selected. Its reference designator, footprint, position, angle and side are displayed.
- The currently-active routing layer (" Top")

5.5 Layer List and Selection Mask

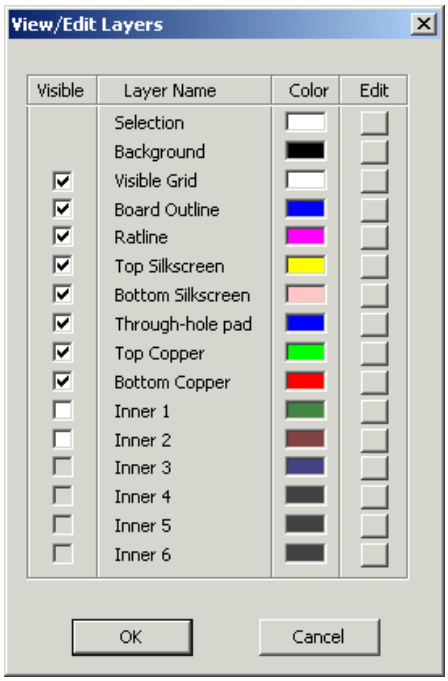
These are two vertical lists in the FreePCB window, to the left of the layout area.

5.5.1 Layer List



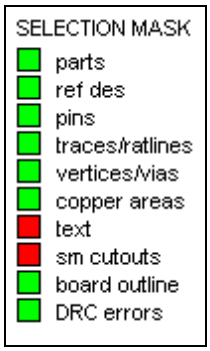
This is a list of all of the drawing layers in use by the project. Next to the name of each layer is a small square showing its color. If the layer has been made invisible, the square will be white with an "X" through it (such as "inner 1" and "inner 2" in the example). You can toggle each layer from visible to invisible by clicking on the square.

The active layer for routing ("top copper" in the example below) will be identified with an arrowhead. You can change the active routing layer by pressing a numeric key on the keyboard (i.e. keys "1" through "8"), or by clicking on the name of the layer in the list. The active routing layer will always be drawn "on top" of the other layers in the layout window.



The color for each layer can be modified by selecting **Layers** from the **View** menu, which pops up the **View/Edit Layers** dialog, as shown below. Click on the buttons in the **Edit** column to change the colors. You can also change the visibility of layers in this dialog.

5.5.2 Selection Mask



Items in the layout are selected for editing by clicking on them or drawing a rectangle around them with the mouse. Since items may overlap one another, sometimes it may be difficult to select exactly the item(s) that you want. In this case, the selection mask can be used to enable or disable the selection of different item types. It consists of a list of these types, with a colored box next to each one. A green box means that the item type can be selected, while a red box means that it can't. You can toggle the state of each box by clicking on it.

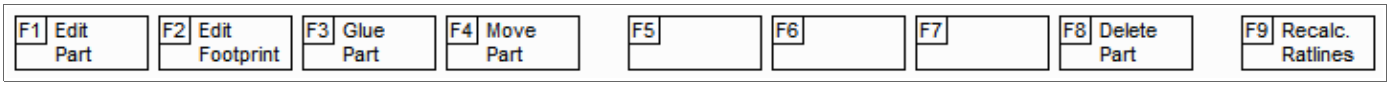
5.6 Function Key and Context Menus

Most Windows drawing programs make heavy use of "click-dragging" with the mouse for editing. This means holding the left mouse button down while moving the mouse. For example, the user may move an item by click-dragging it to a new position. An item may be resized by clicking on it, which brings up a selection box with "handles", which may then be click-dragged to stretch or shrink the item. The endpoints of lines may be click-dragged to new positions, etc., etc.

However, click-dragging doesn't work very well for routing traces, which is the most time-consuming part of PCB layout. Traces are basically polylines, consisting of multiple connected line segments. Placing each segment requires a separate mouse-click, which implies that the segments must be dragged without holding the mouse button down.

Also, while routing traces or placing parts, it is useful to be able to pan and zoom, which I like to do with the scroll wheel on the mouse. It is very difficult to use the scroll wheel while simultaneously holding down a mouse button.

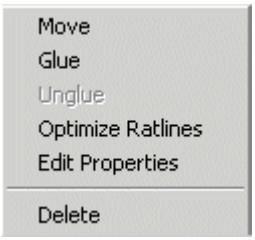
Therefore, FreePCB uses a slightly different approach. Generally, an item will be selected by clicking on it. This will highlight the item, to indicate that it has been selected, and a description of the item will appear on the status bar. An editing operation can then be performed by pressing a function-key on the keyboard. The operations assigned to each function-key are always listed across the bottom of the FreePCB window. This list is context-sensitive, and will change depending on which item has been selected. For example, the menu when a part is selected is shown below.



If the operation requires dragging, this will commence as soon as the function-key is pressed. The operation is usually completed by left-clicking the mouse, or it can be canceled by right-clicking. While dragging, the function-keys remain active, so that additional operations can be performed. For example, while dragging a part, pressing F3 will rotate it, while pressing F2 will flip it to the opposite side of the board.

This style of user interface requires that you keep your left hand on the keyboard and your right hand on the mouse while editing (assuming that you are right-handed). In the days of DOS, many CAD programs used this style, and some still do. If you have any doubt about the speed and efficiency of the "left hand on keyboard, right hand on mouse" approach, just spend a few minutes watching someone play one of the popular first-person shooter games for the PC, such as Doom, Quake, Unreal Tournament, Half-life, Tribes, etc. They ALL use this style of user interface. And routing a PCB is usually easier than circle-strafing a horde of rampaging Nazi mutants while dodging flying chainsaws in zero-G!

The assignment of editing operations to the function keys is not completely random, as it might first appear. Since the fingers of the left hand will normally rest on keys F1-F4, these are used for the primary editing operations. F4 (under the index finger) is used for the most common operation on the selected item (such as moving a part), while F2-F3 are used for less common operations. F1 is often used to pop up some sort of dialog to edit the properties of the selected item. The keys F5-F8 are usually used for deleting or unrouting, since they require moving the hand and are less likely to be hit accidentally. F9 is reserved for recalculating ratlines. This arrangement is obviously optimized for users who are right-handed (like me). I expect to add a "left-handed" option in the future.



If you prefer not to use the function keys, right-clicking the mouse will pop up a more traditional Windows menu, as shown below. This is called the "context menu", and its contents will change depending on which item is selected.

The context menu is not available while you are dragging an item with the mouse, so function keys must be used to perform operations while dragging.

5.7 Panning and Zooming

Notice that there are no scroll bars in the FreePCB window. Using scrollbars for panning is not very practical for a PCB layout program, since panning must often be performed while routing traces or moving parts. It is inefficient to have to move the cursor away from the work area to the scroll bar.

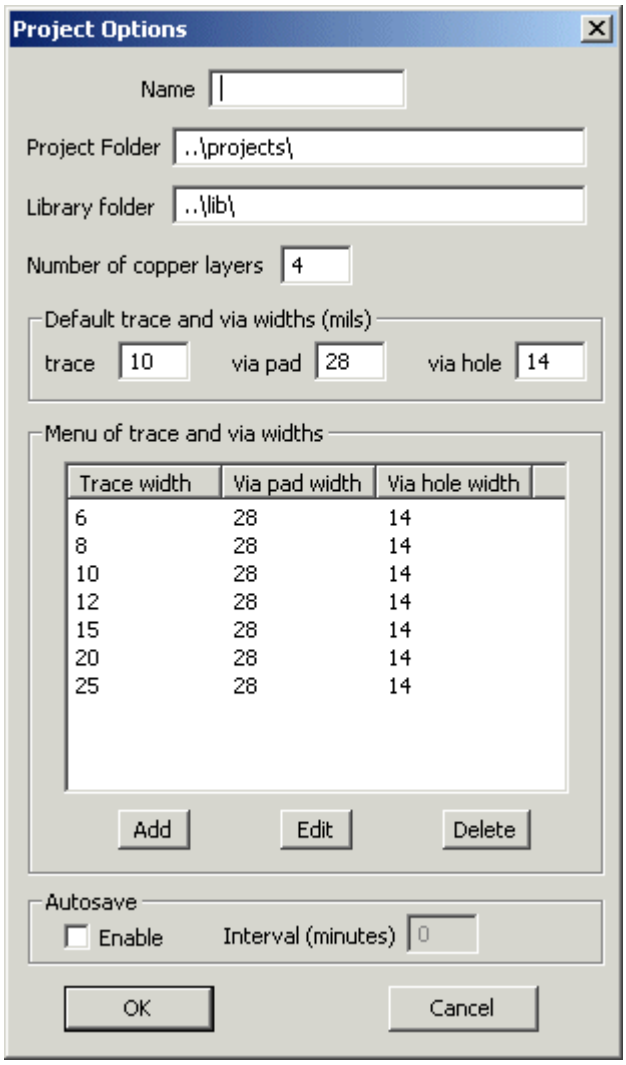
Therefore, panning and zooming in FreePCB are performed with either keyboard keys or (preferably) the scroll wheel on the mouse. To pan, move the cursor to a point on the PCB which will be the new center of the image, and press the spacebar or roll the scroll wheel one detent in either direction. The image will recenter on the cursor position. To zoom in, press the "Page Up" key or roll the scroll wheel forward more than one detent. To zoom out, press "Page Down" or roll the scroll wheel backward. If you don't have a mouse with a scroll wheel, I would recommend getting one.

You can display the entire PCB outline using the **View->Show board outline** menu item. You can show all of the elements in the project (which may be outside of the PCB outline) by selecting **View->Show all**, or by pressing the "Home" key. You can select a part and pan to its location using **View->Show part...**

5.8 Projects

In FreePCB, PCB layouts are referred to as projects. All of the information describing a project is stored in a single text file with the extension **.fpc**, which is usually stored in a folder with the same name as the project. For example, the project named "Motor" would be stored in the file **Motor.fpc** in the folder **C:\FreePCB\projects\Motor**

A new project is created with **File > New**. This will launch the **Project Options** dialog, which is shown below.



Note that most of the fields of the dialog are already filled in. These default settings are taken from a file called **default.cfg** which must be in the same folder as the **FreePCB.exe** application. If you don't want to use the defaults for your project, you can change them in the dialog. If you would like to change the defaults for all future projects, you can edit the **defaults.cfg** file with any text editor. The format of this file is described in Section 8: File Formats.

Each field in the dialog is explained below:

- **Name**: This is the name of the project. It should not contain spaces or any characters that are illegal in filenames.
- **Project Folder**: This is the folder where your project file will be saved. It is already filled in with the path to the default parent folder for new projects. As you enter the name of your project in the **Name** field, each character will automatically be appended to the path. For example, if you enter "Motor" in the **Name** field, the **Project Folder** field will become "..\projects\Motor". If you would like to use a different project folder, you can override the default by entering a path directly into the **Project Folder** field.
  - **Note**: The default path begins with "..\", indicating that it is a relative path. Actually, it is relative to the location of the **FreePCB.exe** application. If you have performed the default installation into **C:\FreePCB**, the application will be in **C:\FreePCB\bin**, and the project folders will be in **C:\FreePCB\projects**. If you installed FreePCB somewhere else, the default path will still work as long as **\bin** and **\projects** are in the same folder. If not, you may want to edit the **default.cfg** file. For example, when I am working on the FreePCB source code, I run the application from **E:\allan\SVNwork\FreePCB\Debug**, but I still want to create my projects in **C:\FreePCB\projects**. Therefore, in **E:\allan\SVNwork\FreePCB\Debug\default.cfg**, I changed the line

    ```
    parent_folder: "..\projects\"
    ```

    to:

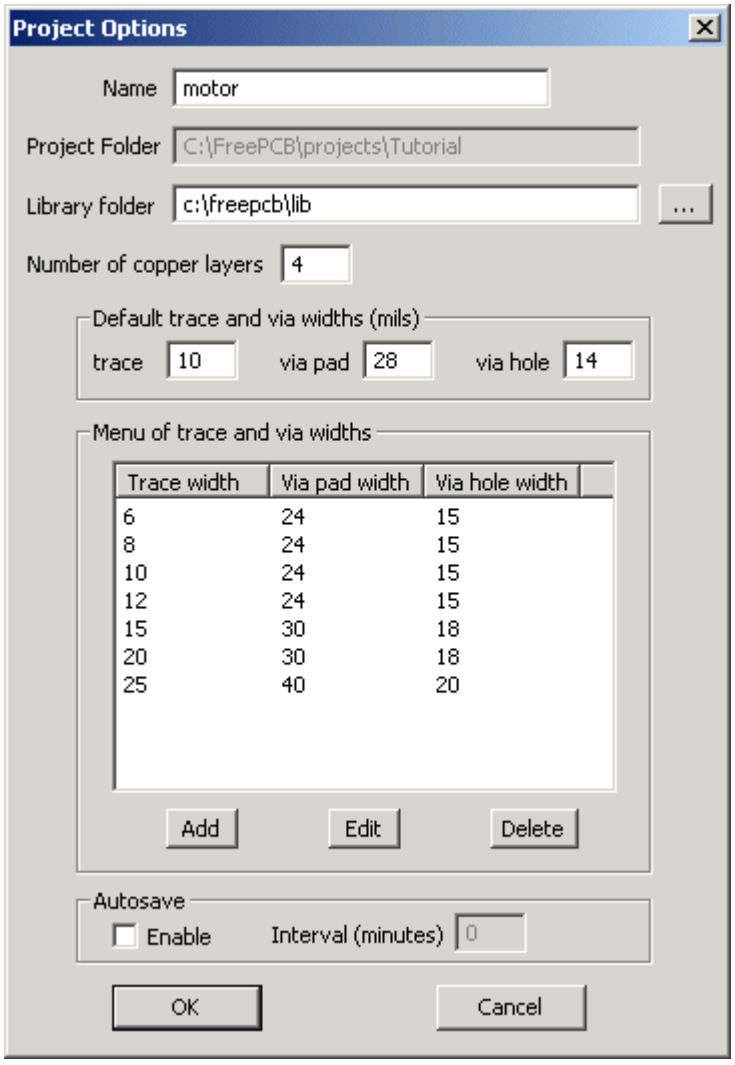
    ```
    parent_folder: "C:\FreePCB\projects\"
    ```

- **Library Folder**: This is the path to the folder which contains the footprint library files for FreePCB. The default path is relative to the application folder, but it can be changed in the dialog if necessary, and the default can be changed in **default.cfg** just as for the **Project Folder** (see note above).
- **Number of copper layers**: This is the number of copper layers on the PCB, between 1 and 8.
- **Default trace and via widths (mils)**: These represent the default values for trace width, via diameter and via hole diameter. They may be overridden later on for particular traces or nets.
- **Menu of trace and via widths**: This is a list of trace and via widths which will be offered as a menu if you modify the widths of particular traces or nets. You are not limited to these values, but it is convenient to have them on a menu and it reduces the chance of error. You can use the **Add** button to add new entries to the list. If you select an item you can use the **Delete** button to remove it or the Edit button to modify it.
- **Autosave**: Enabling Autosave causes FreePCB to automatically save the project file every few minutes.

When you are satisfied with your entries, click **OK**. The project folder will be created if it does not already exist (however, the parent folder must exist or you will get an error message). The project file will not be written into the folder until you **Save** from the **File** menu. I usually do this immediately after I have created a new project.

By the way, the project **Name** is only used to create names for the project file and the default project folder. After that, FreePCB uses the name of the project file as the project name. You can rename the project file or use **Save As** from the **File** menu to save the project under a different name. You may find it useful to keep several different versions of your project in the project folder, under different names. For example, you could experiment with some new ideas in a design, but revert back to an earlier version if they don't work out.

As mentioned above, you can modify the defaults for new projects by editing the **default.cfg** file in the application folder. The format for this file is described in Section 8: File Formats, and you shouldn't have any trouble figuring out how to modify it (but save a copy just in case). You can also edit the project file, if you like. This is a more complex file, with sections describing the project options, footprints, parts, nets, text strings, etc. The options section is very similar to the **default.cfg** file. You probably don't want to mess too much with the other sections.



Once a project has been created, it can be closed, saved and opened using the usual selections from the **File** menu.

After a project has been created, the project options can be edited by selecting **Options...** in the **Project** menu. This will bring back the **Project Options** dialog. You can make changes to any of the enabled fields, and save them by clicking **OK**.

5.9 PCB Elements

5.9.1 Individual Elements

The representation of the PCB in the layout window consists of various items, which are listed in this section. Note that most of these may be selected by clicking on them. Once an item has been selected, operations can be performed on it by pressing a function key or by right-clicking and making a selection from the context menu. If the operation involves dragging the item, function keys can sometimes be used to perform additional operations while dragging.

The elements of the PCB are listed below, along with their associated function key operations.

- **Origin**: This is a symbol that identifies the origin of the PCB coordinate system (i.e. the point where X = 0 and Y = 0). It looks like a cross with a small circle at its center. It is always visible, and cannot be selected. It can be moved using Tools > Move origin.
- **Visible grid**: Dots in a regularly-spaced array, used for visual reference. The grid spacing is set by a drop-down menu in the taskbar.
- **Board outline**: This is a closed polyline, consisting of a set of points (corners), with lines between them (sides). The sides may be straight lines or arcs. The corners and sides are selectable, and may be edited as follows:
  - **Corner**:
    - F1 (Set Position) - pop up a dialog that allows you to edit the X and Y coordinates of the corner explicitly.
    - F4 (Move Corner) - start dragging the corner with the cursor.
    - F5 (Delete Corner) - remove the corner.
    - F8 (Delete Outline) - remove the entire board outline.
  - **Side**:
    - F1 (Straight Line) - make the side a straight line.
    - F2 (Arc CW) - make the side a clockwise arc.
    - F3 (Arc CCW) - make the side a counterclockwise arc.
    - F4 (Add Corner) - insert a corner into the side and starts dragging it.
    - F8 (Delete Outline) - remove the entire outline.

- **Part footprint**: This is a compound symbol consisting of a set of copper pads, a part outline, a text string for the reference designator of the part, and possibly other text strings. The entire footprint, a single pad or the reference designator may be selected for editing.
    - **Entire footprint**:
        - F1 (Edit Part) - pop up a dialog to edit the properties of the part.
        - F2 (Edit Footprint) - switch to the Footprint Editor window, with the footprint of the part already imported for editing.
        - F3 (Glue/Unglue Part) - a "glued" part can't be moved without "ungluing" it.
        - F4 (Move Part) - start dragging the footprint to move it.
            - While dragging:
                - F2 - flip part from one side of the board to the other.
                - F3 - rotate part 90 degrees clockwise.
        - F8 (Delete Part) - remove the part from the PCB and the partlist.
        - F9 (Recalc. Ratlines) - reassign ratlines for all nets which connect to the part to minimize their length.
    - **Pad**:
        - F1 (Set Net) - pop up a dialog to assign the pad to a net.
        - F3 (Start Stub) - start dragging a new stub trace.
        - F4 (Connect Pin) - start dragging a ratline to another pad.
        - F9 (Recalc. Ratlines) - reassign ratlines for the net which connects to this pad to minimize their length.
    - **Reference designator text**:
        - F1 (Set Size) - pop up a dialog that lets you change the size and direction of the text string.
        - F4 (Move Ref Text) - start dragging the text string to move it.
            - While dragging:
                - F3 (Rotate Ref Text) - rotate the text string 90 degrees clockwise.

- **Trace**: This is a polyline representing a connection between two pads. It consists of one or more straight-line segments with vertices between the segments. The segments may be routed (i.e. physically present on a copper layer) or unrouted. Unrouted segments are called ratlines.
    - **Ratline**: This is a line representing an unrouted segment of a trace.
        - F1 (Set Width) - pop up a dialog that lets you set the change the trace and via widths of the trace or net.
        - F3 (Lock/Unlock Connect) - a locked ratline can't be eliminated by the "Recalc. Ratlines" operation.
        - F4 (Route Segment) - start dragging the endpoint of a routed segment to replace the ratline.
            - While dragging:
                - F4 (Complete Segment) - extend the routed segment being dragged to the endpoint of the ratline.
        - F5 (Change Pin) - change the pin to which the ratline is connected.
        - F6 (Unroute trace) - unroute any routed segments in the trace.
        - F8 (Delete Connect) - remove the connection (start_pin to end_pin) from the net.
        - F9 (Recalc. Ratlines) - reassign ratlines for the net to minimize their length.
    - **Segment**: This is a line representing a routed segment of a trace.
        - F1 (Set Width) - pop up a dialog that lets you set the change the trace and via widths of the trace or net.
        - F2 (Change Layer) - change the layer of the segment.
        - F4 (Add Vertex) - insert a vertex into the segment and start dragging it.
        - F5 (Unroute Segment) - unroute the segment, converting it to a ratline.
        - F6 (Unroute Trace) - unroute the entire trace, converting it to a ratline.
        - F8 (Delete Connect) - remove the connection (start_pin to end_pin) from the net.
        - F9 (Recalc. Ratlines) - reassign ratlines for the net to minimize their length.
    - **Vertex**: This is a point at the junction of two segments, or a segment and a ratline. If both segments are routed on different layers, there will be a via at the vertex.
        - F1 (Set Position) - pop up a dialog that allows you to edit the X and Y coordinates of the vertex explicitly.
        - F3 (Connect Pin) - start dragging a ratline to connect the vertex to a pin for a branch trace.
        - F4 (Move Vertex) - start dragging the vertex to move it.
        - F6 (Unroute Trace) - unroute the entire trace.
        - F7 (Delete Vertex) - remove vertex, unrouting the adjacent segments.
        - F8 (Delete Connect) - remove the connection (start_pin to end_pin).
        - F9 (Recalc. Ratlines) - reassign ratlines for the net to minimize their length.

- **Branch Trace**: A branch is a trace between a vertex of another trace and a pin. The functions for ratlines, segments and vertices are the same as for a regular trace.

- **Stub Trace**: A stub trace starts on a pad but doesn't end on a pad. It usually ends with a via, which is used to connect the trace to a copper area on another layer. It may contain ratlines, segments and vertices like a regular trace, but the last segment and last vertex are treated specially.
    - **End Segment**: The last routed segment.
        - F1 (Set Width) - pop up a dialog that lets you set the change the trace and via widths of the trace or net.
        - F2 (Change Layer) - change the layer of the segment.
        - F4 (Add Vertex) - insert a vertex into the segment and start dragging it.
        - F5 (Unroute Segment) - unroute the segment, converting it to a ratline.
        - F6 (Unroute Trace) - unroute the entire trace, converting it to a ratline.
        - F7 (Delete Segment) - delete the segment.
        - F8 (Delete Connect) - remove the entire stub trace.
        - F9 (Recalc. Ratlines) - reassign ratlines for the net to minimize their length.
    - **End Vertex**: The last vertex (i.e. the end-point of the stub trace).
        - F1 (Set Position) - pop up a dialog that allows you to edit the X and Y coordinates of the vertex explicitly.
        - F2 (Add Segment) - start dragging a new segment to extend the stub trace.
        - F3 (Add/Delete Via) - add or remove a via at the end of the stub trace.
        - F4 (Move Vertex) - start dragging the end vertex to move.
        - F5 (Add/Delete Via) - add/remove the via.
        - F7 (Delete Vertex) - remove the vertex.
        - F8 (Delete Connect) - remove the entire stub trace.
        - F9 (Recalc. Ratlines) - reassign ratlines for the net to minimize their length.

- **Copper Area**: This is a closed polyline which defines an area of solid or patterned copper on one of the copper layers. It is used to created ground planes, heatsinks, etc. It consists of a set of corners with sides between the corners. It is filled with a hatch pattern of diagonal lines. Note that this hatch pattern is for visual identification of the area only, and does not represent the actual copper pattern which will be applied to the PCB. It may also contain **cutouts**, which are "holes" in the copper area. Cutouts are edited just like copper areas.
    - **Corner**:
        - F1 (Set Position) - pop up a dialog to set the corner position explicitly.
        - F4 (Move Corner) - start dragging the corner to move it.
        - F5 (Delete Corner) - remove the corner.
        - F7 (Add Cutout) - start drawing a new polygon for a cutout in the copper area.
        - F8 (Delete Area) - remove the entire area.
    - **Side**:
        - F1 (Straight Line) - if the side is an arc, convert to straight line.
        - F2 (Arc (CW)) - convert side to clockwise arc.
        - F3 (Arc (CCW)) - convert side to counterclockwise arc.
        - F4 (Add Corner) - add a new corner and start dragging it.
        - F7 (Add Cutout) - start drawing a new polygon for a cutout in the copper area.
        - F8 (Delete Area) - remove the entire copper area.

- **Text String**: This is a string of alphanumeric characters, used for adding labels, copyright notices, etc. It may be placed on a silk-screen or copper layer.
  - F1 (Edit Text) - pop up a dialog to edit the string and its properties, such as size and stroke width.
  - F4 (Move Text) - start dragging the string to move it.
  - F8 (Delete Text) - remove the entire string.

- **Solder Mask Cutout**: This is a closed polyline that defines a cutout in a solder mask layer.
  - **Corner**:
    - F1 (Set Position) - pop up a dialog to set the corner position explicitly.
    - F4 (Move Corner) - start dragging the corner to move it.
    - F5 (Delete Corner) - remove the corner.
    - F8 (Delete Area) - remove the entire cutout.
  - **Side**:
    - F1 (Straight Line) - make the side a straight line.
    - F2 (Arc CW) - make the side a clockwise arc.
    - F3 (Arc CCW) - make the side a counterclockwise arc.
    - F4 (Add Corner) - add a new corner and start dragging it.
    - F8 (Delete Area) - remove the entire cutout.

5.9.2 Groups of Elements

A group of elements can be selected by drawing a rectangle around them with the mouse. All of the items in the group will be highlighted. Individual items can be added to the group or removed from the group by clicking on them with the Ctrl key held down. Operation which can be performed on groups are:

- F4 (Move Group) - start dragging group
- F8 (Delete Group) - delete the group from the project
- ctrl-C - copy group to clipboard
- ctrl-X - copy group to clipboard and delete it from the project

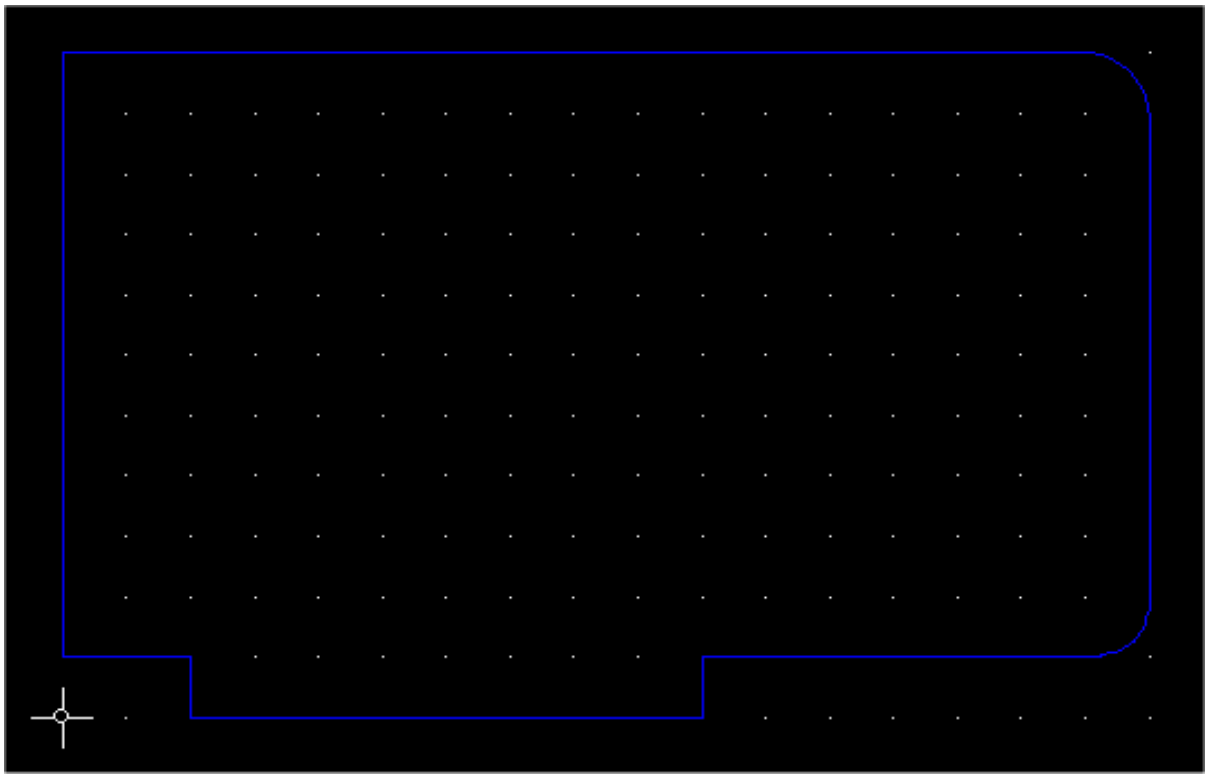
5.9.3 Moving items or groups with the arrow keys

When an item or group is selected, you can move it by selecting a move operation with the function keys or context-menu, and then dragging it with the mouse. Most items and groups can also be moved by pressing the arrow keys on the keyboard. Pressing an arrow key will move the item or group a distance equal to the current setting of the placement grid or routing grid, depending on the item selected. Smaller movements can be made by holding down the Shift key when pressing the arrow key, which will move the item 1 mil or 0.01 mm, depending on the units in use.

5.10 Board Outline

The **board outline** represents the outline of the PCB (duh!). In FreePCB, it consists of a **closed polyline** (or **polygon**) consisting of 3 or more **sides**, with **corners** between the sides. The sides may be straight line segments or **arcs**. An arc is one quadrant of an ellipse whose major axis is parallel to either x or y. The advantage of using this limited definition of an arc is that the arc is completely defined by the position of its endpoints and a direction, which may be clockwise or counterclockwise. A sample board outline consisting of 10 corners and 10 sides is shown below. Note that two of the sides are arcs.

There may be more than one board outline in a project.



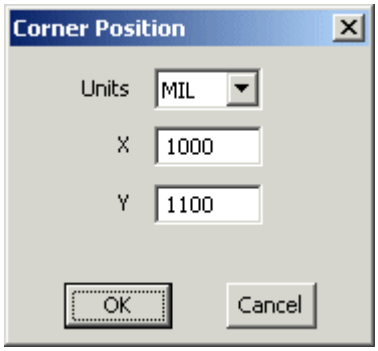
A board outline is created by selecting **Add > Board Outline**. The cursor will change into a cross-hair. Place the starting corner by left-clicking the mouse. You will then find yourself dragging a rubber-band representation of the first side. Left-clicking again will place the second corner, and you will be rubber-banding the second side. Continue until you have placed the last corner, then right-click to close the polyline by drawing a side back to the starting corner.

While drawing, you can change the **style** of the current side from straight to an arc by using the appropriate function keys (F1 for straight, F2 for a clockwise arc and F3 for a counterclockwise arc). The style that you select will remain in effect until you change it. Note that arcs become straight lines if the end-points are vertical or horizontal.

Placement of the corners will be controlled by the Placement Grid, which should be set to an appropriate value before starting to draw the outline. Placement will also be controlled by the Snap Angle. If the Snap Angle is 45 degrees, then arcs will be quadrants of a circle.

Once the outline has been created, you can edit it in the following ways:

- You can reposition a corner by clicking on it to select it. A small white square will appear around the corner indicating that it has been selected, and the status bar will indicate which corner has been selected. Then press F4 ("Move Corner") to start dragging the corner to a new position.



Or press F1 ("Set Position") to pop up a dialog which will let you set the position of the corner explicitly, as shown below:

- You can add a corner by clicking on a side to select it. It will turn white to indicate that it has been selected. Then press F4 ("Add Corner") and you will be dragging a new corner, with rubber-band sides to the adjacent corners. Left-click to place it. This only works with straight sides, so if the side is an arc then you will have to change it to a straight line first (see below).
- You can delete a corner by selecting it and pressing F5 ("Delete Corner"). The corner will disappear, and the adjacent sides will be replaced by one side between the adjacent corners.
- You can change the style of a side by selecting it and then pressing F1 ("Straight Line"), F2 ("Arc CW") or F3 ("Arc CCW").
- The entire outline can be deleted by selecting any side or corner and pressing F8 ("Delete Outline").

5.11 Parts

5.11.1 The Anatomy of a Part

Every PCB will include one or more **parts**. Each part contains the following elements:

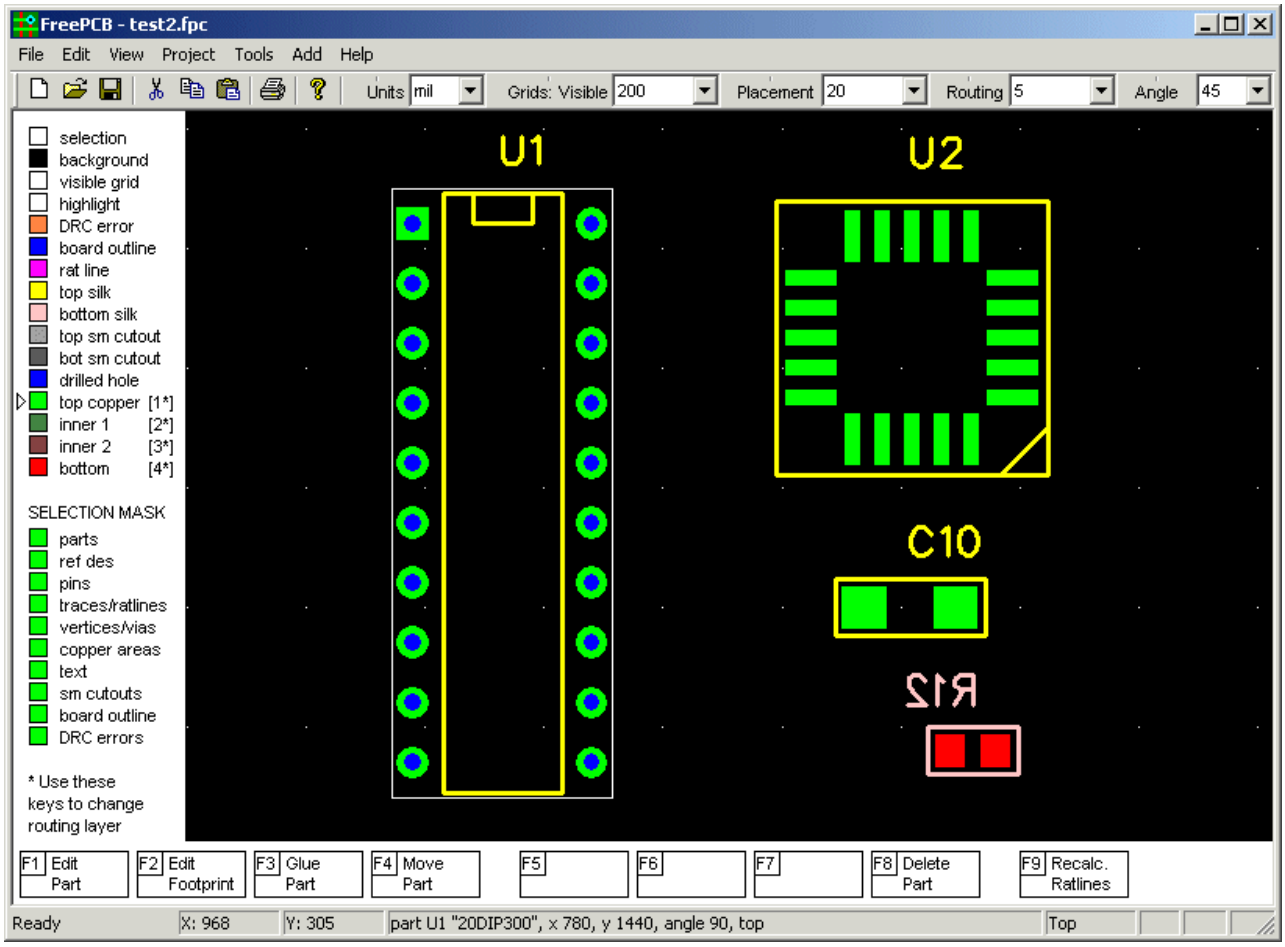
- **reference designator**, such as "U1" - This is usually assigned in a **schematic diagram**, and must be unique.
- **package identifier**, such as "DIP14" - Parts come in various physical packages, and the PCB designer must know which package will be used for each part, because this will determine the **footprint** for the part in the PCB layout.
- **Pins** - The electrical terminals of a part are called pins, even when they are non-pin terminals such as wire leads, tabs, etc. Pins are identified by a name (which is usually a number) which is appended to the reference designator, such as "U1.4" for pin four on part U1. The way pins are named will depend on the package. All pins must be named, even interchangeable pins such as the leads on a resistor.
- **mounting holes** - Mounting holes may be used to attach parts to a PCB. A fastener such as a machine screw may be passed through the hole, or the part may have built-in fasteners such as press-fit posts or solder tabs. Mounting holes may be insulated, or may have pads and be plated-through. It is not unusual to use a plated-through mounting hole to make an electrical connection to a part, such as a ground connection to a connector shield. Therefore, FreePCB treats all mounting holes as pins. The pin names assigned to mounting holes are usually numbers that are higher than the numbers used for conventional pins. For example, a 9-pin D-SUB connector with 2 mounting holes would have pin numbers 10 and 11 assigned to the mounting holes.
- **Footprint** - This is the pattern of copper pads and other elements which will become part of the PCB layout.

  Actually, when we use the term "part" during PCB layout, we are usually referring to the part's footprint, because this is what we are working with in the layout. The actual part is the electronic component which will be soldered to the footprint when the PCB is assembled.

Footprints have the following elements:

- **pads** - These are the copper elements to which pins will be soldered. There are two types:
  - **Through-hole pads** have a hole drilled through them so that the pin can be passed through the hole before soldering. For a through-hole pad on a multilayer board, there is usually a pad on each copper layer, connected together by plating the hole with copper. This arrangement is called a **padstack**. Through-hole pads are usually round or square. A common arrangement is to use a square pad for pin 1 and round pads for the rest of the pins.
  - **Surface-mount pads** do not have a hole, and are only on one layer. Nevertheless, they are still represented by a padstack, although it is a pretty simple one with only one pad. Surface-mount pads are almost always rectangular.
- **part outline** - This is a graphical representation of the part on the silk-screen layer.
- **reference text** - This is a text string for the reference designator on the silk-screen layer.
- **other text** - Text strings such as pin numbers, etc. can be added to the silk-screen layer.
- **selection rectangle** - This is invisible until the part is selected, then it is shown in the selection layer color (usually white). It determines the boundary of the part for selection purposes.
- **origin** - This is an invisible point which serves as the origin for the internal coordinate system of the footprint. When a footprint is created, the position of each element will be specified relative to the origin. When a footprint is dragged during editing, it will be "attached" to the cursor at the origin. When the position of a part is reported in the status bar or in dialogs, it is actually the position of the origin which is reported. The center of pin 1 is often used as the origin.

Some typical footprints are shown below.

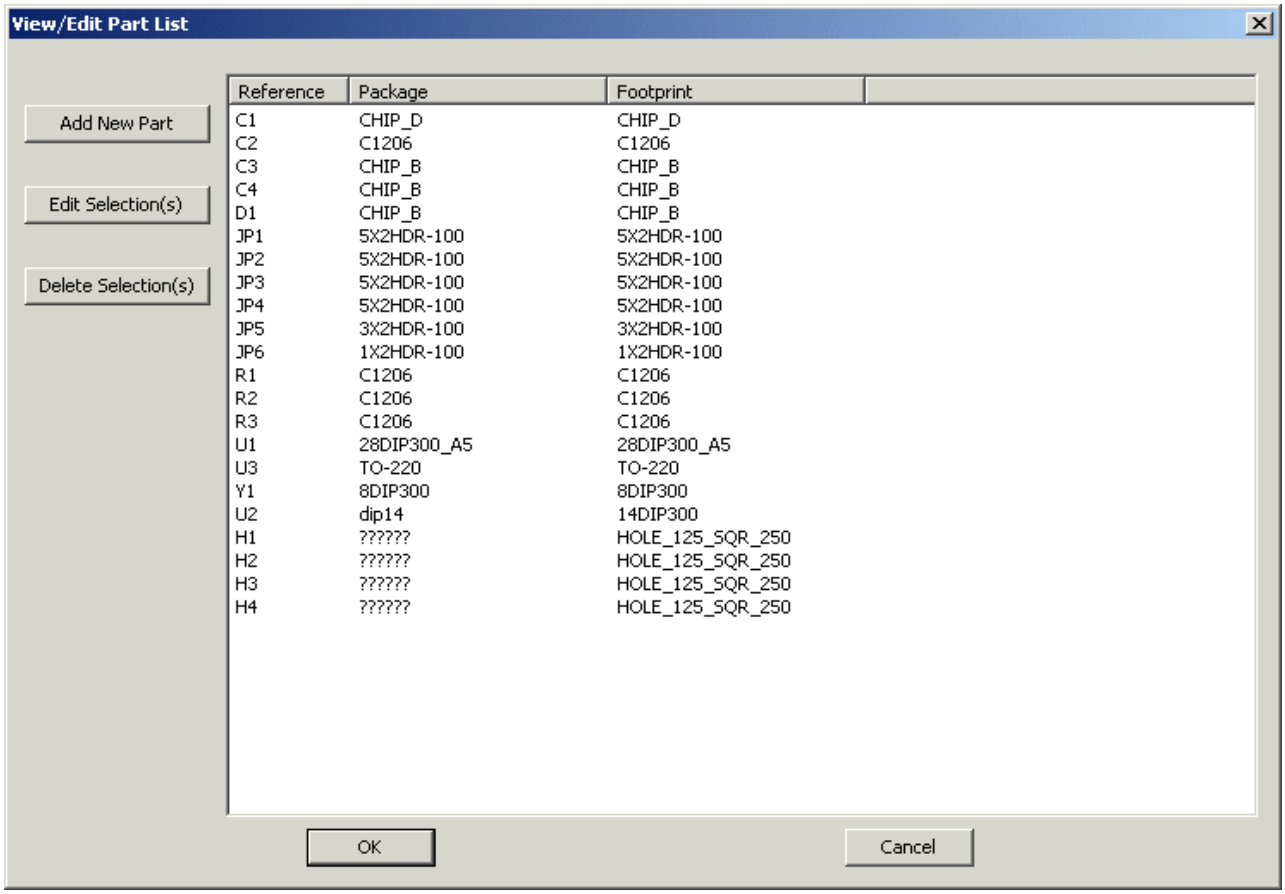


In the screenshot above, U1 is an IC in a 20-pin DIP package, with though-hole pins. It has been placed on the top (or component) side of the board, so the silk-screen elements of the footprint are on the top silk-screen layer. It has been selected for editing, so its selection rectangle is visible. Since the top copper layer is the active routing layer, its top pads are shown in the color for the top copper layer.

U2 is an IC in a 20-pin surface-mount PLCC package, also on the top side of the board. Its pads are drawn in the color for the top copper layer. C10 is a surface-mount capacitor in a chip package. R12 is a surface-mount resistor in a different chip package, on the bottom (or solder) side of the PCB. Its pads are drawn in the color of the bottom copper layer and its silk-screen elements are drawn on the bottom silk-screen layer. Also, its reference text is drawn as a mirror-image, so that it will be readable from the bottom of the board.

5.11.2 The Partlist

In FreePCB, all of the parts in a project are contained in a data structure called the **partlist**, which is empty when the project is first created. Normally, parts are added to the partlist by importing a **netlist file** from a schematic editor. This will be covered in **Section 5.14: Importing Netlist Files**. You can view or edit the partlist by selecting **Parts...** from the **Project** menu, which pops up the **View/Edit Part List** dialog, as shown below.



Each part in the project is listed by reference designator, package and footprint. Individual parts can be selected from the list by clicking on them, or multiple parts can be selected by clicking with the Ctrl key held down. The partlist can be modified using the **Add New Part**, **Edit Selection**, and **Delete Selection(s)** buttons. These will be described in the following sections.

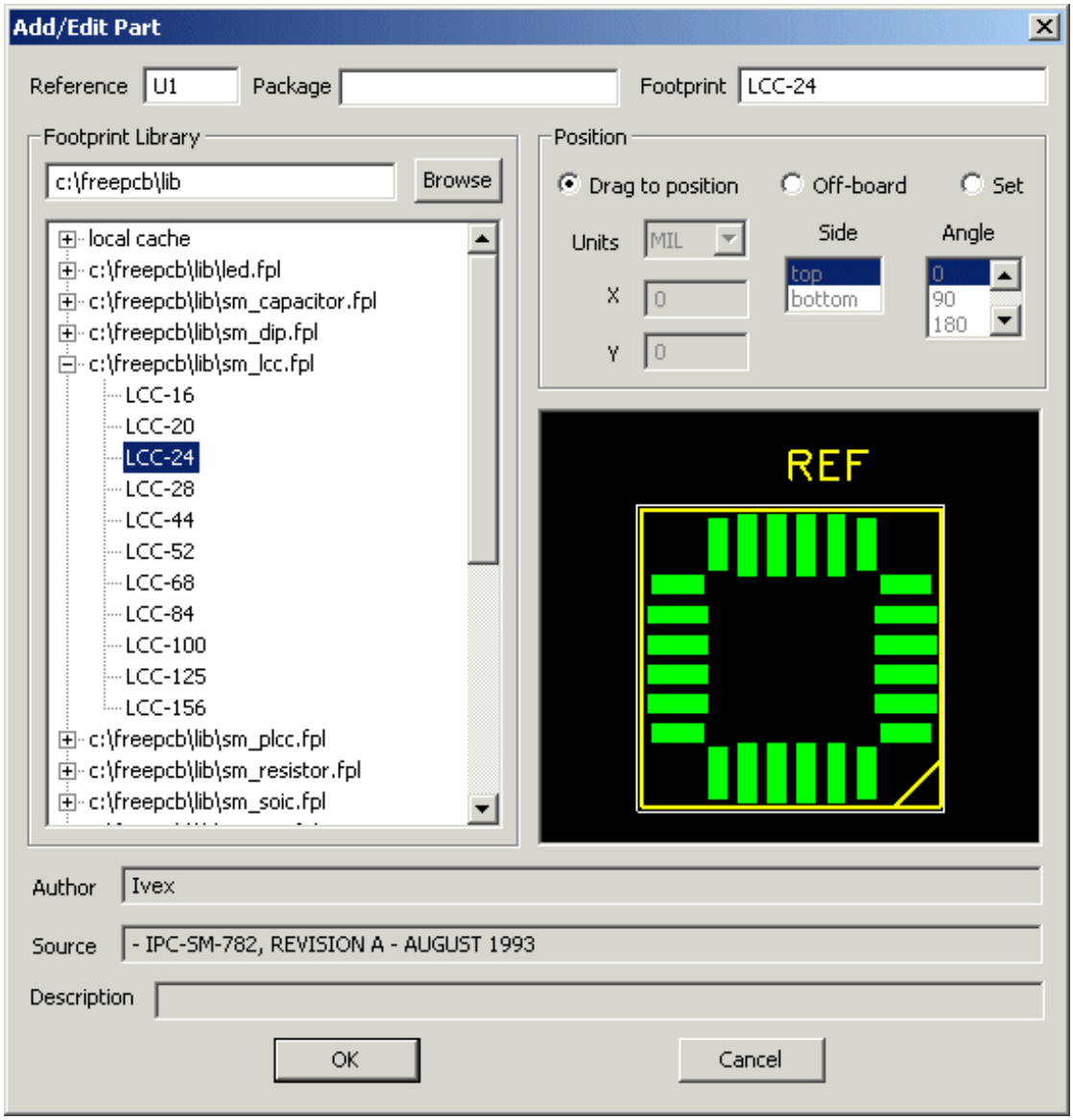
Important Note: You should be careful when modifying the partlist, for two reasons.

- If you are working from a schematic, editing the partlist can introduce discrepancies between the schematic and the PCB. This should not be a problem if you are just assigning footprints, but could be very confusing if you add or delete parts or change reference designators. If you do make any changes like these, you should **back-annotate** the schematic to reflect them.
- Once you click **OK** in the **View/Edit Part List** dialog, the changes that you made CANNOT be undone with the **Undo** function from the **Edit** menu. Therefore, you should probably save a copy of your project beforehand.

5.11.3 Editing Parts

Parts can be added, deleted or modified from either the **View/Edit Part List** dialog (shown above) or the layout window, using the methods described below.

- New parts can be added to the partlist using either the **Part** selection from the **Add** menu, or the **Add New Part** button in the **View/Edit Part List** dialog. Either method pops up the **Add/Edit Part** dialog, shown below



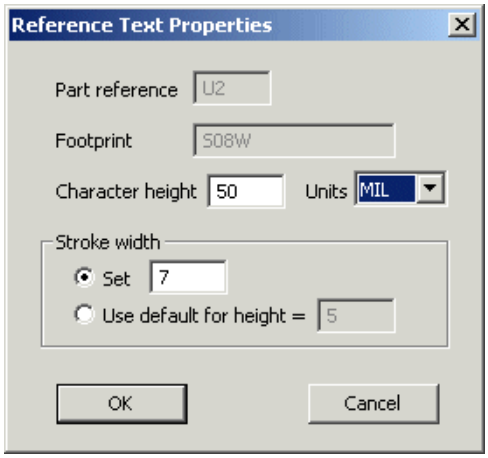
- A new part is created by entering a reference designator into the **Reference** field, and selecting a footprint from the large tree control which occupies the left side of the dialog. This control contains a list of all of the files in the selected library folder. You can change library folders using the **Browse...** button. There is also an entry for "local cache", which contains all of the footprints which are already loaded into the project. You can expand a library by clicking on the "+" next to it. In the example above, the library **C:\FreePCB\lib\sm_plcc.fpl** has been expanded. The footprint "LCC-24" has been selected and is shown in the preview pane. Its name has been automatically entered into the **Footprint** field of the dialog. Alternatively, you can just type the footprint name directly into this field.
- If you leave the **Position** radio button set to **Drag to position**, you will be dragging the new part when you click **OK**. If you want to position the part explicitly, you can use the **Set position** radio button and the **X**, **Y**, **Angle** and **Side** fields in the **Position** box. Then when you click **OK**, the new part will be placed at the position that you set. If you select **Off-board**, the part will be placed to the left of the origin.

◆ Parts can be deleted or edited from the layout window by selecting them, and using the function-key or context menus. With a part selected, the following menu items are available:

- **F1 ("Edit Part")** - this pops up the same **Add/Edit Part** dialog which was shown previously, except that the **Reference** and **Footprint** fields are filled-in. You can use this dialog to change the footprint or position of the part.
- **F3 ("Glue Part")** - this "glues" a part in place so that it cannot be accidentally moved without "ungluing" it. For a part which is already glued, this command changes to "Unglue Part".
- **F4 ("Move Part")** - this is the most commonly used command. It initiates dragging the part with the cursor. Then left-click to "drop" the part at a new position. While dragging, the following function-keys are active:
  - **F2 ("Change Side")** - flip the part from the top side to the bottom or vice-versa
  - **F3 ("Rotate Part")** - perform a 90 degree clockwise rotation of the part
- **F7 ("Delete part")** - this deletes the part from the partlist. You will be asked if you also wish to delete all references to the part from the netlist. Normally you would answer "Yes" unless you think that you might add the part back at a later time.
- **F8 ("Recalc. Ratlines")** - this recalculates the ratlines for all nets connected to the part, to minimize their total length. By default, this command is executed automatically each time a part is moved.

- Parts may also be deleted or edited from **View/Edit Part List** dialog, as follows.
  - You can delete one or more parts from the list by clicking on them (holding the Ctrl key down for multiple selections), and then click the **Delete Selection(s)** button. References to the part(s) will NOT be removed from the netlist.
  - A single part can be edited using the **Edit Selection** button, which pops up the **Add/Edit Part** dialog, which was shown earlier. The main reason for using the **View/Edit Part List** dialog to edit parts, instead of just selecting them in the layout window and using the function key menu, is to assign footprints to parts which don't already have them. A part without a footprint can't be displayed, so it can't be selected in the layout window. This situation often arises when importing netlist files, as discussed in **Section 5.14: Importing Netlist Files**.

5.11.4 Moving or Resizing the Reference Designator



The reference designator for a part may be moved by selecting it and pressing F4 ("Move Ref Tex"). Then you can drag it to a new position. While dragging, it can be rotated 90 degrees clockwise by pressing F3 ("Rotate Ref Text").

You can change the size of the text string by selecting it and pressing F1 ("Set Size"). This pops up the dialog shown below. You can change the character height and the stroke width. If you select **Use default for height**, the stroke width will automatically be set to 1/10 of the height.

5.11.5 Making the Reference Designator Invisible

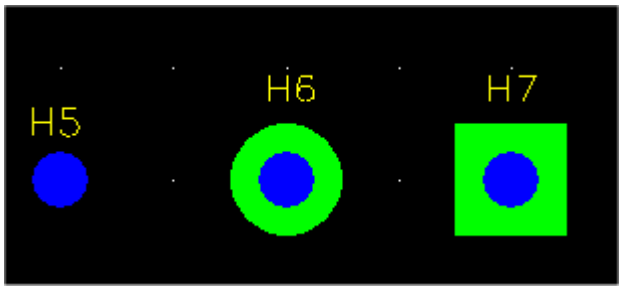
If you set the character height of the reference text to "0", the reference designator will become invisible. To make it visible again, select the part and right-click, then select **Set Ref. Text Size** from the context menu that pops up. Then set the character height to a non-zero value.

5.12 Mounting Holes

Mounting holes are used to attach a PCB to an external structure such as a bracket or enclosure. In FreePCB, a mounting hole is a part which consists of a single through-hole pin. The padstack for the pin may include pads which can be used to connect the mounting hole to a net, just like any other pin. The actual hole in the padstack will be shown in the color for the drilled-hole layer. If the mounting hole has pads, they will be shown in the color for their copper layer.

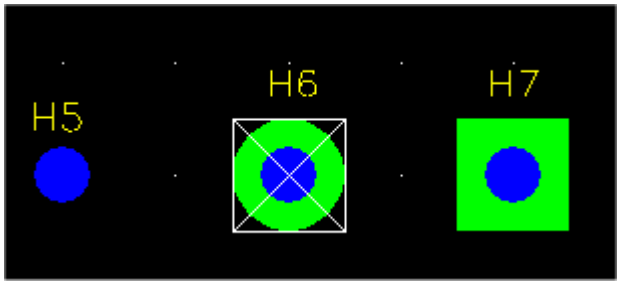
Since the mounting hole is a part, it can be added to the partlist like any other part. It can be included in the netlist file, or it can be added using the **Add > Part** or **Project > Parts...** menu selections. Some mounting holes of different sizes are included in the library file **th_mounting_hole.fpl**.

Examples of mounting holes are shown below. H5 has no pad, H6 has a round pad and H7 has a square pad.

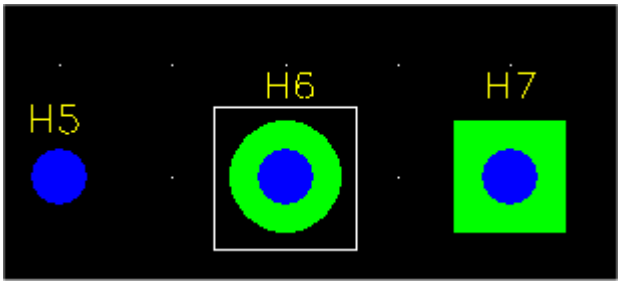


When using the mouse to select a mounting hole, you can select either the part or the pin. You would select the part if you wanted to move it, for example, while you would select the pin to add a connection to a net. The selection rectangles for the part and the pin are almost exactly the same size. Since the pin has priority, you will select it first when you click on it. You must click again to select the part. You can tell whether you have selected the pin or the part by looking at the selection rectangle or by checking the status bar.

In the examples below, the left image shows the pin for H6 selected. The right image shows the part for H6 selected, after clicking again.



Pin selected



Part selected

Selecting the reference designator for a mounting hole may be difficult if it overlaps the pad or the hole for the pin, since clicking on it with the mouse will select either the pin or the part. To get around this, you can use the selection mask to disable selection of parts and/or pins.

Since the reference designator of a mounting hole is not very useful, you may want to make it invisible by selecting and it and using F1 ("Set Size") to set its character height to zero. You can also set the character height by selecting the part and using **Set Ref. Text Size** from the right-click menu, which is how you would make the reference designator visible again if you wanted to.

5.13 Nets, Ratlines and Routing

5.13.1 Nets

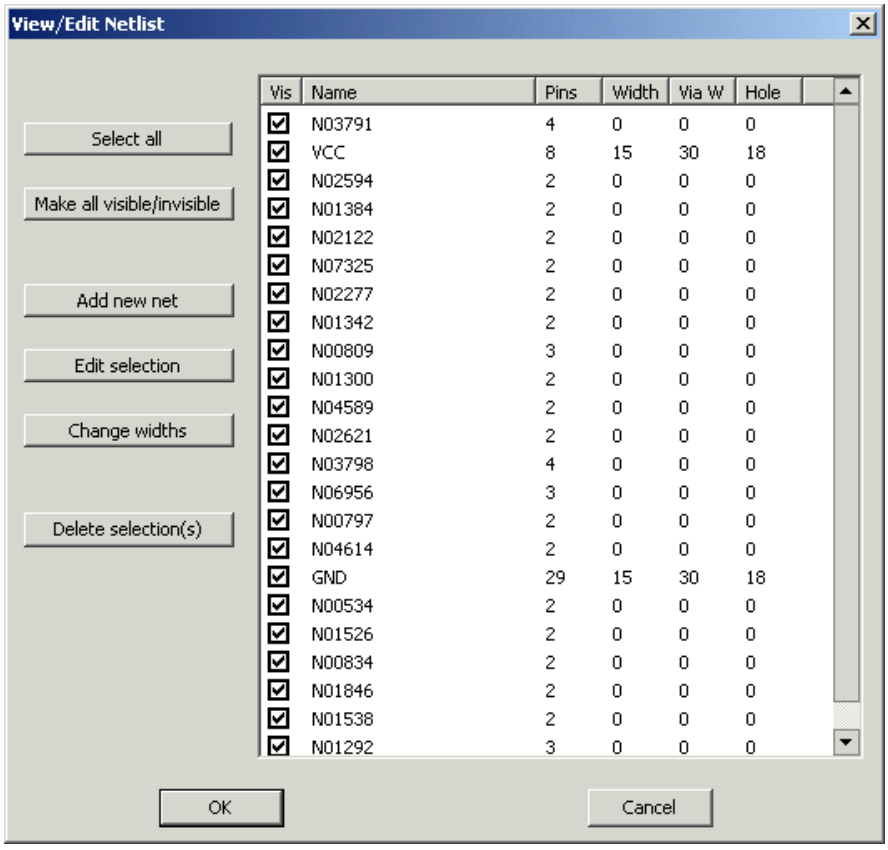
A **net** is a set of pins which will be connected together on the PCB. Each net must have a unique name. These names may be descriptive (such as "GND" or "video_in") or merely distinctive (such as "N06744"). In FreePCB, net names are limited to 40 characters in length, and may contain special characters. Each **pin** in a net is identified by a string consisting of the **reference designator** of the part containing the pin, the character "**.**", and the **pin name**.

Nets are usually read from a **netlist file**, which also includes a list of **parts** in the design (see **Section 5.14: Importing Netlist Files**). Two examples of nets are shown below.

Name: N00834
Pins: U2.4 JP1.9

Name: GND
Pins: U2.7 C2.2 U3.3 C1.2 U1.19 Y1.4 JP5.3 JP4.10
JP4.4 R2.2 C4.2 JP1.10 JP1.5 R1.2 C3.2 U1.8
JP6.2 JP3.3 JP2.3 JP3.10 JP3.4 JP3.6 JP3.8 JP2.10
JP2.8 JP2.2 JP2.4 JP2.6 JP3.2

5.13.2 The Netlist



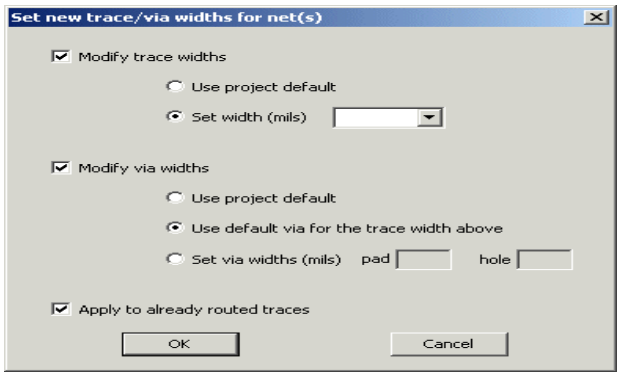
Vis	Name	Pins	Width	Via W	Hole
☑	N03791	4	0	0	0
☑	VCC	8	15	30	18
☑	N02594	2	0	0	0
☑	N01384	2	0	0	0
☑	N02122	2	0	0	0
☑	N07325	2	0	0	0
☑	N02277	2	0	0	0
☑	N01342	2	0	0	0
☑	N00809	3	0	0	0
☑	N01300	2	0	0	0
☑	N04589	2	0	0	0
☑	N02621	2	0	0	0
☑	N03798	4	0	0	0
☑	N06956	3	0	0	0
☑	N00797	2	0	0	0
☑	N04614	2	0	0	0
☑	GND	29	15	30	18
☑	N00534	2	0	0	0
☑	N01526	2	0	0	0
☑	N00834	2	0	0	0
☑	N01846	2	0	0	0
☑	N01538	2	0	0	0
☑	N01292	3	0	0	0

In FreePCB, all of the information about nets is stored in a data structure called the **netlist**. It can be viewed and modified by selecting **Nets** from the **Project** menu. This pops up the **View/Edit Netlist** dialog.

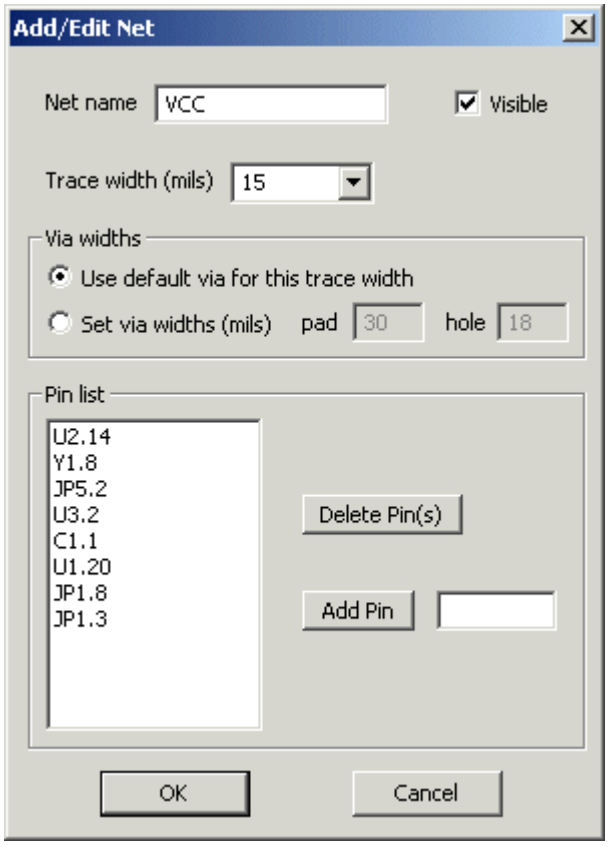
This dialog contains a list of all of the nets in the project. The name, number of pins and the default trace width, via width and via hole diameter are shown for each net.

The checkbox next to each name determines the **visibility** of the net. If the visibility box is unchecked, then ratlines for that net will NOT be displayed in the layout window. This is useful when you are working with a particular net and do not wish to be distracted with other ratlines. The **Make all visible/invisible** button checks or unchecks all of the boxes.

In the example above, the trace and via widths are set to zero for most of the nets. This indicates that the project defaults are to be used. The widths for the VCC and GND nets have been set to non-zero values which will be used for traces in those nets. Widths can be changed by selecting one or more nets, and clicking on the **Change widths** button. This pops up the following dialog, which allows setting the new widths.



You can perform more extensive editing of a single net by selecting it and clicking the **Edit selection** button, which pops up:



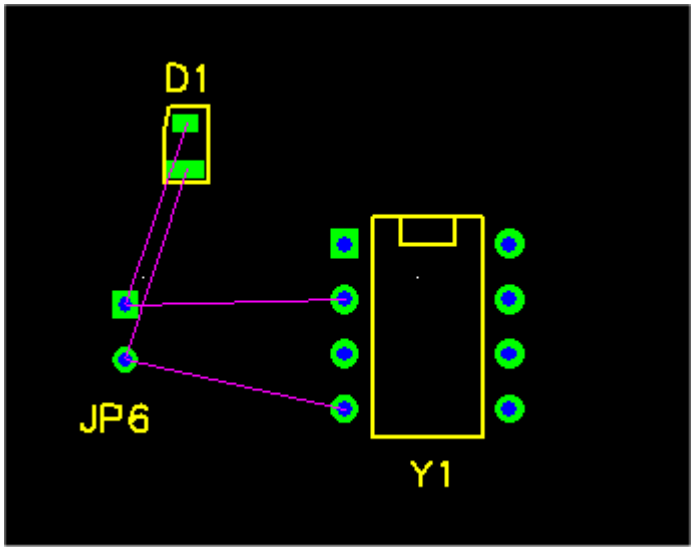
From here you can change the name, visibility and widths of a net, and add or delete pins.

5.13.3 Ratlines

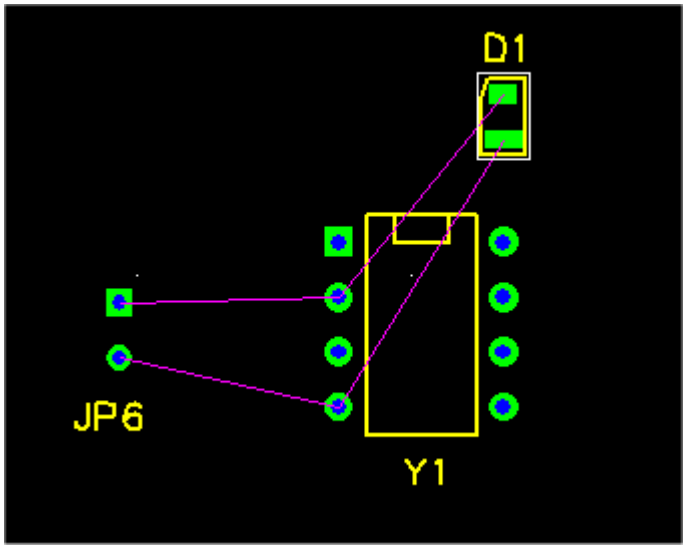
Usually, nets will be routed as a series of traces, where each trace is between 2 pins. For any net containing more than 2 pins, there will be multiple ways that the net could be routed. For example, a net consisting of pins U1.1, U2.2 and U3.3 could be routed as:

- (U1.1 to U2.2) and (U2.2 to U3.3) ***OR***
- (U1.1 to U3.3) and (U2.2 to U3.3) ***OR***
- (U1.1 to U3.3) and (U1.1 to U2.2)

For each net, FreePCB automatically generates a set of connections which attempts to minimize the total distance between the connected pins. These connections are shown as **ratlines** between pins. An example of two nets connecting three parts is shown below.



After most editing operations, FreePCB recalculates the ratlines for each affected net. This is most obvious when you are moving parts or adding copper areas. For example, moving D1 to the right causes the connections to change, as shown below.



To route the net, you can select each ratline and route it as a **trace**. Alternatively, you can ignore the ratlines and route traces from pin to pin using **stub traces**. You can also use **branching traces**. Each of these methods is described in more detail below.

5.13.4 Modifying Ratlines

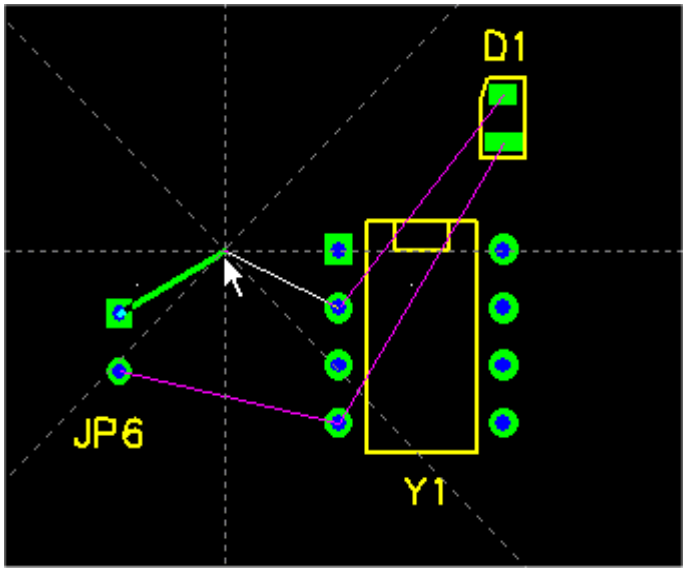
Most of the time, you will use the ratlines which FreePCB calculates. However, there may be situations when you would like more control over the ratlines. FreePCB provides several editing options for ratlines:

- A ratline can be **locked** by selecting it and pressing F3 ("Lock Connect"). The appearance of the ratline doesn't change, but "(L)" will appear in its description on the status bar to indicate that it is locked. A locked connection will NOT be removed when FreePCB recalculates ratlines. A locked connection may be unlocked by selecting it and pressing F3 ("Unlock Connect").
- A connection may be added to a net by selecting the pad for the starting pin of the connection, and pressing F4 ("Connect Pin"). You can then draw a ratline to another pin on the same net. Normally, you should lock the ratline after adding it, otherwise it will probably be removed by FreePCB the next time the ratlines are recalculated. You can also make a connection to an unassigned pin, in which case the new pin will be added to the net.
- A ratline may be removed from a net by selecting it and pressing F7 ("Delete Connect"). If the ratline is locked, you will be prompted to unlock it. If you remove a ratline and don't replace it by adding and locking another one, it will probably reappear the next time the ratlines are recalculated.
- The ratlines for each net are usually recalculated after each editing operation which affects the net. The exceptions are those operations which directly modify the ratlines, such as the ones described above. To force FreePCB to regenerate ratlines, press F8 ("Recalc. Ratlines"). The scope of this command depends on which item was selected when you pressed F8:
    - If nothing was selected, ratlines will be regenerated for every net in the project.
    - If a part was selected, ratlines will be regenerated for each net attached to the part.
    - If a pin, ratline, or trace was selected, ratlines will be regenerated for the net attached to that pin, ratline or trace.

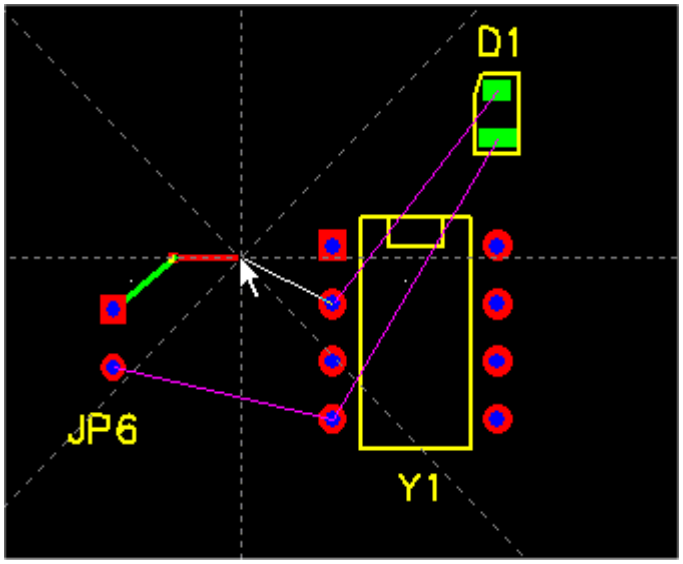
5.13.5 Routing with Ratlines

This method consists of converting ratlines to copper traces. Traces consist of one or more straight-line **segments**, connected together with **vertices** at their connection points. If two adjacent segments are on different layers, a **via** will be placed at the vertex between them.

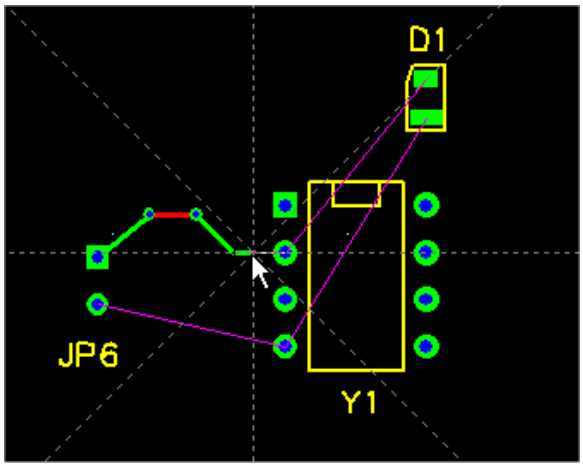
To begin routing, select a ratline by clicking on it. It will change to the selection color to indicate that it has been selected. Then press F4 ("Route Segment"). You will find yourself dragging a trace segment from the pin of the connection that was closest to the cursor when you pressed F4. The image below shows a segment being routed from pin 1 of JP6.



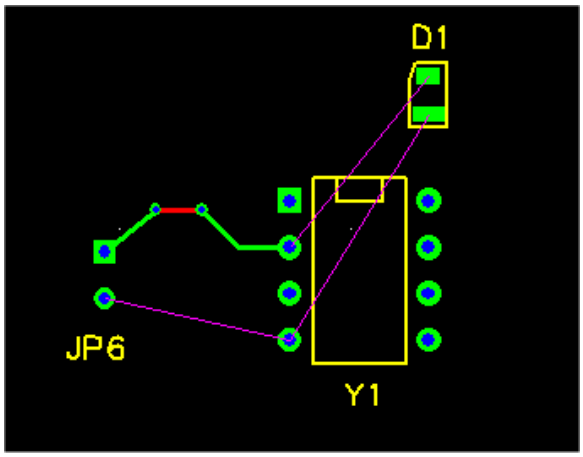
Note that the segment is on the top copper layer because that was the active layer when we started routing. You can change the active layer by pressing one of the numeric keys "1" through "8", which represent the available layers in top-to-bottom order. For example, for a four-layer board, "1" would select the **top copper** layer, "2" would select the **inner 1** layer, "3" would select the **inner 2** layer, and "4" would select the **bottom copper** layer. If you change layers while routing a segment, it will immediately switch to the new active layer. While a segment is being dragged, the **Routing Grid** will be in effect. If the starting point for the segment is on the routing grid, then the **Snap Angle** will also be in effect.



Left-clicking will place a vertex at the cursor position, end the first segment at the vertex, and initiate routing a second segment from the vertex, as shown in the next image. We have also switched layers by pressing the numeric key "4", so that the new segment is being drawn on the bottom copper layer.

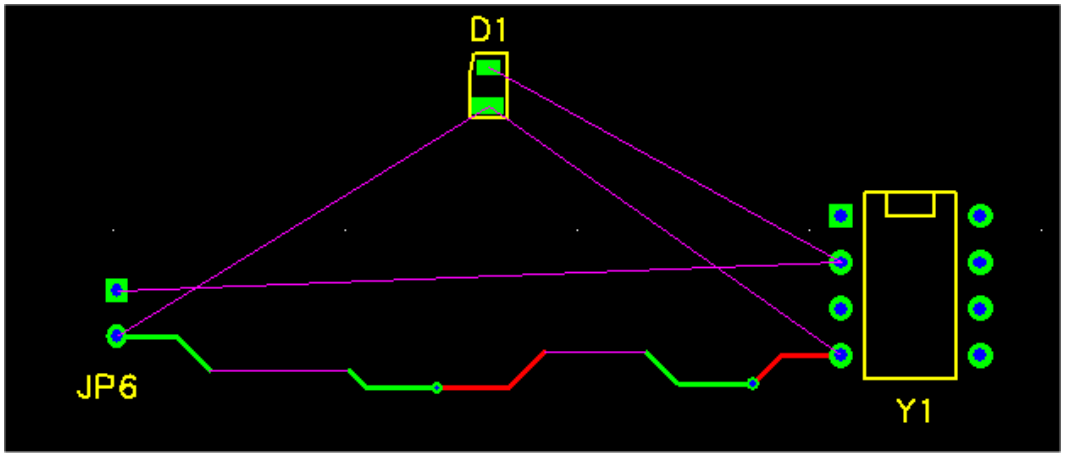


Left-click again to place the next vertex and start routing the next segment. In this way you can draw a trace consisting of multiple connected segments on multiple layers. The image at left shows our trace nearing completion with several segments on different layers, with 2 vias.



To finish the trace, press F4 ("Complete Segment") while routing the last segment. The segment will automatically be extended to the end-pin of the trace. Alternatively, you can place the cursor over the end-pin and left-click. Our final trace is shown to the left.

While routing a trace, you can right-click to stop routing before the trace is completed. The unrouted portion of the connection will be shown as a ratline between the last vertex and the end-pin. You can finish the trace later by selecting the ratline and routing it from either end. A trace can even have multiple unrouted portions, as shown below.

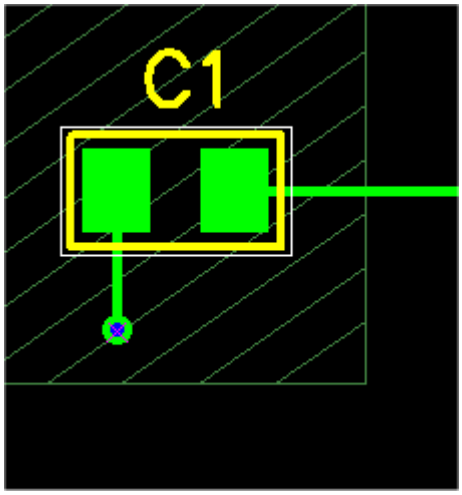


You can change the end-pin of a partially-routed trace by selecting the ratline to the pin and pressing F5 ("Change Pin"), then drawing a new ratline to a different pin.

5.13.6 *Deleted*

5.13.7 Routing with Stub Traces

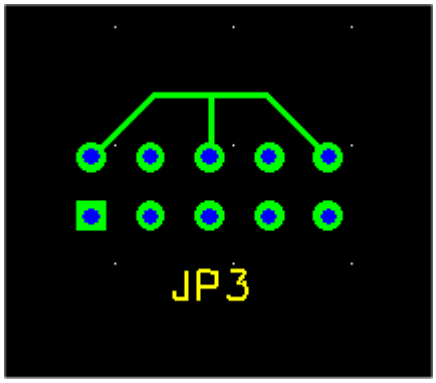
Besides traces based on connections between pins, FreePCB supports **stub traces**. These are traces that start at a pin, but do not end at a pin. Instead, they end blindly, usually with a via. Stub traces are mainly used to connect SMT pins to copper areas, which will be described in **Section 5.15: Copper Areas**. To create a stub trace, select the starting pin. Press F3 ("Start Stub") to start dragging a trace segment from the pin. The segment will be on the currently active layer, unless the pin is a SMT pin, in which case the active layer will change to the layer of the pin. Left-click to place a vertex. If you want more than one segment, add additional vertices. Then terminate the stub trace by right-clicking. The trace will end at the last vertex that you placed, and a via will be added at the vertex. If you don't want the via, select it and press F3 ("Delete Via"). If the via passes through a copper area on the same net as the starting pin, a thermal relief will be placed to connect the via to the copper area. An example of a stub trace connecting a pin to a copper area is shown below. The thermal relief is represented by the "X" drawn on the via in the ratline color.



Stub traces can also be used for pin-pin routing. Start routing the stub trace from a pin as described above. Instead of terminating it by right-clicking, route it to a pin that is on the same net or that is not assigned to any net. When you place the end-point of a segment anywhere on the pin, the stub trace will automatically be converted to a regular pin-pin trace. This can be very useful if you are designing a PCB "on-the-fly", without a netlist.

5.13.8 Routing with Branching Traces

Once you have created a trace, branch traces can be added to it that connect it to other pins. An example is shown below, where the main trace is between the pins at either end of JP3, and the branch is between the middle pin and a vertex in the main trace.



There are two ways of adding branches to a trace. The first method is to select a vertex in the trace and press F3 ("Connect Pin"). Then draw a ratline from the vertex to the pin that you want to add, and route the ratline as usual. The second method is to route a stub trace from the pin to the vertex.

Notice that there are no visible differences between the main trace and the branch trace. However, there will be differences in some editing operations. For example, if you delete the branch trace then the main trace will remain, but if you delete the main trace then both traces will disappear. You should keep this in mind when creating branching traces that you might want to change later.

You can tell whether segments and vertices are on the main trace or a branch by selecting them and looking at the status bar. The main pin-pin trace will be described as a trace, while a branch trace will be described as a branch. The vertex that a branch connects to is called a "tee-vertex". If you select a tee-vertex, the status bar description will end with a parameter such as "(T1234)", where the "T" indicates a tee-vertex and the "1234" is an identifier for the tee. Branch traces that connect to this tee will show the same parameter in the status bar.

Besides pin-pin traces, branches can be added to stub traces and even to other branch traces.

5.13.9 Vias

Vias are used to connect copper elements on different layers. In FreePCB, they are considered part of a trace, and can only appear at trace vertices.

Vias can be **forced** or **unforced**. An unforced via is generated automatically while routing a trace. It will appear whenever necessary to connect trace segments on different layers, and will disappear if the trace is modified so that it is no longer necessary. Usually, this is the behavior that you will want. However, there are times when you would like a via to be present when the adjacent trace segments are on the same layer. For example, you might be relying on a via to connect the trace to a copper area. In this case, you can "force" a via by selecting a vertex and pressing the "f" key. A forced via will not be removed automatically. You can tell if a via is forced by selecting it and looking at the status bar, which will show "(F)" for a forced via. A forced via can be unforced by pressing the "u" key. Note that the vias that are created automatically at the end of stub traces are forced by default.

When vias are created, default values will be used for the pad and hole diameters. These will be the default values for the net, or if these are unspecified then the project default values will be used. The pad and hole widths for vias can be changed using the **Set Trace and/or Via Widths** dialog, which is described in the following section.

5.13.10 Modifying Traces

Traces can be modified by selecting segments, vertices or ratlines and using function-key commands. You can select an entire trace by clicking on a segment or vertex while holding down the "t" key. You can select an entire net by clicking on a segment, vertex or pin while holding down the "n" key.

Here are descriptions of the function-key commands:

- With a ratline selected:
  - F1 (Set Width) - pop up the **Set Trace and Via Widths** dialog, which is described below.
  - F3 (Lock/Unlock Connect) - lock or unlock the ratline. A locked ratline cannot be eliminated by the "Recalc. Ratlines" operation.
  - F4 (Route Segment) - start dragging the endpoint of a routed segment to replace the ratline.
  - F5 (Change Pin) - this is only available if you have the selected a ratline connecting a partially-routed trace to a pin. It allows you to change the pin that the ratline connects to.
  - F7 (Delete Connect) - remove the connection from the net.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

- While routing a segment:
  - F4 (Complete Segment) - extend the segment to the endpoint of the ratline, and stop routing.

- With a copper segment selected:
  - F1 (Set Width) - pop up the **Set Trace and Via Widths** dialog, which is described below.
  - F2 (Change Layer) - pop up a dialog that lets you change the layer of the segment, trace or net.
  - F5 (Unroute Segment) - unroute the segment, converting it to a ratline.
  - F6 (Unroute Trace) - unroute the entire trace, converting it to a ratline.
  - F7 (Delete Connect) - remove the connection from the net.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

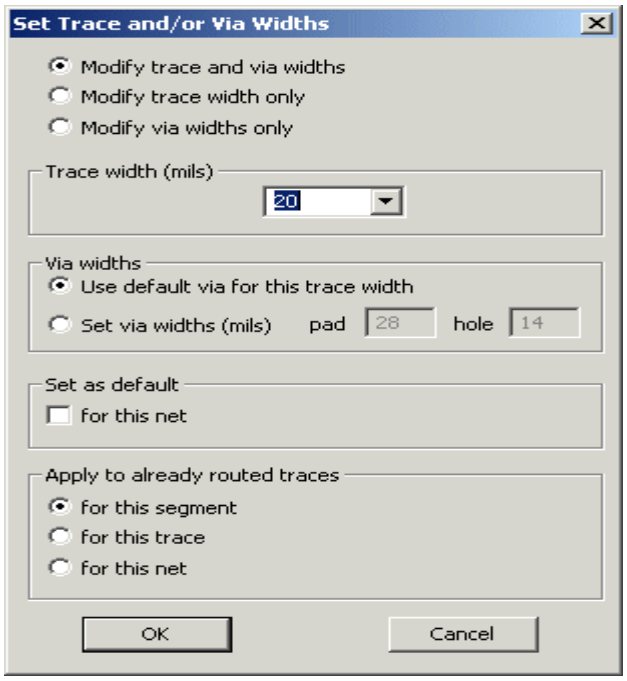
- With a vertex selected:
  - F1 (Set Position) - pop up a dialog that allows you to edit the X and Y coordinates of the vertex explicitly.
  - F4 (Move Vertex) - start dragging the vertex to move it.
  - F5 (Delete Vertex) - remove vertex, unrouting the adjacent segments into a ratline.
  - F6 (Unroute Trace) - unroute the entire trace, converting it to a ratline.
  - F7 (Delete Connect) - remove the connection from the net.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

- With the end-vertex of a stub trace selected:
  - F1 (Set Position) - pop up a dialog that allows you to edit the X and Y coordinates of the vertex explicitly.
  - F2 (Add Segment) - start dragging a new segment from the vertex.
  - F3 (Add/Delete Via) - add a via, or delete one that is already present.
  - F4 (Move Vertex) - start dragging the vertex to move it.
  - F5 (Delete Vertex) - remove vertex, unrouting the adjacent segments into a ratline.
  - F7 (Delete Connect) - remove the entire stub trace.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

- With an entire trace trace selected:
  - F1 (Set Width) - pop up the **Set Trace and Via Widths** dialog, which is described below.
  - F2 (Change Layer) - pop up a dialog that lets you change the layer of the trace or net.
  - F6 (Unroute Trace) - unroute the entire trace, converting it to a ratline.
  - F7 (Delete Connect) - remove the connection from the net.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

- With an entire net selected:
  - F1 (Set Width) - pop up the **Set Trace and Via Widths** dialog, which is described below.
  - F2 (Change Layer) - pop up a dialog that lets you change the layer of the net.
  - F3 (Edit Net) - add a via, or delete one that is already present.
  - F8 (Recalc. Ratlines) - recalculate ratlines for the net to minimize their length.

The **Set Trace and Via Widths** dialog, which is invoked by pressing F1 with a net, trace, segment or vertex selected, is shown below.



This dialog allows you to modify the trace and via widths for a segment, trace, or net. You can also set the net default to the new widths, so that they will be used for any future routing of the net.

5.13.11 Swapping Pins

You can swap pins in a part by selecting one of them, holding down the "s" key and clicking on the other pin. You will be prompted to swap the connections to the pins.

5.14 Importing Netlist Files

5.14.1 Netlist Files

After a new project is created, parts and nets can added to it by importing a **netlist file**. This file is usually produced by a schematic editor, but you can create it by hand if necessary.

A sample netlist file is shown below. This file will be used later in the tutorial (see **Section 7: Tutorial**). It was produced by a schematic editor called Orcad Capture, in PADS-PCB format. This is a common format, and most schematic editors support it. Most importantly, FreePCB can read it.

```
*PADS-PCB*
*PART*
C1 CHIP_D
C2 C1206
C3 CHIP_B
C4 CHIP_B
D1 CHIP_B
JP1 5X2HDR-100
JP2 5X2HDR-100
JP3 5X2HDR-100
JP4 5X2HDR-100
JP5 3X2HDR-100
JP6 1X2HDR-100
R1 C1206
R2 C1206
R3 C1206
U1 28DIP300
U2 dip14
U3 TO-220
Y1 8DIP300

*NET*
*SIGNAL* N00834
U2.4 JP1.9
*SIGNAL* GND
U2.7 C2.2 U3.3 C1.2 U1.19 Y1.4 JP5.3 JP4.10
JP4.4 R2.2 C4.2 JP1.10 JP1.5 R1.2 C3.2 U1.8
JP6.2 JP3.3 JP2.3 JP3.10 JP3.4 JP3.6 JP3.8 JP2.10
JP2.8 JP2.2 JP2.4 JP2.6 JP3.2
*SIGNAL* N01846
Y1.5 U1.9
*SIGNAL* VCC
U2.14 Y1.8 JP5.2 U3.2 C1.1 U1.20 JP1.8 JP1.3
*SIGNAL* N02621
U2.6 U1.21
*SIGNAL* N02594
U2.3 U1.22
*SIGNAL* N02122
JP2.1 U1.18
*SIGNAL* N01526
JP4.6 U1.15
*SIGNAL* N01538
JP4.8 U1.14
*SIGNAL* N02277
U1.17 JP3.1
*SIGNAL* N00797
JP1.4 U2.1
*SIGNAL* N00534
```

```
D1.1 JP6.1
*SIGNAL* N01300
U1.27 JP5.5
*SIGNAL* N01342
U1.28 JP5.4
*SIGNAL* N01384
JP5.1 U1.1
*SIGNAL* N00809
U2.5 JP1.7 U1.23
*SIGNAL* N03791
C3.1 R1.1 JP2.7 U1.3
*SIGNAL* N01292
U2.2 U1.24 JP1.2
*SIGNAL* N03798
C4.1 R2.1 JP3.7 U1.2
*SIGNAL* N04614
U1.13 JP3.5
*SIGNAL* N04589
JP2.5 U1.12
*SIGNAL* N06956
C2.1 U3.1 R3.2
*SIGNAL* N07325
R3.1 D1.2
*END*
```

The listing is fairly self-explanatory. In the ***PART*** section of the file, there is a line for each part, containing a reference designator (such as "**U1**") and a package identifier (such as "**28DIP300**"). In the **NET** section, each net starts with the ***SIGNAL*** keyword, followed by the name of the net. All of the pins in the net are then listed on subsequent lines.

In order for FreePCB to assign footprints, the package identifier for each part must match a footprint in one of the FreePCB libraries. Since there is no universal standard for package identifiers, you will probably have to assign these explicitly.

Basically, there are 3 ways to do this:

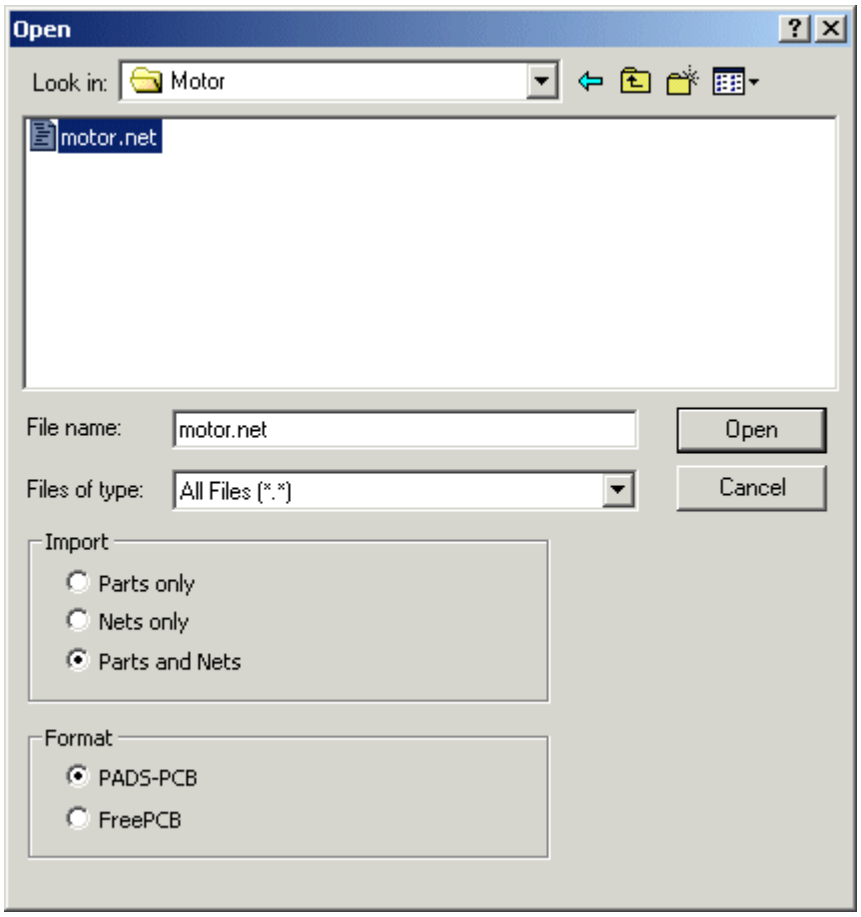
1. In the schematic editor, assign the package identifier as an attribute for each part, using package identifiers that match FreePCB footprints. Then export the netlist, and import it into FreePCB.
2. Export the netlist, edit it using a text editor, and then import it into FreePCB.
3. Export the netlist, import it into FreePCB, and then fix any incorrect or missing footprints within FreePCB.

Option 1 is definitely the way to go if you think that you might make changes to your schematic after you start routing the board, and want to import those changes into FreePCB.

Important note: For FreePCB to find the footprints referenced in the netlist file, the libraries containing those footprints must be in the default library folder for the project (usually ..\lib). If you are using footprints from libraries in other folders, such as ..\lib_contrib or ..\lib_extra, you should move those libraries into ..\lib before importing the netlist file. If you forget to do this, you can always assign the footprints later, using the **Project > Parts...** dialog.

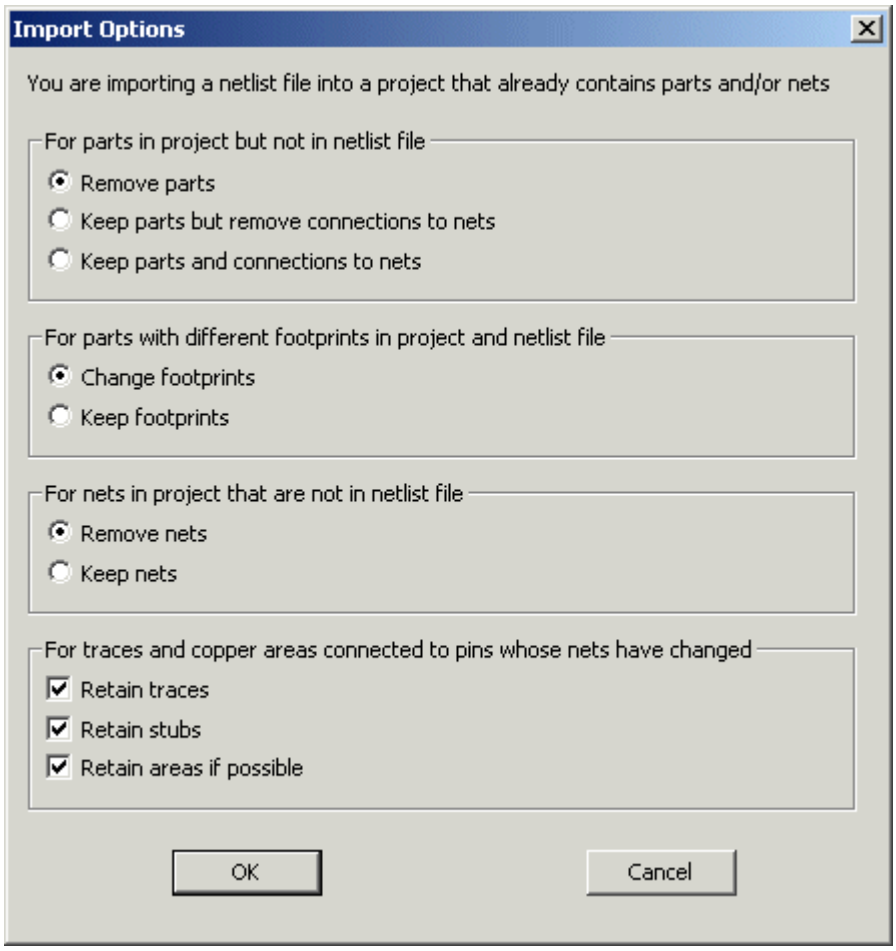
5.14.2 Importing Netlist Files into a Project

To import a netlist file, select **File > Import netlist...**. This will pop up the following dialog.



Select the netlist file by navigating to the correct folder (if necessary) and clicking on the file name. I usually move or copy the netlist file into the project folder before importing it, but this is not necessary. Using the radio buttons, you can choose to import only the parts from the file, only the nets, or both. You can also select the file format (although currently only the PADS-PCB format is supported). Then click **OK**.

If you are importing the netlist file into an empty project, without existing parts or nets, the file will load immediately. Otherwise, the **Import Options** dialog will appear, as shown below.



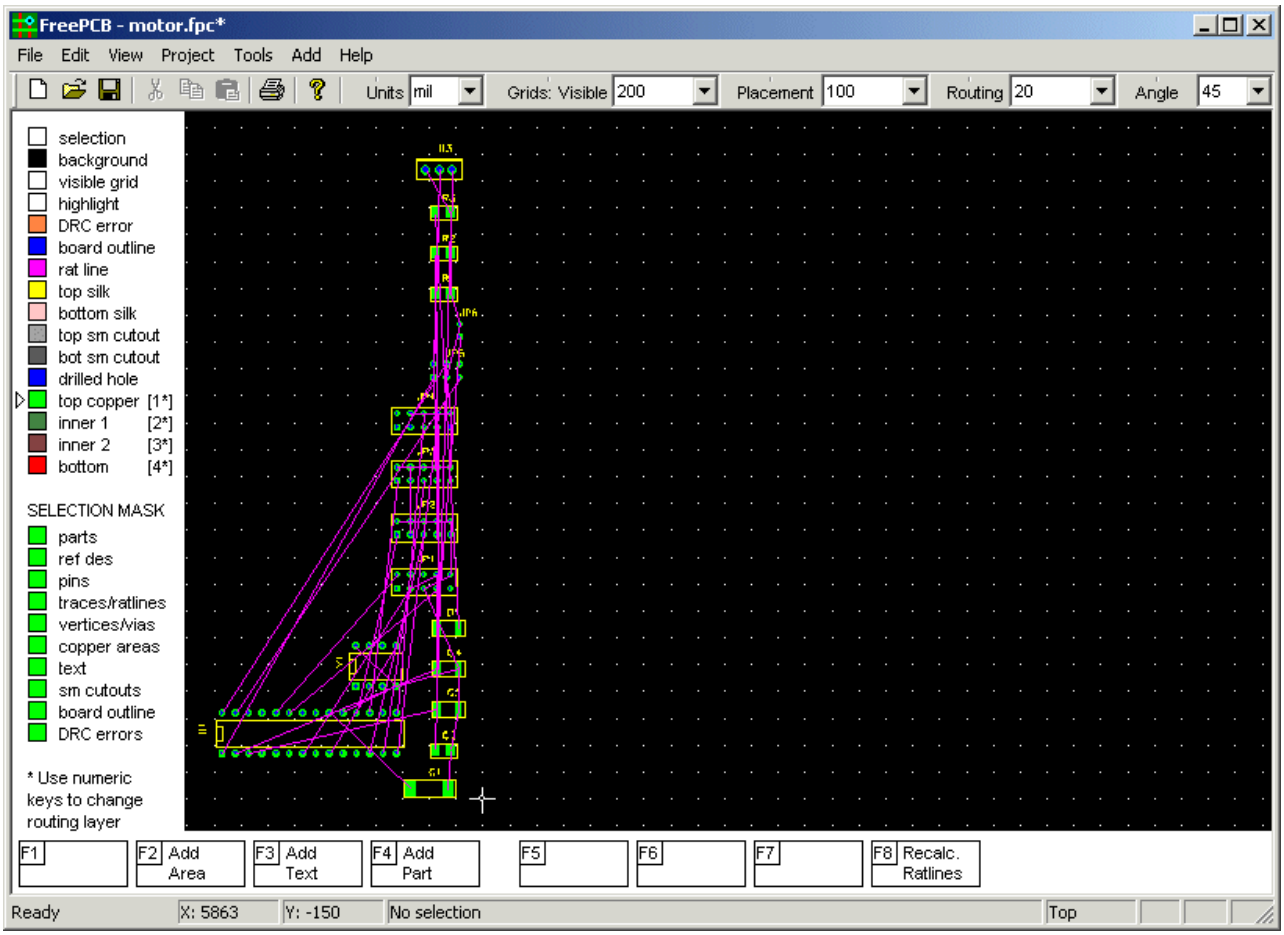
This dialog determines how FreePCB handles conflicts that may arise when importing a netlist file into a project that already contains parts and/or nets. This situation usually arises when you are using a schematic editor. For example, suppose that after importing your initial netlist and laying out some or all of the PCB, you make some changes in the schematic and wish to import those changes. With the the default settings shown below, any parts, footprints or nets in the project that don't match the new netlist file will be changed or deleted, and the project will exactly match the netlist file. Most of the time, this will be what you want.

However, suppose that you have added some parts to the project (such as mounting holes, for example) that aren't in your netlist file, and you have connected them to a net. Then you might want to select **For parts in project but not in netlist file: Keep parts and connections to nets** so that you won't lose these parts and connections. Or, if you changed some of the footprints after importing the netlist, you might want to select **For parts with different footprints in project and netlist file: Keep footprints** so that you don't revert back to the old footprints after the import. If you have added nets and wish to keep them, you might select **For nets in project that are not in netlist file: Keep nets**.

You should use these options carefully as they may cause unexpected results. For example, if you inadvertently change a net name in the schematic editor and then import the netlist with the **Keep nets** option selected, you will wind up with 2 nets with different names, with all of the pins in the new net except for any pins that were in the old net but not in the new one. Also, you will lose all of the routing on the new net (see note below). Or if you changed the reference designator of a part in the schematic editor and you select one of the **Keep parts** options, you will wind up with a duplicate part.

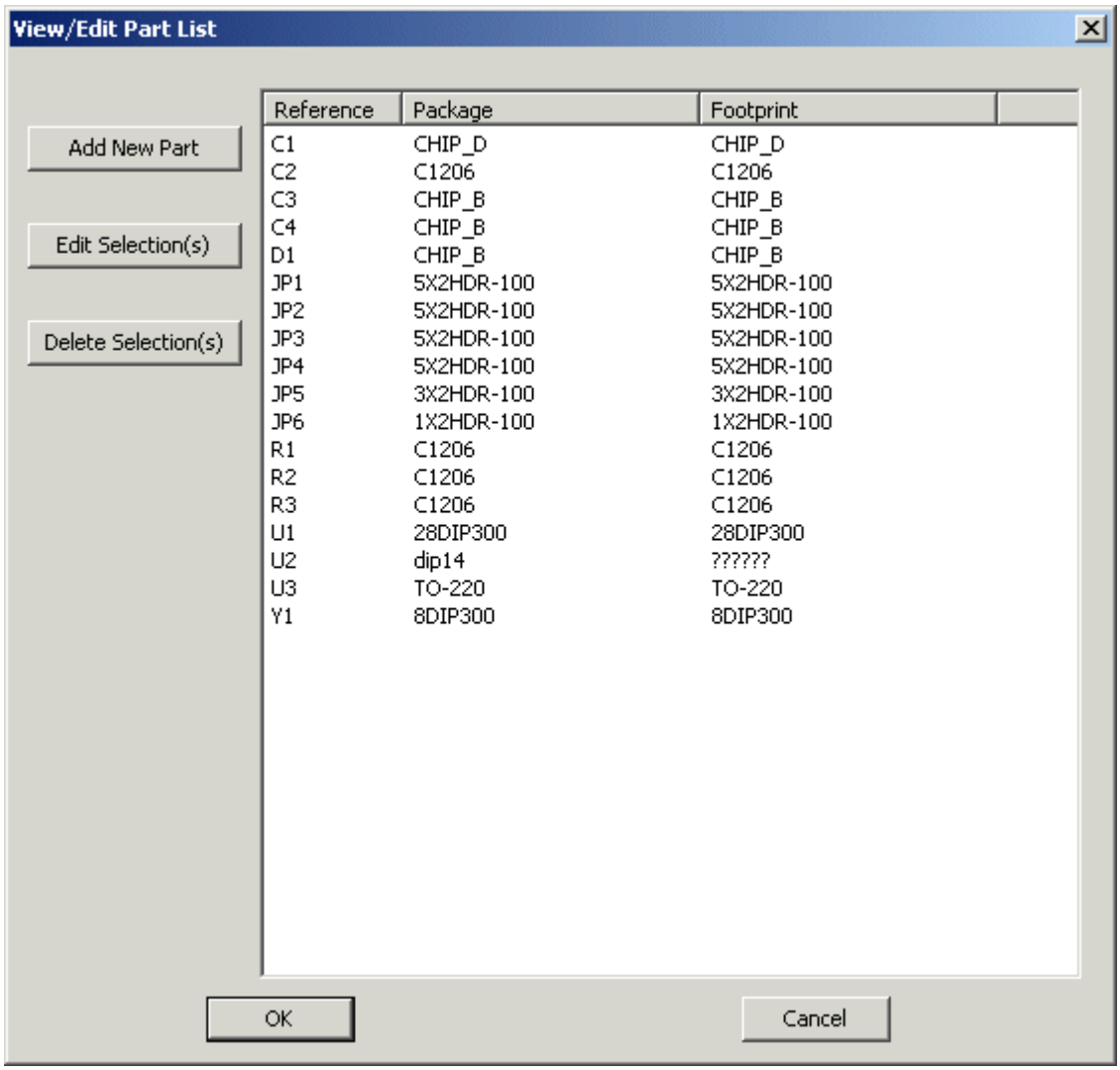
The last three options in the dialog (**Retain traces**, **Retain stubs**, **Retain areas**) determine whether FreePCB will attempt to retain routed traces and copper areas whose net names have changed in the netlist file, by analyzing the pin lists.

Once you have selected the **Import Options**, click **OK**. The netlist file will load and all of the new parts will be placed in the layout window to the left of the origin. To see them, you can select **All Parts** from the **View** menu (or press the "Home" key), which will adjust the layout window to make all of the parts visible, as shown below.



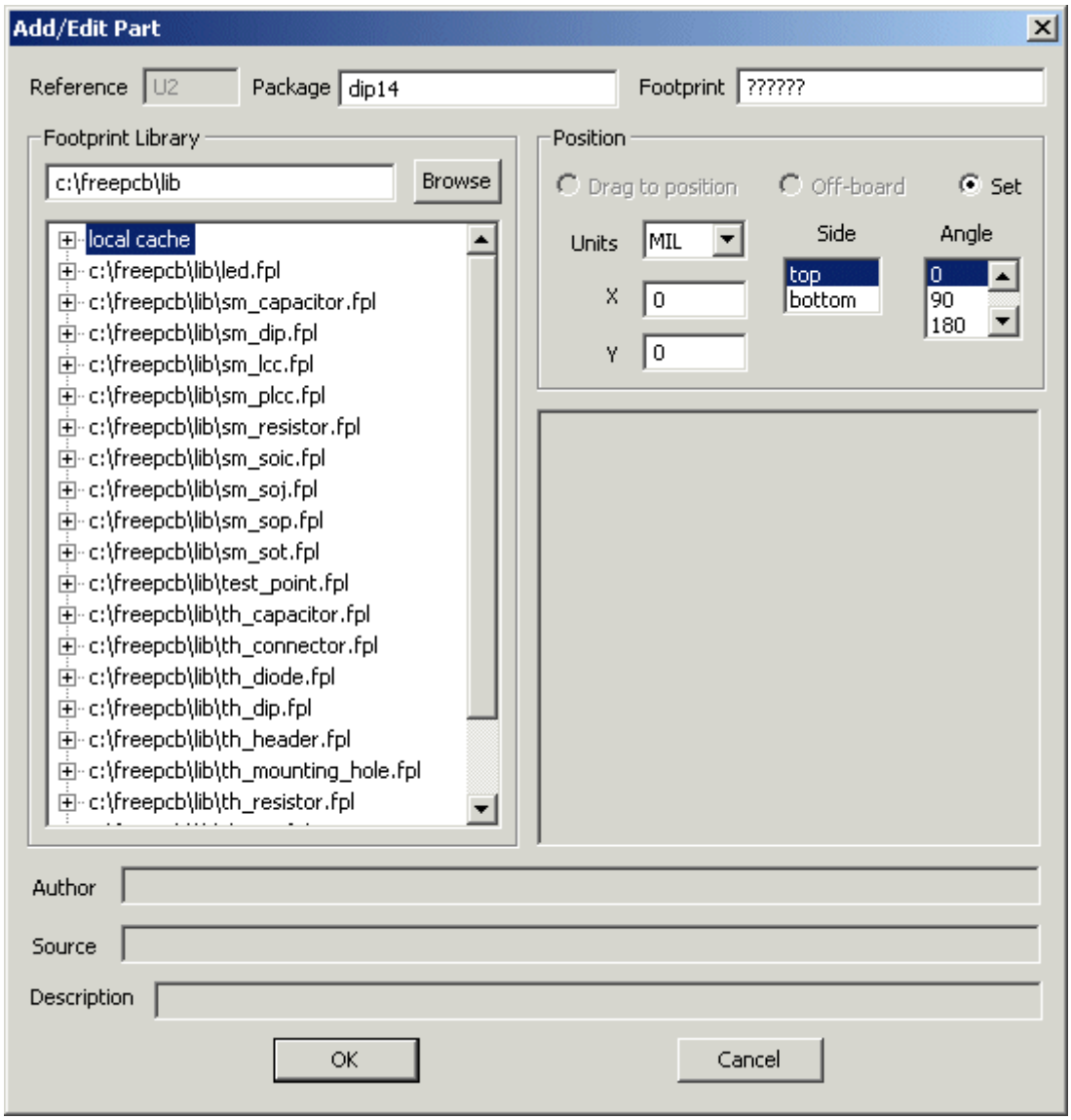
If FreePCB is unable to match any of the package identifiers with footprints from its libraries, the following message will appear.

In this case, you can check the partlist by selecting **Parts...** from the **Project** menu. This will pop up the **View/Edit Part List** dialog, as shown below.

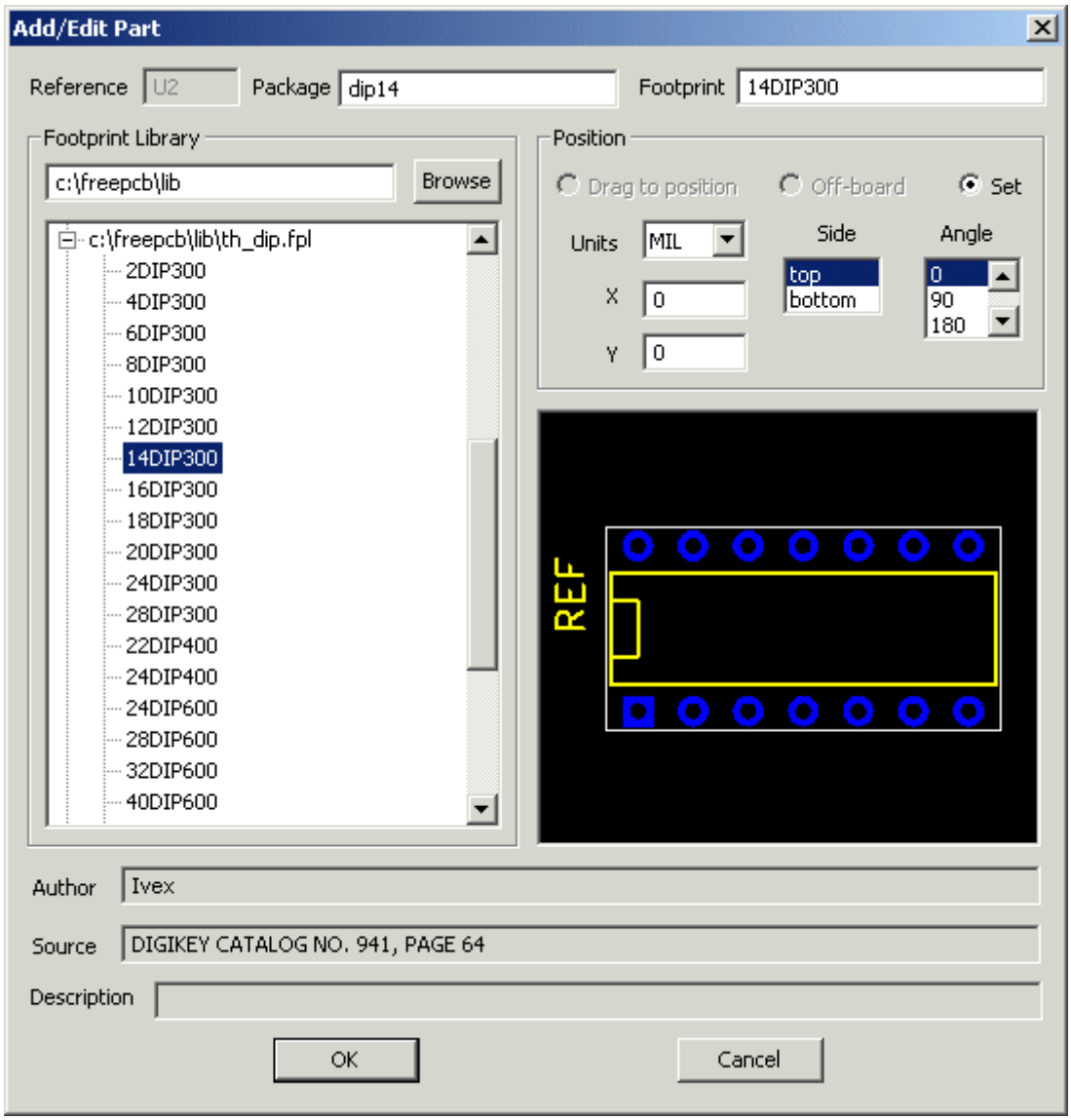


Reference	Package	Footprint
C1	CHIP_D	CHIP_D
C2	C1206	C1206
C3	CHIP_B	CHIP_B
C4	CHIP_B	CHIP_B
D1	CHIP_B	CHIP_B
JP1	5X2HDR-100	5X2HDR-100
JP2	5X2HDR-100	5X2HDR-100
JP3	5X2HDR-100	5X2HDR-100
JP4	5X2HDR-100	5X2HDR-100
JP5	3X2HDR-100	3X2HDR-100
JP6	1X2HDR-100	1X2HDR-100
R1	C1206	C1206
R2	C1206	C1206
R3	C1206	C1206
U1	28DIP300	28DIP300
U2	dip14	??????
U3	TO-220	TO-220
Y1	8DIP300	8DIP300

Note that the footprint for U2 is shown as "??????", indicating that FreePCB was unable to find a footprint with the identifier "dip14". A footprint can be assigned to this part by selecting it and clicking the **Edit Selection** button, which pops up the **Add/Edit Part** dialog, shown below.



You can assign a footprint to U2 by expanding the library files in the dialog until you find the footprint that you want, and then clicking on it to copy it into the **Footprint** field of the dialog, as shown below. If you know the name of the footprint, you can just enter it directly.



U2 should now have a footprint in the layout window, placed at whatever position you set in the dialog (X = 0, Y = 0 in the example above).

5.14.3 Exporting Netlist Files

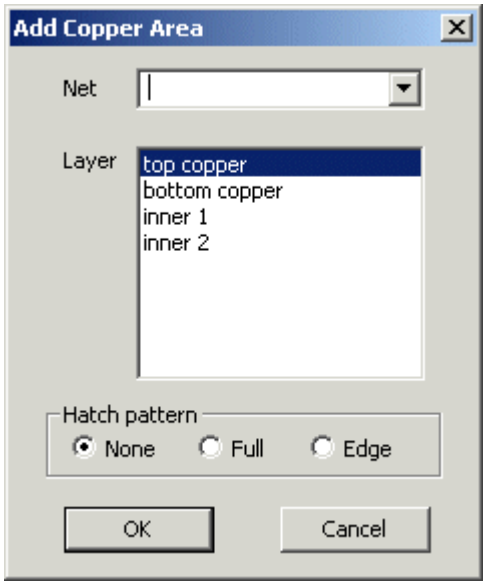
You can export a netlist file from a project using the **File > Export netlist...** menu selection. In the resulting file, the package for each part will be set to the name of the actual footprint that was used in the project.

5.15 Copper Areas

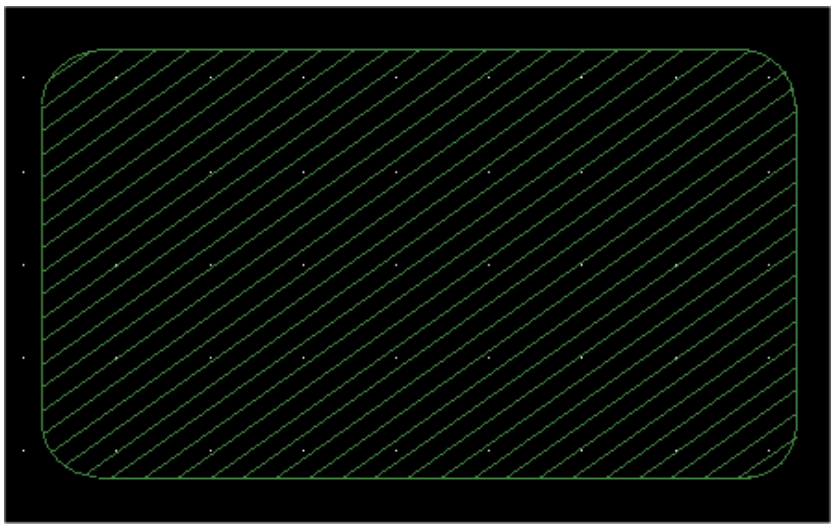
Besides traces, **copper areas** can be used for connections on a PCB. These are commonly used for power and ground planes, in which case they may occupy an entire layer of the board. In FreePCB, all copper areas must be assigned to a net.

Copper areas are drawn as closed polylines (or polygons), similar to the board outline. They can be displayed as a simple outline, an outline with internal hatching, or an outline with a little bit of hatching along its inner edge. These hatch patterns are for visual reference only, as the actual copper area on the PCB will be solid copper. FreePCB will automatically create clearances around any pads or traces which pass through the copper area on the same layer. However, these clearances will not be displayed, but will be created in the Gerber files.

To create a new copper area, select **Copper Area** from the **Add** menu. The following dialog will appear.



Select the net for the copper area from the **Net** drop-down menu, or type the net name directly into the field. Select the layer from the **Layer** list. Choose the **Hatch pattern**. Then click **OK** to start drawing the area. The cursor will change to cross-hairs, and the **Routing Grid** will be in effect. Move to the desired position of the starting corner, and left-click to place it. Now you will be dragging the first side of the polyline. As with the board outline, you can change the style of the side from straight to a clockwise or counterclockwise arc by using the function-key menu. Continue left-clicking to place the rest of the corners, and right click to close the polyline. The copper area will then fill with a diagonal hatch pattern, to make it more visible.

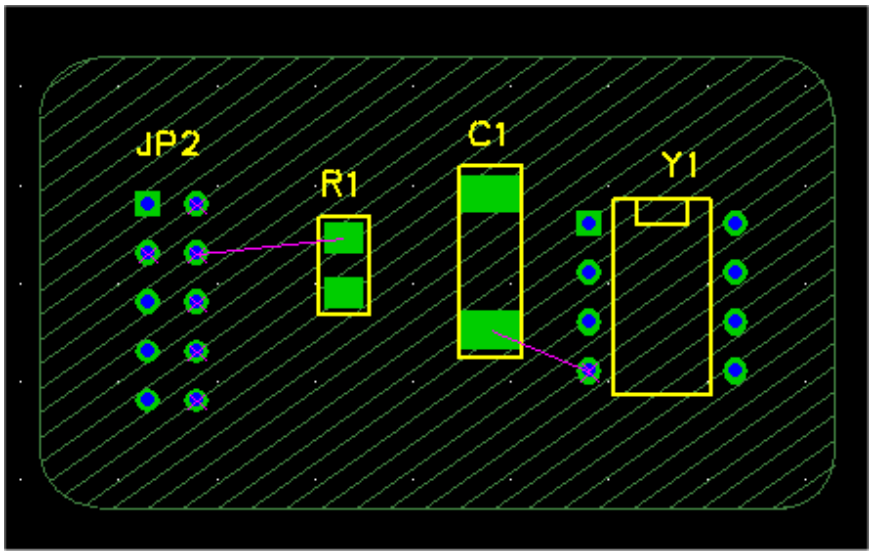


An example of a small copper area on the inner 1 layer. This area was assigned to the GND net, and it uses the full hatch pattern.

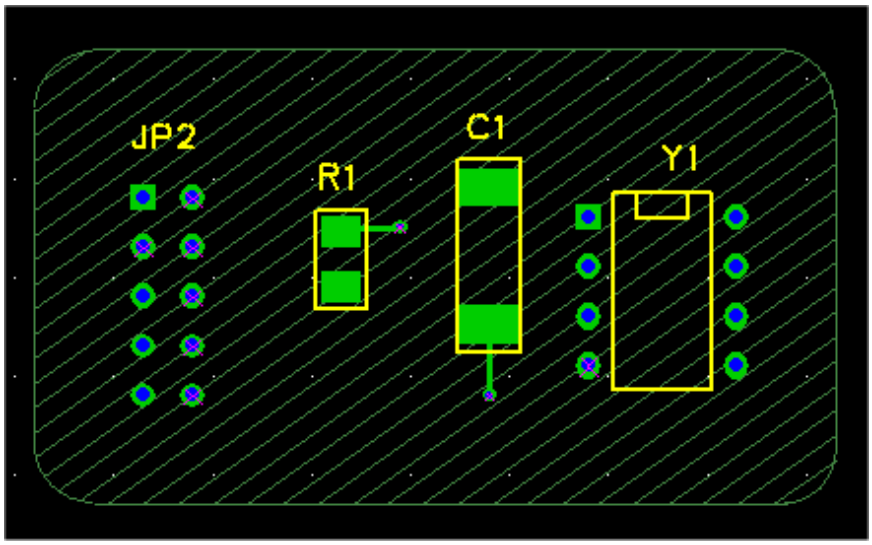
Once created, a copper area can be edited by selecting corners or sides and using the function-key menu. This is exactly the same as editing the board outline, so you can just refer to **Section 5.10: Board Outline** for instructions. You can change the hatch pattern by selecting a side and right-clicking, then selecting **Hatch style** from the context menu.

Connections between parts and copper areas can be made two ways:

- Through-hole pins which pass through the copper area can connect using thermal reliefs. These will be placed automatically for pins on the same net as the copper area. They are indicated by an "X" drawn on the pin in the ratline color.
- SMT pads or through-hole pins which do not pass through the copper area can be connected using stub traces which terminate with vias. A thermal relief will be placed automatically on the via.



To illustrate, 4 parts have been placed, overlying the copper area in our previous example. Only the GND net has been made visible, which is the net assigned to the copper area. Note that pins Y1.4, JP2.2, JP2.3, JP2.4, JP2.6, JP2.8, JP2.10 have all been automatically connected to the copper area by thermal reliefs.



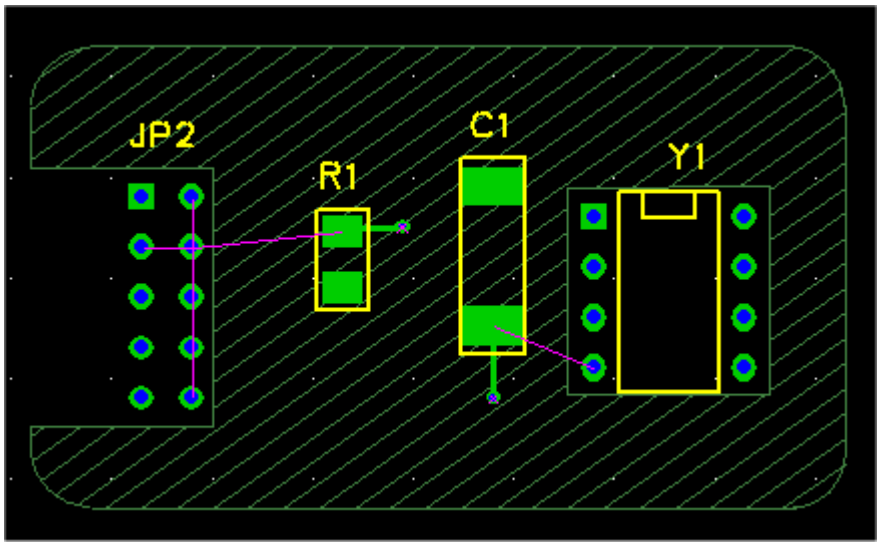
To connect R1.1 and C1.2, we will use stub traces. These have been added below. Note that the vias on the ends of the stubs have been connected with thermal reliefs. Now the entire GND net has been connected using the copper area.

The hatch pattern in the copper area is only intended as a visual aid, to show the extent of the area while allowing other features to show through. When the Gerber files are generated, the area will be solid copper with clearances for any non-connected pads and traces which pass through it.

In the tutorial, you will use copper areas to create power and ground planes which occupy entire layers of the PCB.

5.15.1 Copper Area Cutouts

Sometimes it is useful to "cut out" a portion of a copper area. This may be done to create an opening inside the copper area, or to modify its outline. To create a cutout, select a side or corner of the area and press F6 (Add Cutout). Then draw the cutout just as you would draw a new area. The image below shows our copper area with cutouts around Y1 and JP2.

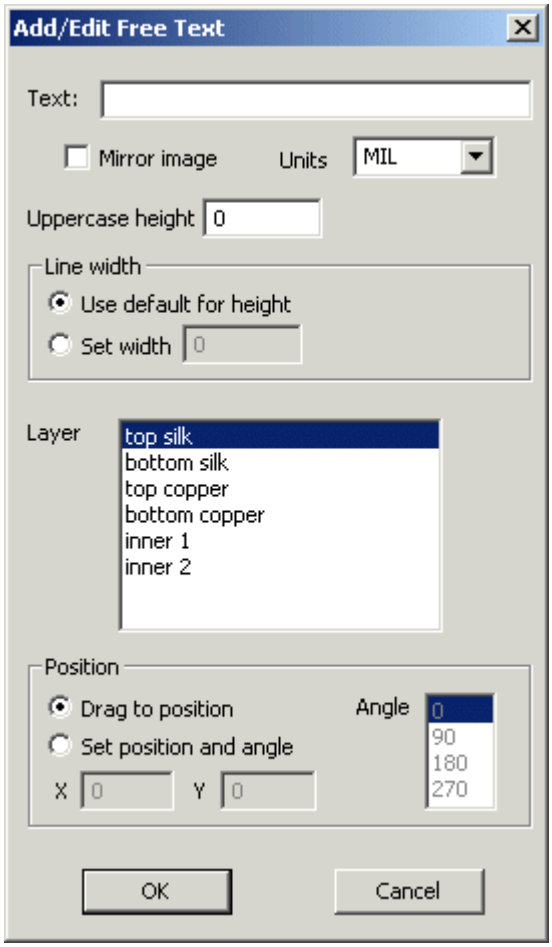


Copper areas can be placed inside cutouts in other copper areas.

5.16 Text

Text strings can be added to the silk-screen or copper layers of a PCB. These are useful for showing information such as labels, revision numbers, copyright notices, etc.

To add a text string to a PCB, select **Text** from the **Add** menu. The following dialog will pop up.



Enter the desired text string into the **Text:** field. Spaces and most special characters are allowed. Use the other fields to set the layer, character height and stroke width. The character height refers to the maximum height of a character, not counting descenders. The default stroke width is 10% of the character height. Checking the **Mirror image** box will draw the text as a mirror image, so that it can be read from the bottom copper or bottom silk-screen layer. If you leave the **Drag to position** radio button selected, you will be dragging the text when you click **OK**. Otherwise, you can set the position and angle of the text string explicitly. The position refers to the lower left corner of the string (not including descenders).

5.17 Solder Mask Cutouts

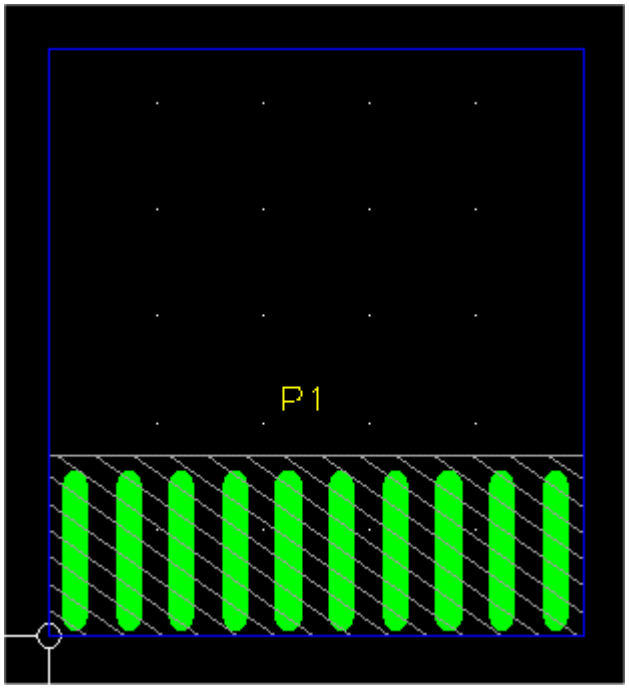
Solder mask is a coating that is applied to a PCB to prevent solder from sticking to it. It is not really necessary for hand-soldered boards, but is essential for boards that are soldered automatically using solder bath or reflow techniques. In the CAM process, the solder mask is considered to be a "layer" of the PCB, and is described by a Gerber file similarly to copper or silkscreen layers. There will usually be top and bottom solder masks, with separate Gerber files for each.

Openings must be made in the solder mask for pads, since they require solder. FreePCB creates these openings automatically in the Gerber files for the solder mask layers, using the clearance specified in the CAM dialog.

Sometimes it may be necessary to create additional openings in the solder mask. For example, board edge connectors should have an opening around them to prevent "dams" of solder mask between the fingers of the connector. Or, you might want to have an area of bare copper for application of shielding or a heatsink. In FreePCB, these openings are referred to as **cutouts**. You can add cutouts to the solder mask layers using **Add > Solder Mask Cutout**. This pops up the following dialog:

You must select the top or bottom solder mask layer, and the hatch pattern. Clicking **OK** allows you to draw the cutout as a closed polyline, similarly to the board outline or a copper area. It will be shown in the color for the "top sm cutout" or "bot sm cutout" layer in the layer list.

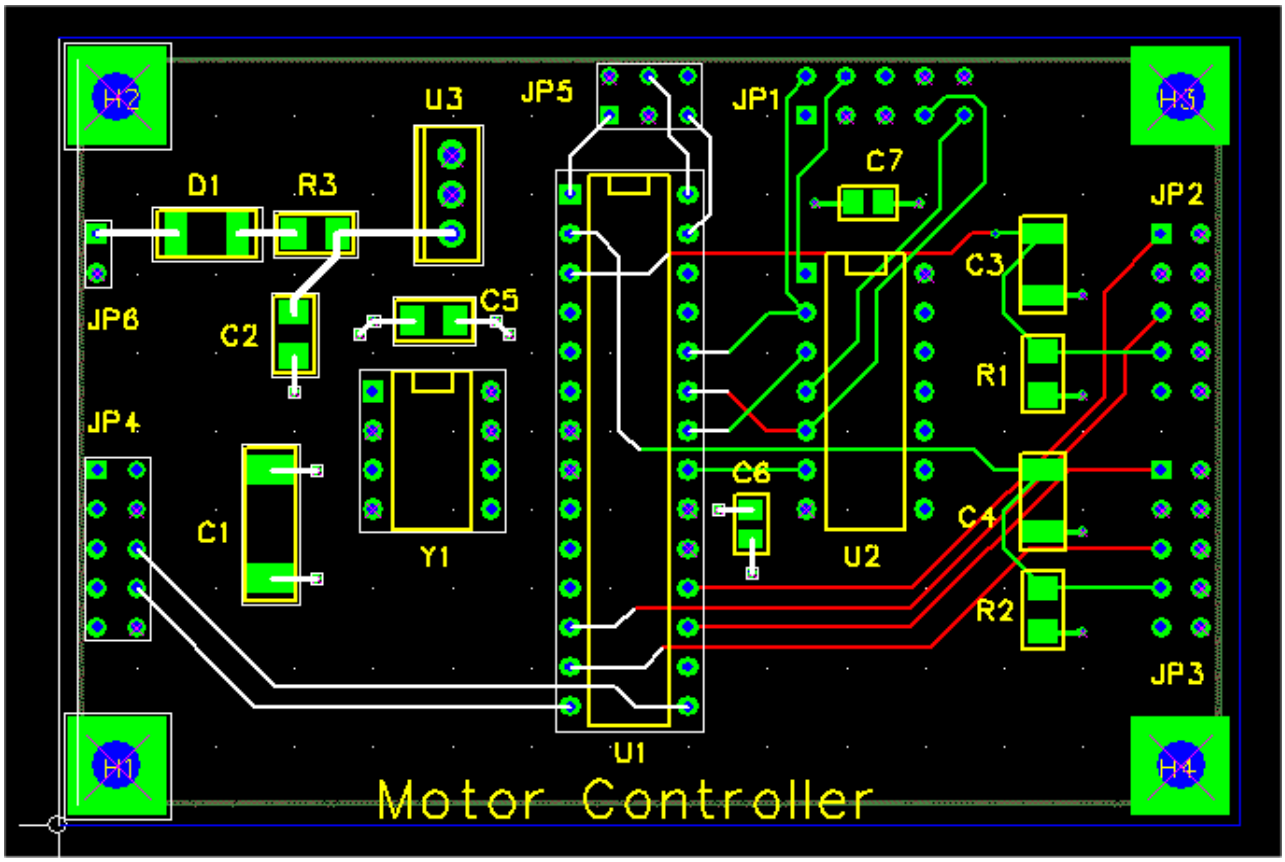
An example of a solder mask cutout around an edge connector is shown below.



For a double-sided edge connector, cutouts should be made on both the top and bottom solder mask layers.

5.18 Groups

Besides manipulating individual PCB elements, you can also perform operations on **groups** of elements. A group can contain any of the elements of the PCB. To select a group of elements, click-drag a rectangle to enclose them. They will be highlighted, as shown in the screenshot below:

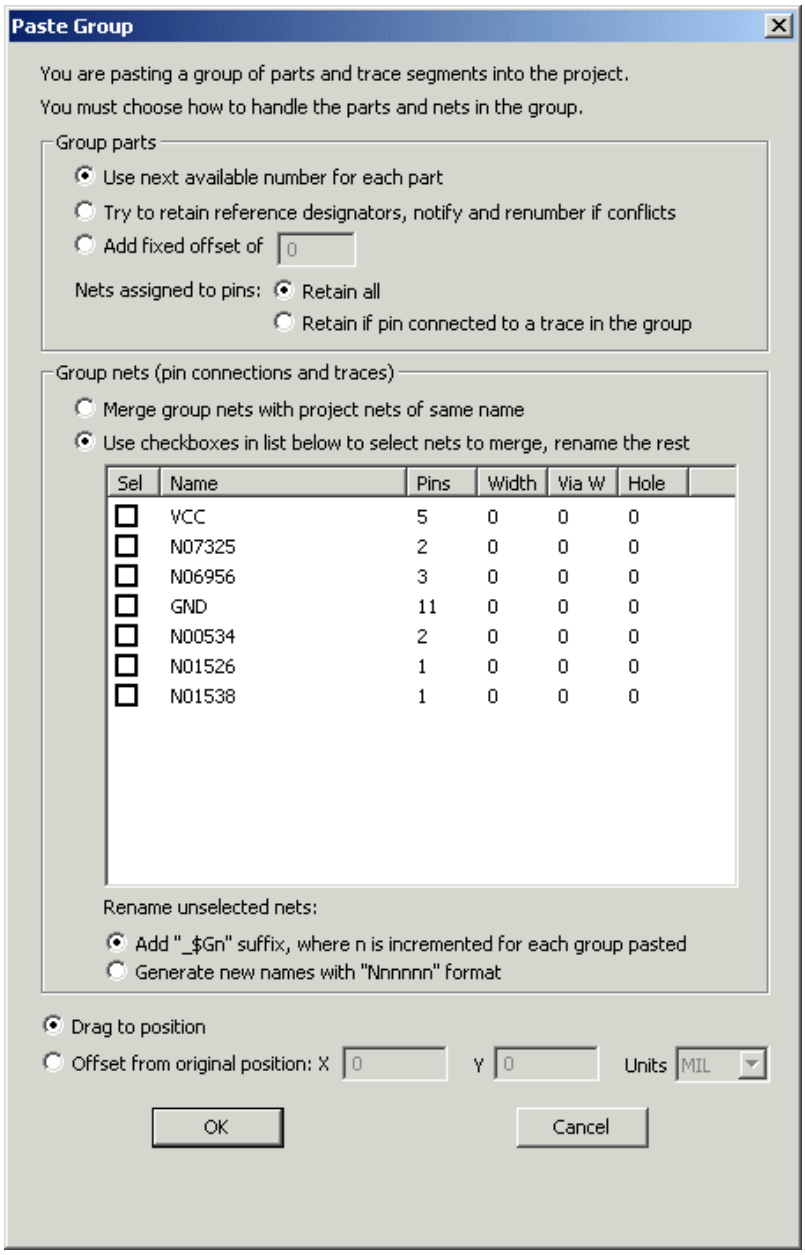


With a group selected, individual elements can be added to it or removed from it by clicking on them with the "ctrl" key pressed. The selection mask can be useful when doing this.

The following operations can be performed on groups:

- **Move** - To move a group, press F4 ("Move Group"). The group elements will disappear from the project window, and you will be dragging an outline representation of the group. Left-click to place it in a new position.
- **Delete** - Press the "Delete" key or F8 ("Delete Group") to remove a group from the project. Note that some elements of the group may not be removed. In general, polygon elements such as the board outline, copper areas, etc. will only be removed if the entire polygon is included in the group. Trace segments will be unrouted rather than removed.
- **Copy** - To copy a group to the clipboard, use ctrl-C or select **Edit > Copy**. Not all elements may be copied. Polygon elements will only be copied if the entire polygon is included in the group. Traces will only be copied if the part(s) containing the pins that they are connected to are also copied.
- **Cut** - For this function, use ctrl-X or select **Edit > Cut**. This is just a combination of Copy and Delete.

- **Paste** - After a group has been copied to the clipboard, it can be pasted into the project by pressing ctrl-V or selecting **Edit > Paste**. Pasting is a complex operation, because the elements in the group may have to be renamed before they are added to the project, to avoid conflicts with existing names. The **Paste Group** dialog will pop up, as shown below. For parts in the group, you have the option of renaming them or trying to reuse the original reference designators. For nets, you have the choice of renaming them or merging them with existing nets in the project with the same name.



- **Save to file** - Use **Edit > Save to File...** to save the group as a project file. This file can be opened for editing with FreePCB, or pasted into another project (see below).
- **Paste from file** - Use **Edit > Paste from File...** to paste a project file into the current project as a group. The **Paste Group** dialog will appear, as shown above.

With a little creativity, you can find lots of uses for group operations in your projects. For example, to unroute all of the traces in a project, set the selection mask to select traces, select all of the traces in the project, and press "Delete".

5.19 Design Rule Checking

In an ideal world, PCBs would always be fabricated exactly as specified in the Gerber and drill files. In real life, of course, this isn't true. Due to the inexact nature of the etching process, copper features may turn out slightly larger or smaller than specified. Layers on multilayer boards may not line up exactly, and the size and position of holes may vary due to manufacturing tolerances. In practical terms, this means that there are lower limits on trace widths, pad sizes, and clearances between copper features and/or holes. These limits are called **Design Rules**, and will vary depending on the manufacturing process.

The PCB manufacturer should provide design rules for each process that it offers. For example, the design rules posted on the Internet for the Advanced Circuits low-cost process are shown below:

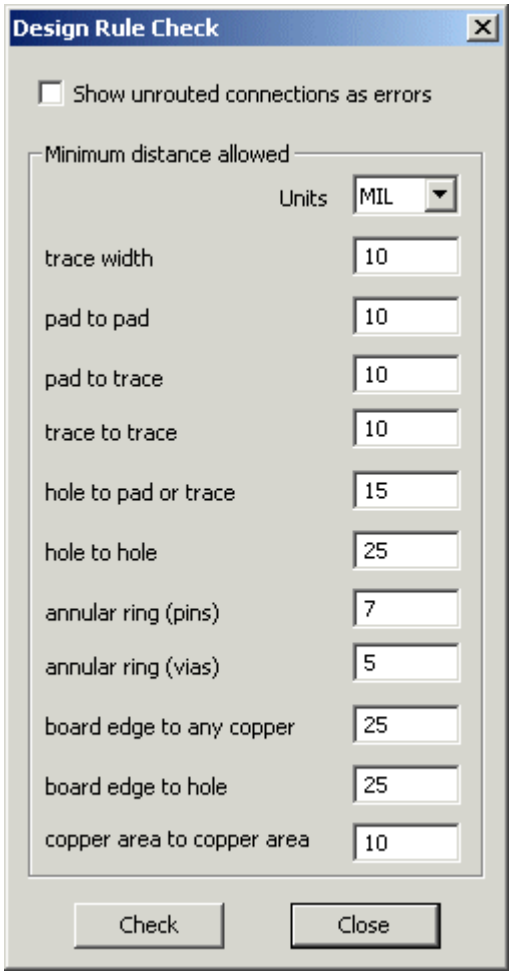
Minimum trace width	0.008 inch
Minimum clearance between copper features	0.008 inch
Minimum distance from copper to edge of PCB	0.014 inch
Minimum annular ring width (pins)	0.007 inch
Minimum annular ring width (vias)	0.005 inch
Minimum silkscreen line width	0.008 inch

The design rules for the PCB Express low-cost process are:

Minimum trace width	0.007 inch
Minimum clearance between copper features	0.007 inch
Minimum distance from copper to edge of PCB	0.020 inch
Minimum space between pads (using solder mask clearance of 0.004 inch)	0.013 inch
Minimum annular ring width (pads and vias)	0.0085 inch
Minimum clearance from inner layer holes to copper	0.0175 inch
Minimum silkscreen line width	0.007 inch

If you are making the board yourself, you will have to come up with your own design rules.

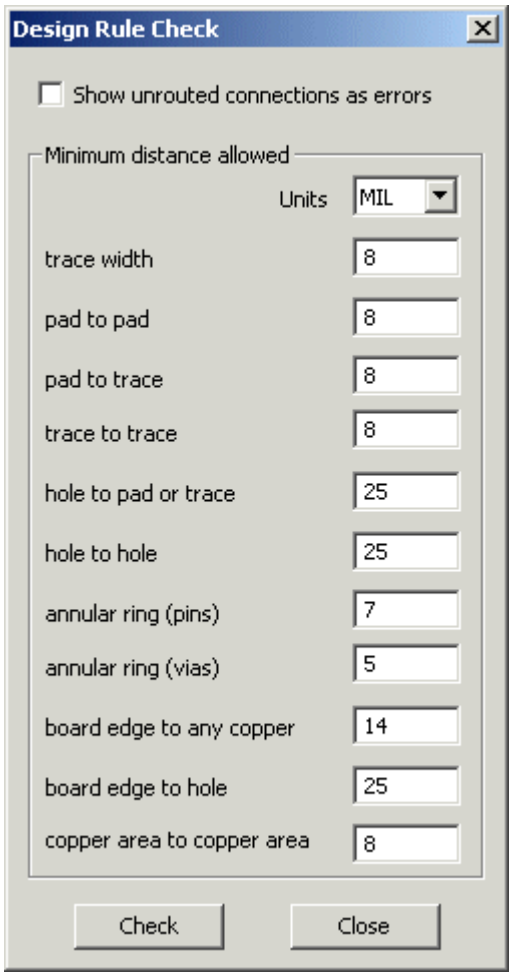
FreePCB has a **Design Rule Checker** that checks your project for compliance with a set of design rules. Selecting **Tools > Design Rule Check** pops up the following dialog:



The **Show unrouted connections as errors** checkbox allows you to treat connection errors as DRC errors. The other fields in the dialog are explained below:

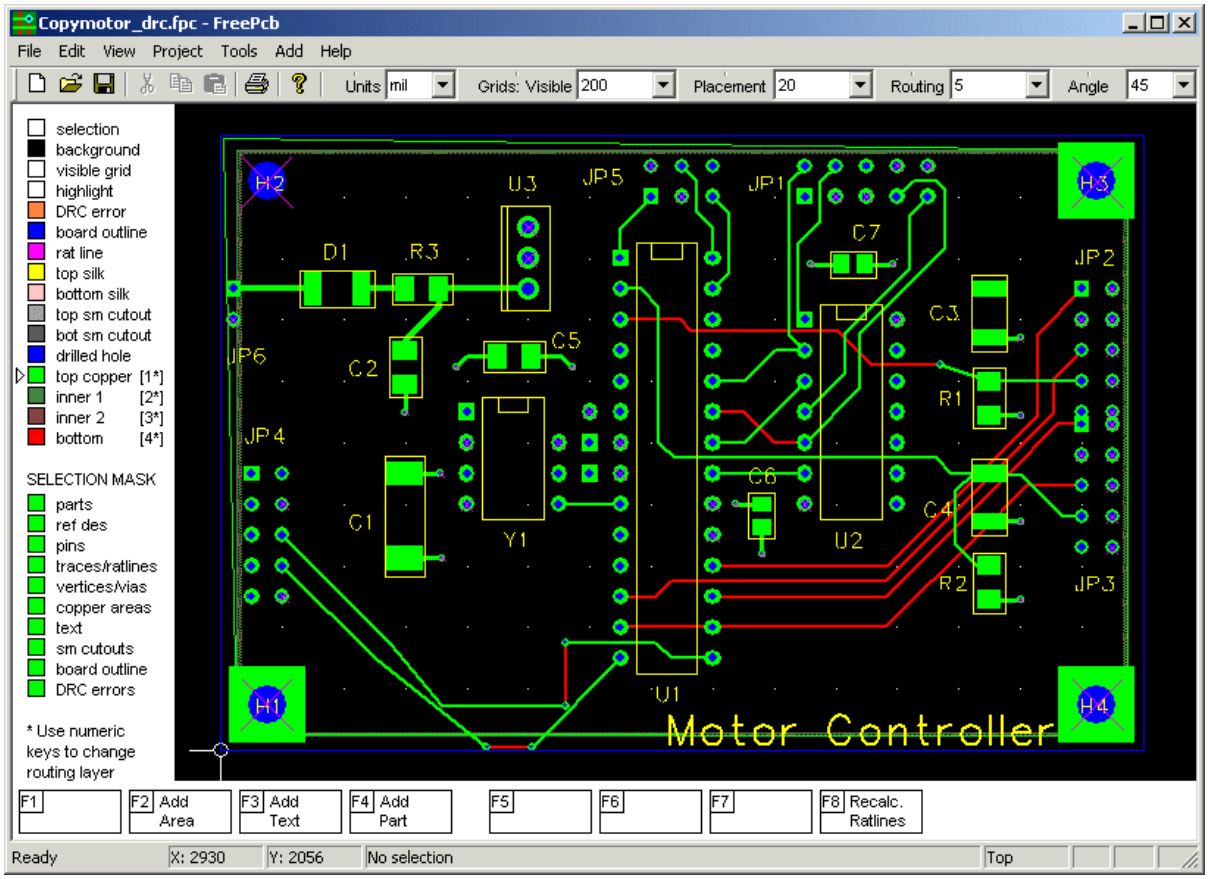
trace width	The minimum trace width allowed
pad to pad	The minimum distance from the edge of one pad to another on a different net
pad to trace	The minimum distance from the edge of a pad to a trace on a different net
trace to trace	The minimum distance from the edge of a trace to a trace on a different net
hole to pad or trace	The minimum distance from the edge of a hole to a pad or trace on a different net
hole to hole	The minimum distance from the edge of a hole to the edge of another hole
annular ring (pins)	The minimum width of copper surrounding a hole for a pin
annular ring (vias)	The minimum width of copper surrounding a hole for a via
board edge to any copper	The minimum clearance between any copper feature and the edge of the board
board edge to hole	The minimum clearance between the edge of a hole and the edge of the board
copper area to copper area	The minimum clearance between copper areas

You should set these fields for the design rules that you are using. If the PCB manufacturer doesn't give the value for a particular field, you will have to guess at a reasonable value. For example, the settings that I would use for Advanced Circuits are shown below:

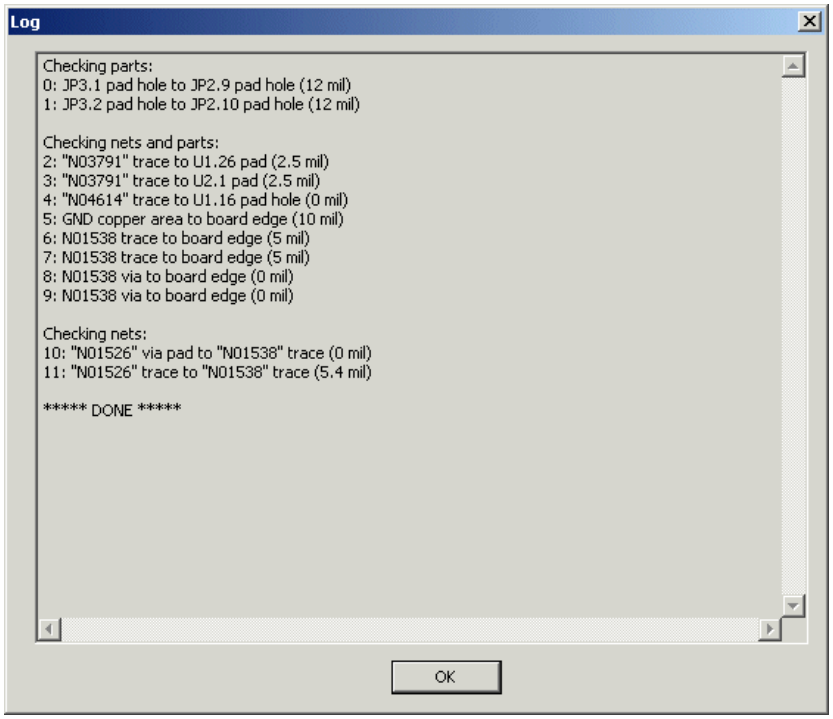


The settings for **trace width**, **pad to pad**, **pad to trace**, **trace to trace**, **annular ring (pins)**, **annular ring (vias)**, **board edge to any copper**, **hole to pad or trace** and **copper area to copper area** were taken from the design rules posted on the Internet. Since there were no rules provided for **hole to pad or trace**, **hole to hole** or **board edge to hole**, I used 25 mils which seems like a reasonable value. If necessary, I could confirm this with the PCB manufacturer.

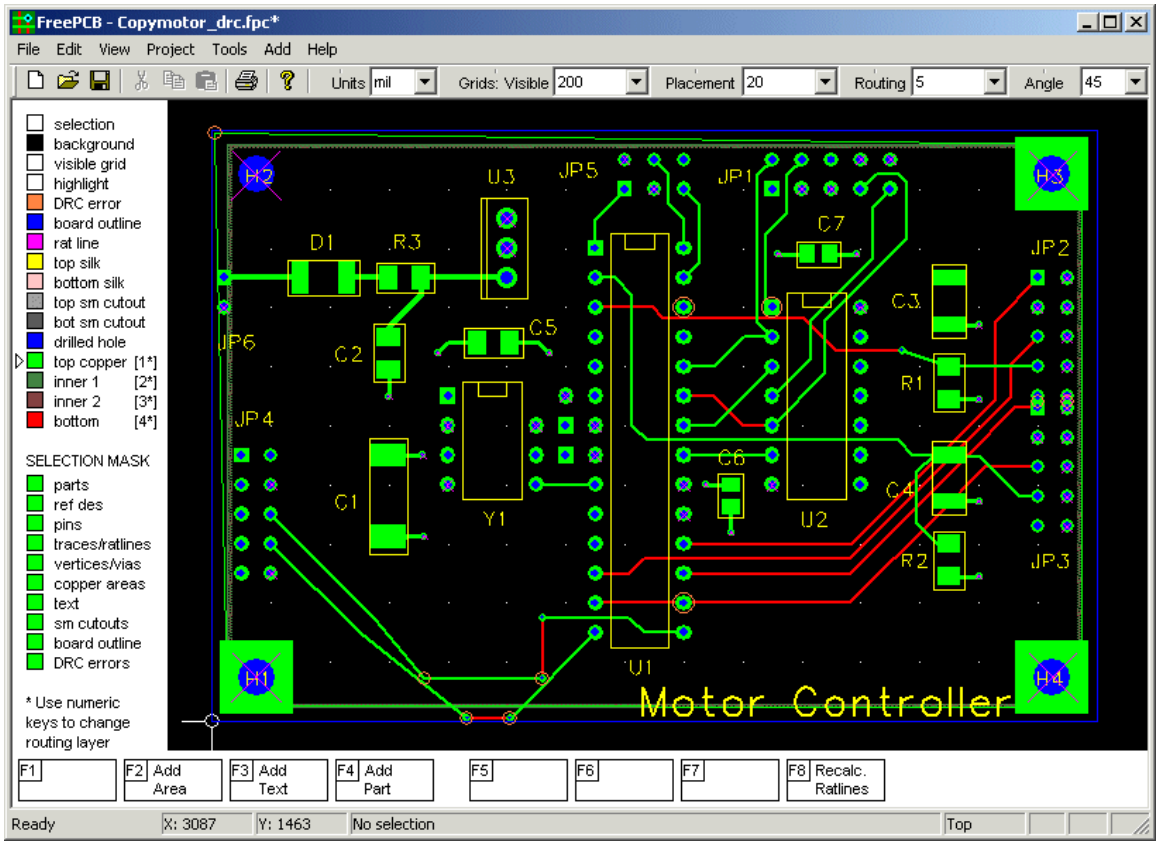
As an example, let's check the project shown below, which contains multiple intentional design rule violations.



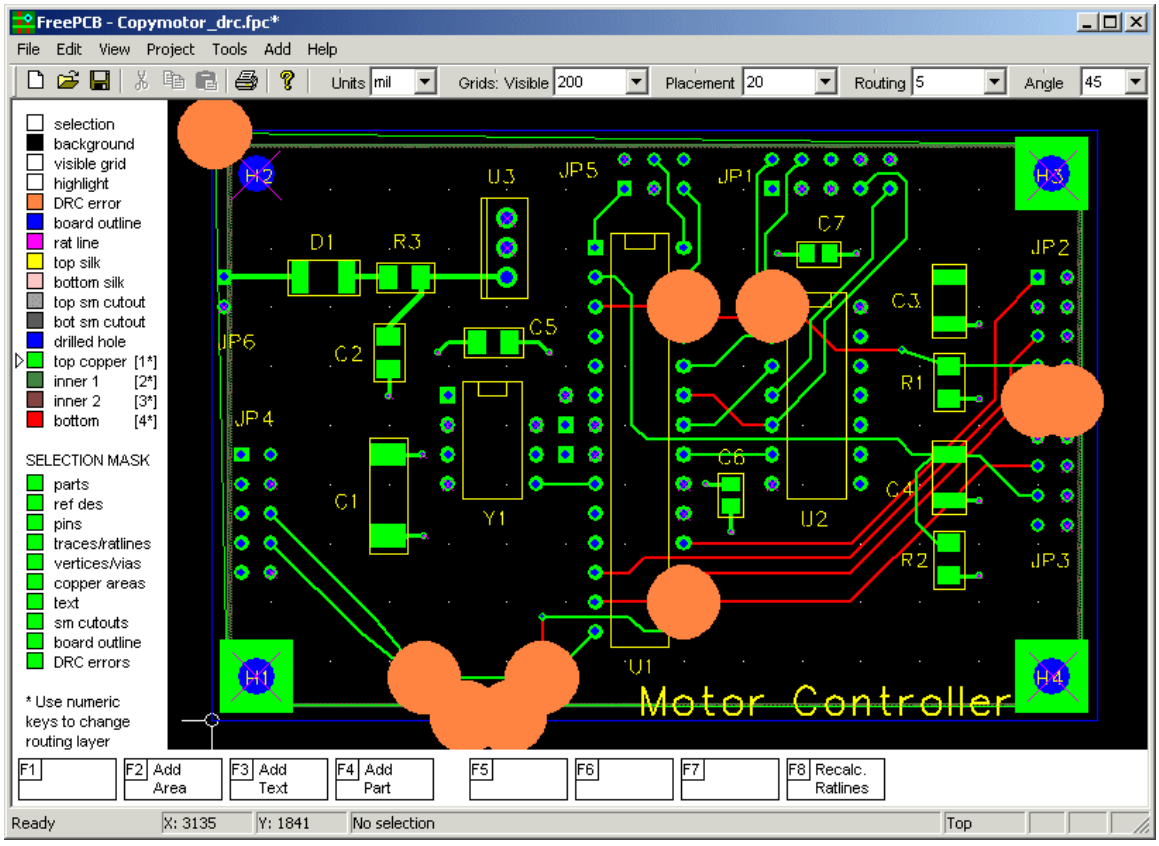
To check the design, select **Tools > Design Rule Check** and set the design rules as described above. Clicking **Check** starts the checker and brings up the following dialog that lists all of the violations.



When you close the **Design Rule Check** dialog, each violation will be indicated in the layout window by a small ring in the color for DRC errors, as shown below.

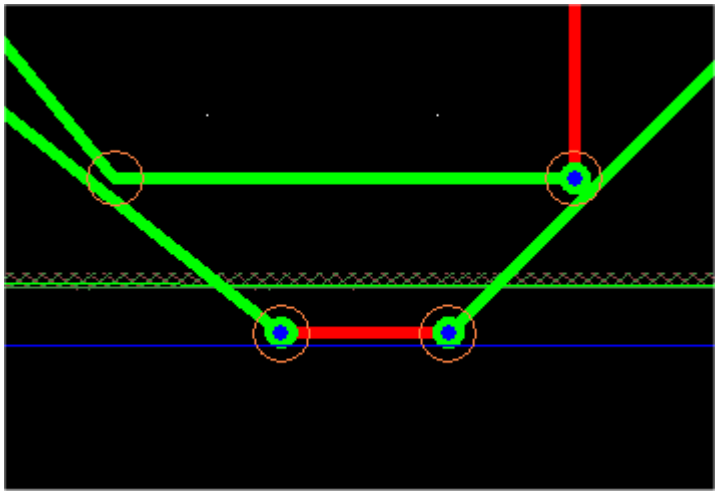


These small rings may be hard to see in a dense design.

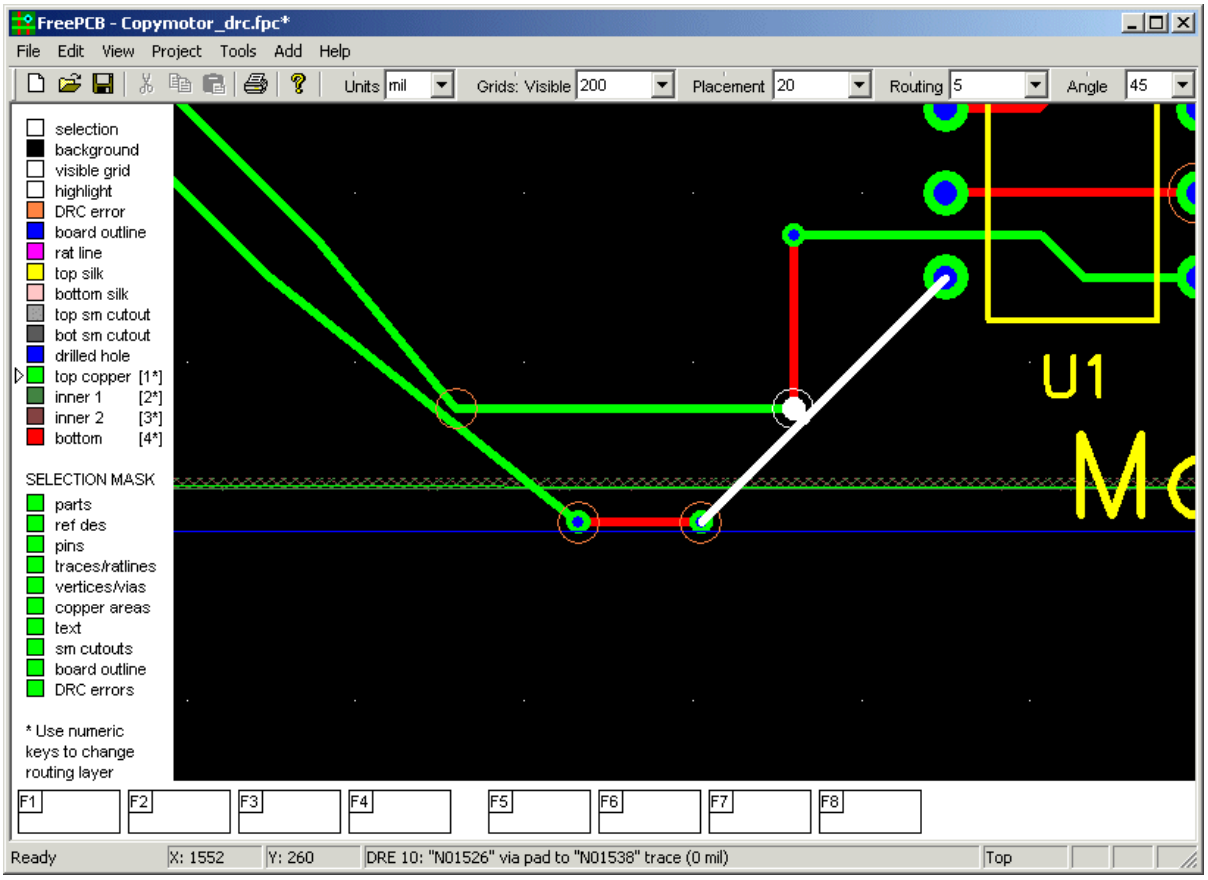


If you hold down the "d" key, each ring is converted to a much larger solid circle, as shown.

Now you can zoom in on individual errors and fix them. The group of errors near the bottom of the board is shown below.

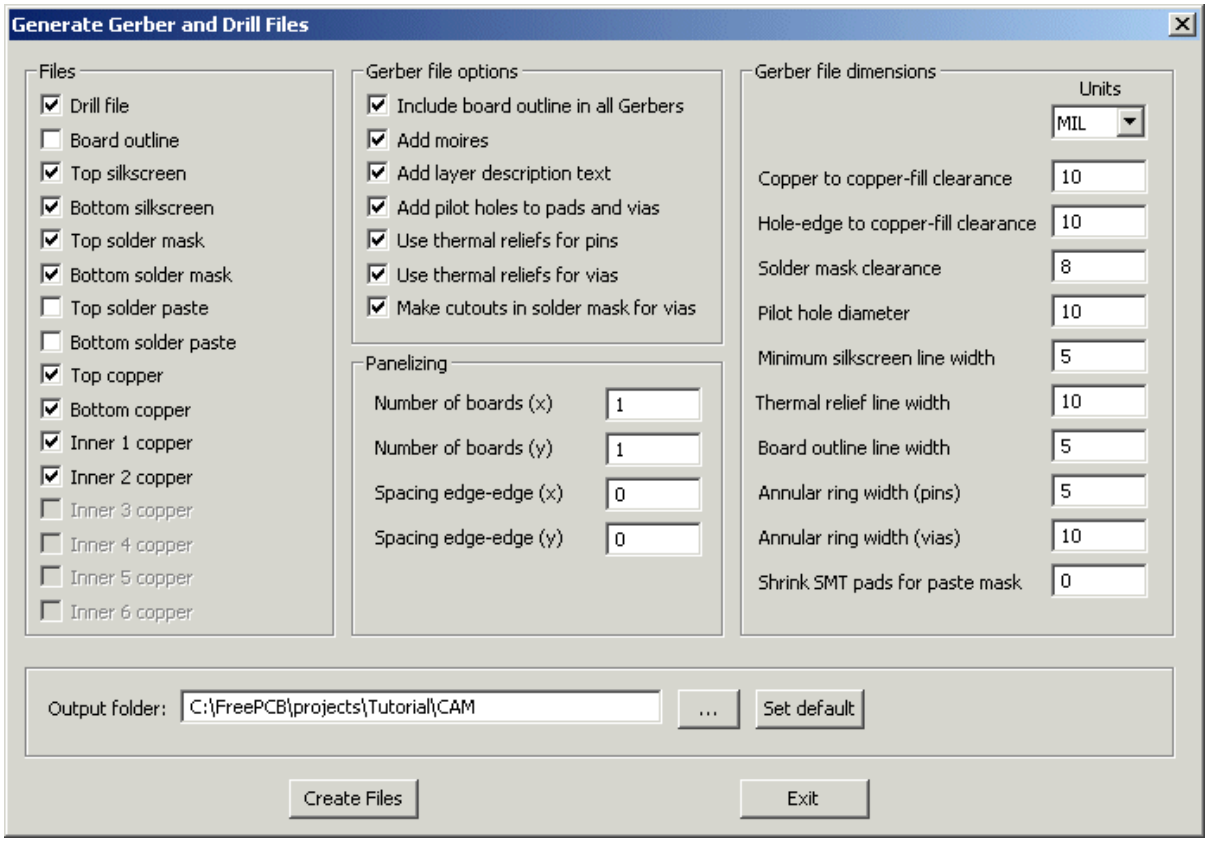


You can select one of the errors by clicking on the ring. It will be highlighted, along with the elements that caused the error, and the status bar will show a description of the error. In the screenshot below, I clicked on the right-most error.



In this case, the error is a pad-to-trace distance violation, where the distance is 0 mils instead of at least 8 mils. We can fix the error by moving the pad or trace. The error ring will not automatically disappear when you fix the error, but you can delete it by selecting it and pressing the "delete" key. You can delete all of the errors by selecting **Tools > Clear DRC Errors**.

Important note: Compliance with some design rules also depends on your Gerber file settings. For example, the settings shown below could create violations of the minimum silkscreen line width (which should be at least 7 mils for Advanced Circuits) and the annular ring width for pins (which should be at least 7 mils), so they should be changed to match the design rules.



Another note: If you found this section confusing, some PCB manufacturers provide information about design rules on their websites.

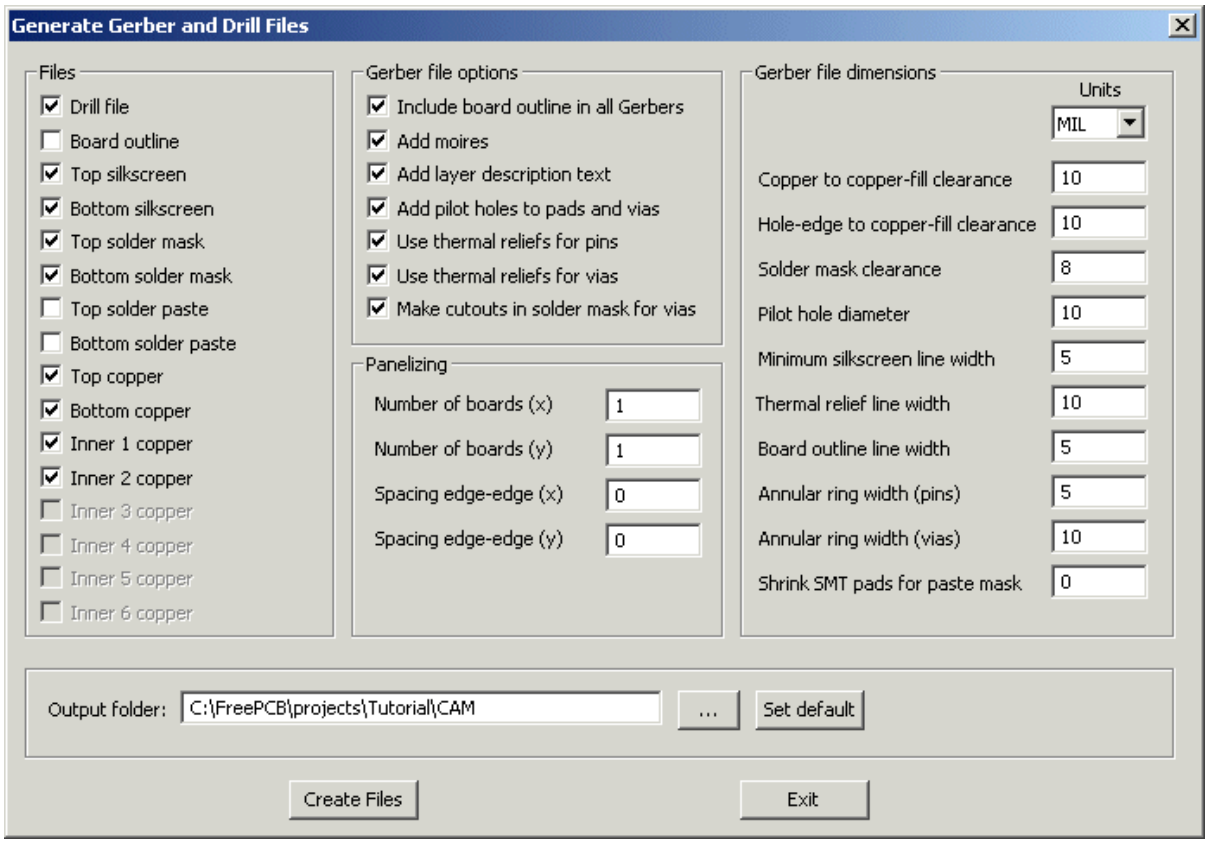
5.20 Exporting Drill and Gerber Files

5.20.1 Creating Files

The final step in creating a PCB layout is to export the files that the board house will use to produce your boards. These files are described below.

- Gerber files - These files are used to create the photoplots that will be used to make the copper layers, silk-screens and masks for the PCB. There is one file for each layer or mask. The format is extended Gerber RS274X.
- Drill file - This will be used by the machine that drills the holes in the PCB. It uses Excellon format, which is the industry standard. Basically, it starts with a list of drill sizes (in inches), and then gives the coordinates of the center of each hole. **There is an important issue regarding drill sizes, which is discussed in Section 5.19.3: Drill Sizes. Please read this section before you send your files to the PCB house for fabrication.**

To export drill and Gerber files, select **Generate CAM files...** from the **File** menu. The following dialog will pop up.



In the **Files** section, select the files that you want to generate by checking or unchecking the boxes next to each one.

The **Gerber file options** section allows you to select or deselect the following options:

- **Include board outline** - check this box to include the board outline in all Gerbers.
- **Add moires** - add moire symbols (sometimes called "targets") for layer registration.
- **Add layer description text** - add a text string to each Gerber file indicating the layer that it represents.
- **Add pilot holes to pads and vias** - add a pilot hole to through-hole pads and vias on the top and bottom layers, to aid in drilling.
- **Use thermal reliefs for pins** - when connecting through-hole pins to internal copper layers, use a thermal relief.
- **Use thermal reliefs for vias** - when connecting stub traces to internal copper layers with vias, use a thermal relief.
- **Make cutouts in solder masks for vias** - create openings in the solder masks around via pads.

The **Gerber file dimensions** section allows you to set values for the following:

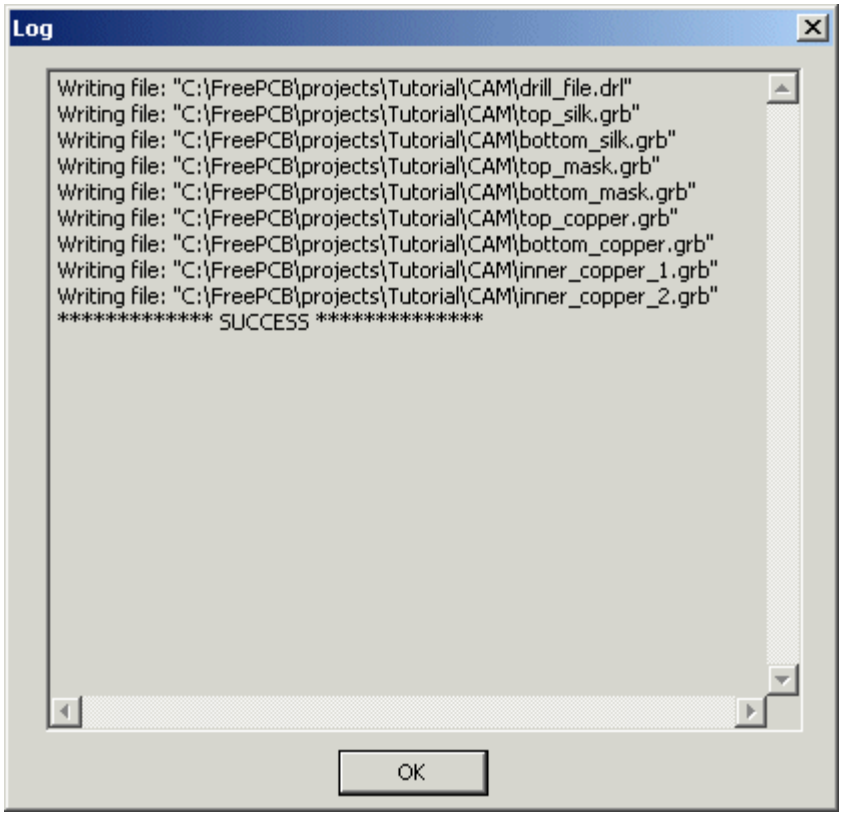
- **Copper to copper-fill clearance** - is the clearance that FreePCB will create around traces or vias that pass through copper layers.
- **Hole-edge to copper-fill clearance** - is the clearance that will be created around drill holes. Please refer to **Section 5.194: Drill Clearances** for an important note about these clearances.
- **Solder mask clearance** - the space that FreePCB will provide around pads in the solder masks.
- **Pilot hole diameter** - the diameter of the pilot holes, if you selected them.
- **Minimum silkscreen stroke width** - the minimum stroke width that FreePCB will use for the silkscreen layers, which is usually recommended by the board house.
- **Thermal relief line width** - the width of the lines that will be used to connect pads or vias to copper fill areas using thermal reliefs.
- **Board outline line width** - the width of the board outline, if used.
- **Annular ring width (pins)** and **Annular ring width (vias)** - the width of annular copper rings placed around inner layer pin and via holes with thermal relief connections to copper areas.

The Panelizing section allows you to panelize your project by creating multiple copies of the PCB in the Gerber files:

- **Number of boards (x)/(y)** - the number of copies in each axis.
- **Spacing edge-edge (x)/(y)** - the clearance between the edges of the copies in each axis.

When you click **Create Files**, the files will be written to the **Output folder**. By default, this is a subfolder of the project folder called **CAM**. It will be created if it doesn't already exist. You can change it if you want, or reset it to the default path with the **Set default** button.

Assuming that all goes well, you will see the following **Log** dialog, with "*** SUCCESS ***" as the final line. If errors occur, there will be messages in the log. If an error occurs while creating a file, FreePCB will usually abort that file and go on to the next file. Each file is named according to its content, such as **top_copper.grb** or **drill_file.drl**.



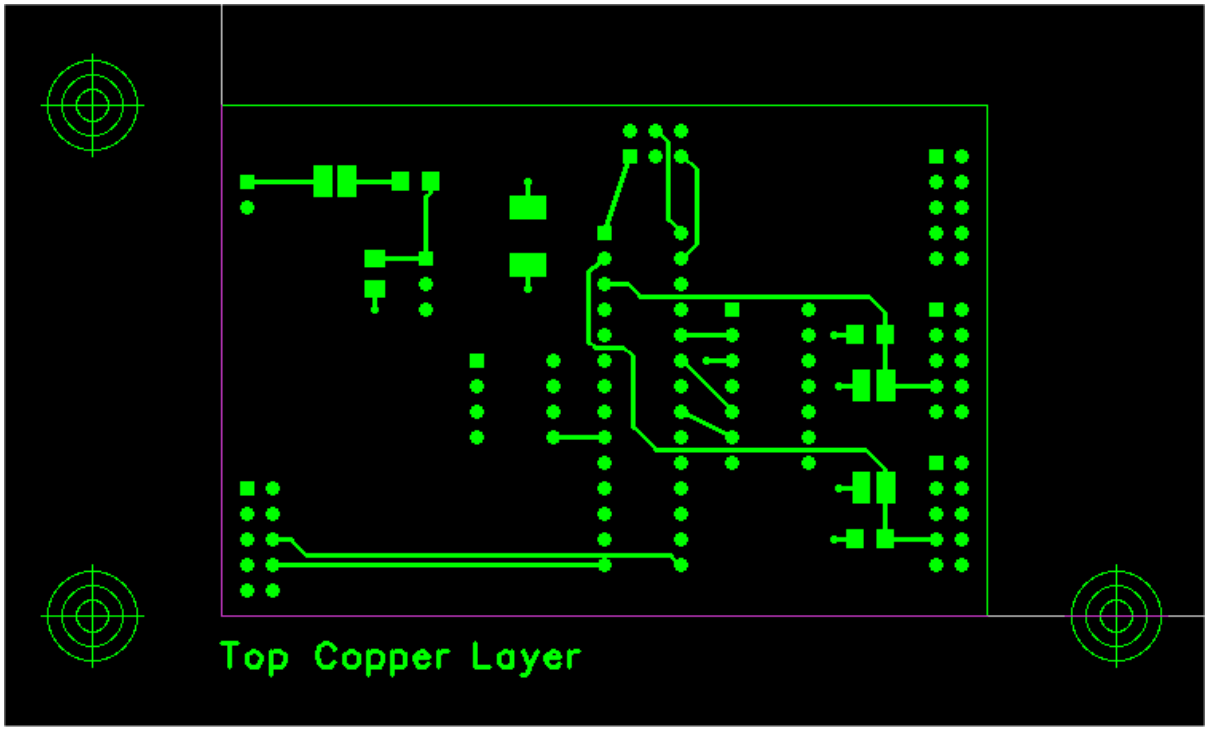
Click **OK** to dismiss the log.

When you are finished creating files, click **Exit** in the **Generate Gerber and Drill Files** dialog.

5.20.2 Viewing and Printing Files

There are a number of free programs that let you view Gerber and drill files. Before sending your files to a board house, I would STRONGLY suggest that you check them with one of these programs. I have been using **ViewMate** by **PentaLogix** (formerly Lavenir). You can download this at www.pentalogix.com/Download/download.html. ViewMate lets you view both Gerber and drill files, and also lets you make check-plots by printing them to a Windows printer. Since the printing functions in FreePCB are not yet implemented, this is currently the only way to print your designs.

An example of a FreePCB Gerber file as displayed by ViewMate is shown below. This Gerber file includes the board outline, moirés and layer description text ("Top Copper Layer").



5.20.3 Drill Sizes

The drill file specifies the size and position of each hole in the PCB. An issue that arises with these files is whether the hole size refers to the actual drill bit size (in which case the final diameter of the hole will be a few mils smaller after plating) or the finish size (or plated size) which is the final hole size after plating. To me, it makes more sense to specify the finish size and let the board house figure out which drill bit to use, since they know how thick their plating will be. FreePCB footprints use finish sizes, so these are the sizes which appear in the drill file.

You MUST check with your board house, however, as some of them expect actual bit sizes, which must be larger than the finish size. In this case, you may have to edit the drill file and increase the drill sizes appropriately. Your board house should be able to help with this, as long as they know what you want.

A sample FreePCB drill file is shown in the listing below.

```
;Holesize 1 = 24.0 PLATED MILS
;Holesize 2 = 18.0 PLATED MILS
;Holesize 3 = 14.0 PLATED MILS
M48
INCH
T01C0.024
T02C0.018
T03C0.014
%
G05
G90
T01
X028000Y018000
X029000Y018000
X028000Y017000
X023000Y010000
X023000Y011000
X023000Y012000
T02
X012000Y017000
X006000Y012000
X012000Y012800
X024000Y011000
X024200Y009000
X024200Y005000
X024000Y003000
T03
X019000Y010000
M30
```

Lines that begin with "**;**" are comments. Each different hole size is described in a comment, such as

```
;Holesize 1 = 24.0 PLATED MILS
```

which indicates a hole diameter of 24 mils, after plating. The line

```
T01C0.024
```

is the actual instruction to the machine that establishes the drill size, in inches.

If you need to change a drill size to allow for plating, you should change both the comment and the instruction. For example, to use a 28 mil bit instead of a 24 mil bit, change the comment to

```
;Holesize 1 = 28.0 UNPLATED MILS
```

and the instruction to

```
T01C0.028
```

Some board houses will ask for a separate text file which lists the drill sizes. You can tell them to look at the comments at the beginning of the drill file, or you can copy these into a separate file.

5.20.4 Drill Clearances

Board houses will often recommend a "drill clearance". This is the minimum recommended size of clearance holes in inner layer copper that should be created around drill holes, to allow safe passage of the hole without shorting. Typically, they will specify something like "drill diameter plus 0.025 inches". This refers to the **DIAMETER** of the clearance hole. FreePCB uses the **Hole-edge to copper-fill clearance** value to create these clearances. However, because FreePCB treats this clearance as the distance between the edge of the hole and the edge of the copper, it will be the difference in **RADIUS** between the holes, which is half of the difference in diameter. Therefore, to create a drill clearance of "drill diameter plus 0.025 inches", you should use a **Hole-edge to copper- fill clearance** of one half of 0.025 inches, or 12.5 mils. This would be rounded up to 13 mils.

6. Footprints and Libraries

6.1 Footprint Libraries

Footprints are stored in **library files**, which have the extension **.fpl**. Each file contains multiple footprints, usually for packages of the same type. FreePCB comes with a set of library files, most of which were obtained by converting libraries from **Ivex Design International Inc.,** a maker of commercial ECAD software which went out of business last year. These files are organized into three groups:

- Core libraries - These are the most commonly used libraries. They will be located in the **C:FreePCB\lib** folder if you did the standard installation.
- Extra libraries - Libraries with less commonly used parts. They will be located in **C:FreePCB\lib_extra**.
- Contributed libraries - Libraries contributed by users, located in **C:FreePCB\lib_contrib**.

When you create a project in FreePCB, you will select a library folder to use for the project. Normally, this will be the core library folder. If you need to use any of the other libraries, you should copy them into the core library folder. If you want, you can copy ALL of the files into the core library folder so they will all be in one place.

You can create your own custom footprints using the **Footprint Wizard** or the **Footprint Editor**, which are covered in **Section 6.2: Footprint Wizard** and **Section 6.3: Footprint Editor**. If you create your own footprints, I would recommend saving them in separate library files instead of the standard files that come with FreePCB. That way, if there are updates to the standard libraries, you can replace them without losing your custom footprints.

Each library file is documented in a PDF file, which has the same name as the library file except for the extension **.pdf** instead of **.fpl**. If you are searching for a particular footprint, the PDF files are the best place to look. You can also browse for footprints in the **Add > Part** dialog, but the PDF files contain more information such as dimensions and pad sizes.

If you create or modify your own libraries, you can make PDF files for them using the **Tools > Make PDF from Library File...** menu item in the Footprint Editor.

The library files that are currently supplied with FreePCB are listed in the sections below. Note that files containing footprints with through-hole pads start with "th_" while those with SMT pads start with "sm_".

6.1.1 Core libraries

Filename (.fpl)	Source(s)	Type	Description
th_transistor	Ivex, JEDEC	through-hole	JEDEC TO-series transistor and IC packages
th_diode	Ivex, JEDEC	through-hole	JEDEC DO-series diode packages
th_capacitor	Ivex	through-hole	Capacitors, polarized and non-polarized
th_resistor	Ivex	through-hole	Resistors, including potentiometers
th_connector	Ivex	through-hole	Connectors
th_header	Ivex	through-hole	Headers
th_sip	Ivex	through-hole	SIP (single-in-line) packages (100 mil spacing)
th_dip	Ivex	through-hole	DIP (dual-in-line) packages (100 mil spacing)
sm_resistor	Ivex, IPC	surface-mount	Chip resistors
sm_capacitor	Ivex, IPC	surface-mount	Chip and tantalum capacitors
sm_soic	Ivex, IPC	surface-mount	SOIC (small outline IC) packages
sm_sop	Ivex, IPC	surface-mount	SOP and TSOP (small outline package) packages
sm_soj	Ivex, IPC	surface-mount	SOJ (small outline J-lead) packages
sm_lcc	Ivex, IPC	surface-mount	LCC (leadless chip carrier) packages
sm_plcc	Ivex, IPC	surface-mount	PLCC (plastic leaded chip carrier) packages
sm_sot	Ivex, Siemens	surface-mount	SOT (small outline transistor) packages
led	Ivex	mixed	Light-emitting diodes
test_point	Ivex	mixed	Test points

6.1.2 Extra libraries

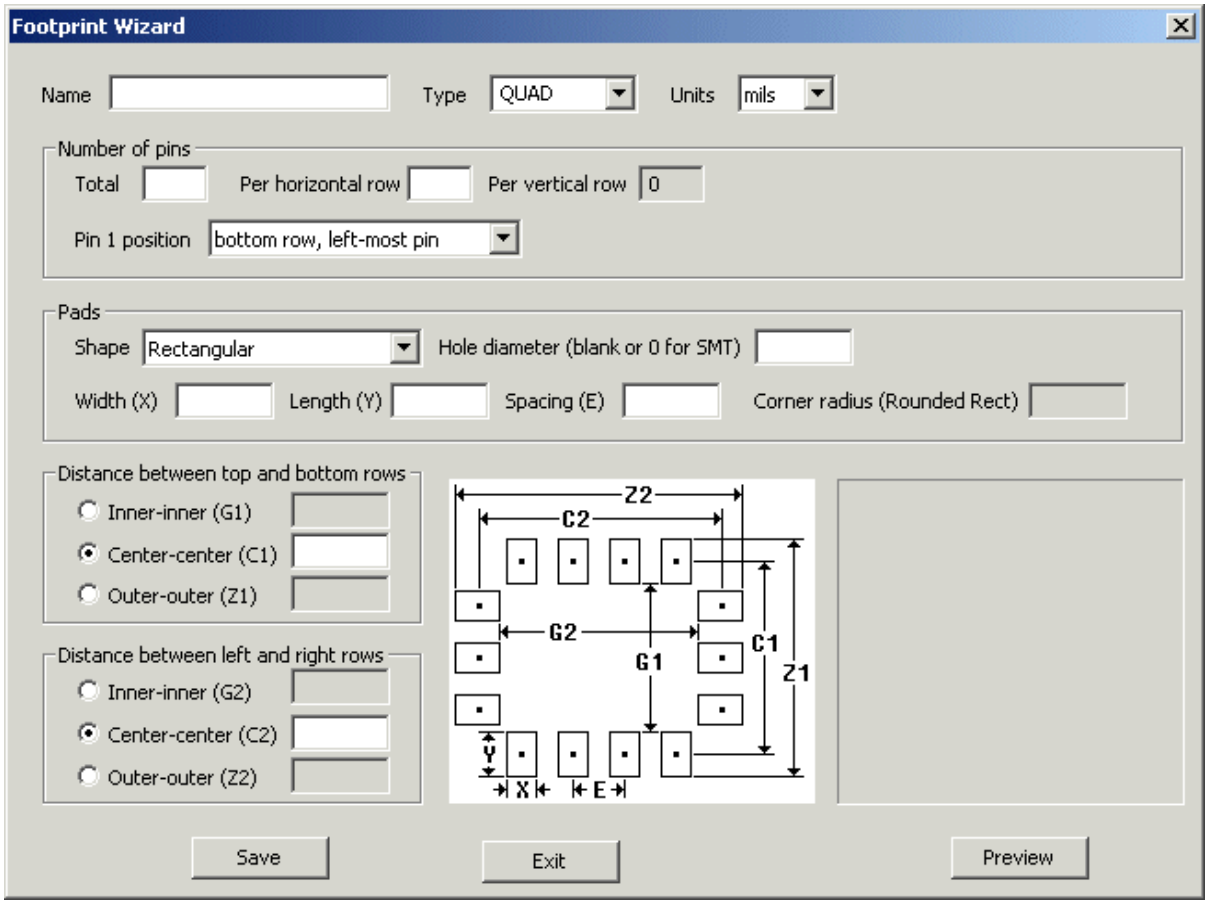
Filename (.fpl)	Source(s)	Type	Description
sm_qfp	Ivex, IPC	surface-mount	PQFP, SQFP and QFP (quad flat-pack) packages (this file is not in the core library folder because it is so large)
device	Ivex	mixed	Various devices
sm_tantalum_cap	Ivex	surface_mount	Tantalum capacitors from AVX
elfa_cap	Ivex	mixed	Capacitors from the ELFA catalog
elfa_chk	Ivex	mixed	Various devices from the ELFA catalog
elfa_pot	Ivex	mixed	Potentiometers from the ELFA catalog
elfa_res	Ivex	mixed	Resistors from the ELFA catalog
flatpack	Ivex, JEDEC	mixed	Quad flatpacks
sm_ipc782	Ivex, IPC	surface-mount	Various devices from IPC not in other libraries
siemens2	Ivex, Siemens	mixed	Siemens LCC packages
siemens3	Ivex, Siemens	mixed	Siemens SO packages
siemens4	Ivex, Siemens	mixed	Siemens QFP packages
transformer	Ivex	mixed	Transformers
tx_inst	Ivex	mixed	Various Burr-Brown and Texas Instruments packages

6.1.3 Contributed libraries

Filename (.fpl)	Source(s)	Type	Description
flintstone	Anonymous	mixed	Various relays, pots, transformers and connectors

6.2 Footprint Wizard

There will be times when you need a footprint that is not in one of the libraries that come with FreePCB. The **Footprint Wizard** is a tool for generating footprints for a variety of standard packages. It is invoked with the **Tools > Footprint Wizard** menu item. This pops up the following dialog:

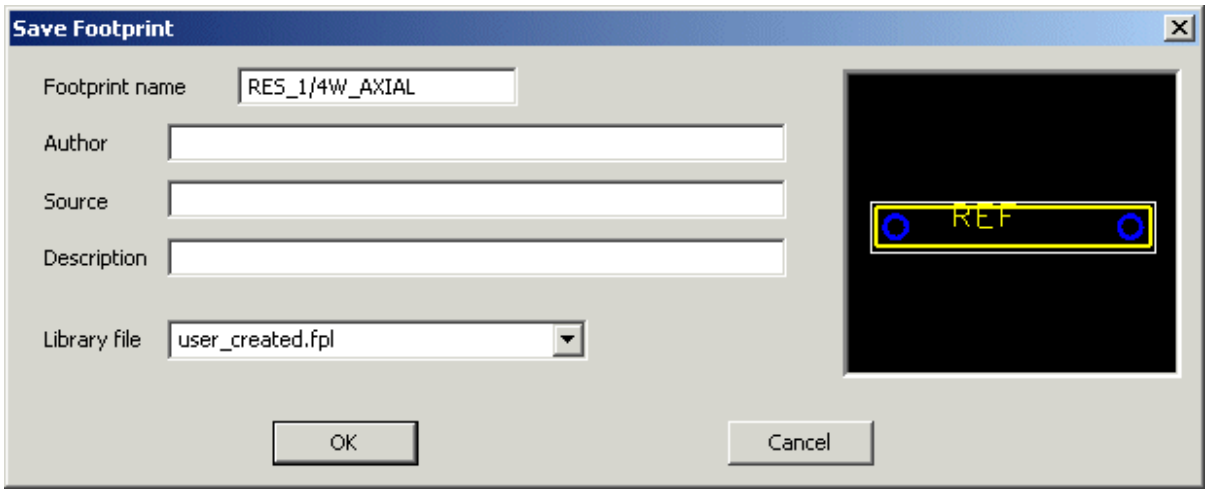


A footprint is created by setting the various controls in the dialog. These are described below:

- **Name** - This is the name which will be assigned to the footprint when it is saved (e.g. DIP28).
- **Type** - The type of package. A line diagram of the selected type will be shown in the dialog (QUAD in the example above).
  - **SIP** (for Single Inline Package) - this consists of a single row of one or more pins.
  - **DIP** (for Dual Inline Package) - two rows of pins, numbered as for a DIP IC package.
  - **QUAD** - four rows of pins arranged as a rectangle (as shown above) and numbered as for a PLCC or QFP package.
  - **HEADER** - pins arranged in a grid and numbered sequentially by column.
  - **HEADER2** - same as **HEADER** except pins numbered sequentially by row.
  - **PGA/BGA** (for Pin Grid Array or Ball Grid Array) - pins are arranged in a rectangle, and named using the convention for grid arrays.
- **Units** - mils or mm.
- **Number of pins**
  - **Total** - the total number of pins in the package.
  - **Per horizontal row** (for QUAD, HEADER and PGA/BGA packages only) - the number of pins per horizontal row.
  - **Per vertical row** (for QUAD, HEADER and PGA/BGA packages only) - this is filled-in automatically, for information.
  - **Pin 1 position** (for QUAD packages only) - there are three options:
    - **top row, center pin**
    - **bottom row, left-most pin**
    - **top row, left-most pin**
- **Pads**
  - **Shape** - there are four options:
    - **Rectangular** - rectangular pads
    - **Pin 1 square, others round** - a square pad for pin 1, others round
    - **Round** - round pads
    - **Square** - square pads
    - **None** - used to create a hole without a pad (for a mounting hole, for example).
  - **Hole diameter** - the diameter of the hole in the pad for through-hole pads (blank or "0" for SMT pads).
  - **Width (X)** - the pad width (shown as "X" in the line diagram).
  - **Length (Y)** - the pad length (shown as "Y" in the line diagram).
  - **Spacing (E)** - the center-to-center spacing of pads within a row ("E" in the line diagram).
- **Distance between top and bottom rows** - by selecting one of the three radio buttons, this distance can be set as:
  - **Inner-inner (G1)** - the distance between the inner margins of the pads ("G1" in the diagram).
  - **Center-center (C1)** - the distance between the centers of the pads ("C1" in the diagram).
  - **Outer-outer (Z1)** - the distance between the outer margins of the pads ("Z1" in the diagram).
- **Distance between left and right rows** - by selecting one of the three radio buttons, this distance can be set as:
  - **Inner-inner (G2)** - the distance between the inner margins of the pads ("G2" in the diagram).
  - **Center-center (C2)** - the distance between the centers of the pads ("C2" in the diagram).
  - **Outer-outer (Z2)** - the distance between the outer margins of the pads ("Z2" in the diagram).
- **Preview** - clicking this button creates a preview of the footprint in the dialog.

The best source for the dimensions of a particular package is the manufacturer's datasheet. This is usually available on the Internet. In some cases, the manufacturer will even supply a drawing of the suggested PCB footprint (for SMT parts, these are often referred to as "land patterns").

After you have entered the settings for the footprint, click on **Preview** to see an image of the footprint. If it looks OK, you can save it to a library file by clicking Save. This will pop up the Save Footprint dialog, shown below.

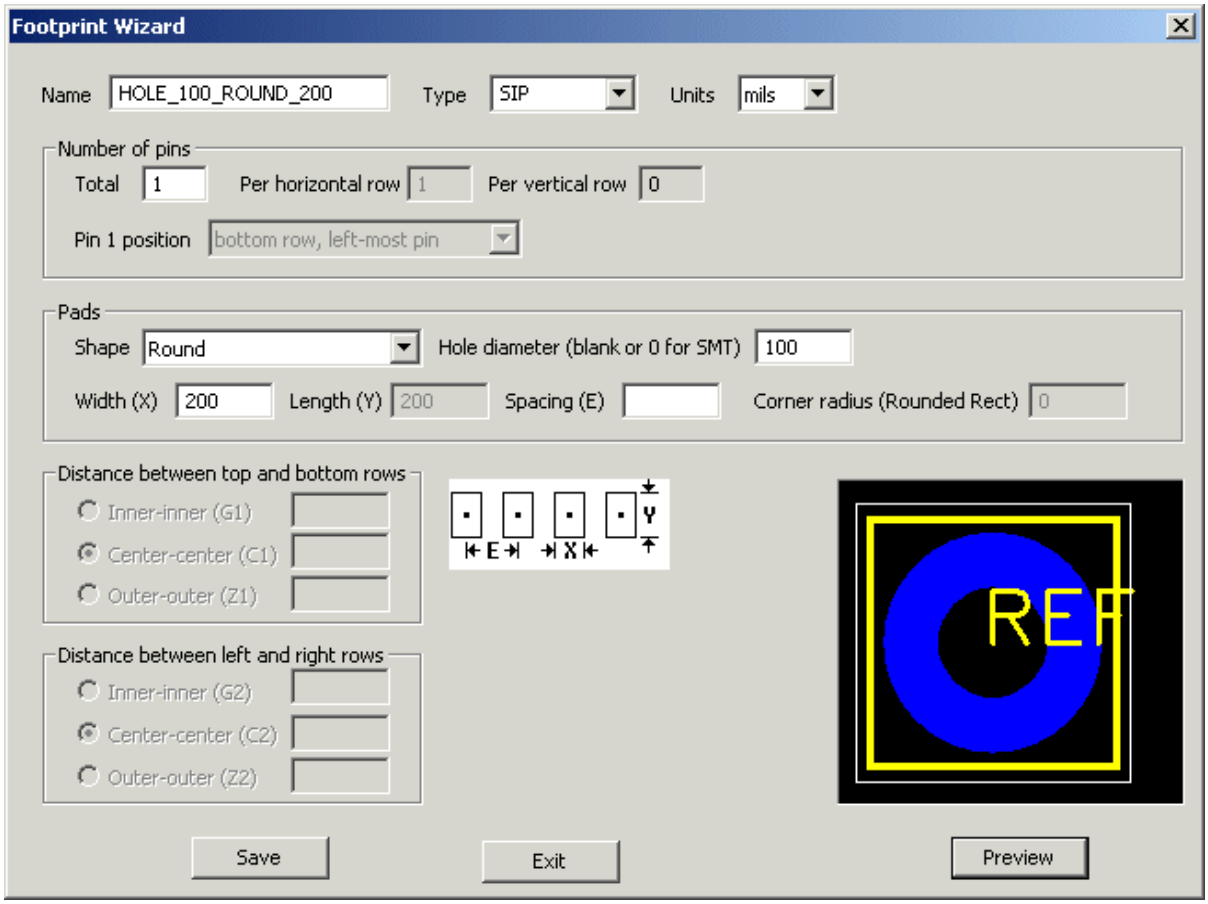


The **Footprint name** is copied from the Wizard, and can be changed here if you like. The **Author**, **Source** and **Description** fields may be filled with text describing the footprint. The **Library file** list box allows you to select a library file, or enter a new file name to create a new one. It will be initialized to **user_created.fpl**.

Several examples of footprints created with the Footprint Wizard are given below.

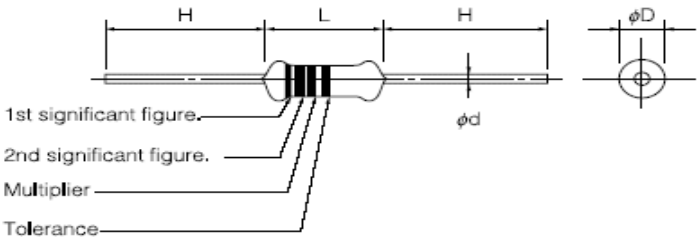
6.2.1 Example 1: Mounting Hole or Test Point with a single pin

To create a single-pin footprint for a mounting hole or test point, set the **Type** to "SIP" and the **Number of pins** to "1". The settings for a mounting hole with a hole diameter of 100 mils and a round pad with a diameter of 200 mils are shown below.



6.2.2 Example 2: Resistor with Axial leads

■ Dimensions in mm (not to scale)

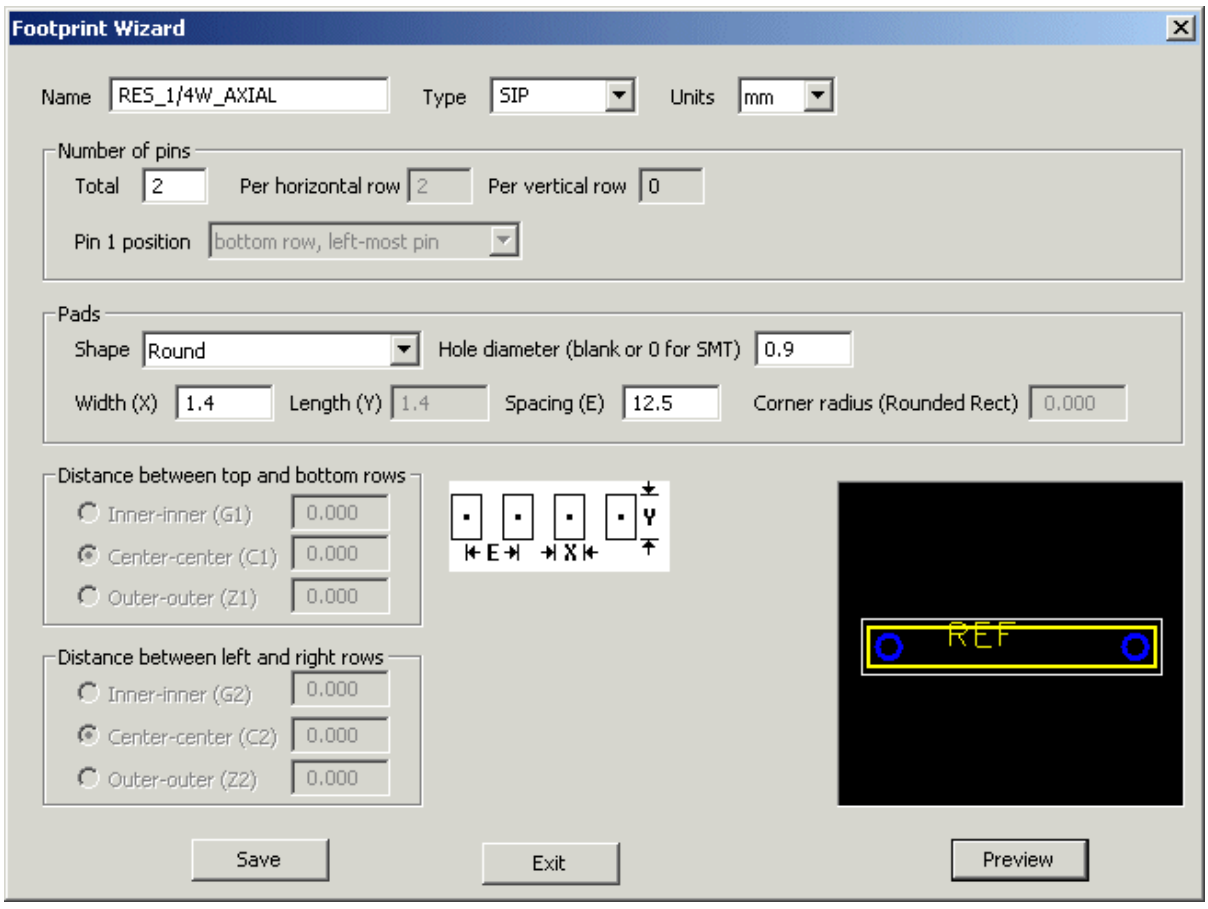


See Page ER150 for color code indication
Standard Quantity : 2000 pcs.

Type	Dimensions (mm)				Mass (mg)
	L	φD	φd	H	
ERDS1T ERDS1F	$6.35^{+0.85}_{-0.35}$	$2.30^{+0.50}_{-0.30}$	$0.60^{+0.05}$	20 min.	228
ERDS2T ERDS2F	$3.20^{+0.20}$	$1.70^{+0.20}_{-0.10}$	$0.45^{+0.05}$	20 min.	107
ERD25T ERD25F	$6.35^{+0.85}_{-0.35}$	$2.30^{+0.50}_{-0.30}$	$0.60^{+0.05}$	20 min.	228

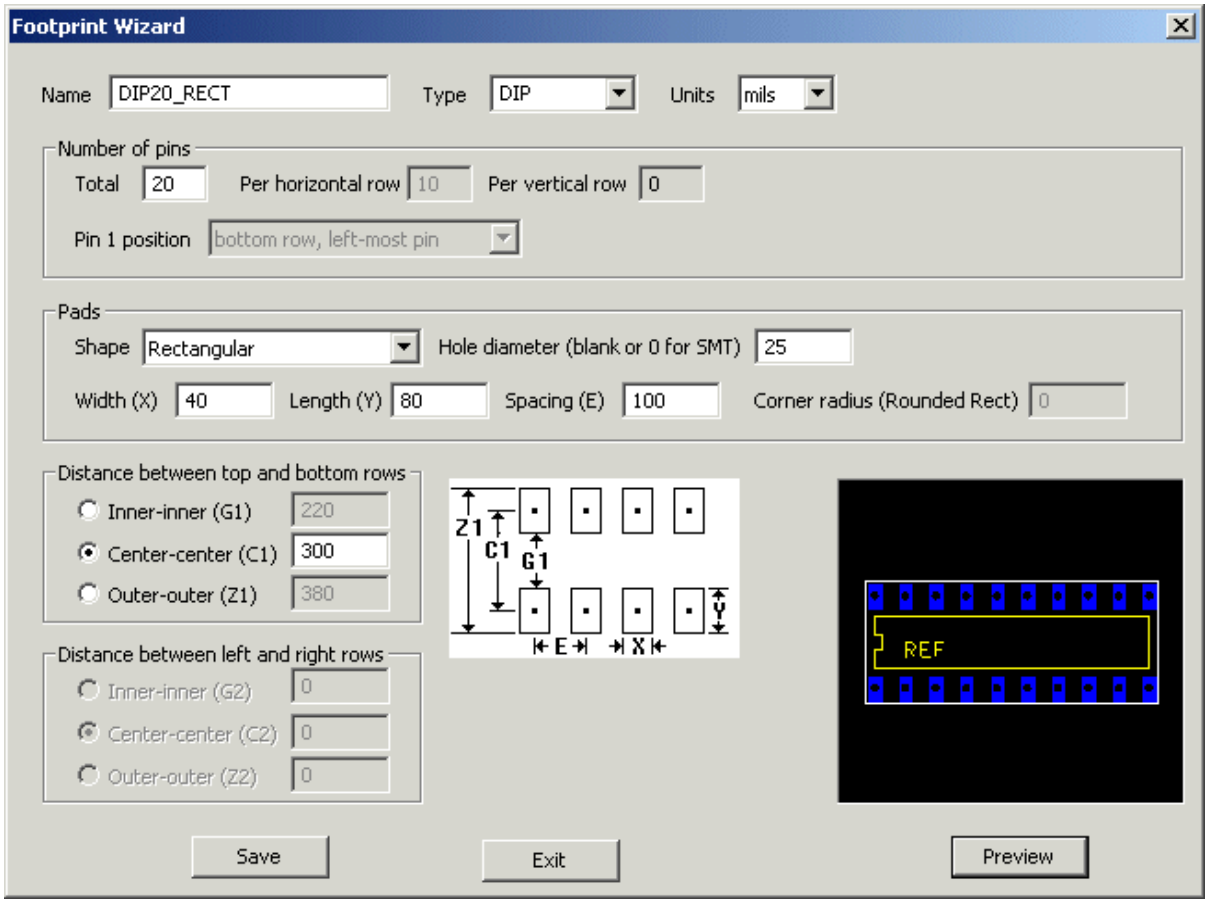
Let's start with a drawing of the resistor, taken from the Panasonic datasheet. This footprint will be for the 1/4 watt type ERD25.

From the datasheet, the package length is 6.35 mm. To allow for bending the leads, we will space our pads 12.5 mm. apart. The lead diameter is 0.60 mm. To allow clearance for insertion, the hole diameter will be 0.9 mm. The pad diameter will be 1.4 mm. The Footprint Wizard settings are shown below.



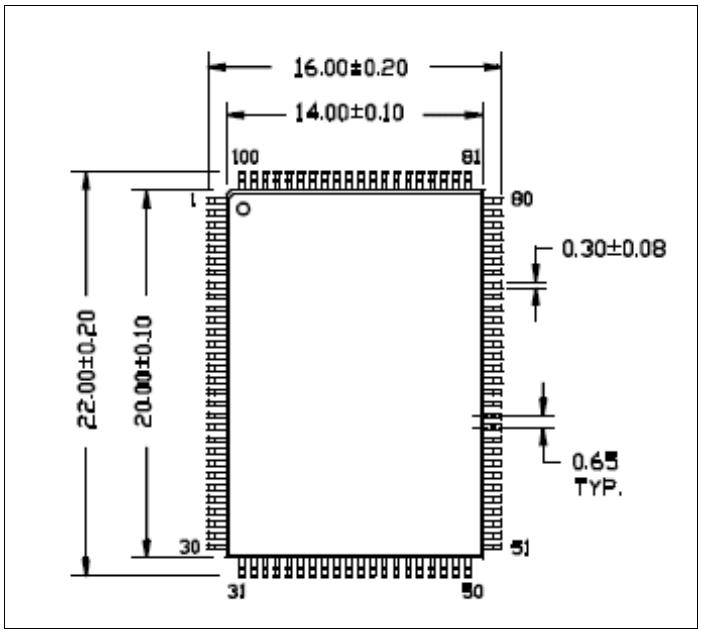
6.2.3 Example 3: DIP package with rectangular pads

Here are the settings for a DIP IC footprint with 20 through-hole pins. The standard DIP package pin spacing is 100 mils and the row spacing is 300 mils. We will use rectangular pads 80 X 40 mils in size with a 25 mil hole diameter.

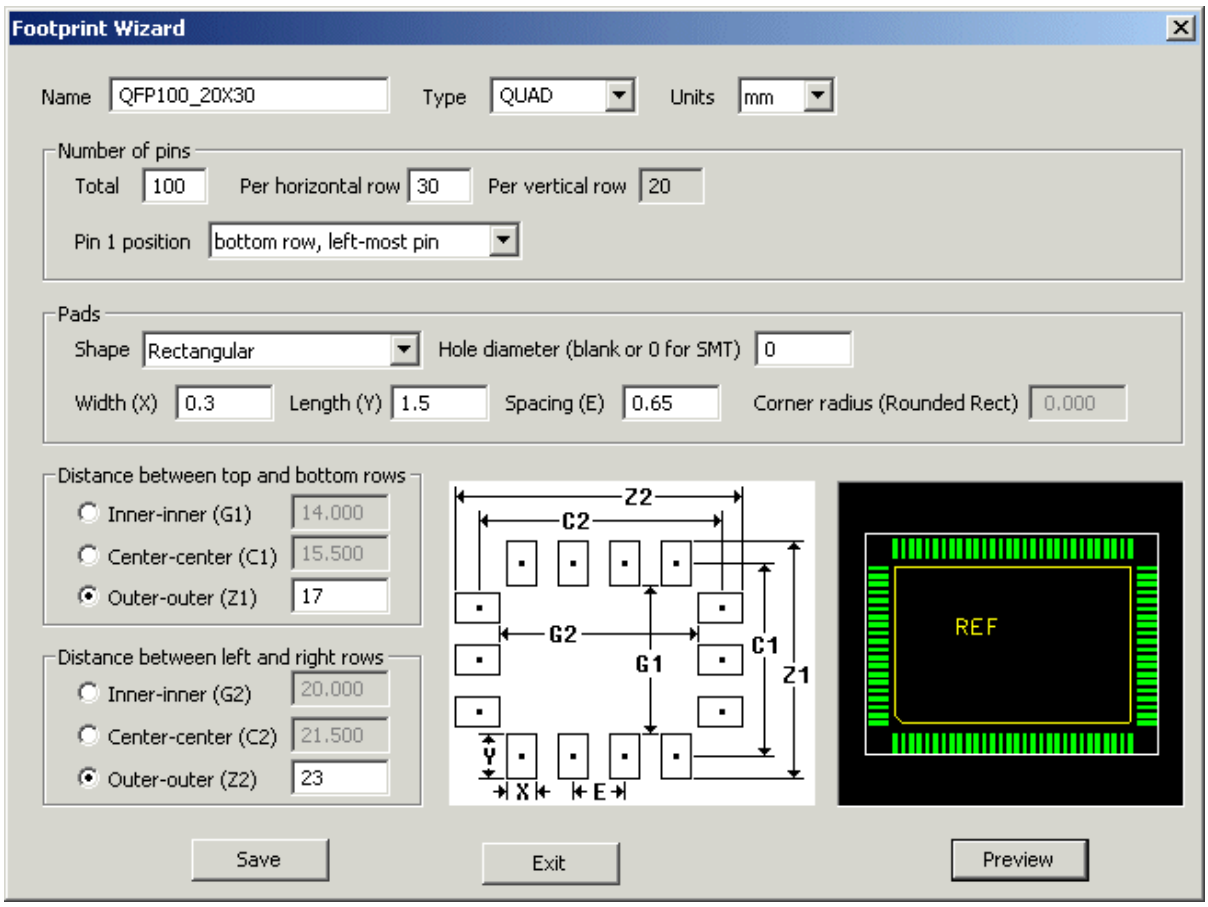


6.2.4 Example 4: QFP package with 100 pins

This footprint is based on the package for the Cypress CY7C1329, a 64K X 32 synchronous cache RAM. The package dimensions are shown in the drawing below, which was copied from the Cypress datasheet.



For the footprint, we want the pads to extend 0.5 mm beyond the end of each pin, so we will add 1.0 mm to the external dimensions of the pin rows. Also, we have turned the package on its side so that pin 1 is in the lower-left corner. The settings for the Footprint Wizard are shown below.



6.3 Footprint Editor

The Footprint Wizard is the easiest way to create new footprints, but it is limited to certain predefined pin patterns. There will be times when you need more flexibility in pin placement. Also, you may want to create more informative or artistic outlines for your footprints.

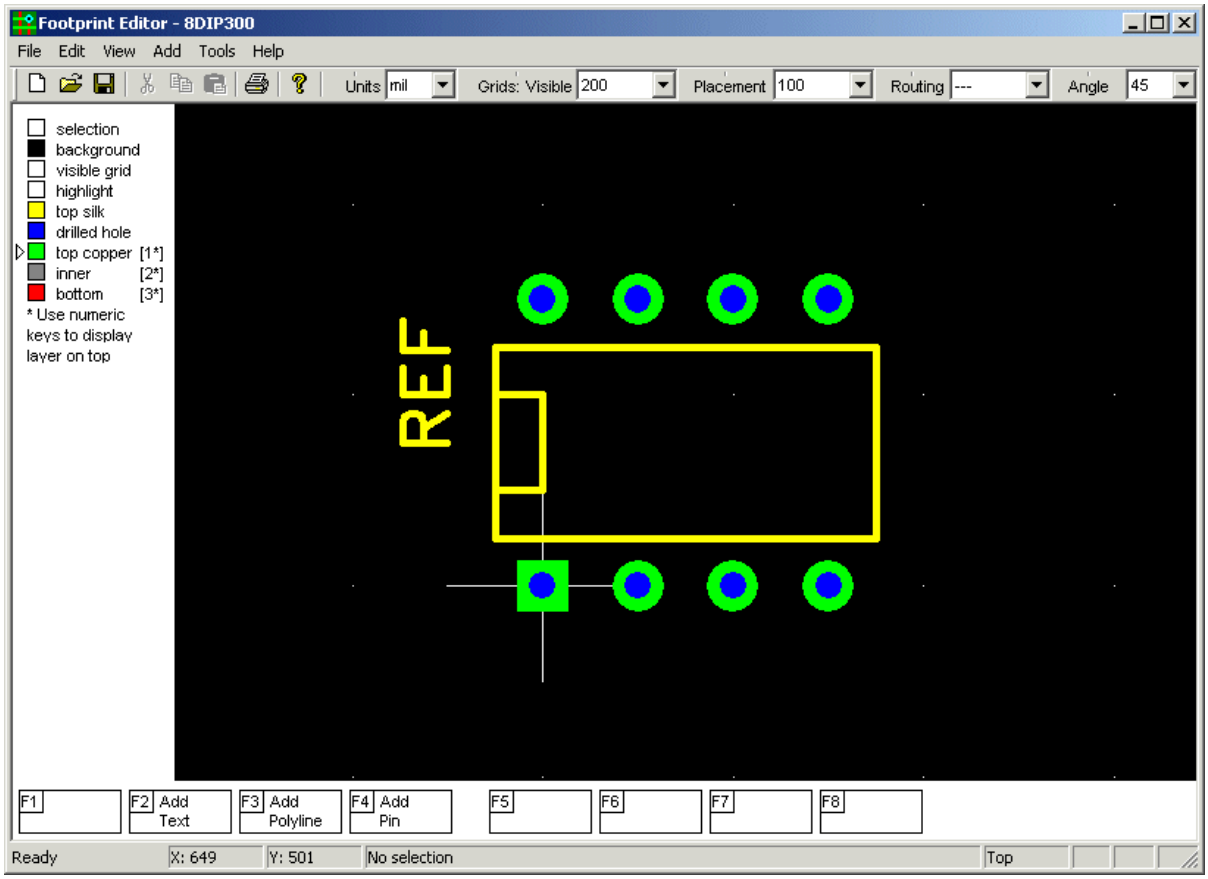
The Footprint Editor lets you do all of these things. It can be opened from FreePCB in several ways:

- By selecting the **File > Open Footprint Editor** menu item - This is available even when there is no project open in FreePCB.
- By selecting the **Tools > Footprint Editor** menu item - This is available only when there is an open project in FreePCB.
- By selecting a part, and pressing F2 ("Edit Footprint") - This opens the Footprint Editor with the footprint for the part already imported. When you leave the Footprint Editor, you will be asked if you wish to replace the part's original footprint with your modified one.

6.3.1 The Footprint Editor Window

When you open the Footprint Editor, it does not create a new window but instead replaces the usual FreePCB window. You can tell that you are in the Footprint Editor by looking at the window title bar. When you have finished editing footprints, you can return to FreePCB by pressing F8 ("Return to PCB") or selecting the **File > Return to PCB Layout** or **Tools > Return to PCB layout** menu items, or by clicking in the "Close" box in the upper right corner.

A screenshot of the Footprint Editor window is shown below, with a DIP8 footprint imported.



As you might expect, the menus in the Footprint Editor are different from the FreePCB menus. They are listed below:

- **File** -
  - **New footprint** - Start editing a new footprint..
  - **Import Footprint** - Import a footprint from a library file.
  - **Save Footprint As** - Save the footprint to a library file.
  - **Return to PCB layout** - Close the Footprint Editor and return to FreePCB.
- **Edit** -
  - **Undo** - Undo the last editing operation (if possible).
- **View** -
  - **Entire footprint** - Resize and recenter the window to display all of the footprint elements.
- **Add** -
  - **Pin** - Add a new pin or row of pins.
  - **Polyline** - Add a new polyline to the silk-screen layer.
  - **Text** - Add a text string
- **Tools** -
  - **Footprint Wizard** - Invoke the Footprint Wizard.
  - **Make PDF from Library File** - Make a PDF file describing the footprints in a library file.
  - **Return to PCB Layout** - Same as in the File menu.
- **Help** -
  - **About FreePCB** - Display the "About Box" with the version number.

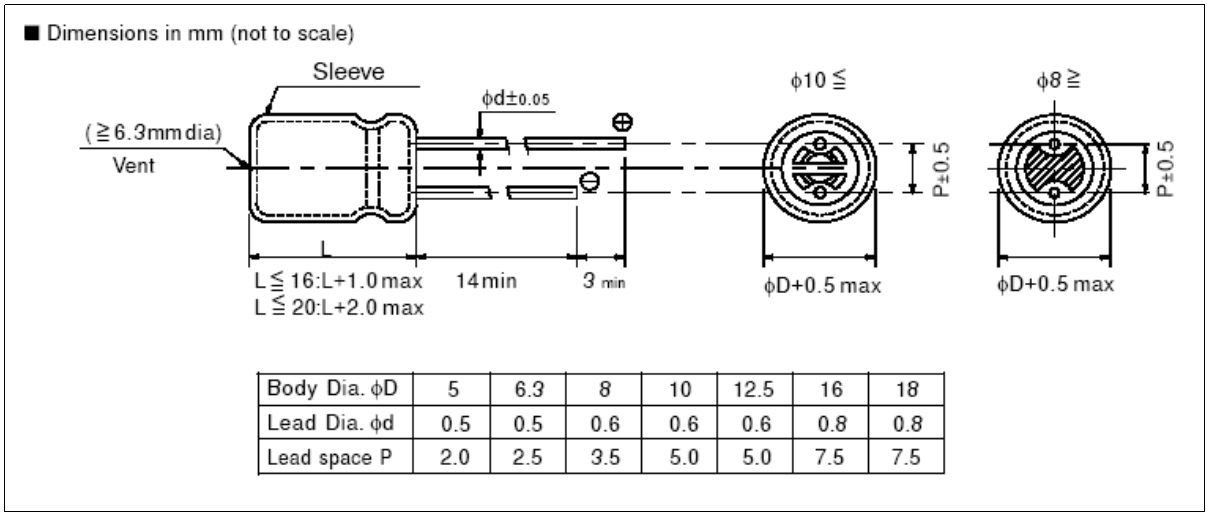
Other elements of the window such as the toolbar, layer list, function key menu, status bar and layout window are pretty much the same as for FreePCB.

6.3.2 Footprint Elements

A footprint consists of 4 basic elements. These are:

- **Pins** - these are implemented as **padstacks**. A padstack contains copper pads for the top, inner and bottom layers as well as an optional hole.
- **Polylines** - these are graphical elements on the silk-screen layer. They may be **closed** or **open**.
- **Reference designator** - this is a text string for the reference designator. Since the actual reference designator is a property of the part and not the footprint, the Footprint Editor uses the string "REF" as a placeholder. It is always present and cannot be deleted, but it can be resized.
- **Selection box** - this is a rectangle that encloses all of the other elements, used for selection of a part during PCB layout. It is generated automatically when saving the footprint.

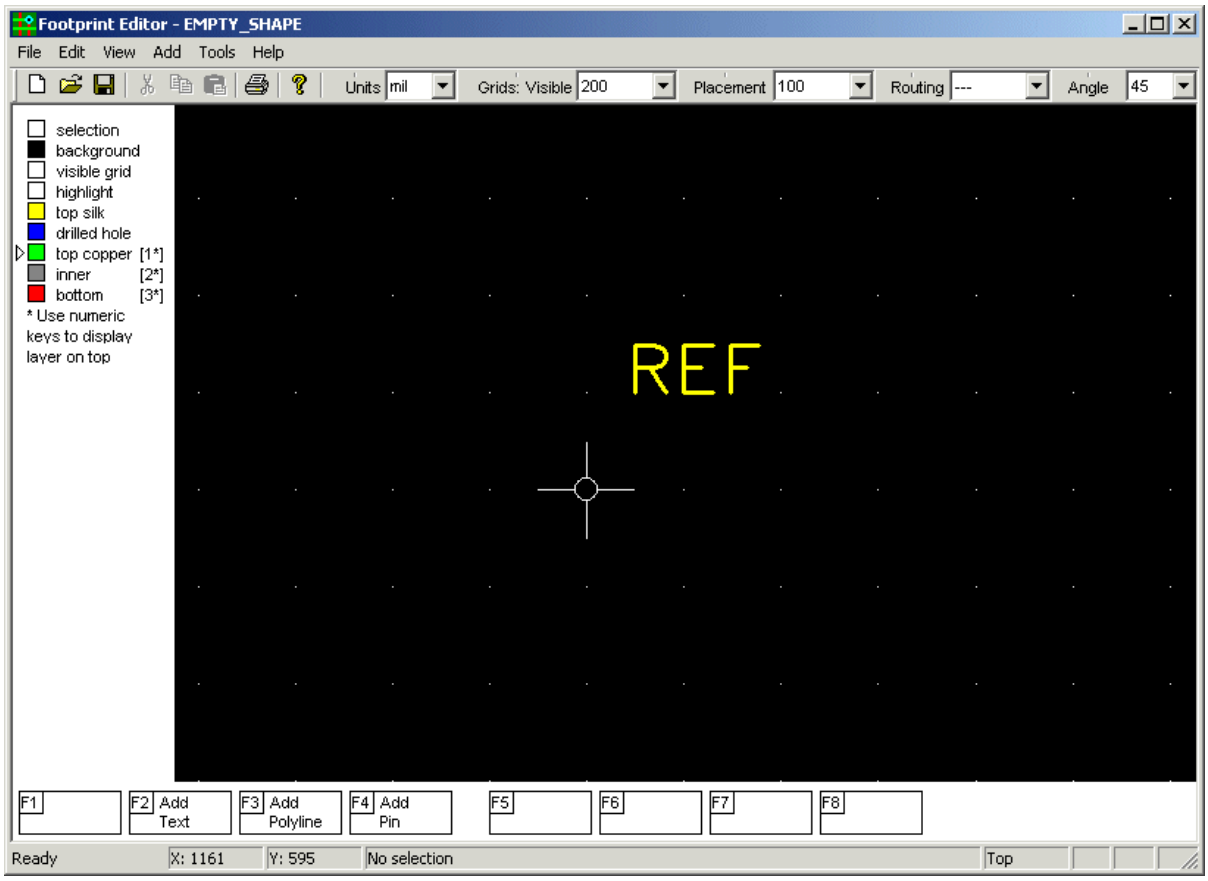
Creating a footprint is actually very similar to laying out a PCB, so if you have used FreePCB then you already have most of the required skills. Therefore, we will take a tutorial approach to the subject. In the next few sections, we will create a footprint for a polarized capacitor. The capacitor dimensions are shown in the drawing below, taken from a Panasonic datasheet. We will create a footprint for the 12.5 mm body diameter capacitor.



Body Dia. ϕD	5	6.3	8	10	12.5	16	18
Lead Dia. ϕd	0.5	0.5	0.6	0.6	0.6	0.8	0.8
Lead space P	2.0	2.5	3.5	5.0	5.0	7.5	7.5

6.3.3 Starting the new footprint

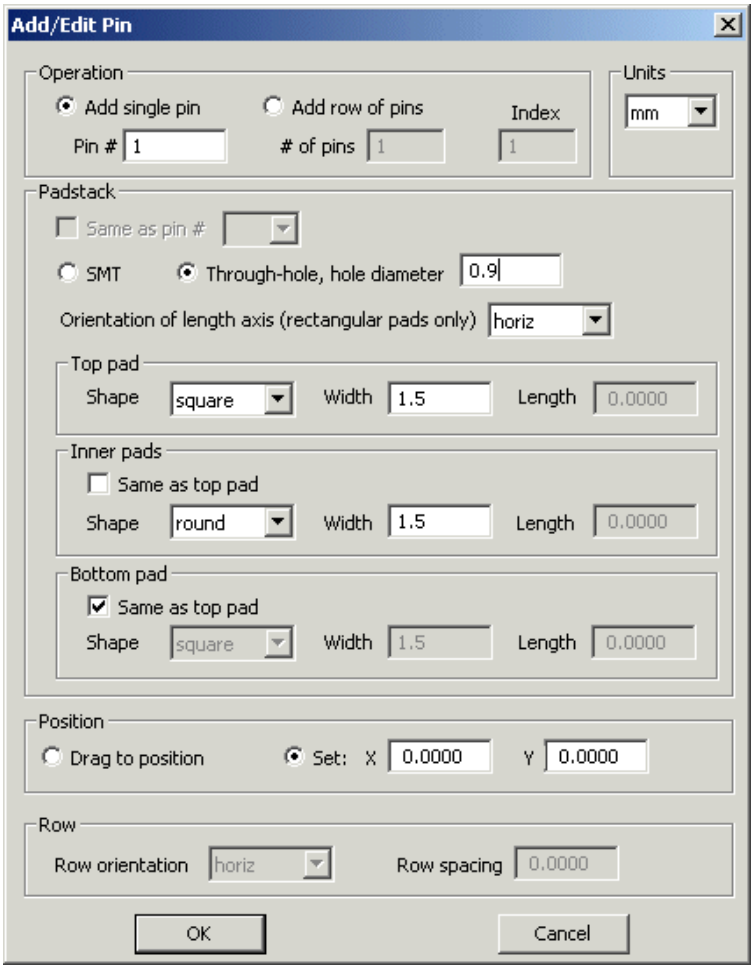
From FreePCB, invoke the Footprint Editor from the **File** or **Tools** menus. You will start with a layout window which is empty except for the origin symbol and the reference designator, as shown in the screenshot below. Since we will be using metric units, make sure that the **Units** are "mm".

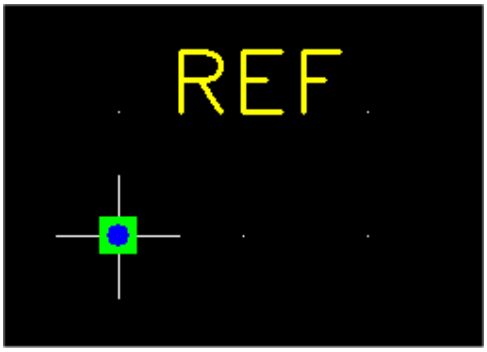


6.3.4 Adding and Editing Pins

Select the **Add > Pin** menu item. This pops up the **Add/Edit Pin** dialog. The controls in this dialog are pretty much self-explanatory. Some of them will be enabled or disabled depending on the pin being added. Since we are adding the first pin, the **Padstack > Same as pin #** check box is disabled because there are no other pins present.

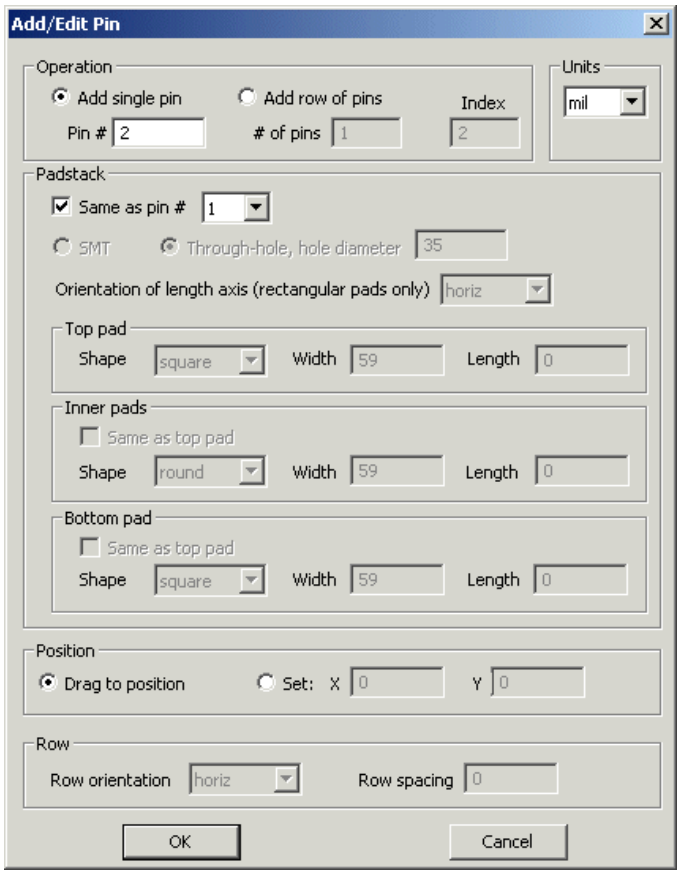
The capacitor has leads with a diameter of 0.6 mm. For clearance, we will use a hole diameter of 0.9 mm. and pad diameters of 1.5 mm. For the first pin, we will use square pads on the top and bottom layers and a round pad on the inner layers. Select the **Padstack > Through-hole** radio button and set the **hole diameter** to 0.9 mm. Set the **Top pad > Shape** to "square" and the **Width** to 1.5 mm. The **Inner pad > Shape** should be "round". The **Bottom pad** should be the same as the **Top pad**. Set the **Position** to **X** = 0.0 and **Y** = 0.0. The dialog should look like this:





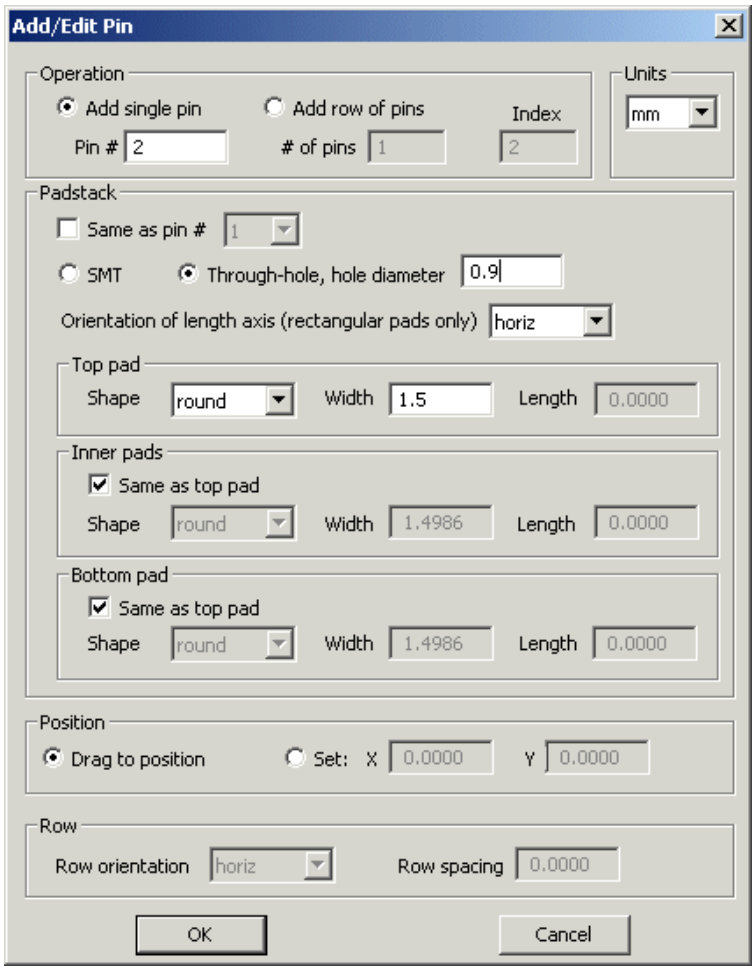
Clicking **OK** should result in the pad being placed at the origin.

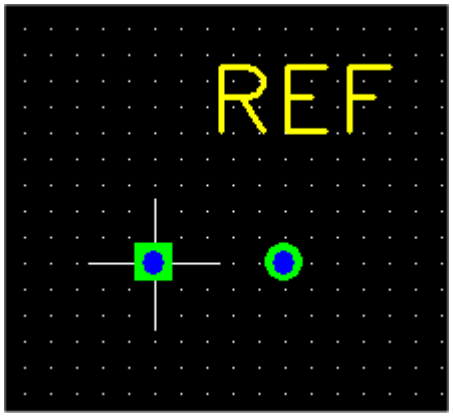
According to the datasheet, the lead spacing is 5 mm. so we will place our second pin at X = 5.0 mm and Y = 0.0 mm. To make this easier, set the **Visible** and **Placement** grids to "1 mm". Now select Add > Pin to add the second pin.



Since there is already one pin in our footprint, the dialog will come up initialized for the second pin, with the **Padstack** the same as pin 1 by default. However, we should use a round pad for pin 2 instead of a square pad. Therefore, uncheck the **Padstack > Same as pin #** check box, and change the **Padstack > Top pad > shape** to "round". Then you can check **Padstack > Inner pads > Same as top pad** and **Padstack > Bottom pad > Same as top pad** to make these pads are the same as the top pad.

Now the dialog should look like:



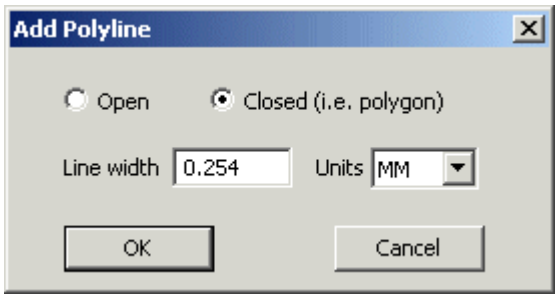


Click **OK** to leave the dialog and start dragging the pad. Place it at X = 5.0 mm and Y = 0.0 mm, as shown.

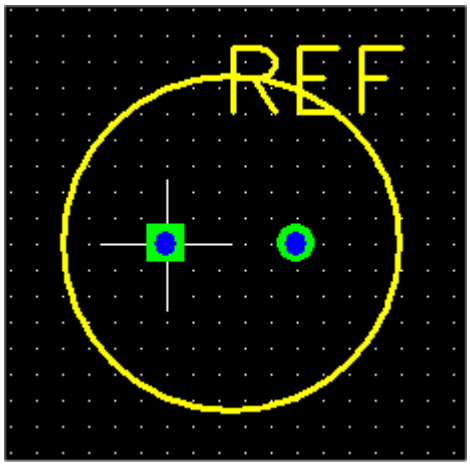
6.3.5 Adding Polylines

Now we will add some silk-screen graphics to our footprint, consisting of a part outline (a circle with about the same diameter as the body diameter of the capacitor), and a "+" to mark the positive terminal, which by convention is pin 1.

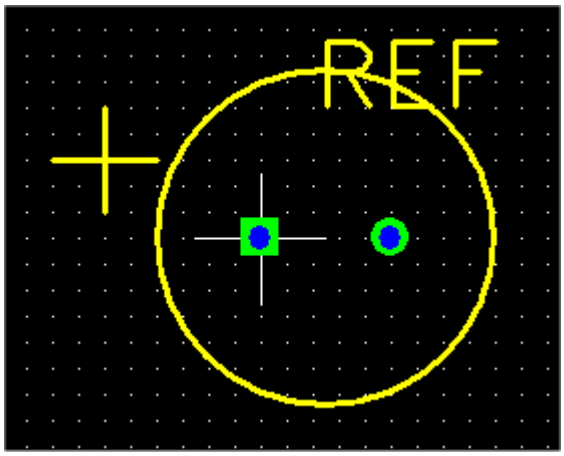
Let's start with the circle. This will be a closed polyline, with arcs for sides. Since we will need to place corners mid-way between the pins on the X-axis, set the **Placement** grid to "0.5 mm". Also, make sure that the **Angle** grid is set to "45". Then select the **Add > Polyline** menu item. This will pop up the **Add Polyline** dialog:



For the circle we need a closed polyline so make sure that the **Closed** button is selected, as shown above. The **Line width** is set to 0.254 mm (or 10 mil) by default. This is a reasonable value so you can leave it alone. Click **OK** to start dragging the first corner of the polyline. Place it at X = -4.0 mm, Y = 0.0 mm. While dragging the second corner, press F2 ("Arc (CW)") to change the side style to a clockwise arc. Place the second corner at X = 2.5 mm, Y = 6.5 mm. Place the third corner at X = 9.0 mm, Y = 0.0 mm and the fourth corner at X = 2.5 mm, Y = -6.5 mm. Then right-click to close the polyline. Your circle should look like:

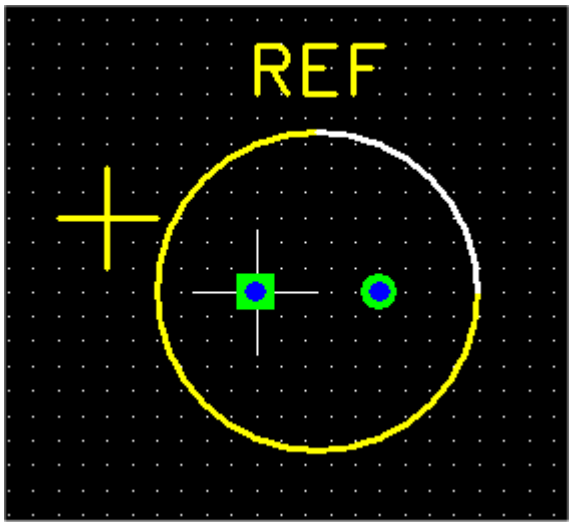


Now let's add the "+". Select **Add > Polyline**, but this time make it an open polyline. Draw a vertical line by placing the first corner at X = -6.0 mm, Y = 5.0 mm and the second corner at X = -6.0 mm, Y = 1.0 mm. Then right-click to stop drawing. Add another open polyline, and draw a horizontal line by placing the first corner at X = -8.0 mm, Y = 3.0 mm and the second corner at X = -4.0 mm, Y = 3.0 mm. Now your footprint should look like:



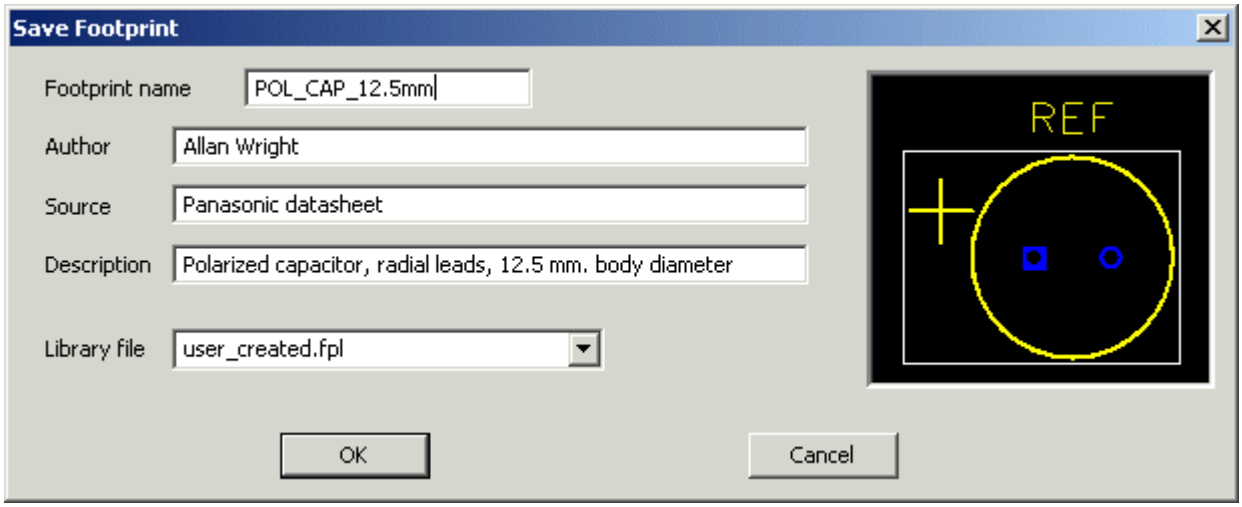
6.3.6 Modifying the Reference Designator

Finally, let's move the reference designator out of our part outline. Click on it to select it, and press F4 ("Move Ref Text") to start dragging it. Move it above the part outline and place it by left-clicking. If you want to make it larger or smaller, you can press F1 ("Set Size"), and then use the **Reference Text Properties** dialog to change the size and stroke width as you see fit. Your final footprint should look something like:



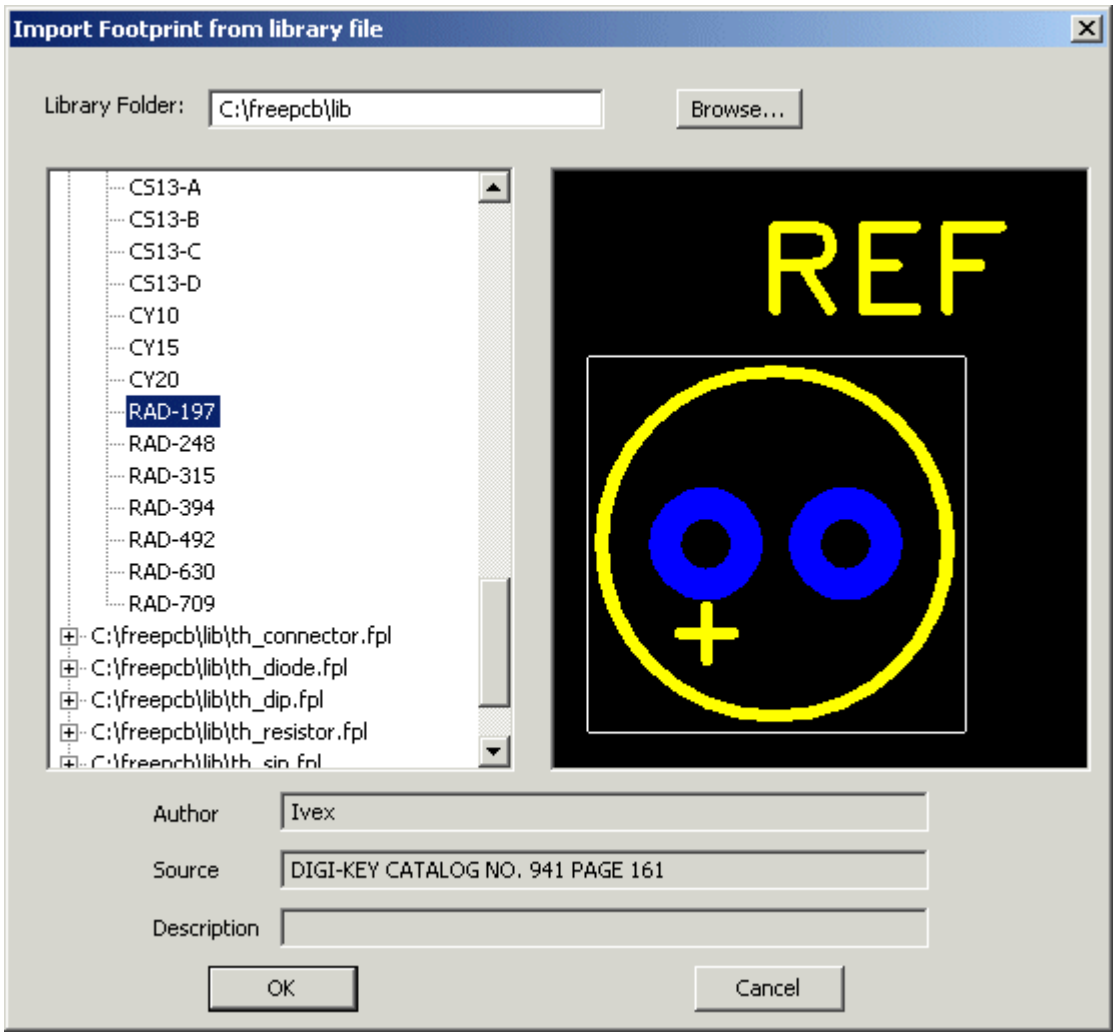
6.3.7 Saving the Footprint

To save your new footprint into a library file, select the **File > Save As** menu item. This will pop up the **Save Footprint** dialog. Enter the **Footprint name** and (optionally) the **Author**, **Source** and a **Description**, as shown below. Then select the **Library file** that you wish to use (or enter a new filename), and click **OK** to save the footprint.



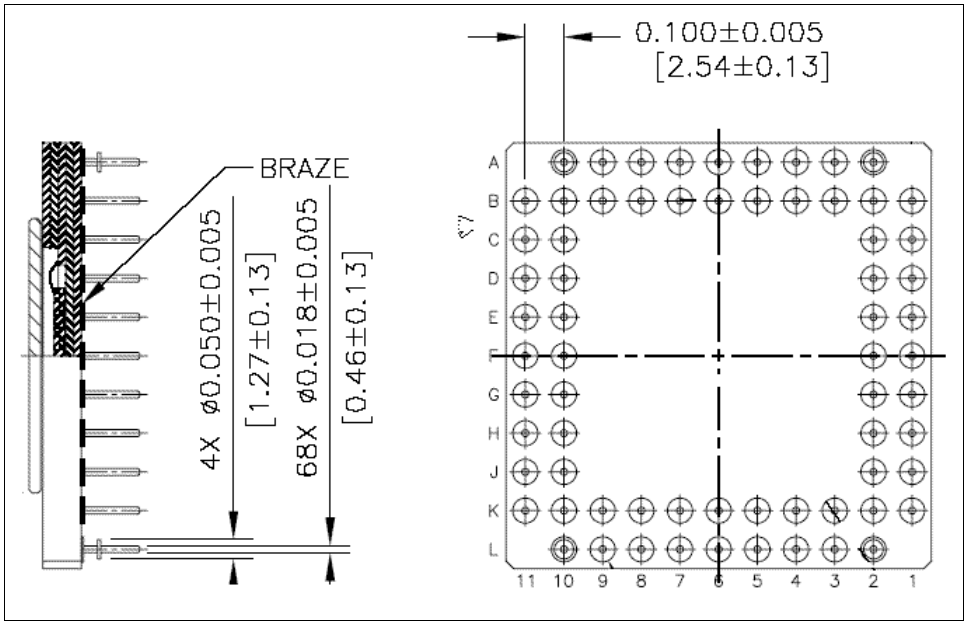
6.3.8 Importing Footprints

So far, we have described how to create a footprint from scratch. In many cases, it will be easier to start with an existing footprint and modify it. You can import a footprint into the Footprint Editor by using the **File > Import footprint** menu item. This pops up the following dialog.



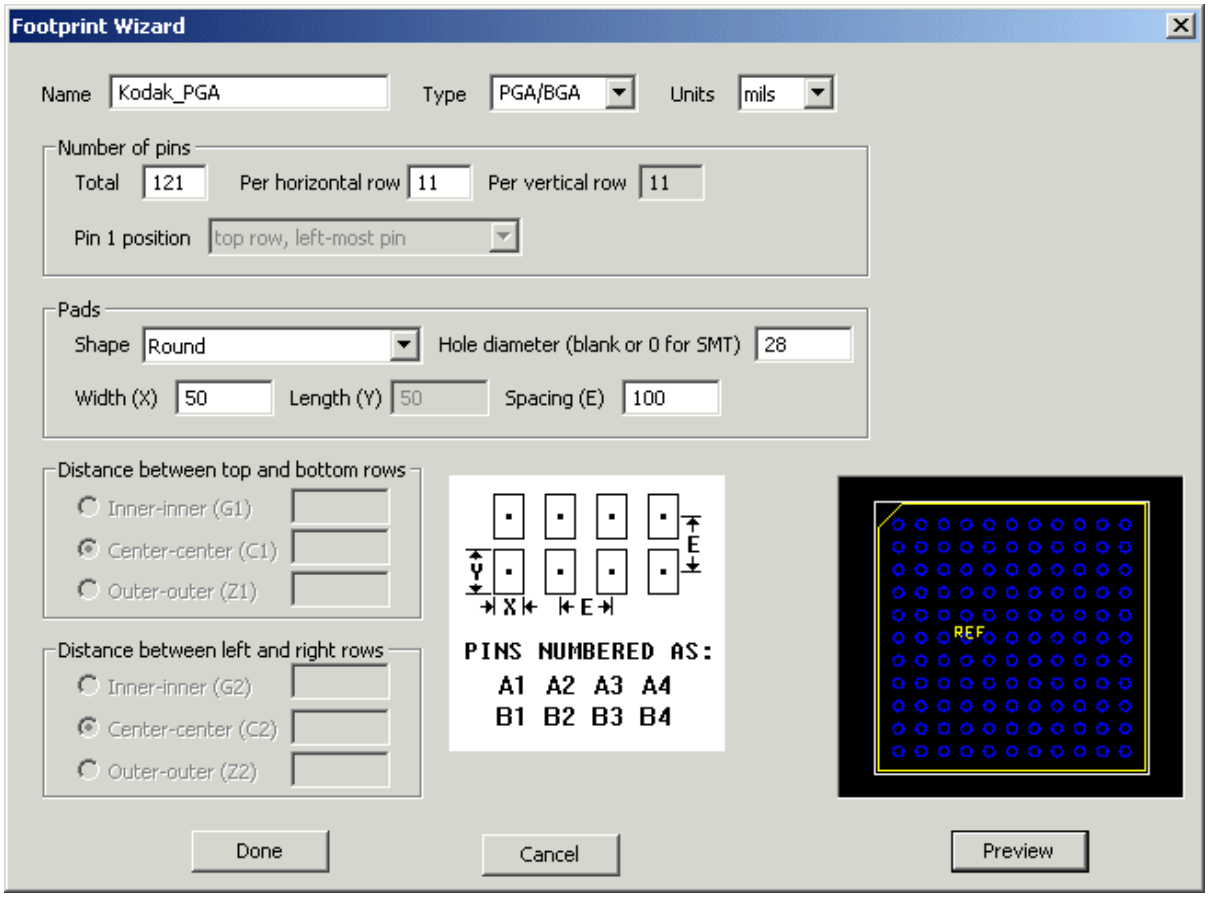
From here, you can select a library folder, open library files and select footprints. You will see a preview of the selected footprint in the dialog. Click **OK** to import the footprint into the Footprint Editor. Then you can edit it.

6.3.9 Using the Footprint Wizard

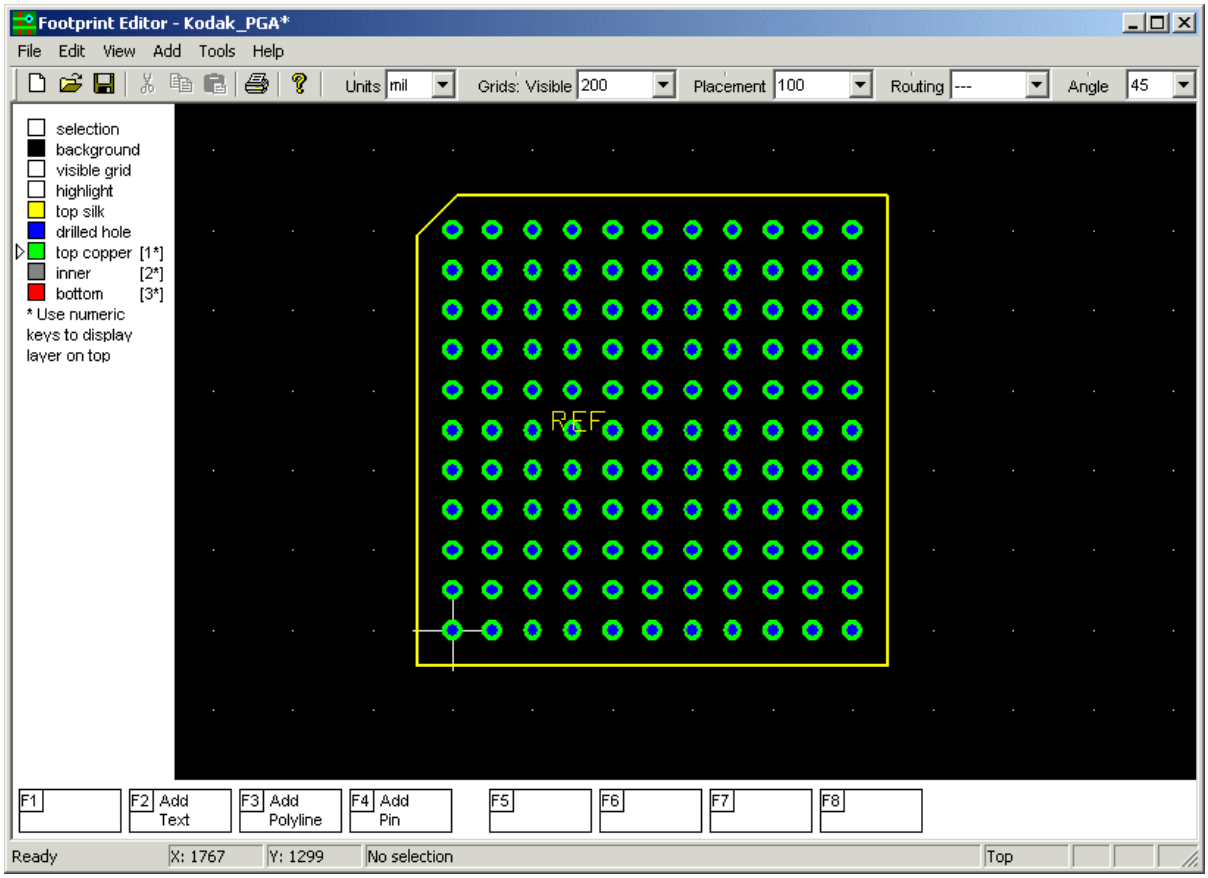


You can also use the Footprint Wizard from within the Footprint Editor. Here's an example. Let's make a footprint for the PGA package shown below. This is a Kodak image sensor. The pin diameter is 18 mils, and the pin spacing is 100 mils. The pins are named as usual for a PGA.

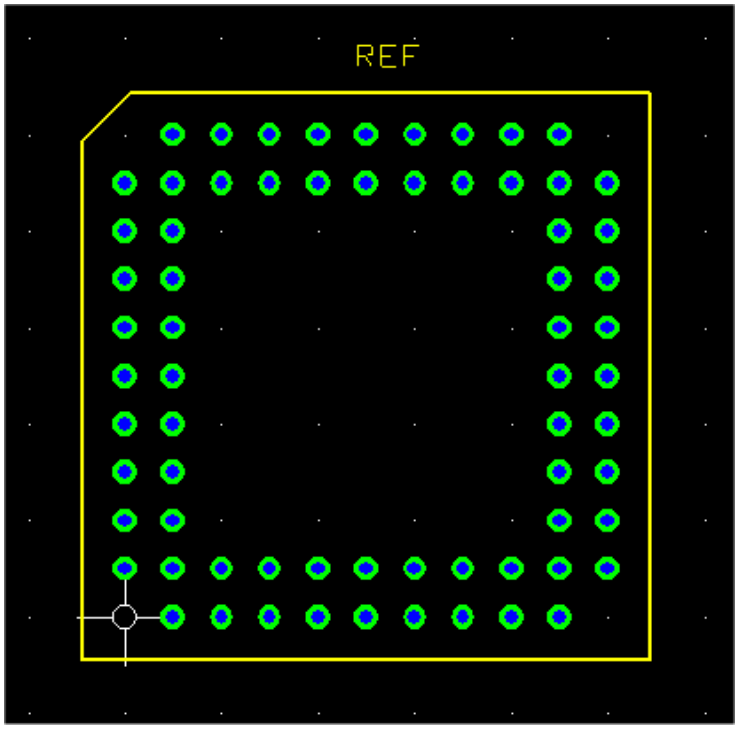
From within the Footprint Editor, select **Tools > Footprint Wizard**. We will use round pads with a diameter of 50 mils, and a hole diameter of 28 mils. Since the pins are arranged in 11 columns and 11 rows, we will initially create our footprint as a matrix of 11 x 11 = 121 pins. Then we will remove the extra ones. Set up the **Footprint Wizard** dialog as shown below.



Click **Done** to create the footprint and import it into the Footprint Editor. It should look like:

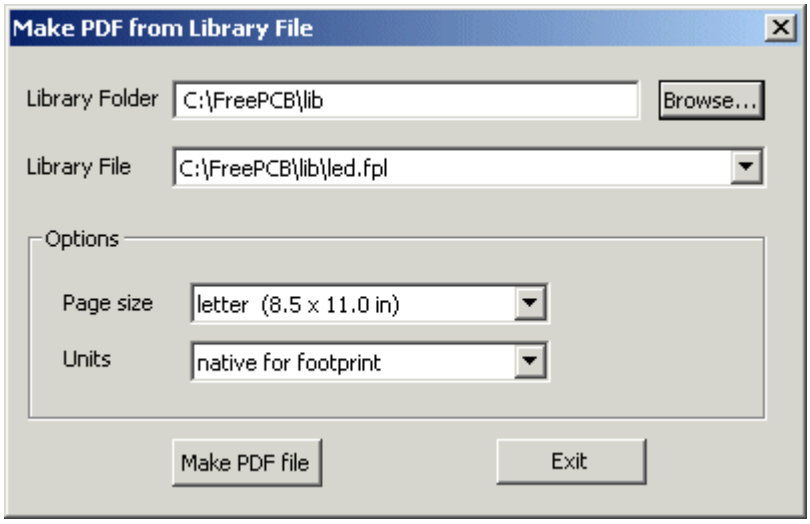


Now, select each extra pin and delete it. Move the "REF" string so that it will be visible with the part in place. Your finished footprint should look like:

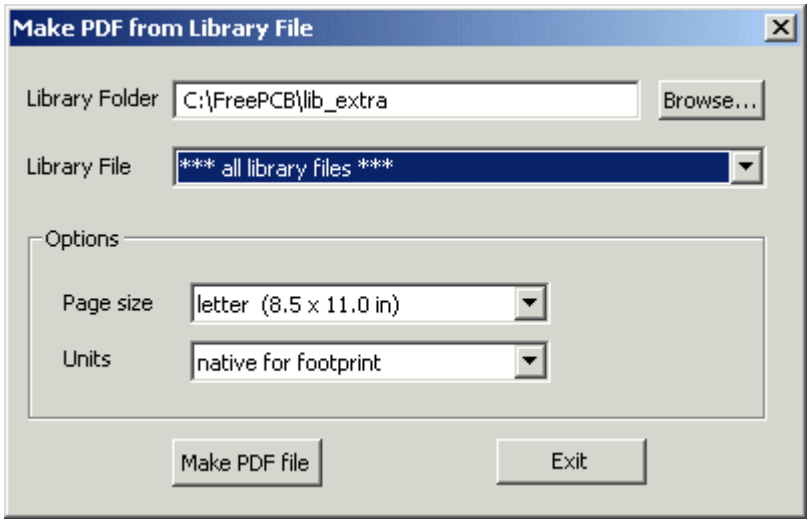


6.3.10 Making PDF Files from Libraries

If you create or modify a library file, you can make a PDF file describing the footprints in the library by selecting **Tools > Make PDF from Library File...** This will pop up the following dialog:



If necessary, use the **Library Folder** field to select a library folder. Use the **Library File** drop-down menu to select the file that you wish to document. To make PDF files from ALL of the library files in the folder, scroll to the bottom of the file menu and select "*** all library files ***", as shown below. You can use the **Page size** menu to select between letter and A4 page sizes, and the **Units** menu to choose the units that will be used in the PDF. Then click **Make PDF file** to make the file(s).

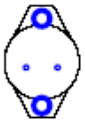


A sample page from a PDF file is shown below.

TO-3

Author: Ivex
Source: JEDEC PUBLICATION NO.95 BOOK ONE PAGE 98
Size: 1050 x 1573 mil

Scale 1:2

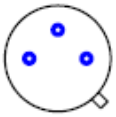


PIN(s)	PAD			
	TYPE	SHAPE	SIZE	HOLE
1-2	TH	Round	95 mil	45 mil
3-4	TH	Round	300 mil	150 mil

TO-5

Author: Ivex
Source: JEDEC PUBLICATION NO.95 BOOK ONE PAGE 99
Size: 370 x 370 mil

Scale 2:1

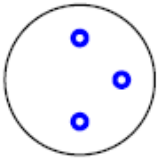


PIN(s)	PAD			
	TYPE	SHAPE	SIZE	HOLE
1-3	TH	Round	50 mil	20 mil

TO-8

Author: Ivex
Source: JEDEC PUBLICATION NO.95 BOOK ONE PAGE 100
Size: 524 x 524 mil

Scale 2:1



PIN(s)	PAD			
	TYPE	SHAPE	SIZE	HOLE
1-3	TH	Round	65 mil	28 mil

TO-8BB-54

Author: Ivex
Source: DIGITAL PRINTED CIRCUIT DESIGN AND DRAFTING, PAGE 398
Size: 320 x 320 mil

Scale 2:1

PIN(s)	PAD			
	TYPE	SHAPE	SIZE	HOLE
1-8	TH	Round	54 mil	20 mil

TO-8BB-62

Author: Ivex
Source: DIGITAL PRINTED CIRCUIT DESIGN AND DRAFTING, PAGE 398
Size: 342 x 342 mil

Scale 2:1

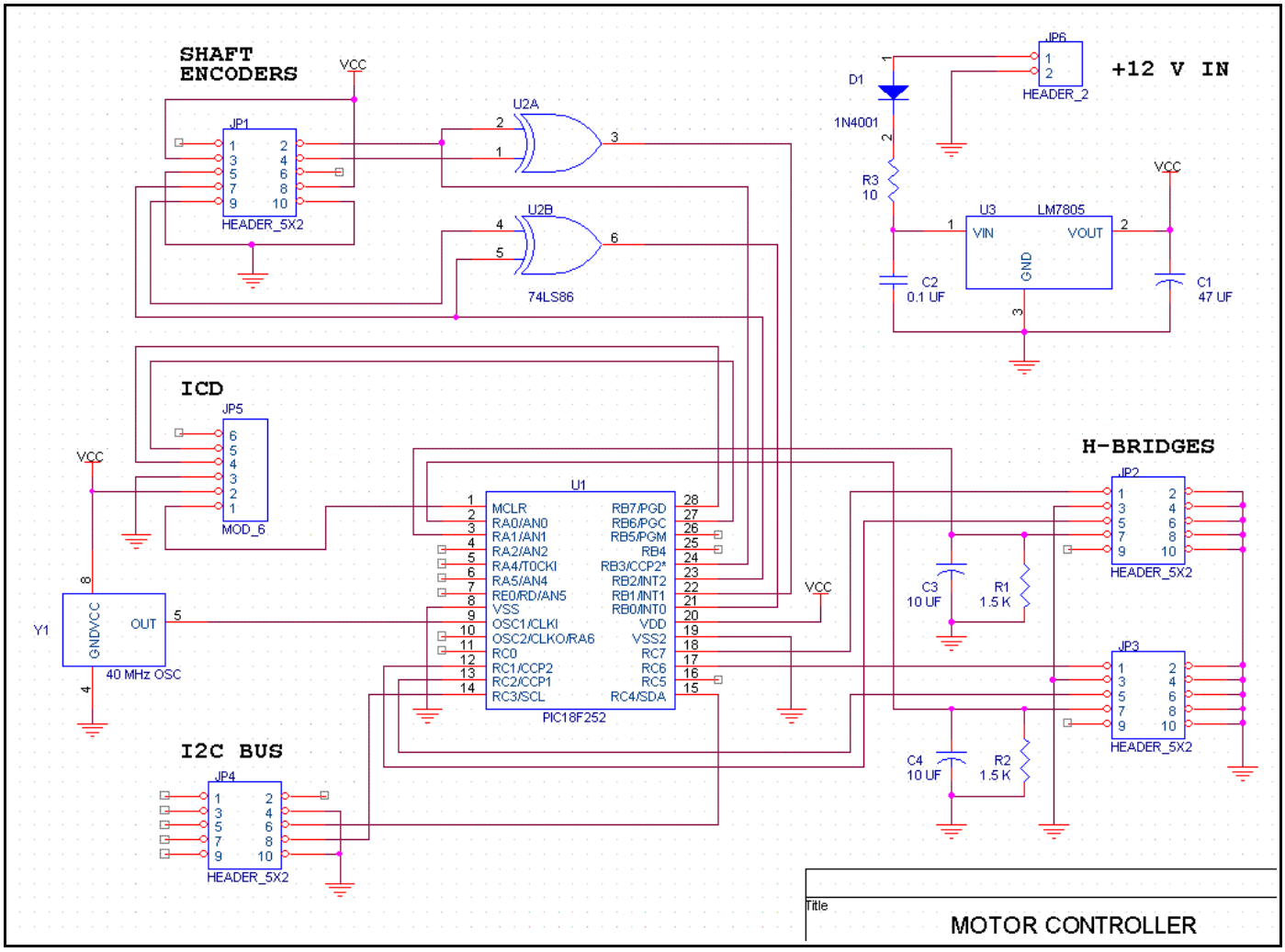
PIN(s)	PAD			
	TYPE	SHAPE	SIZE	HOLE
1-8	TH	Round	62 mil	20 mil

7. Tutorial

7.1 Schematic Diagram

The best way to learn a new CAD program is to try it out. This tutorial will guide you through the layout of a small PCB for a motor controller, whose schematic is shown below. Sorry that this image is rather large.

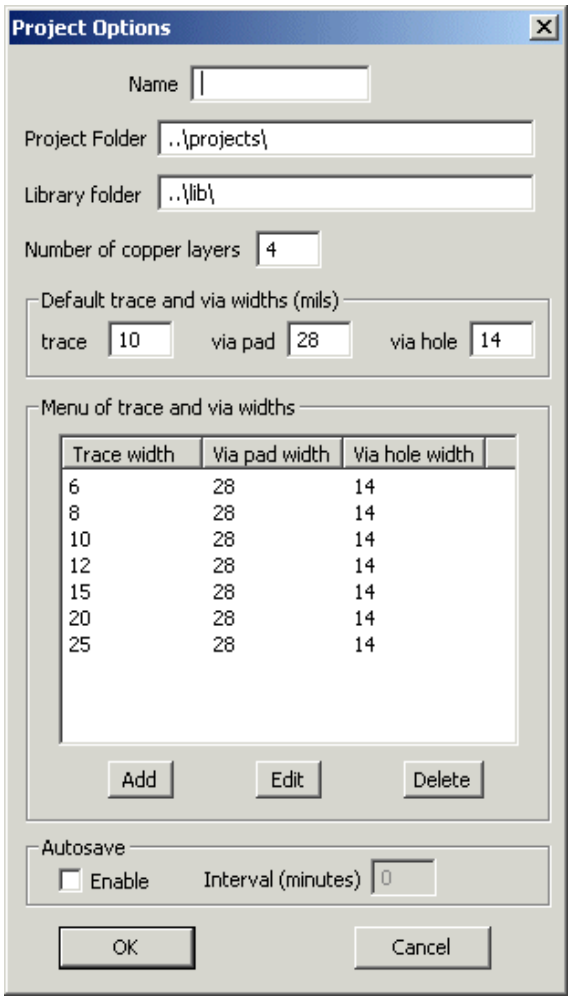
You may wish to print out a copy of the schematic so that you can follow it as we lay out the PCB.



7.2 Creating the Project

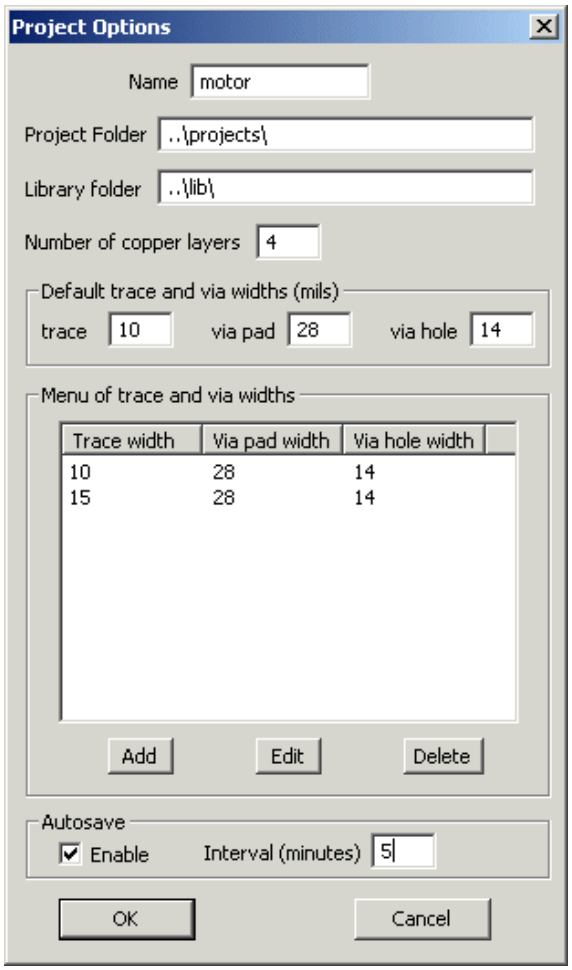
First we will create a new project called "Motor".

- Select **New** from the **File** menu. The following dialog should appear.



- Enter the word "motor" in the **Name** box. This establishes the name of the project.
- Since the project folder normally has the same name as the project, "motor" will be automatically added to the end of the folder path in the **Project Folder** box, which should now read "..\projects\motor". Note that this path is relative to the location of the application, which is normally in **FreePCB\bin**. If you wanted to use some other folder as the project folder, you can override the default by entering a different folder path, which can be either relative to the application folder or absolute. If the project folder doesn't already exist, FreePCB will create it (although its parent folder must exist). Since the "motor" project folder already exists and contains the netlist file for the motor controller, I would suggest leaving this path alone.
- Similarly, you can override the path to the **Library Folder**, which contains the footprint libraries. If you did a standard installation of FreePCB this should not be necessary.
- If you decide to move your folders around, you can change the default paths by editing the file **default.cfg** in the application folder. If you move the library folder after creating the project, you can edit the library path in the project file **motor.fpc**.
- Leave the number of copper layers at "4".
- The **Default trace and via widths** will be applied to any net or trace which doesn't have widths explicitly assigned. Since we will be using 10 mils as our default trace width, you can leave them alone..
- The **Menu of trace and via widths** is a list of values which will be presented if you explicitly assign trace or net widths later on. You are not limited to using these values, but using a selection from the menu is convenient and reduces the chance of error. If you know that you will be using certain widths which are not on the default menu, you can add them by clicking on **Add**. You can also delete items which you know you will not be using with **Delete**, or change an item by selecting it and clicking **Edit**.
- In this tutorial, we will be using trace widths of 10 and 15 mils. Select each of the other widths in the menu and **Delete** them.
- **Autosave** will automatically save your project periodically. **Enable** it and set the save **Interval** to 5 minutes.

- Now your dialog should look like:

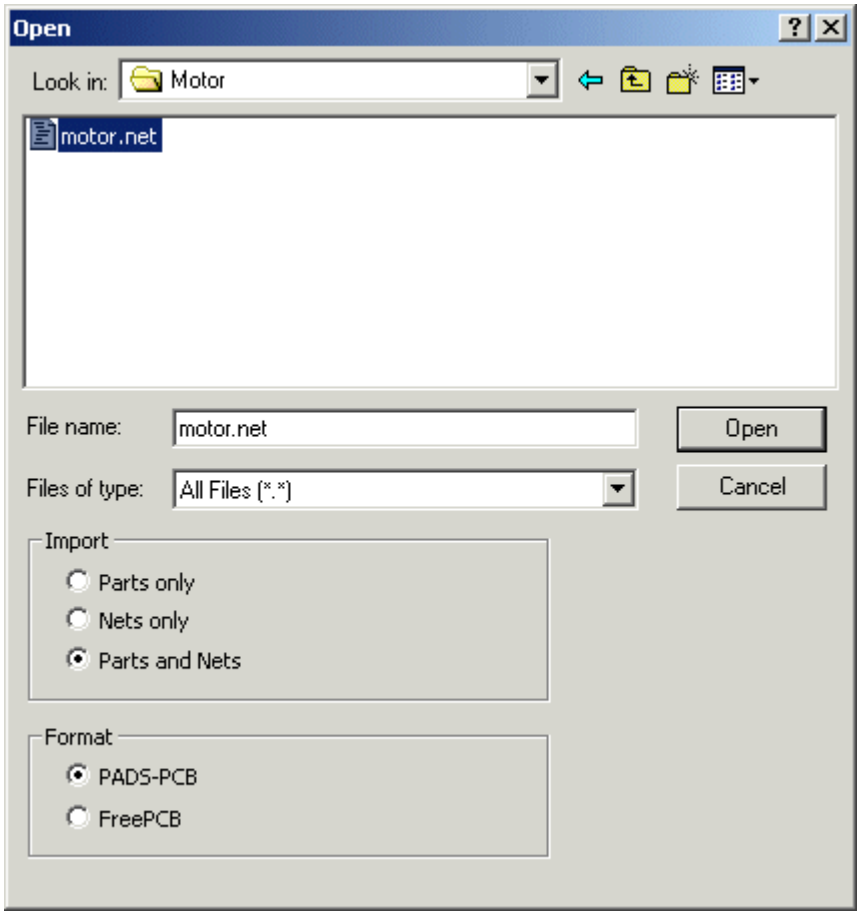


- Click **OK** to exit the dialog.
- The title bar of the FreePCB window should now read "FreePCB - motor.fpc*", where **motor.fpc** is the name of the project file that you will be creating. The asterisk indicates that the project has been modified and the information has not been saved. To make sure that you can save the file, select **Save** from the **File** menu (or press ctrl-s). FreePCB will write the file, and the asterisk will disappear.
- If you want to look at the project file, you can open **motor.fpc** with a text editor such as NotePad. You will see that there is an [options] section of the file containing the options that you set , along with some other default options. Most of the other sections of the file, such as [shapes], [parts], etc. will be empty.
- In the next section, we will start designing our PCB by importing the netlist file.

7.3 Importing the Netlist File

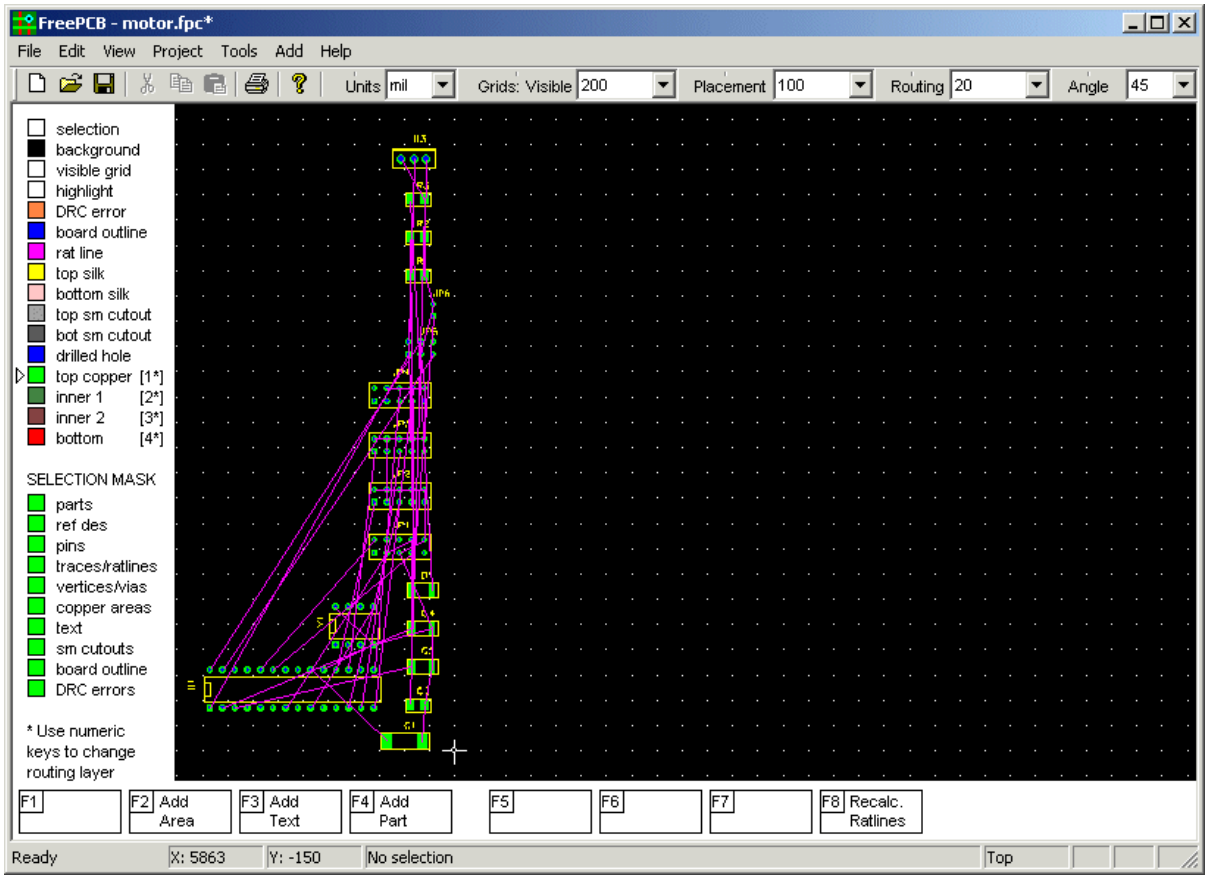
The netlist file that we will be using is listed in **Section 5.14: Importing Netlist Files**. You might want to review that section before continuing. The package for U2 was intentionally left incorrect, so we will have to fix it after the file is imported.

- Select **Import** from the **File** menu. A dialog called **Open** will pop up.

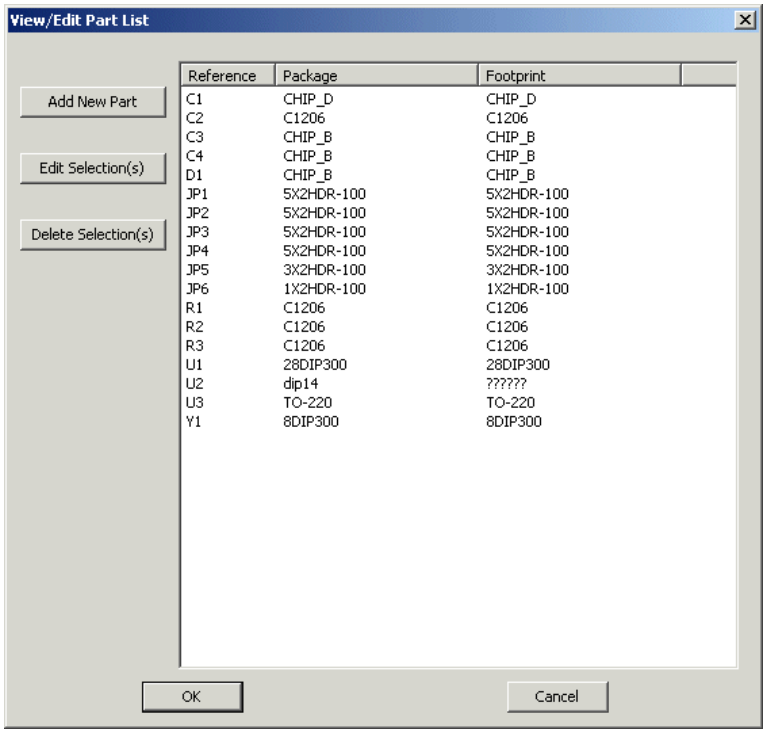


- If necessary, navigate to the "motor" project folder. Select file **motor.net**.
- Select **Parts and Nets** in the **Import** section of the dialog.
- Select **PADS-PCB** in the **Format** section of the dialog.
- Click the **Open** button.
- FreePCB will open a **Log** window and import the netlist, describing its progress in the **Log**. You should see a message indicating that a footprint was not found for U2. Click **OK** to dismiss the **Log**.
- All of the imported parts will be stacked up just to the left of the origin symbol in the layout window.

- Select **View > All Parts** (or press the "Home" key) to see them. Connections between the pins will be shown as ratlines.

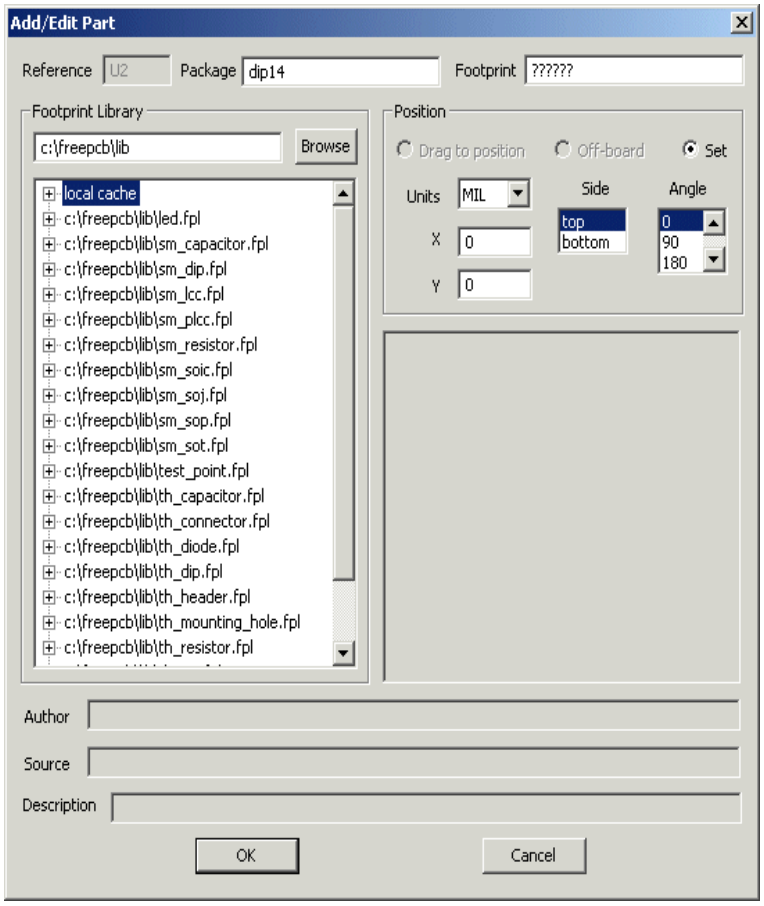


- Now we will fix the missing footprint. Select **Parts...** from the Project menu. The **View/Edit Part List** dialog should appear:

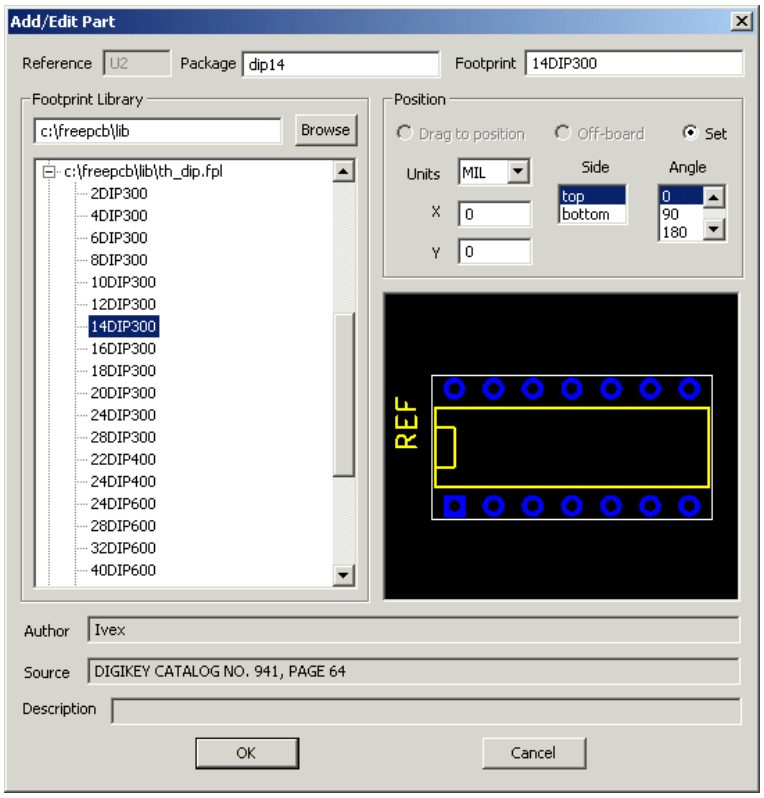


Reference	Package	Footprint
C1	CHIP_D	CHIP_D
C2	C1206	C1206
C3	CHIP_B	CHIP_B
C4	CHIP_B	CHIP_B
D1	CHIP_B	CHIP_B
JP1	5X2HDR-100	5X2HDR-100
JP2	5X2HDR-100	5X2HDR-100
JP3	5X2HDR-100	5X2HDR-100
JP4	5X2HDR-100	5X2HDR-100
JP5	3X2HDR-100	3X2HDR-100
JP6	1X2HDR-100	1X2HDR-100
R1	C1206	C1206
R2	C1206	C1206
R3	C1206	C1206
U1	28DIP300	28DIP300
U2	dip14	??????
U3	TO-220	TO-220
Y1	8DIP300	8DIP300

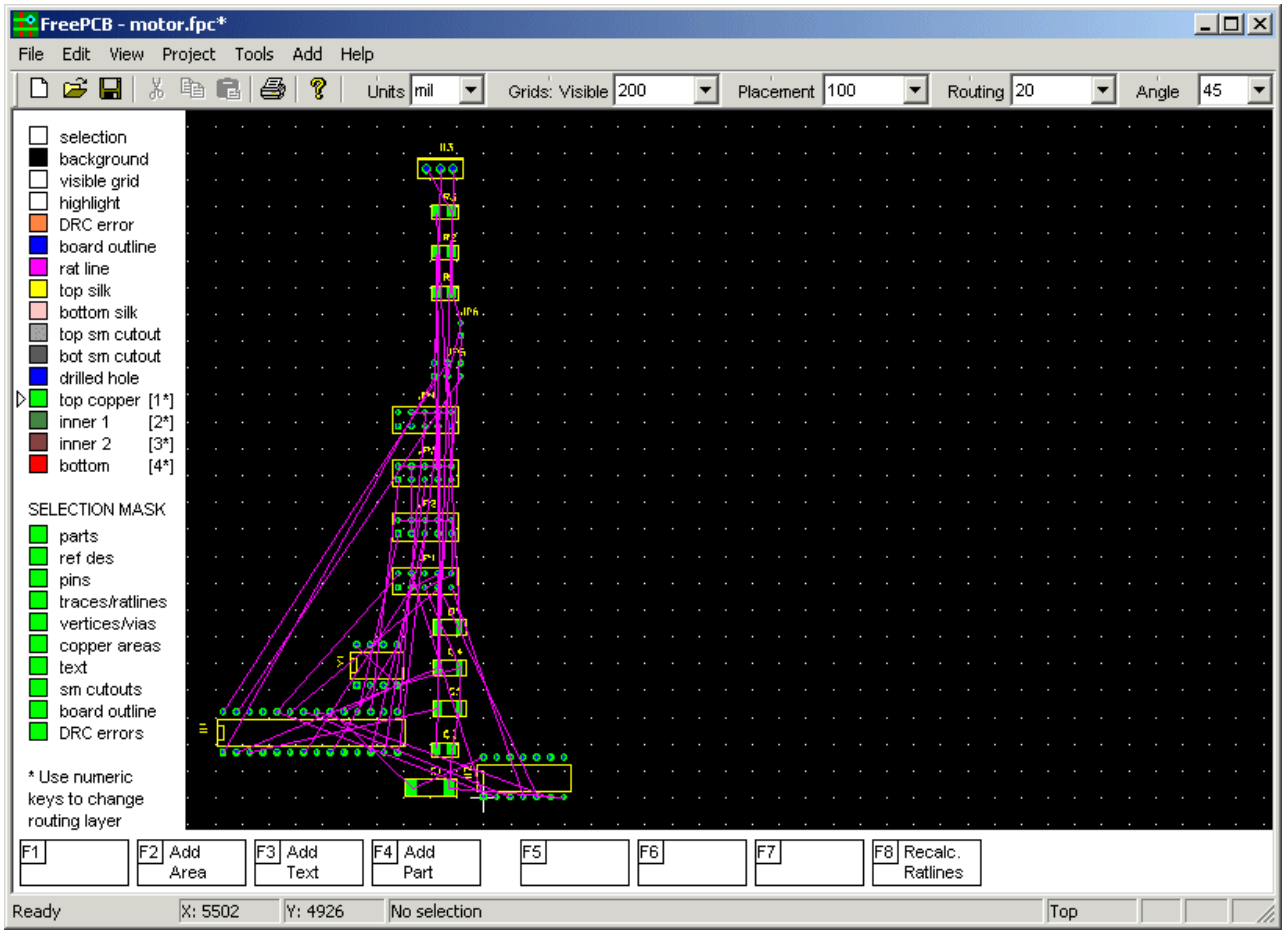
- This dialog shows a list of all of the parts which were imported, with their reference designators, package identifiers and footprints.
- Note that the footprint for U2 is shown as "??????", indicating that the package identifier "dip14" does not match any of FreePCB's footprints. We need to fix this.
- Select the line for U2 by clicking on it.
- Click the **Edit Selection(s)** button. The **Add/Edit Part** dialog should appear as shown below.



- This dialog allows you to edit the parameters for part U2, including its footprint. All of the available footprint libraries are shown in the large tree control occupying most of the dialog. The first entry is "local cache", which is not actually a library but which contains all of the footprints which are currently loaded into FreePCB.
- Expand the "th_dip.fpl" library by clicking on the "+" next to it. This library contains footprints for DIP packages. Its name begins with "th_" to indicate that it contains through-hole footprints.
- Select "14DIP300" from the library by clicking on it. An image of the footprint will appear in the preview window, and "14DIP300" should replace "??????" in the **Footprint** field. Now the dialog should look like like:



- Click on **OK**.
- A message box should appear asking whether you want to replace all instances of "dip14" with "14DIP300", or just U2. Since there is only one instance, you can click on either **YES** or **NO**.
- This should return you to the **View/Edit Partlist** dialog, with "??????" replaced with "14DIP300" for part U2.
- Click **OK**. U2 should now have a footprint, placed at position (0,0), as shown below.

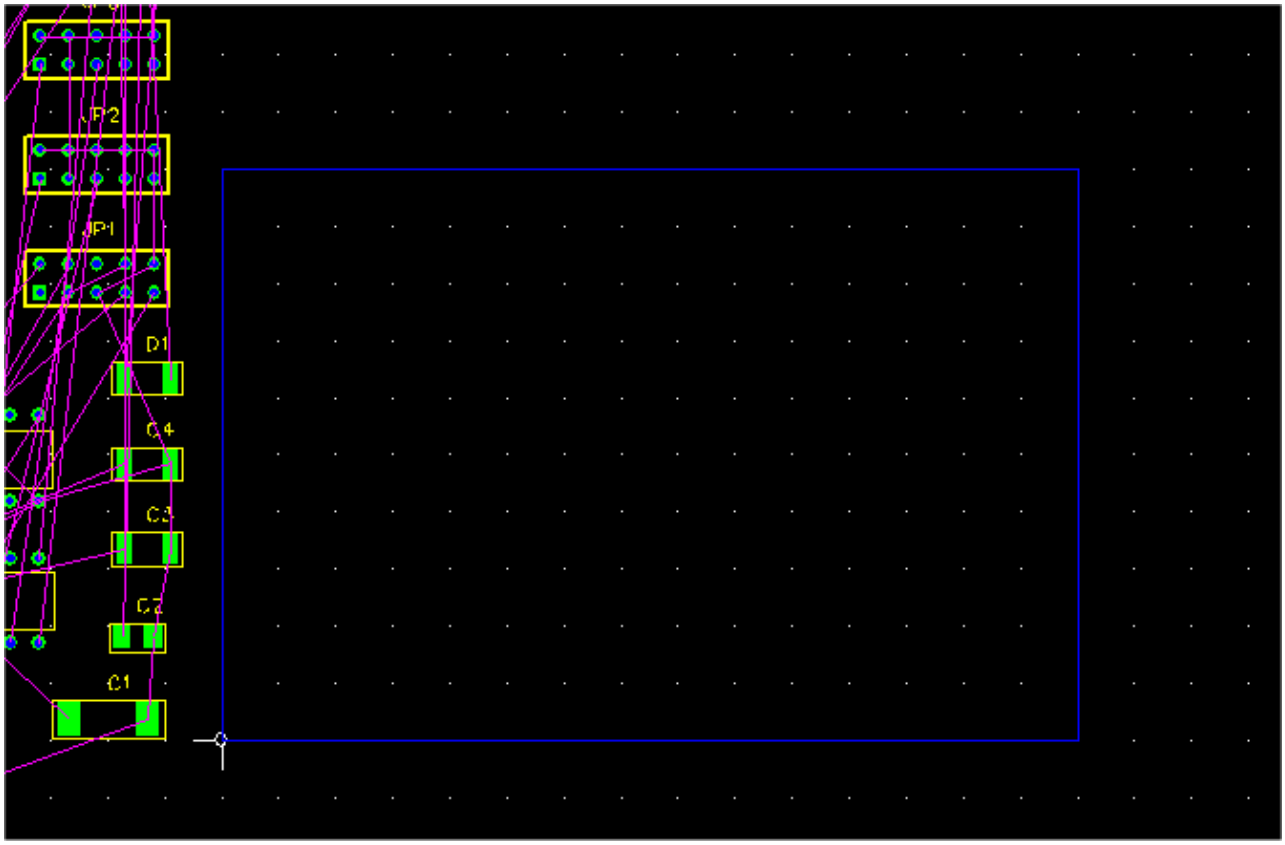


- OK, now all of the footprints are loaded. Save your work. In the next section, we will create the board outline.

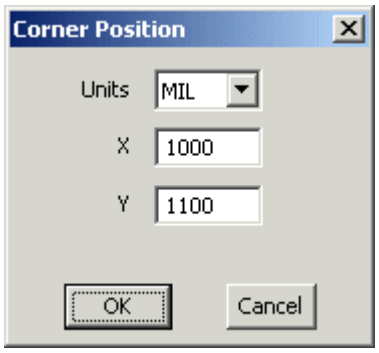
7.4 Drawing the Board Outline

In this section, we will create the board outline for our PCB. It will be a rectangular board, 3 inches long by 2 inches high. You might want to review **Section 5.10: Board Outline** before starting.

- First, let's move U2 out of the way. Zoom in on the board origin by placing the cursor over it and scrolling forward with the mouse (or by pressing PgUp). Select U2 by clicking on it. A white outline should appear around it, indicating that it has been selected. Press F4 to begin moving it, and drag it down and to the left of the origin to make room for the board outline. Scroll the window as needed with the scroll wheel or by pressing "space". (By the way, a window describing all of the keyboard shortcuts is available in **Help > Keyboard shortcuts**.)
- Left-click the mouse to place U2 at its new position.
- When drawing the board outline, the **Placement** grid will be in effect. Set it to a reasonably large number like 100 or 200 mils.
- Select **Board Outline** from the **Add** menu. The cursor should change to a cross-hair. Place the cross-hair at the origin (i.e. X=0, Y=0). Left-click to place the first corner of the outline.
- Now move the cursor vertically upwards.. You should be dragging a blue line segment, representing the left side of the board. Move the cursor to X=0, Y=2000 and left-click to place the upper-left corner of the board.
- In similar fashion place corners at X=3000, Y=2000 and X=3000, Y=0.
- Now right-click to close the board outline by drawing a side from the last corner to the first one. Your board outline should look like:



- You can check the position of a corner by clicking on it to select it. A small white square should appear around the corner to indicate that it has been selected. Then press F1 ("Set Position") to pop up the **Corner Position** dialog. This displays the exact coordinates of the selected corner, and allows you to change them if you want. This dialog is very useful if you are designing a board with unusual dimensions.

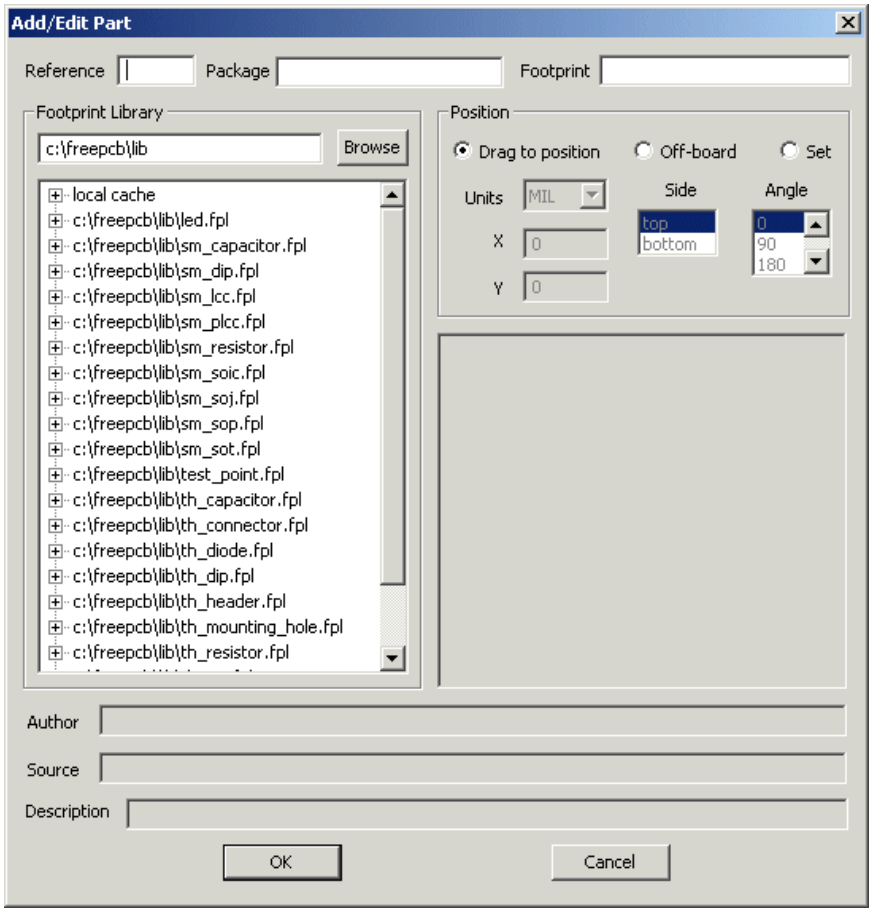


- If you wish, you can try out some of the other commands for editing the board outline, as described in **Section 5.10: Board Outline**. When you are finished, be sure that the outline has been restored to its original rectangular shape.
- **Save** the project from the **File** menu.

7.5 Adding Mounting Holes

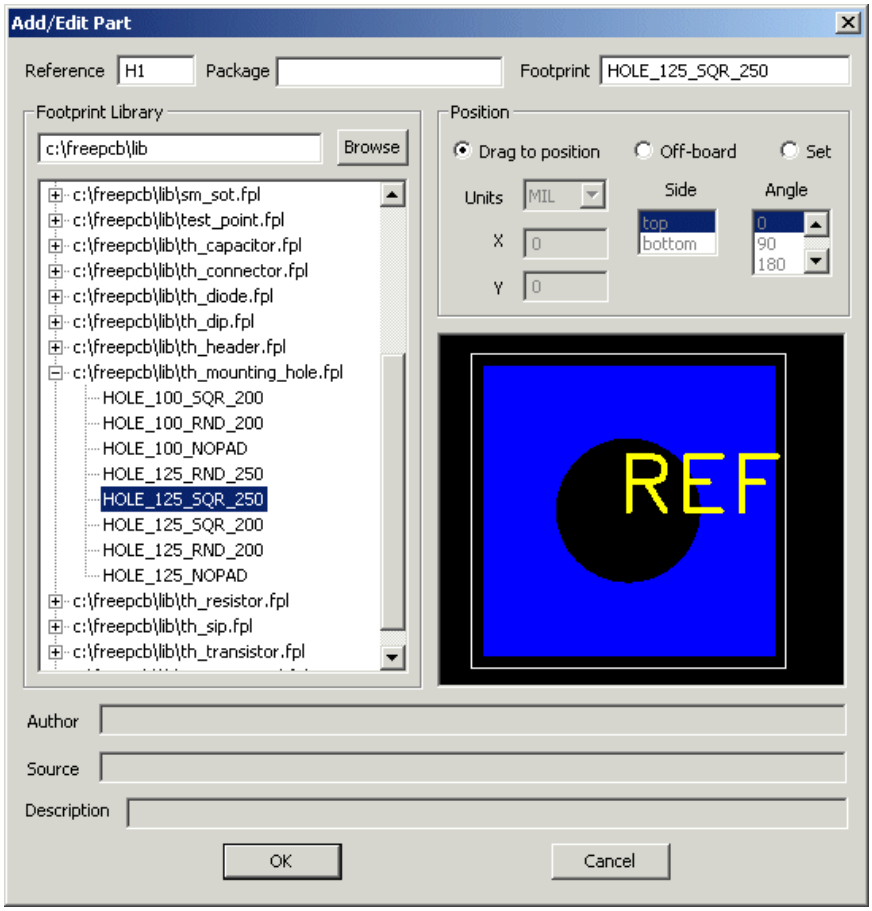
Now we will add mounting holes at the corners of the PCB. You might want to review **Section 5.12: Mounting Holes** before starting.

- Mounting holes are actually a special type of part, consisting of a single through-hole pin. They are added to a project like any other part. They can be included in the netlist file, or they can be added later. Since our netlist file did not include the mounting holes, we will add them using the **Add > Part** menu.
- Set the placement grid to a small number such as 25 mils.
- Select **Part** from the **Add** menu. The following dialog should pop up

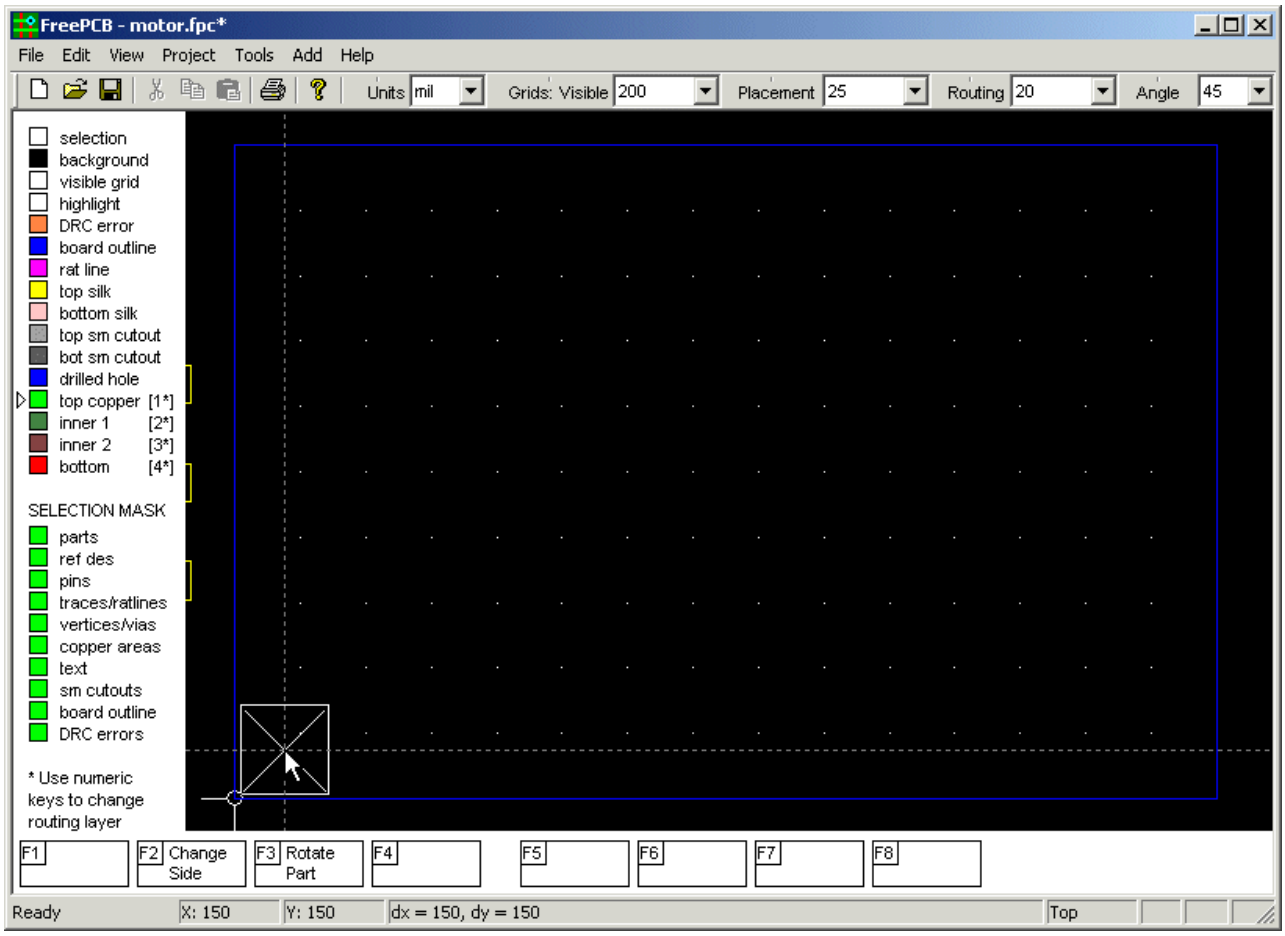


- Enter "H1" in the **Reference** field of the dialog. This will be the reference designator for the first mounting hole.
- Expand the **C:\FreePCB\lib\th_mounting_hole.fpl** library file by clicking on the "+" next to it. Then click on "HOLE_125_SQUARE_250". This is the footprint for a mounting hole 125 mils in diameter, with a square pad 250 mils in diameter. It is about the right size for a #4 machine screw.

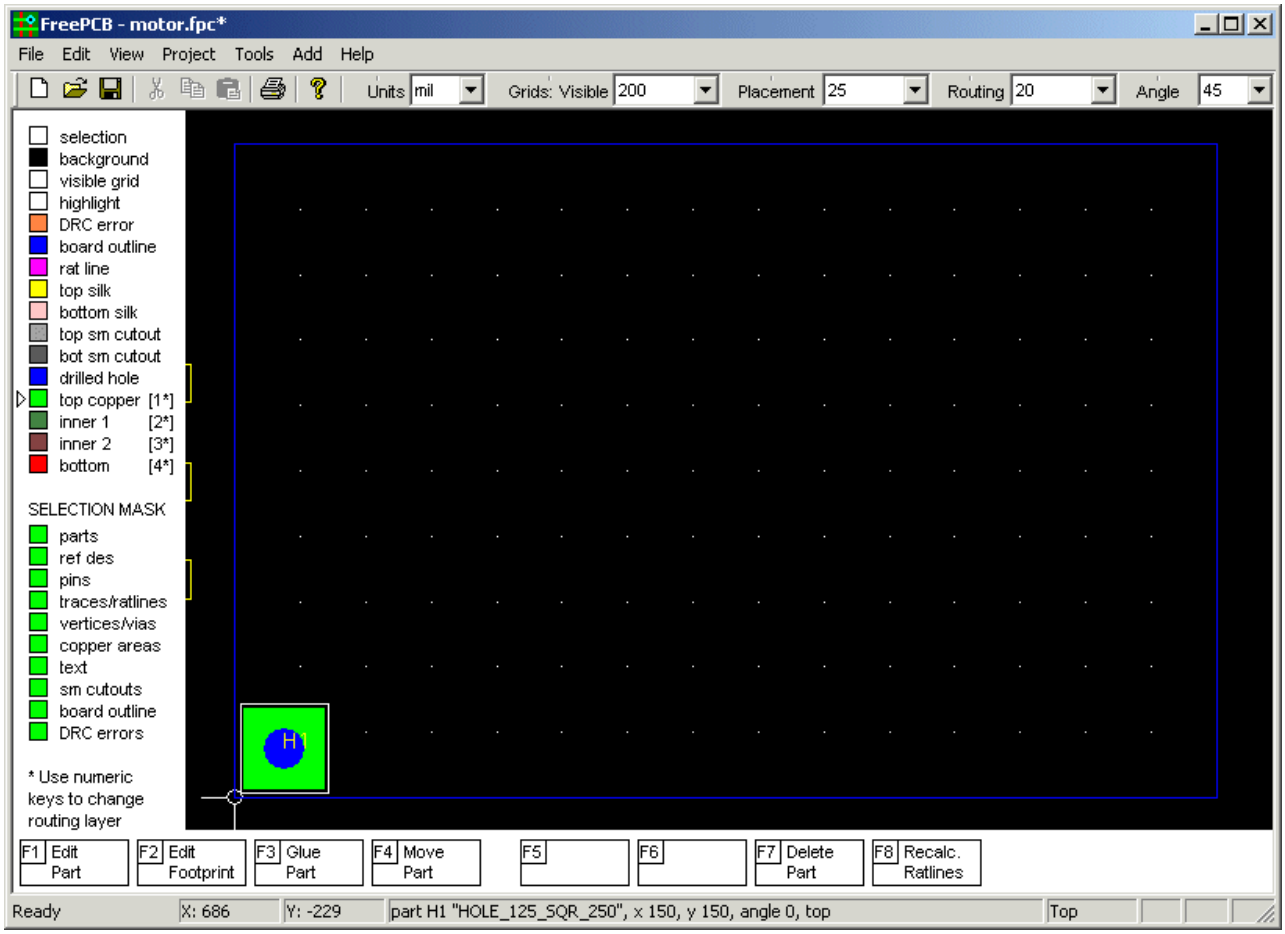
- Now the dialog should look like:



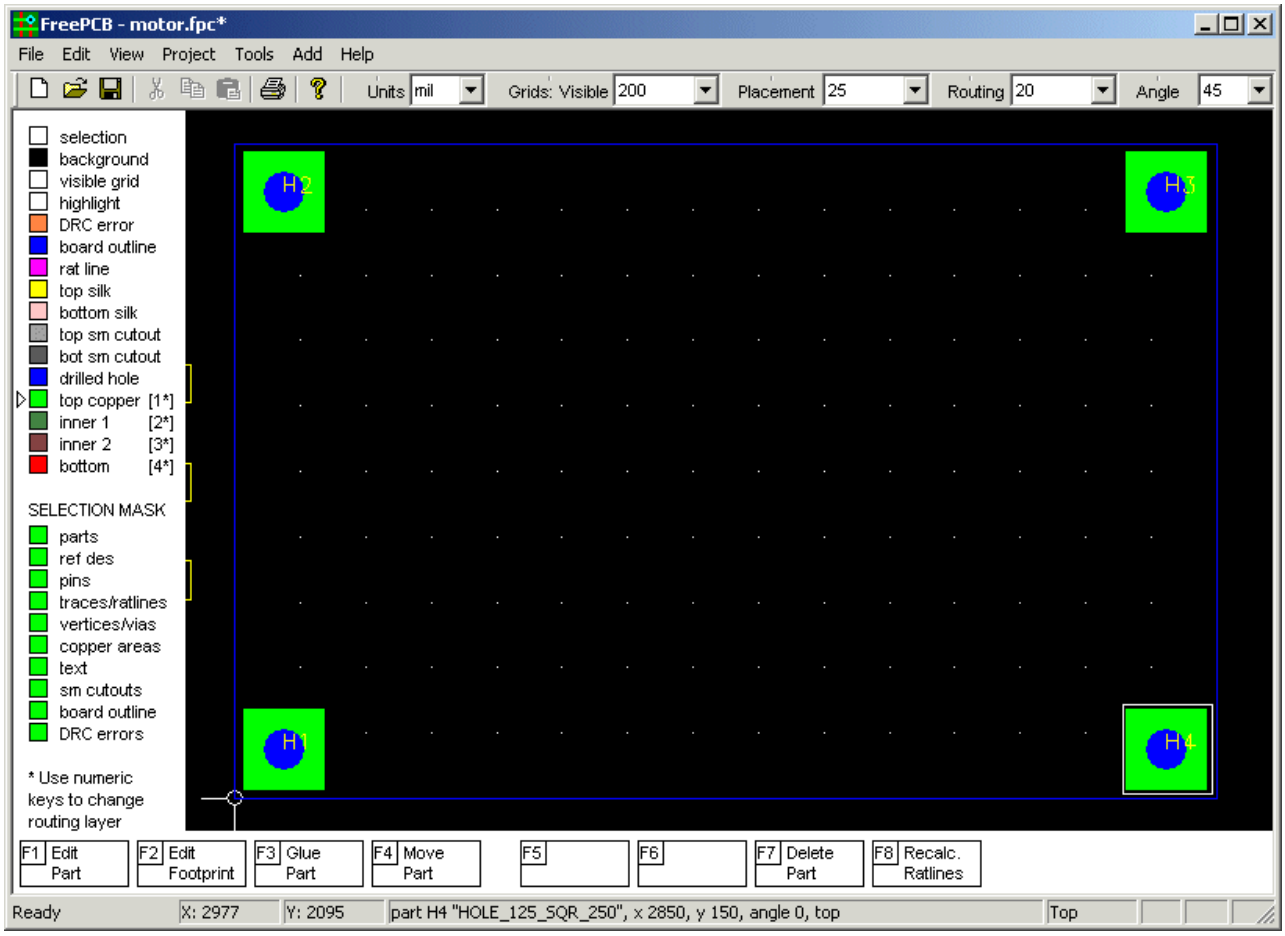
- Click **OK** to close the dialog and begin dragging the mounting hole. Move it to X = 150, Y = 150 as shown below.



- Click the left mouse button to place the hole. Now the layout should look like

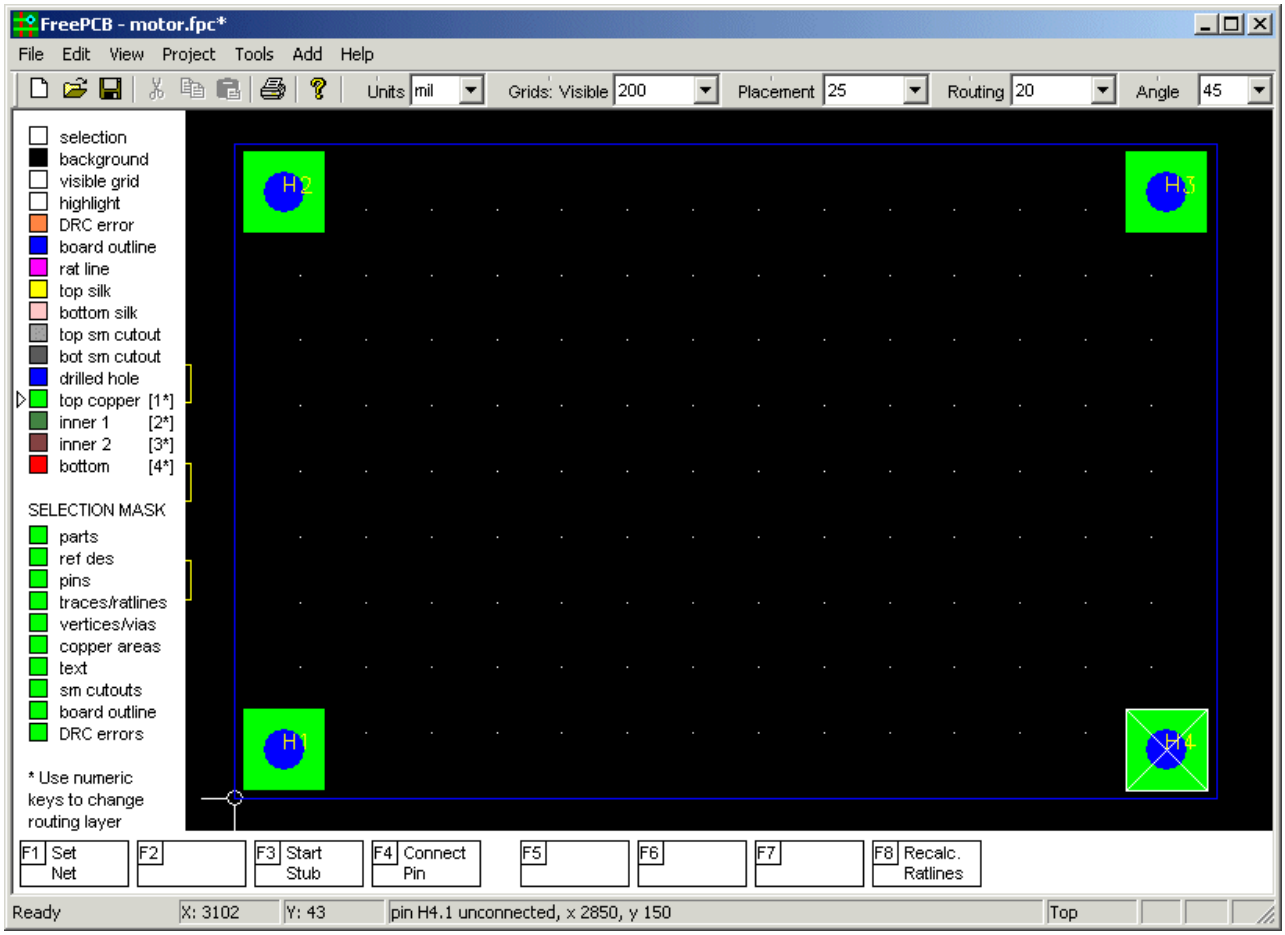


- In similar fashion, add 3 more mounting holes at the other corners of the PCB, using reference designators H2, H3 and H4, as shown below. Note that when you use the **Add > Part** dialog to add these holes, it will come up initialized with the last footprint that you added, with the reference designator incremented by one, so you only have to click OK or press "return" to add the next part.

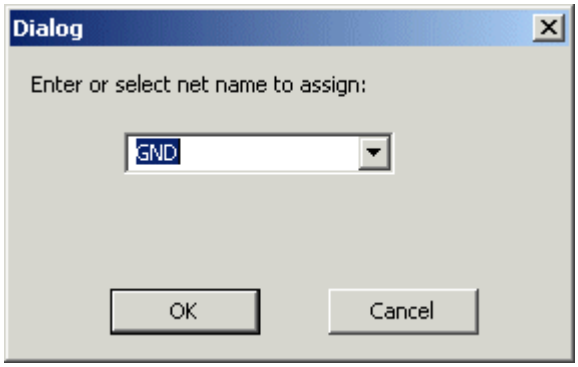


- Now we will connect H4 to the GND net, so that it can be used to ground the PCB to its enclosure. To do this, we must select the **pin** H4.1, instead of the **part** H4, since connections are always made to pins .Notice that in the screen shot above, the selection rectangle around H4 does not have an "X" through it, indicating that the part is selected and not the pin. The status bar also shows that part H4 has been selected.

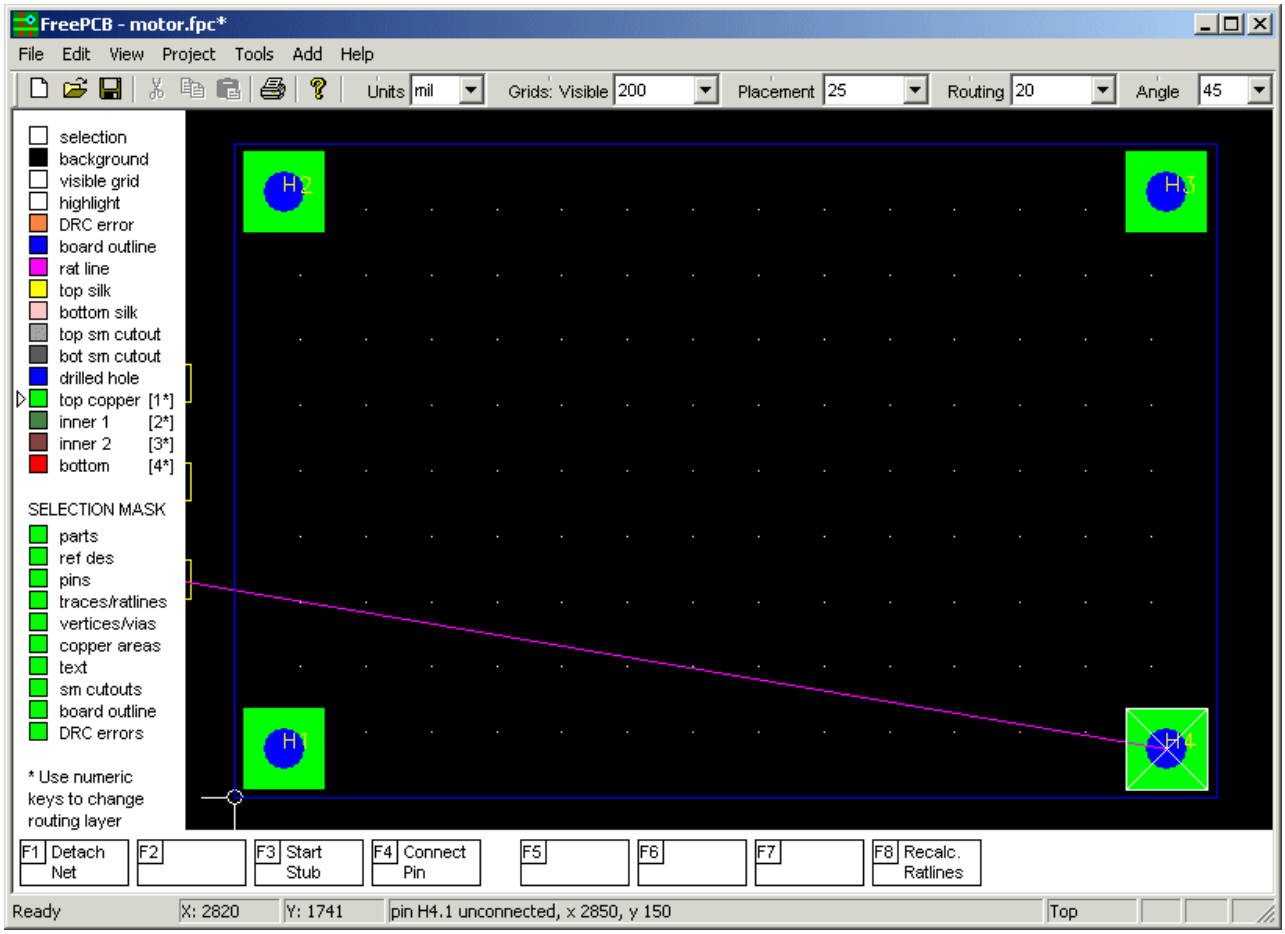
- To select the pin for H4, click anywhere on the pad. Now the selection rectangle should contain an "X", and the status bar should indicate that pin H4.1 has been selected, as shown below. If you wanted to reselect the part (to move it, for example) you would click again on the pad.



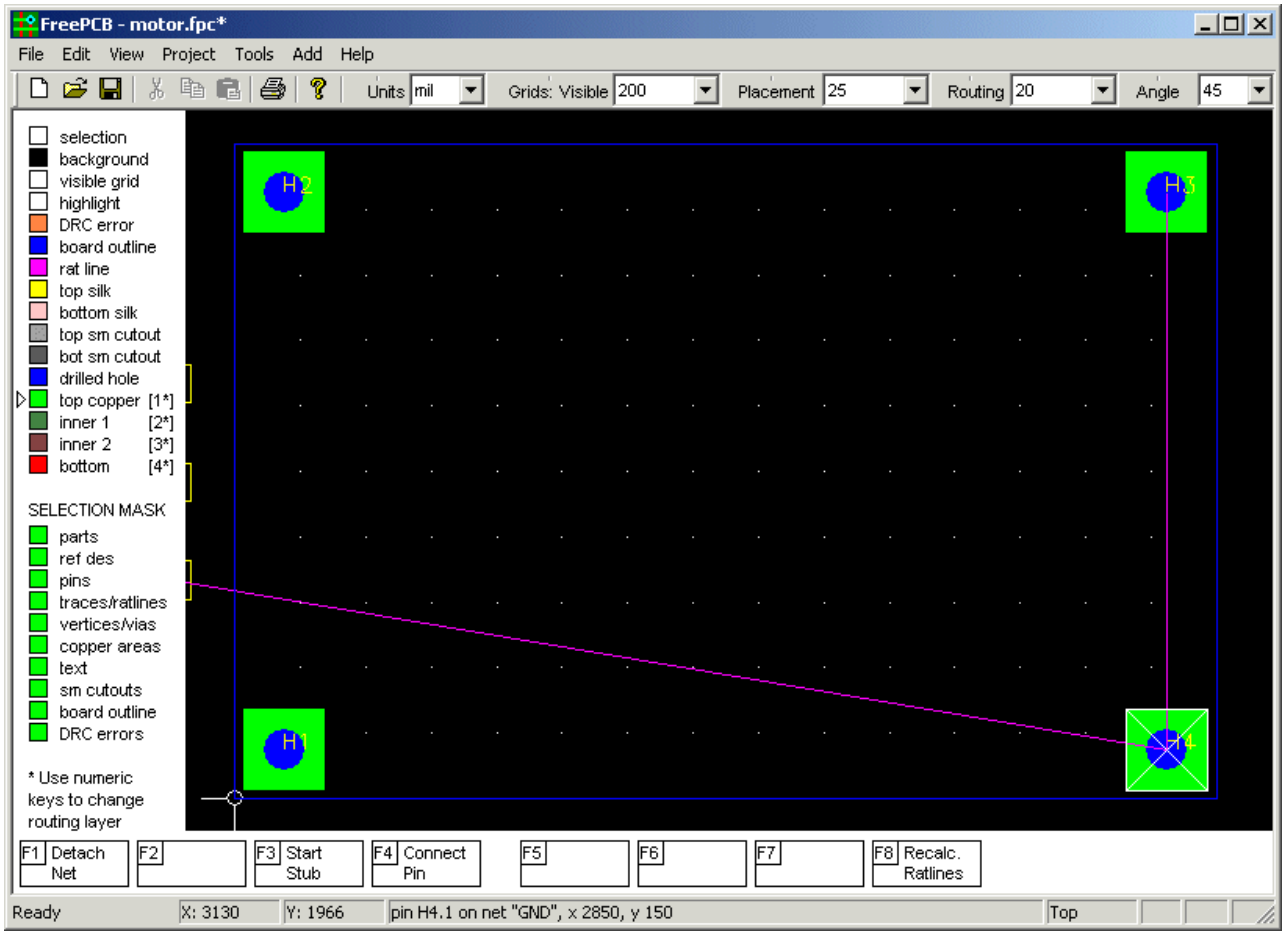
- With the pin selected, press F1 ("Set Net"), which pops up the following dialog. Use the drop-down menu to select the GND net, as shown.



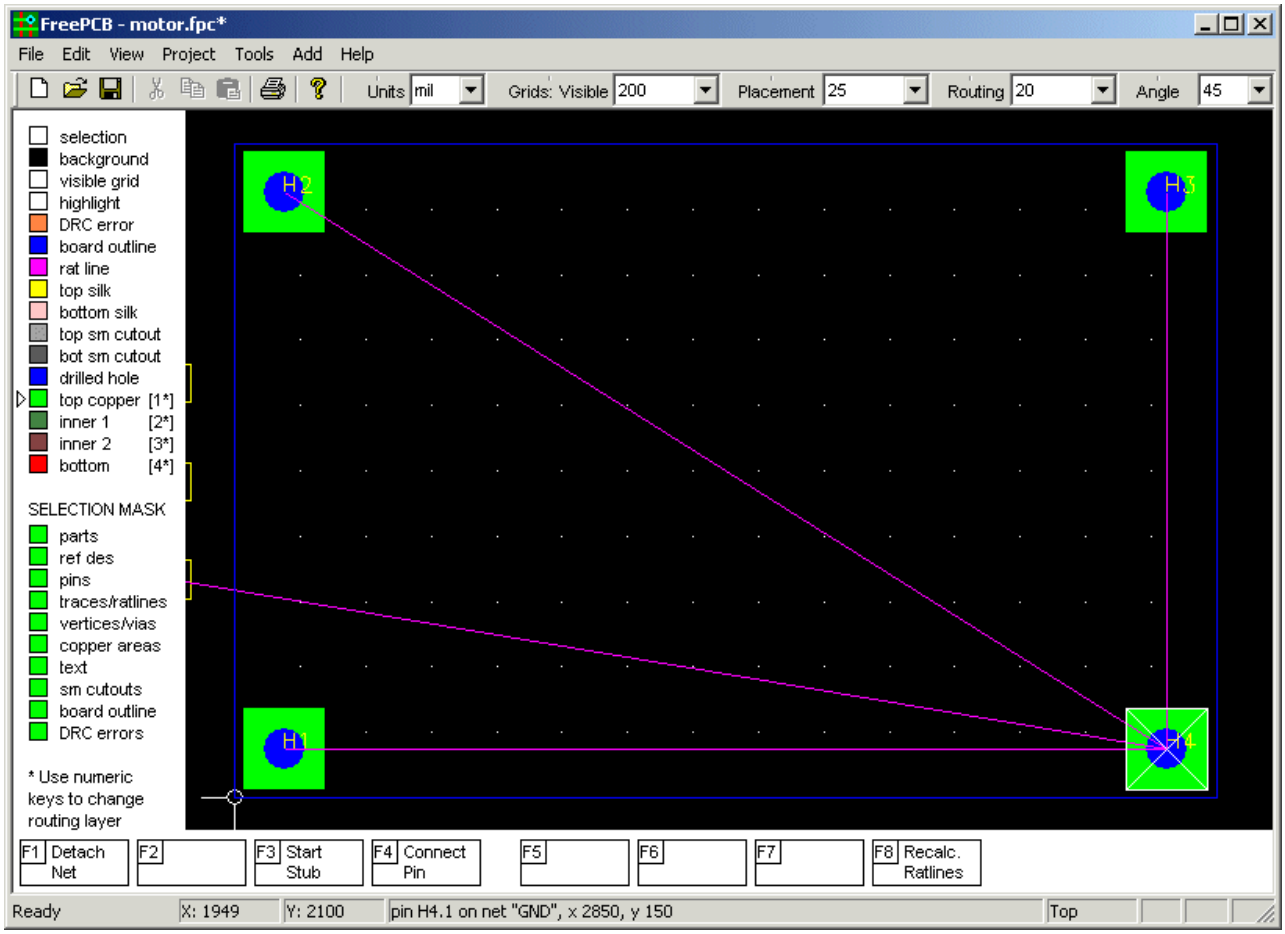
- Click **OK** to assign H4.1 to the GND net. A ratline to the pin should appear.



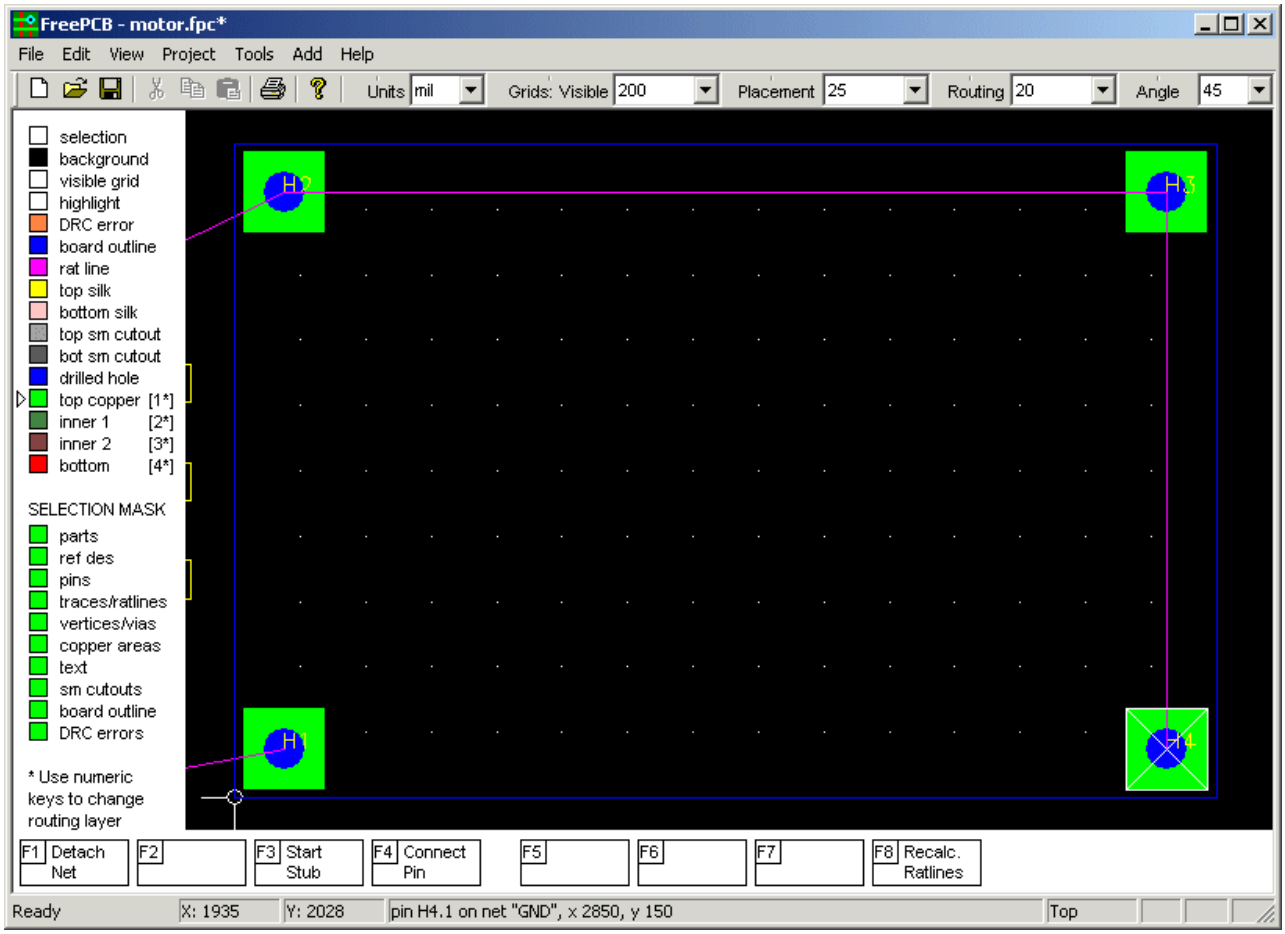
- Now we will connect H3.1 to GND, by drawing a ratline from H4.1 to H3.1. With the pin H4.1 still selected, press F4 ("connect Pin"). Now you will be dragging a ratline from H4.1. Place the end of the ratline over H3 and left-click to connect H3.1 to H4.1, as shown below.



- In similar fashion, connect H4 to H1 and H2.



- Now press F8 ("Recalc ratlines") to redraw the ratlines to minimize their total length.

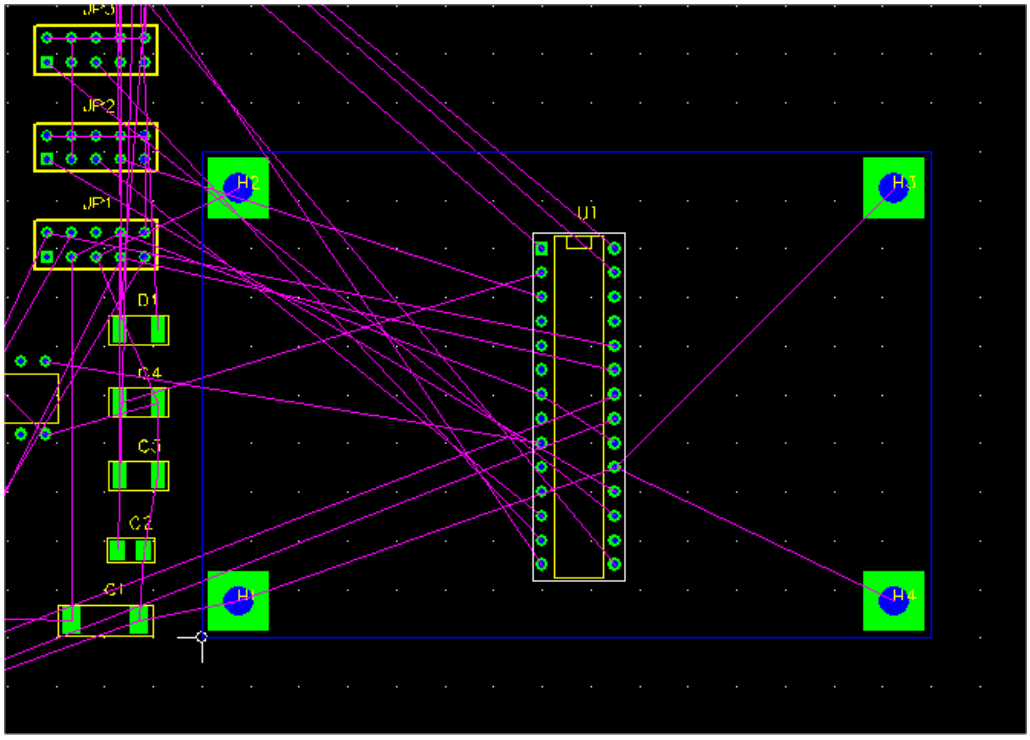


7.6 Placing Parts

In this section, we will place the parts on our PCB.

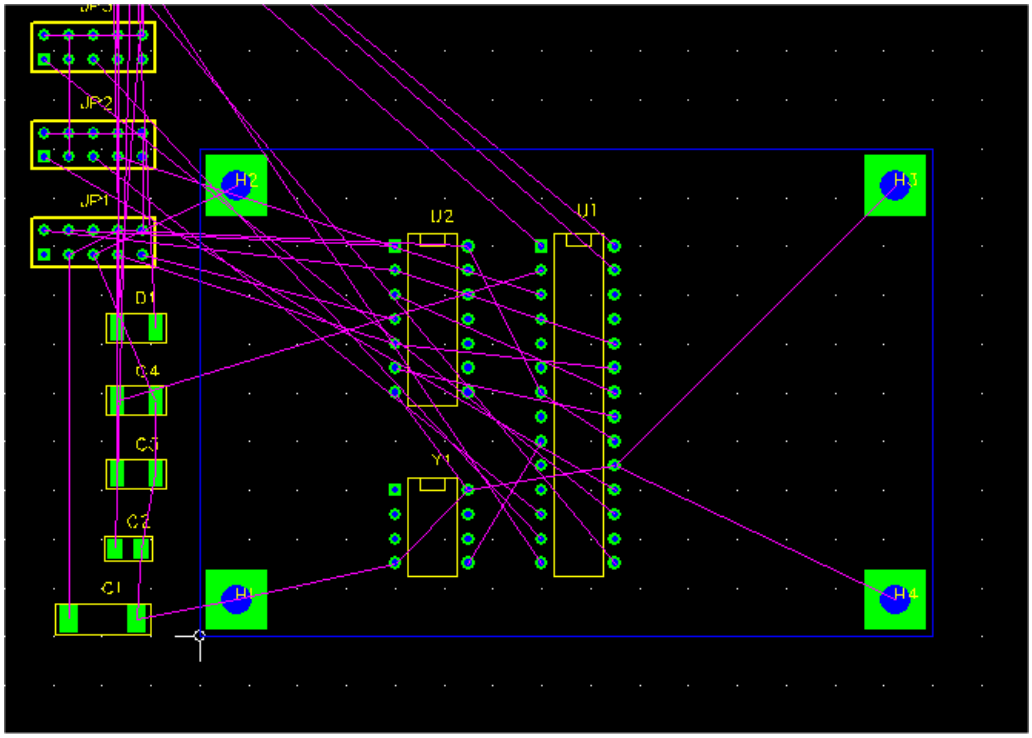
- Set the **Placement** grid to a reasonable value such as 100 mils for placing the through-hole parts, which have a pin spacing of 100 mils. This will make it easier to align the parts. Later, we may use a smaller grid for the surface-mount parts.
- Select **Show all** from the **View** menu (or press the "Home" key) so that you can see all of the parts. It will help if you make the FreePCB window as large as possible (although then you may have trouble seeing this tutorial, unless you have dual monitors).
- Before placing any parts, take a minute to review the schematic. If you haven't already done it, it may be useful to print the schematic so that you can refer to it while designing the PCB. Since U1 is the largest and most "connected" part, I would suggest placing it somewhere near the center of the board, with the other IC's near it. The headers JP1-JP6 should probably be placed near the edges.
- Let's start by placing U1. Select it by clicking on it. A white outline should appear around the part, indicating that it has been selected. Be careful not to click on one of the pads or the reference designator, or you will select that instead of the entire part. If you are having trouble finding U1, you can use **View > Show part** to select it by reference designator.
- **Tip:** Sometimes it can be difficult to select a small part that has relatively large pads, such as a surface-mount capacitor. However, here is a trick that you can use. FreePCB will never select something that is already selected, if there is an alternative. Therefore, if you click on a part and select a pad instead (or any overlapping structure), just click again and you will usually get the part. You have to be sure to allow enough time between clicks that Windows doesn't mistake your two clicks for a double-click, which usually doesn't do anything.

- With U1 selected, press F4 to begin moving it. Drag it towards the center of the board. The cursor will be "attached" to pin 1 of the part. Rotate it by pressing F3 once, so that it is vertical with pin 1 in the upper left corner. When it is near the center of the board, left-click to place it. It should look something like:

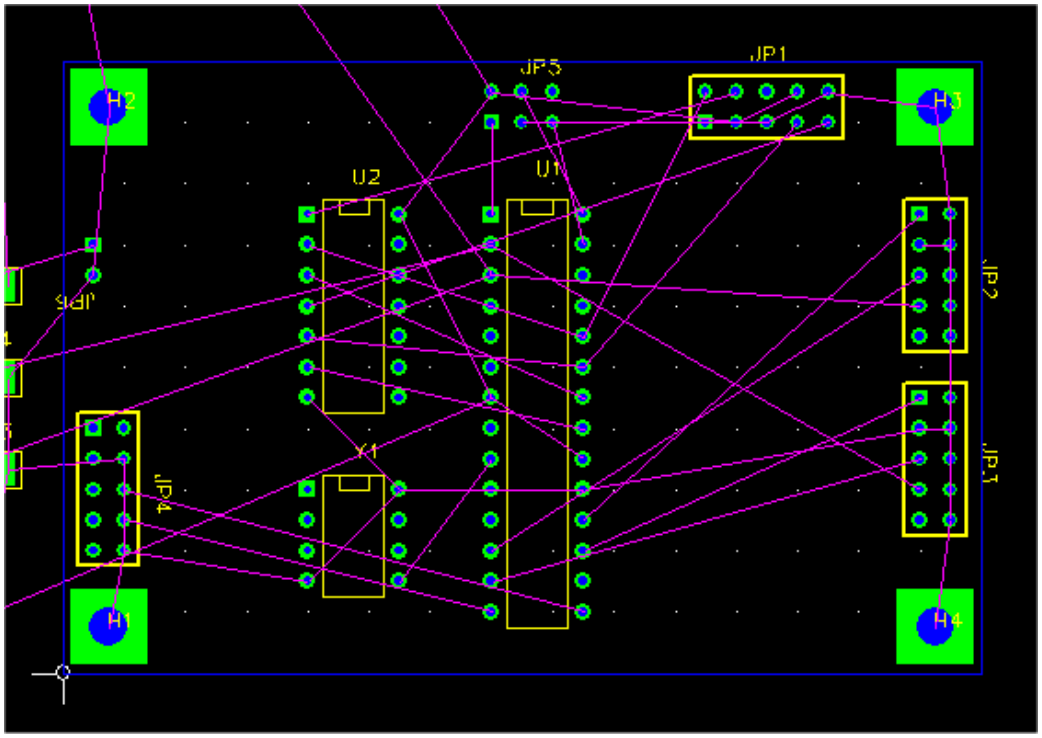


- Notice that the ratlines moved with the part. After you placed it, you might have noticed that some of the ratlines changed their pin connections. This is because FreePCB automatically optimizes the ratlines for minimum total length after moving a part.

- Now let's place the other DIP ICs, U1 and Y1, something like this:

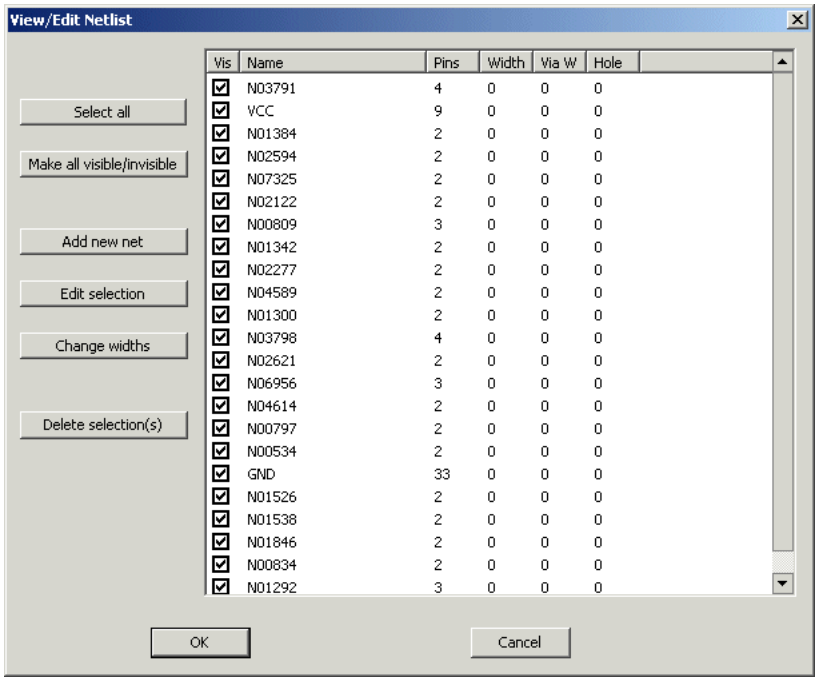


- Now place the DIP headers around the edges of the board, something like:

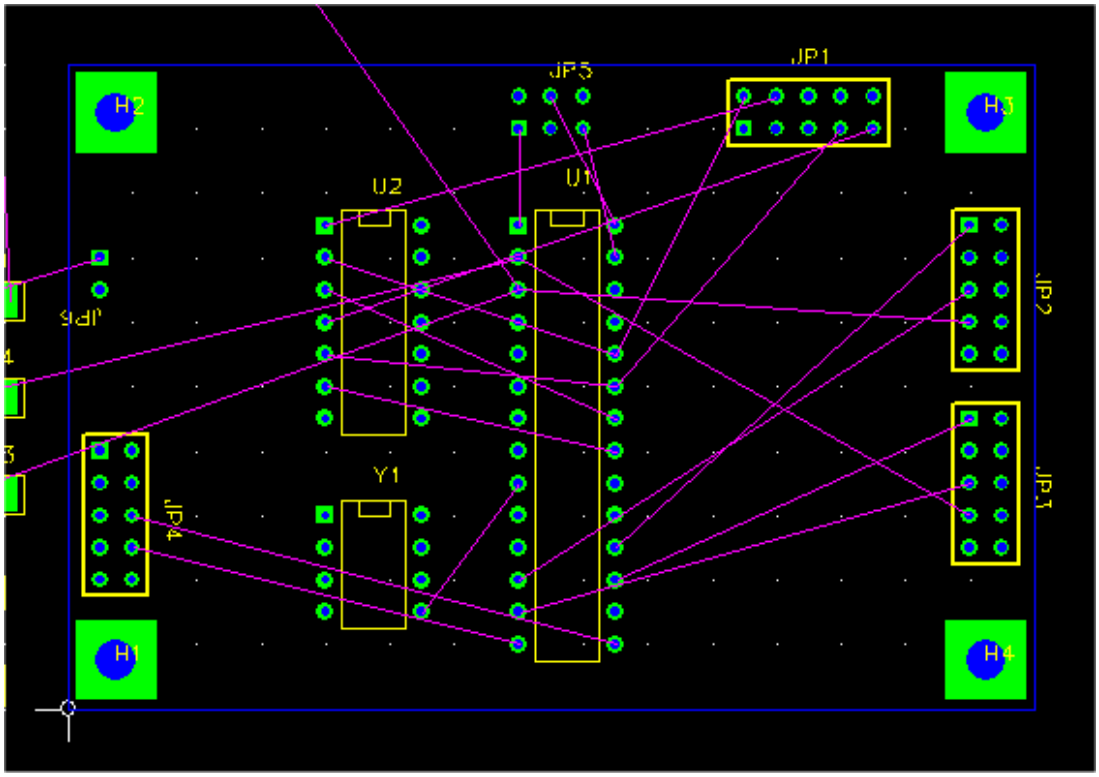


- As you place the parts, it is a good idea to look at the ratlines to try to find an optimal placement for easy routing. However, you don't necessarily have to see at ALL of the ratlines. On this board, we will be using internal power and ground planes for the VCC and GND nets. Therefore, we might as well make the ratlines for these nets invisible.

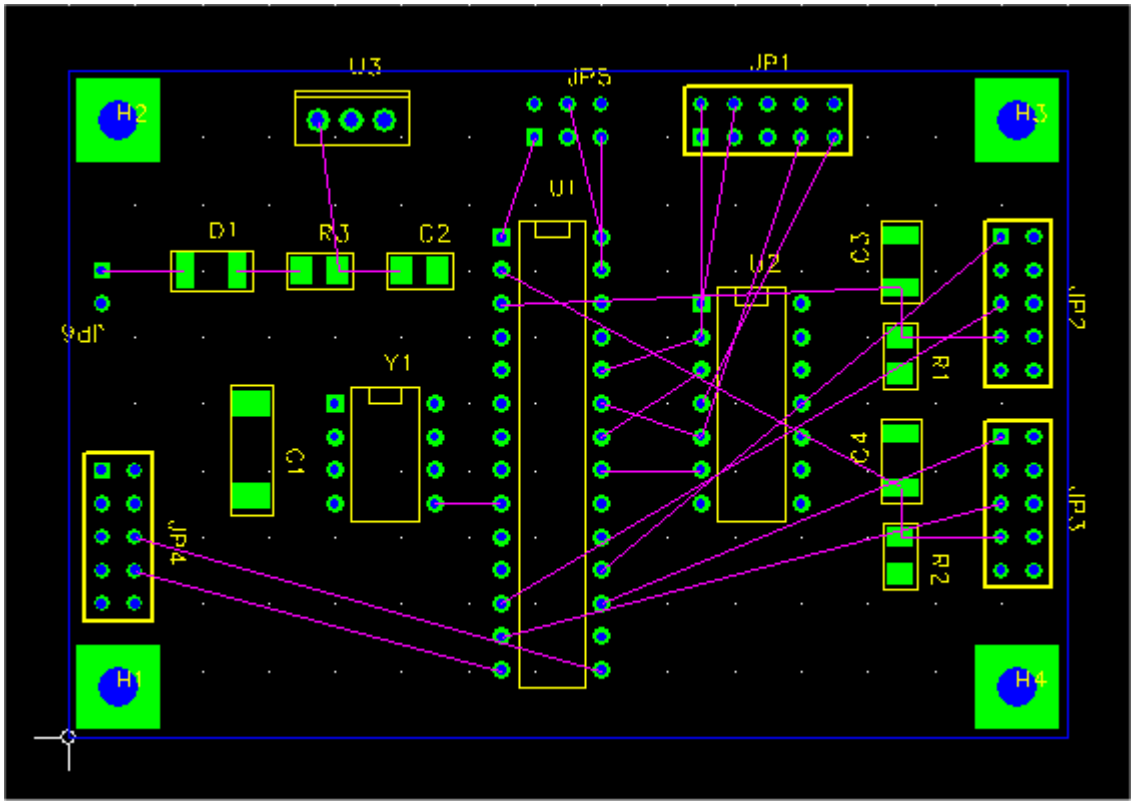
- Select **Nets...** from the **Project** menu. The following dialog should appear:



- Most of the dialog is occupied by a list of all of the nets. The number of pins and the trace and via widths for each net are shown next to the net name. Also, to the left of each name is a checkbox, which determines the visibility of the net ratlines. Currently, all nets are visible. To make the VCC and GND nets invisible, uncheck the boxes next to them. Then click **OK**. Now there are fewer ratlines visible.

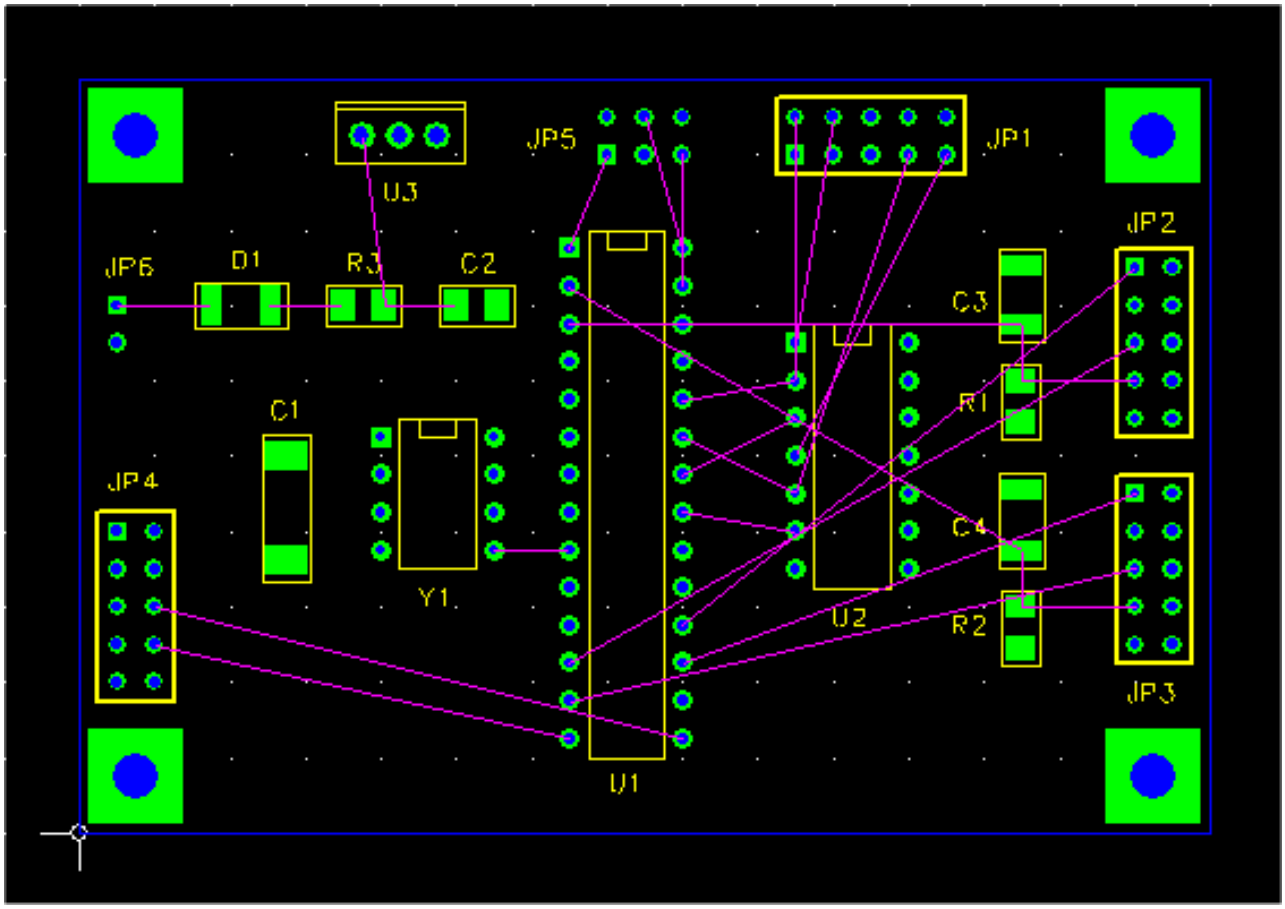


- Now place the rest of the parts. As mentioned above, selecting some of the smaller surface-mount capacitors may be tricky, unless you place the cursor precisely between the pads, or use the two-click method. You may want to use a smaller **Placement** grid, such as 50 mils. As you place each part, you may have to move the other parts around to make room or to improve the efficiency of the connections. Try not to crowd things too much. On a real-life densely populated board, the parts placement may be critical to successful routing, so it pays to take the time to get it right. That shouldn't be much of a problem on this board, however.
- You can see my final parts placement below. Note that I moved U2 to the right of U1 to shorten the ratlines between these parts.



- Now, let's position the reference designators for the parts on the silk-screen layer, so they will be correctly oriented and visible when the parts are soldered on. First, set the **Placement** grid to a small value, such as 10 mils, for finer control of the placement of these small items. Select each reference designator by clicking on it, and move it with the F4 key. As with parts, F3 rotates the reference designator while dragging.
- Since the reference designators for the mounting holes aren't very useful, you can make them invisible by selecting them and pressing F5 ("Set Size"), and then setting their size to zero. Since they are positioned directly over the pads for the mounting holes, you probably won't be able to select them by clicking on them, since you will get either the part or the pin instead. Therefore, use the **selection mask** to disable selection of parts and/or pins, and then click on the reference designator. Alternatively, with the mounting hole part selected, select **Set Ref. Text Size** from the right-click menu. This is also how you would make them visible again by increasing their size.

- Here is the final screenshot.

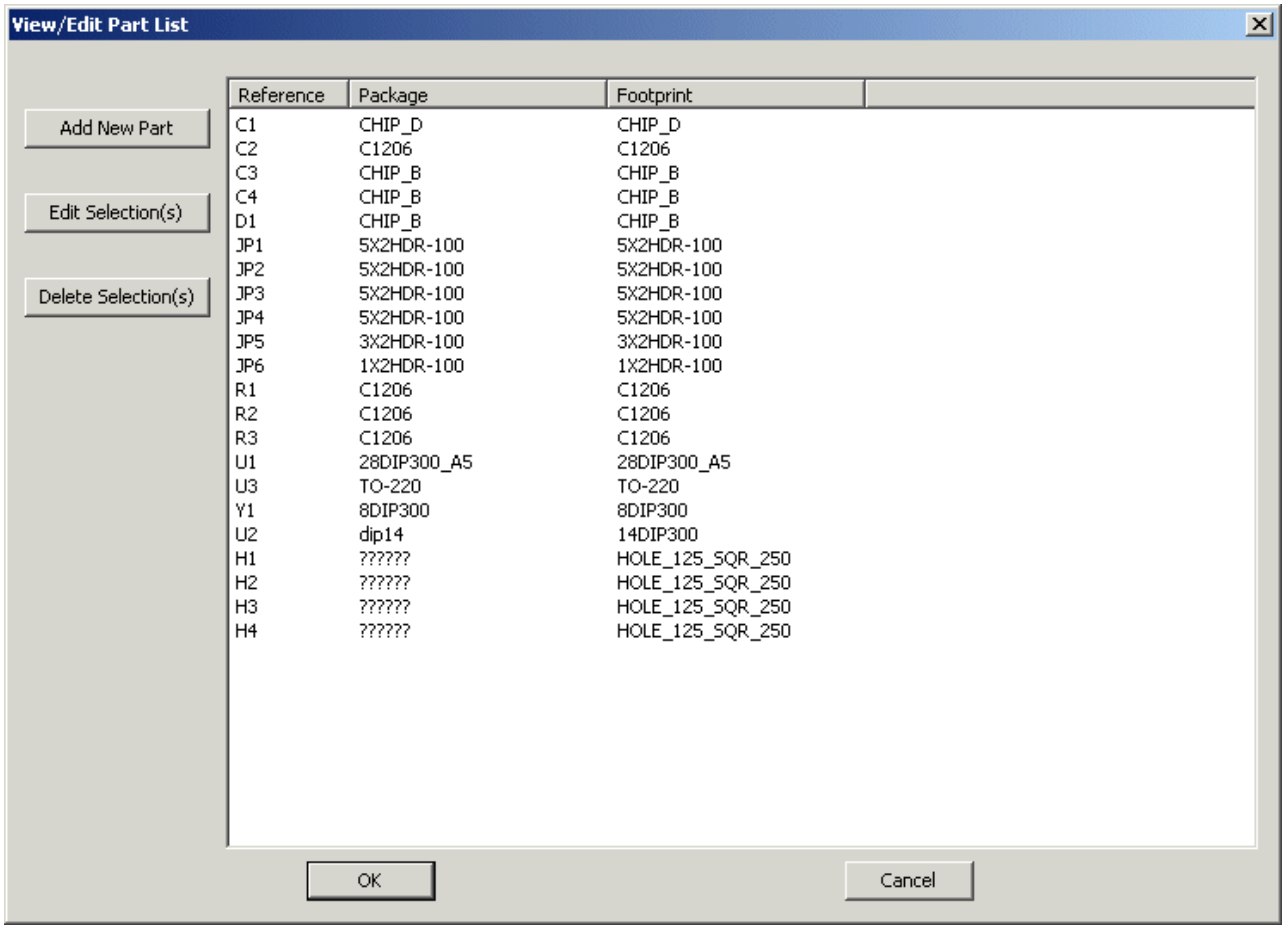


7.7 Adding Parts and Editing Nets

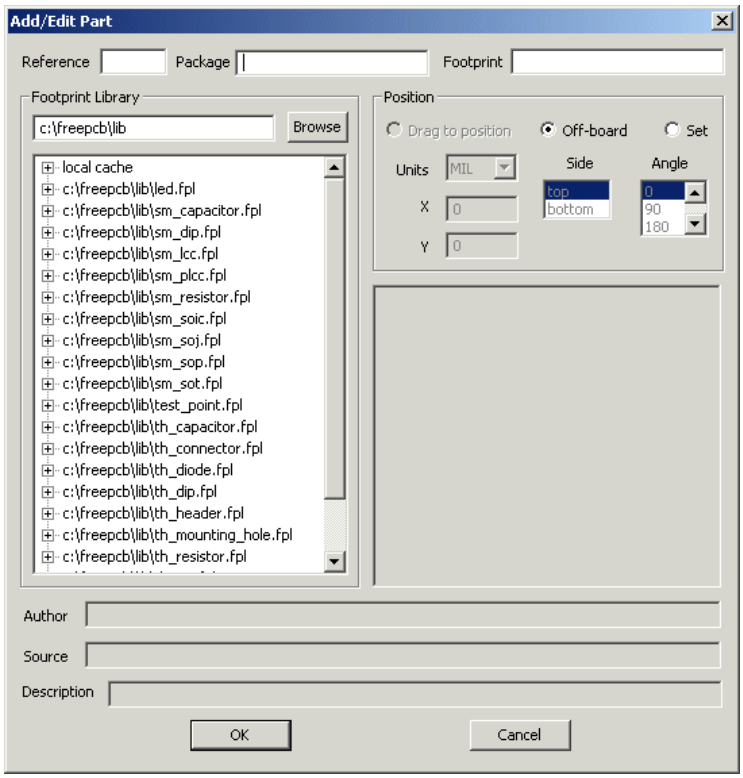
Modifying a PCB design by adding, replacing or deleting parts or changing the netlist is sometimes referred to as "**on-the-fly**" editing. FreePCB gives you virtually unlimited freedom for this type of editing. However, you should use this freedom carefully. Keep in mind that by modifying the partlist and netlist, you are creating a discrepancy between your PCB layout and the original schematic. If you make any significant changes you should carefully **back-annotate** them to the schematic, otherwise troubleshooting the PCB can be a nightmare. In general, "on-the-fly" editing is best used for relatively minor changes. If you need to perform more extensive redesign, you should probably go back to your schematic editor and create a new netlist file.

In this section, we will add three bypass capacitors to our PCB, one for each of the DIP ICs. We will connect them to the VCC and GND nets. Since FreePCB provides several methods for doing this, we will try them all.

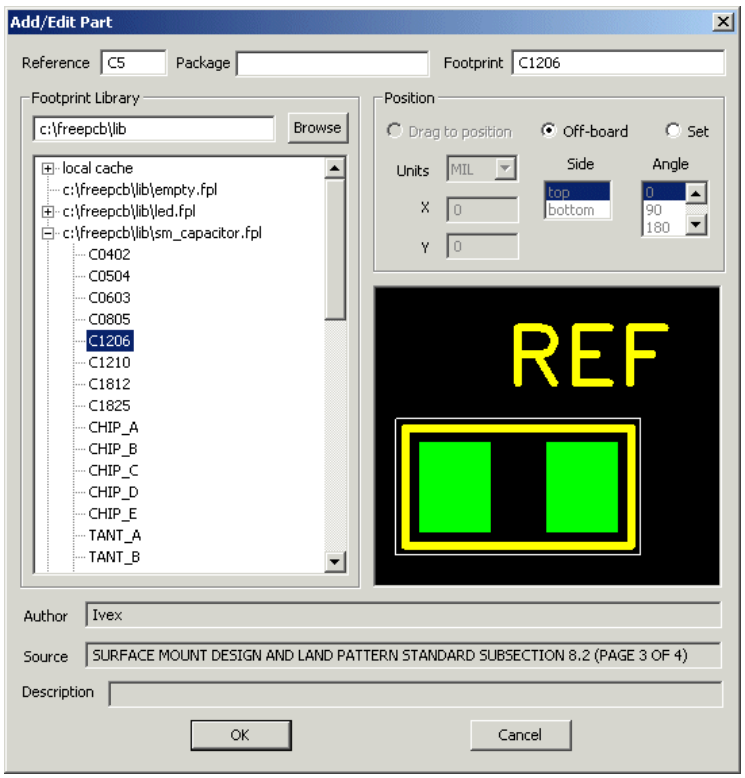
- Set the placement grid to a reasonable value for surface-mount capacitors such as 50 mils.
- Select **Parts...** from the **Project** menu. The **View/Edit Part List** dialog should appear.



- Note that the last capacitor in the part list is C4. We will add C5, C6 and C7.
- Click on **Add New Part**. This will bring up the **Add/Edit Part** dialog.

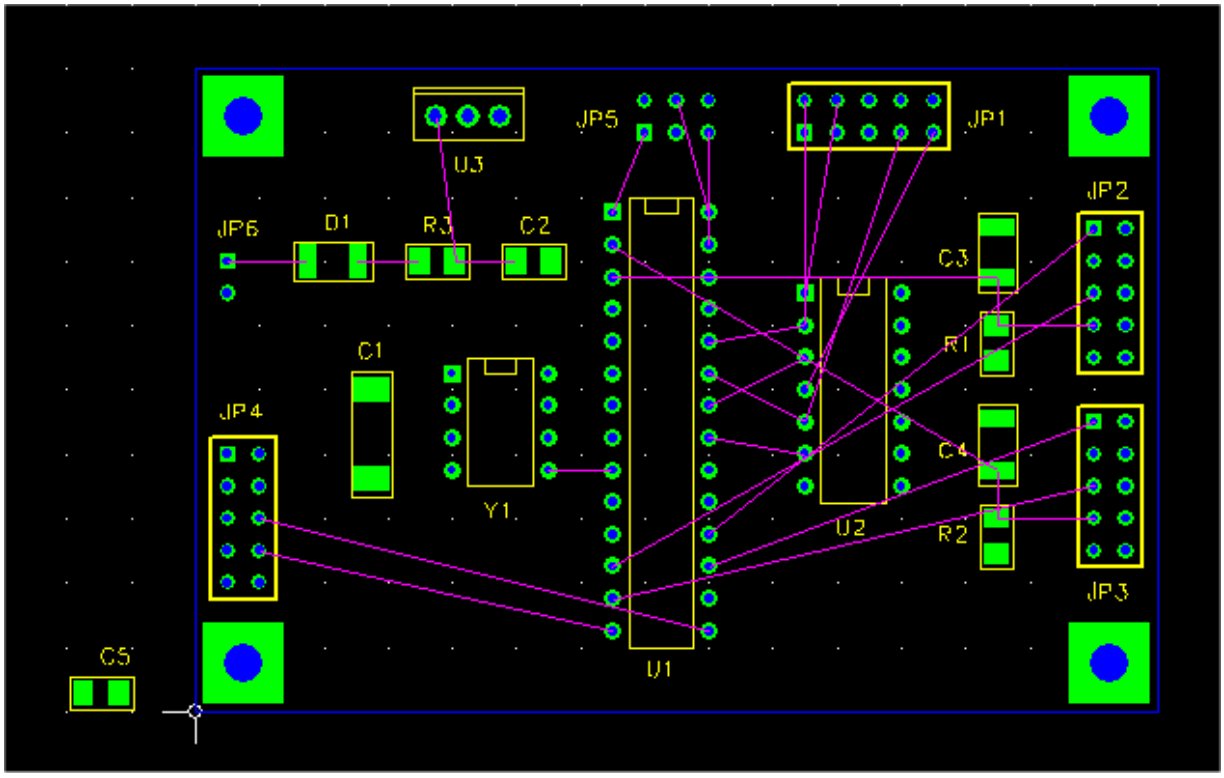


- Expand the library **sm_capacitor.fpl** by clicking on the "+" sign next to it.
- Select the footprint "C1206" by clicking on it. The **Footprint** field should change to "C1206".
- Enter "C5" in the **Reference** field. Now you should see:

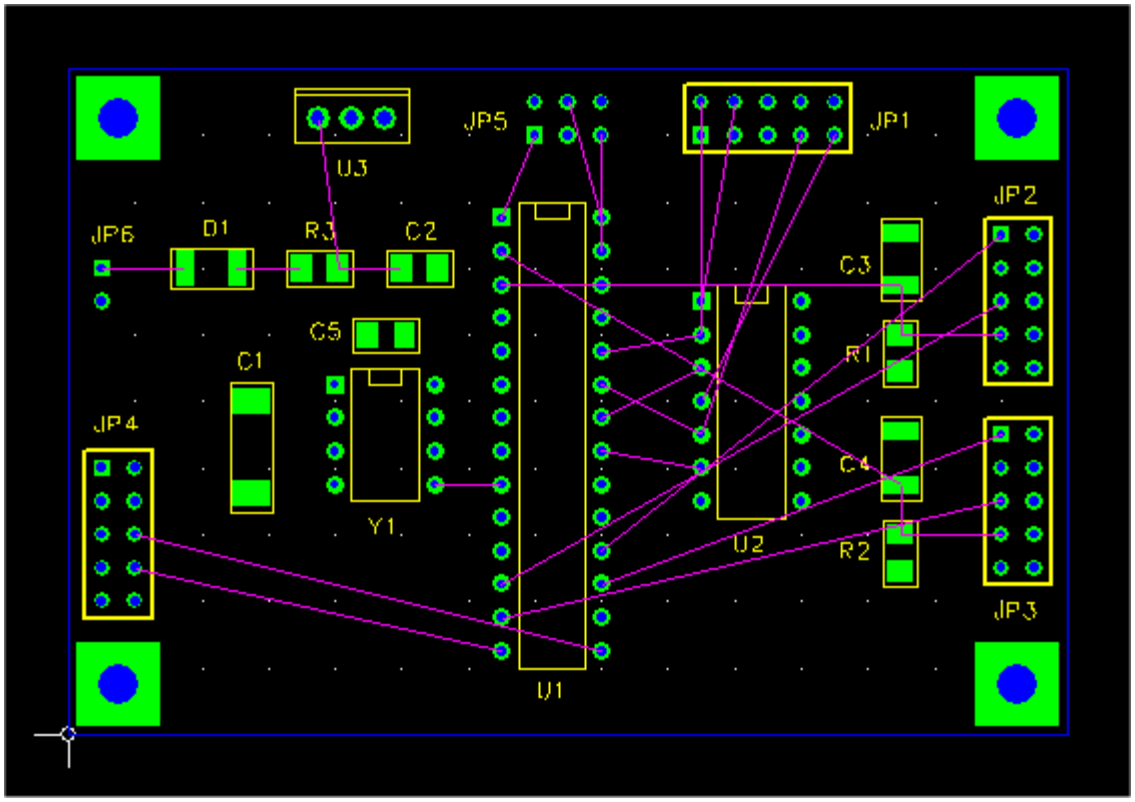


- Click **OK** to add the part to the partlist.

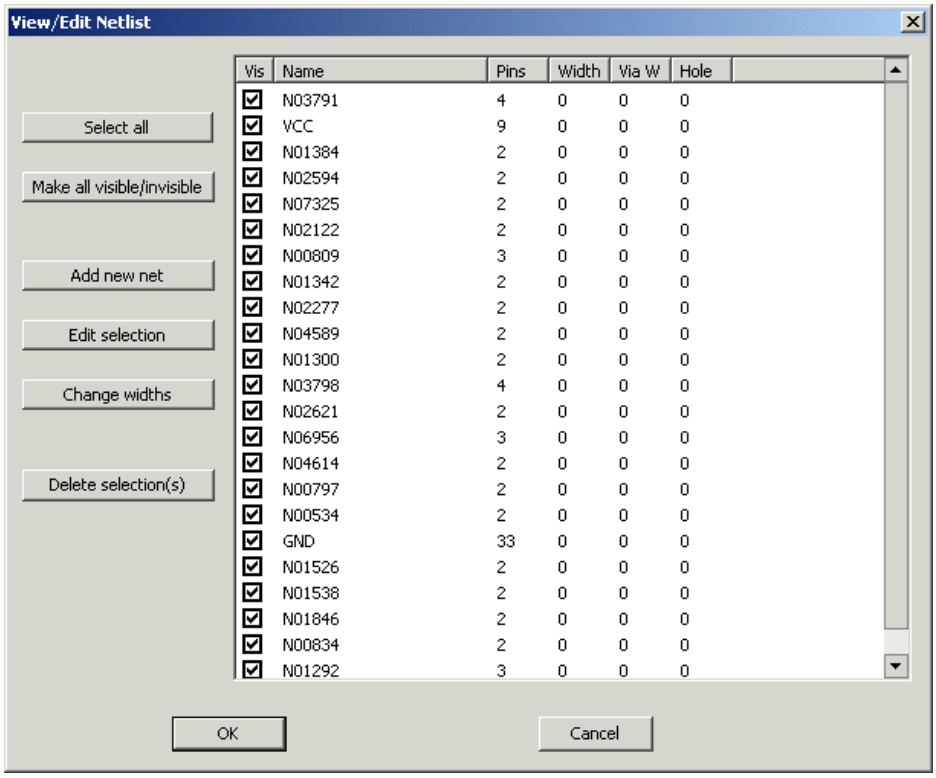
- Now you will be back in the **View/Edit Part List**dialog. Click **OK** to exit. Your new capacitor C5 will be visible just to the left of the origin.



- Select C5 by clicking on it, and move it near the top of Y1. Move the reference designator text if necessary so that it will be visible.

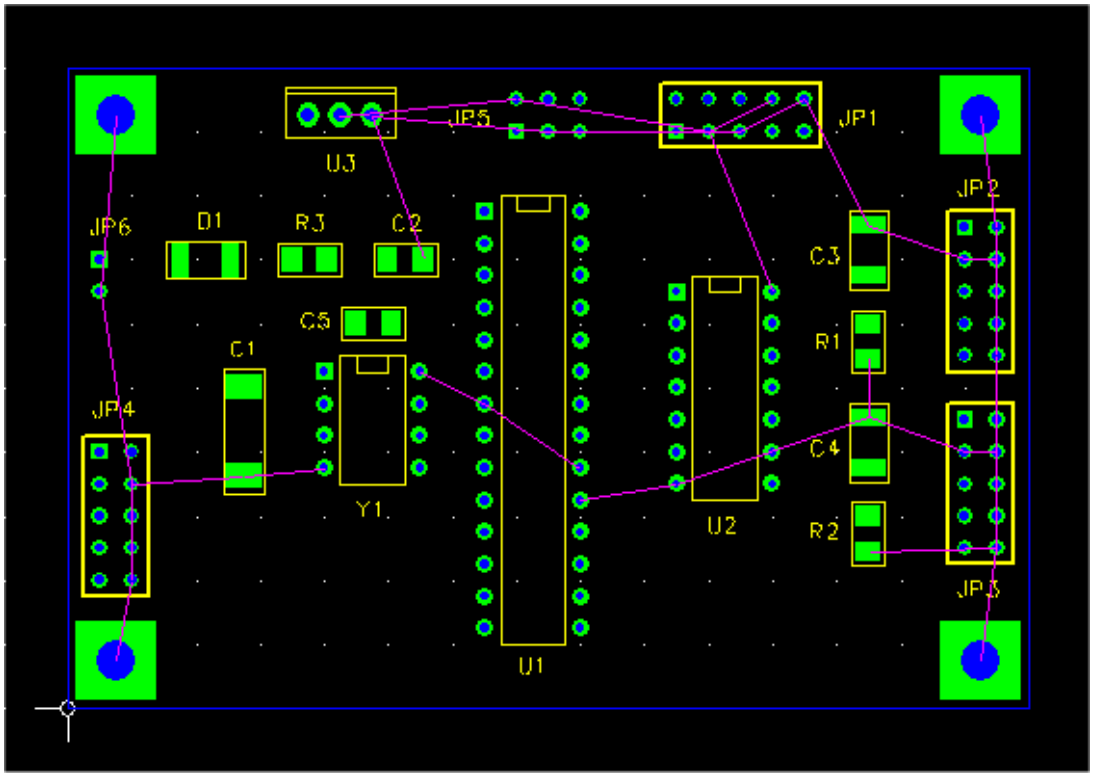


- Now we will connect the pins of C5. Lets start by making only the nets VCC and GND visible. Select **Project > Nets...** to bring up the **View/Edit Netlist** dialog.

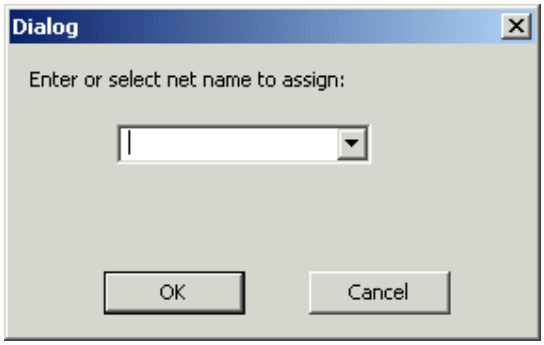


Vis	Name	Pins	Width	Via W	Hole
☑	N03791	4	0	0	0
☑	VCC	9	0	0	0
☑	N01384	2	0	0	0
☑	N02594	2	0	0	0
☑	N07325	2	0	0	0
☑	N02122	2	0	0	0
☑	N00809	3	0	0	0
☑	N01342	2	0	0	0
☑	N02277	2	0	0	0
☑	N04589	2	0	0	0
☑	N01300	2	0	0	0
☑	N03798	4	0	0	0
☑	N02621	2	0	0	0
☑	N06956	3	0	0	0
☑	N04614	2	0	0	0
☑	N00797	2	0	0	0
☑	N00534	2	0	0	0
☑	GND	33	0	0	0
☑	N01526	2	0	0	0
☑	N01538	2	0	0	0
☑	N01846	2	0	0	0
☑	N00834	2	0	0	0
☑	N01292	3	0	0	0

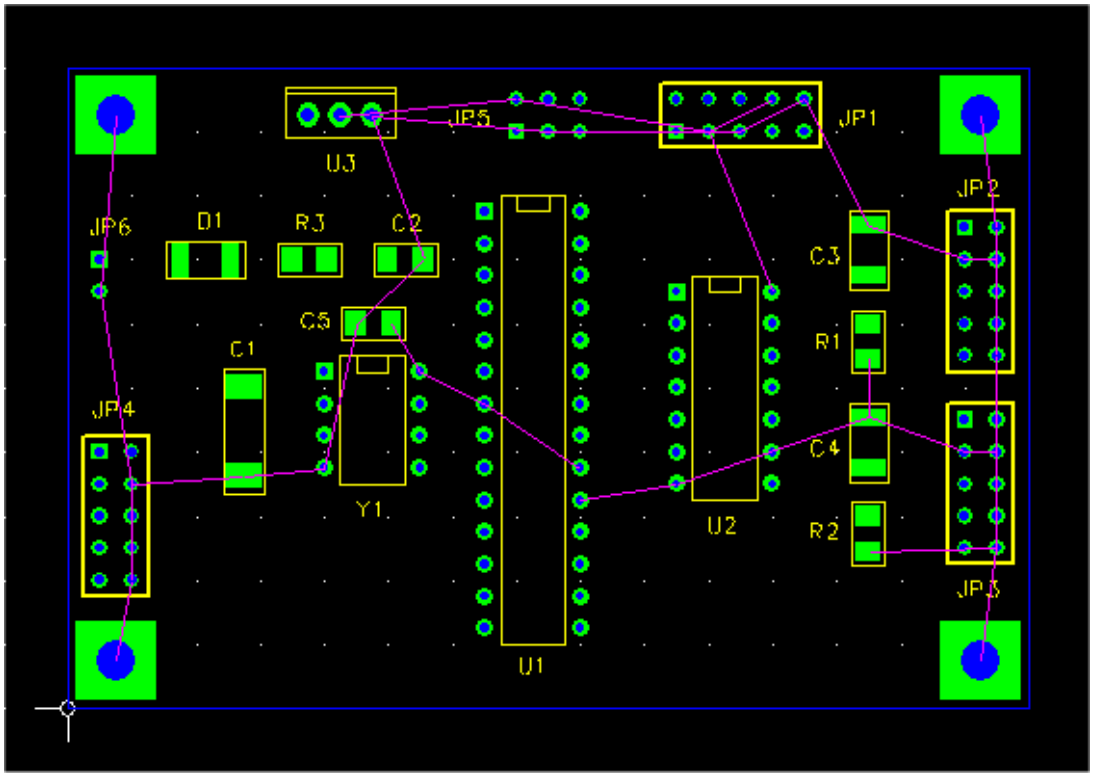
- Click the **Make all visible/invisible** button twice, to clear all of the checkboxes and make all of the nets invisible. Then click on the boxes for VCC and GND, to make these nets visible. Then click **OK**. Now only the VCC and GND ratlines are visible, as shown.



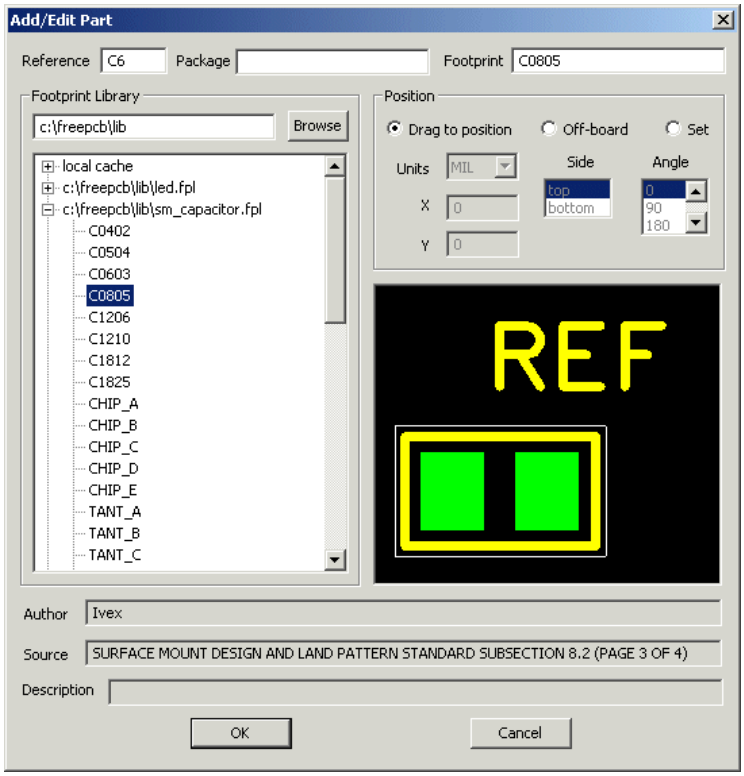
- Click on the right-most pad of C5 to select it. This will be pin 2, unless you rotated the part. Press F1 ("Set Net") to pop up the following dialog:



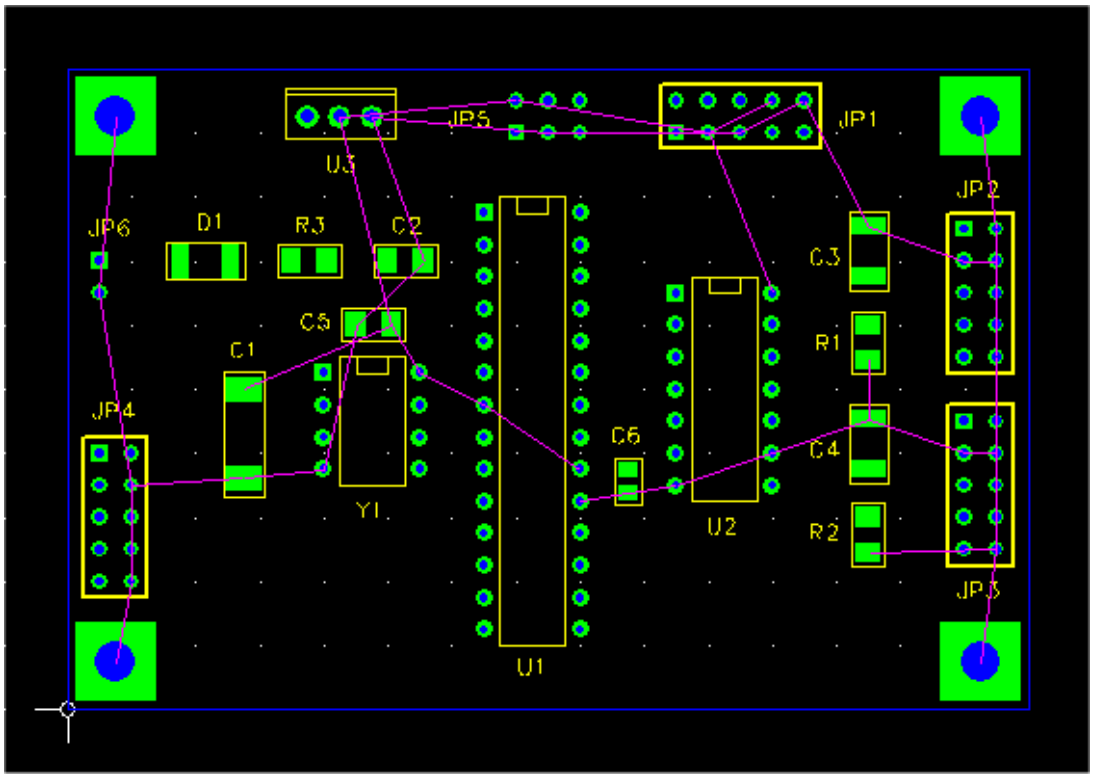
- From the drop-down menu, select VCC, or just type "VCC" into the text box. Click **OK** to connect the pin. Press F8 ("Recalc. Ratlines") to recalculate the ratlines for VCC to include the new pin C5.2.
- Now select the left-most pad of C5, and repeat the above procedure to connect it to GND. Now you should see ratlines to both pads of C5.



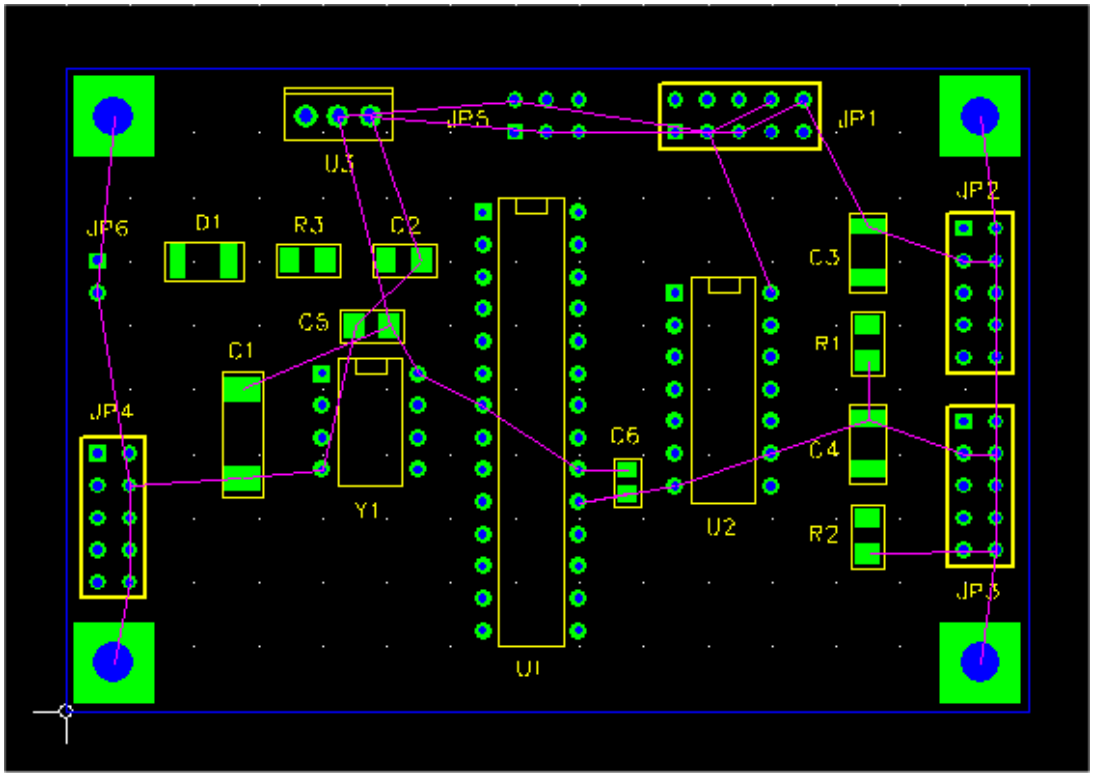
- Let's use a slightly different method to add C6. Select **Add > Part**. The **Add/Edit Part** dialog will pop up. This is the same dialog that you invoked previously from the **View/Edit Partlist** dialog, except that the **Drag to position** button is now enabled and selected. This time, select C0805 for the footprint of the new part. Enter C6 into the **Reference** field.



- Press **OK**. You should find yourself dragging the outline for C6. Place it near pins 19 and 20 of U1, as shown.

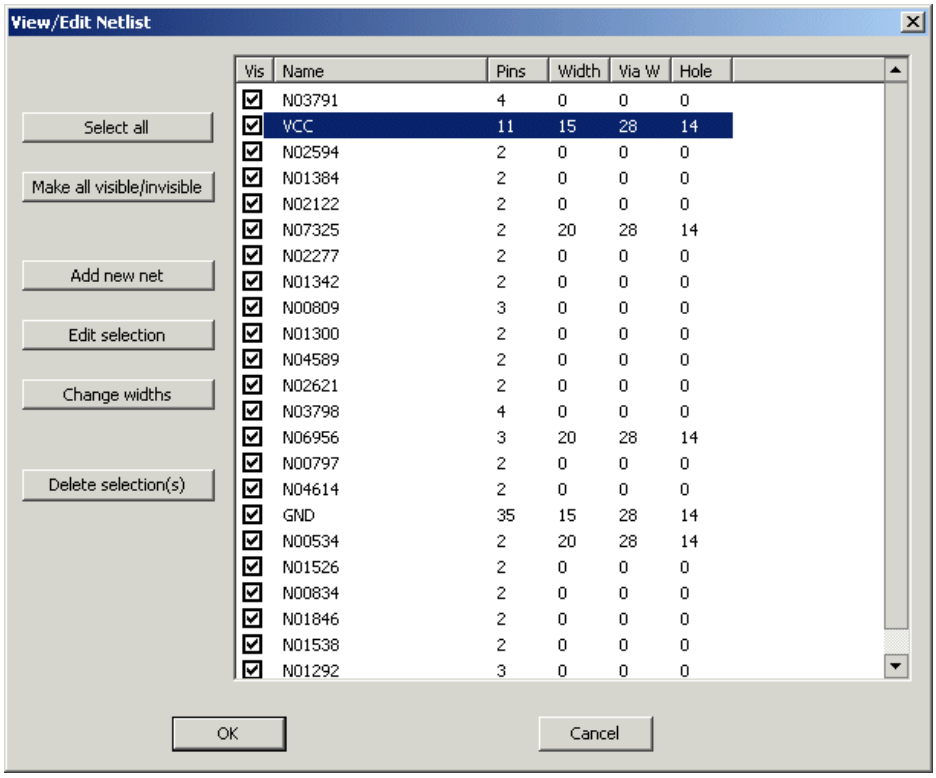


- We will connect C6 to the VCC and GND nets by drawing ratlines between pins. Select the upper pad of C6 by clicking on it. Then press F4 ("Connect Pin"). You should start dragging a ratline from the pad. Move the cursor over the pad for pin 20 of U1 (which is attached to net VCC) and click to connect the ratline to this pin. In similar fashion, connect the lower pad of C6 to pin U1.19. Select C6 by clicking on it, and press F8 ("Recalc. Ratlines") to recalculate the ratlines for C6. Your layout should look like this:



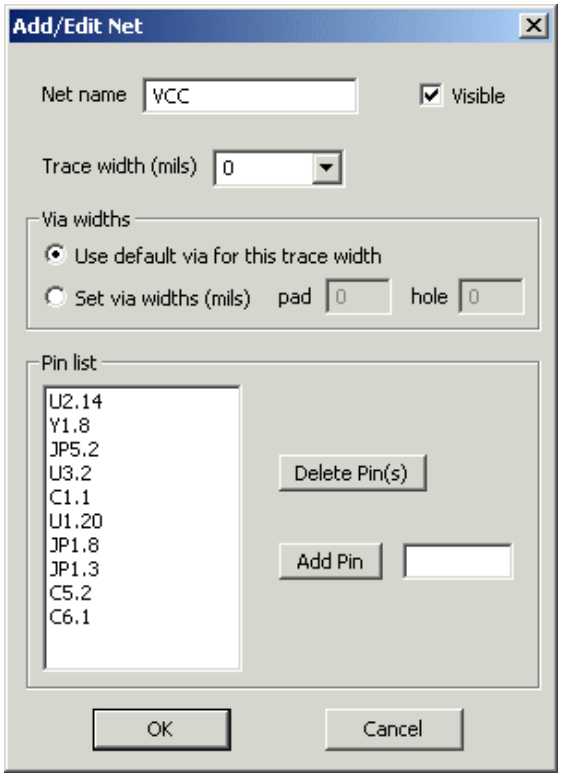
- Finally, add C7 using **Add > Part**, the same way we added C6. Place it near the top of U2, similarly to the way we placed C5 near the top of Y1.

- Now let's connect it to VCC and GND by using **Project > Nets...**, to pop up the **View/Edit Netlist** dialog. Then select the VCC net by clicking on it, as shown.



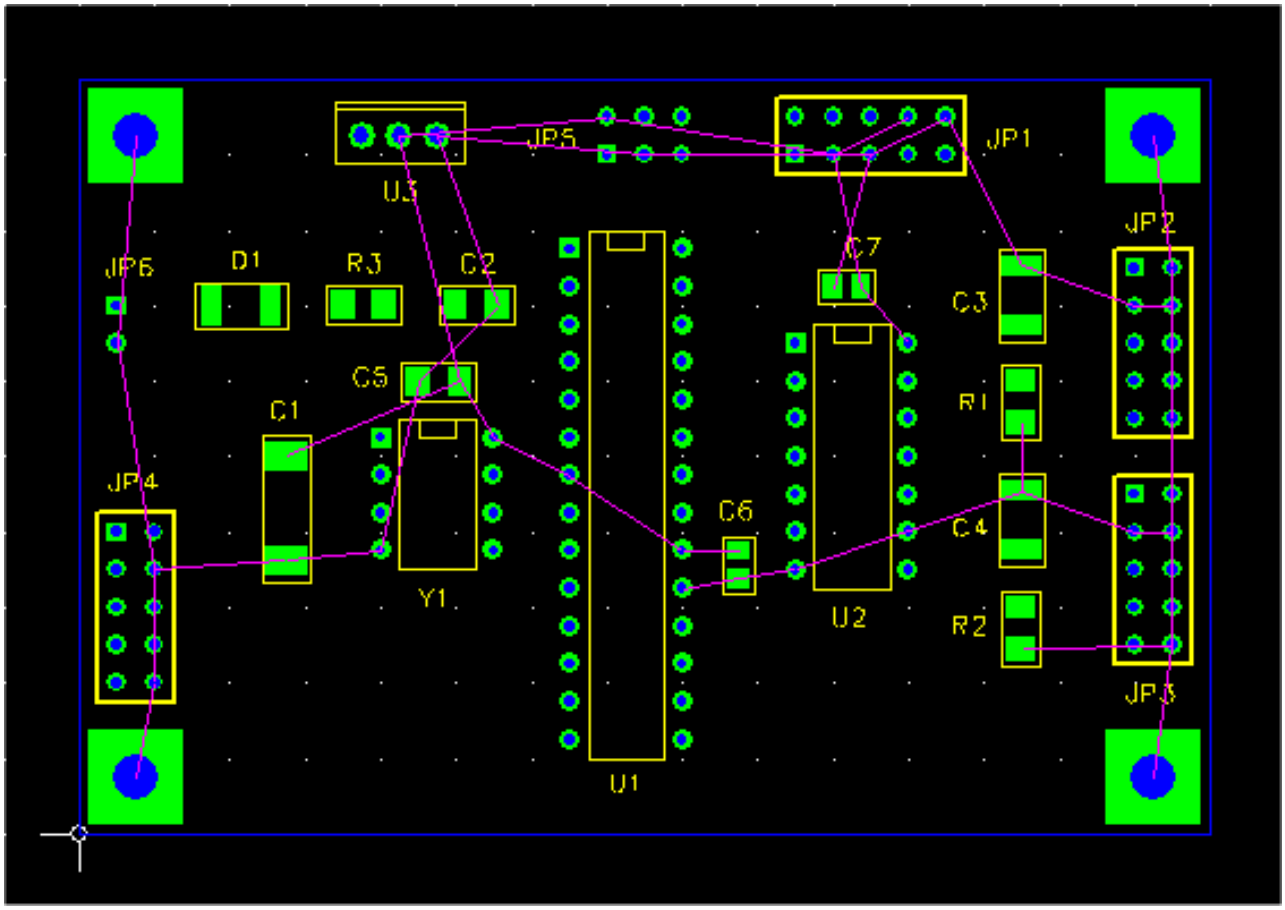
Vis	Name	Pins	Width	Via W	Hole
☑	N03791	4	0	0	0
☑	VCC	11	15	28	14
☑	N02594	2	0	0	0
☑	N01384	2	0	0	0
☑	N02122	2	0	0	0
☑	N07325	2	20	28	14
☑	N02277	2	0	0	0
☑	N01342	2	0	0	0
☑	N00809	3	0	0	0
☑	N01300	2	0	0	0
☑	N04589	2	0	0	0
☑	N02621	2	0	0	0
☑	N03798	4	0	0	0
☑	N06956	3	20	28	14
☑	N00797	2	0	0	0
☑	N04614	2	0	0	0
☑	GND	35	15	28	14
☑	N00534	2	20	28	14
☑	N01526	2	0	0	0
☑	N00834	2	0	0	0
☑	N01846	2	0	0	0
☑	N01538	2	0	0	0
☑	N01292	3	0	0	0

- Now click on **Edit selection**. The **Add/Edit Net** dialog will appear.



- Enter C7.2 into the text box next to the **Add Pin** button. Then click on **Add Pin**. C7.2 should be added to the **Pin list**. Click **OK** to accept it. This will return you to the **View/Edit Netlist** dialog.
- Now select the GND net by clicking on it, and edit the pin list just as you did for VCC, this time adding C7.1.
- **Tip:** When you add a pin to a net using the **Add/Edit Net** dialog, be **sure** that you click on **Add Pin**. Don't just click on **OK**, which exits the dialog without adding the pin.

- Press F8 ("Recalc. Ratlines") to add ratlines to the newly added pins. The layout should now look like:

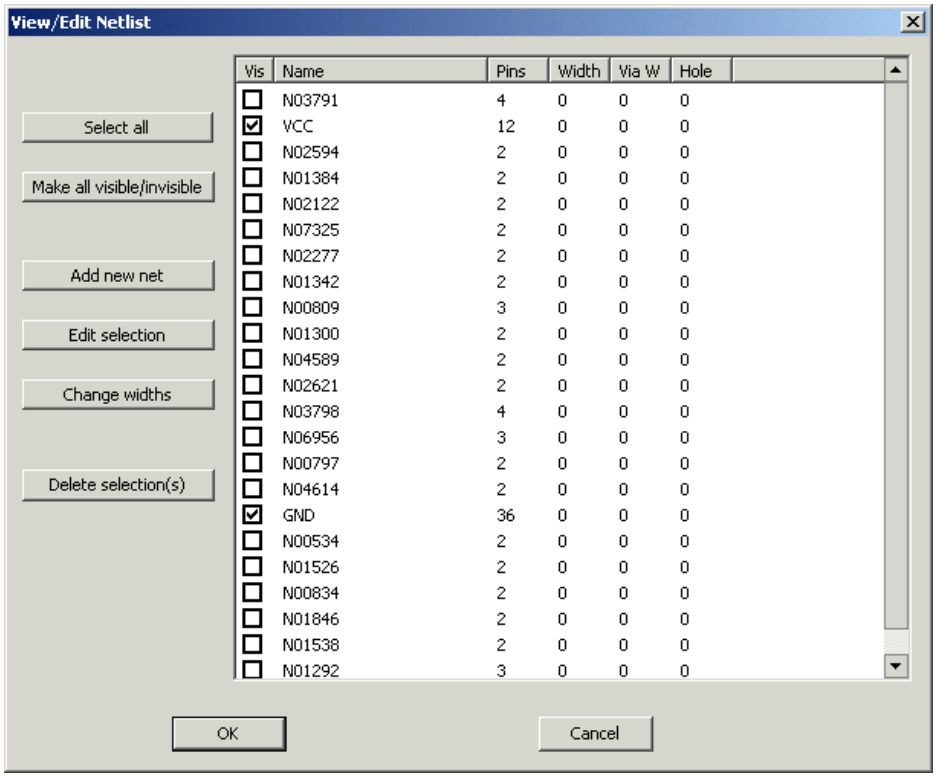


In the next section we will use copper areas and stub traces to create power and ground planes for VCC and GND.

7.8 Adding Copper Areas

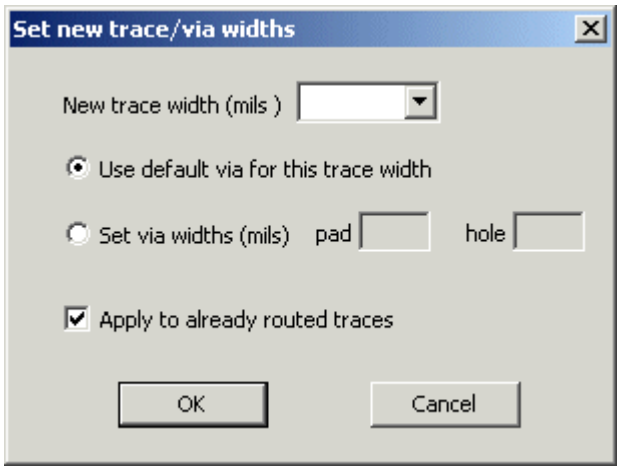
In this section, we will create copper areas on the inner layers for power and ground planes.

- Select **Project > Nets...**. The **View/Edit Netlist** dialog should appear. If you are continuing from the last section, only the VCC and GND nets will be visible, as shown.



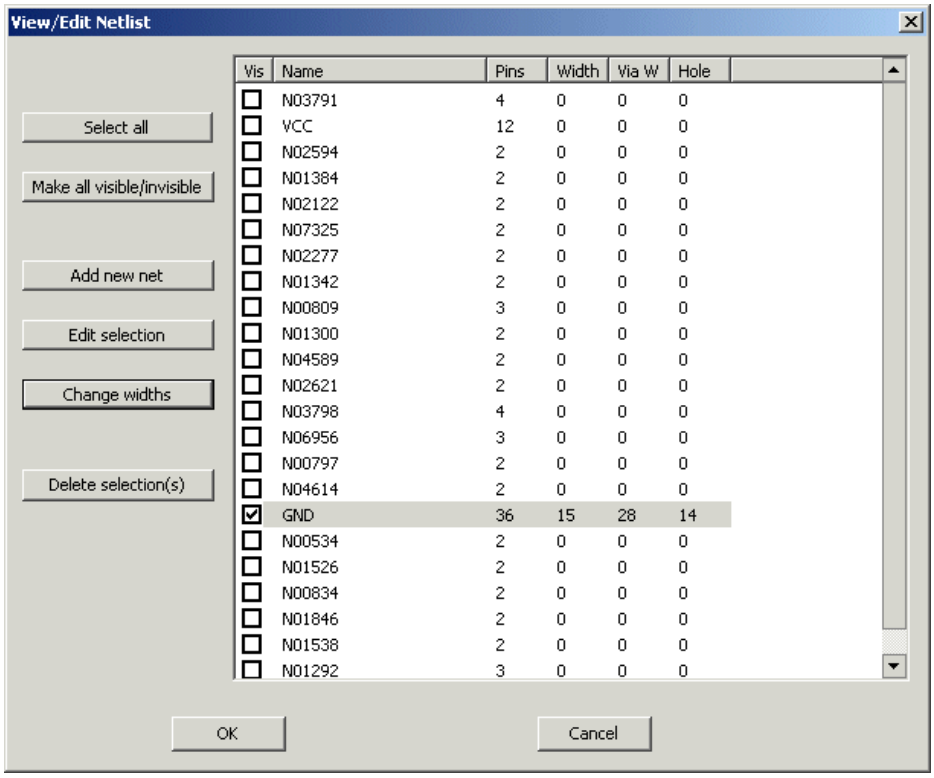
Vis	Name	Pins	Width	Via W	Hole
☐	N03791	4	0	0	0
☑	VCC	12	0	0	0
☐	N02594	2	0	0	0
☐	N01384	2	0	0	0
☐	N02122	2	0	0	0
☐	N07325	2	0	0	0
☐	N02277	2	0	0	0
☐	N01342	2	0	0	0
☐	N00809	3	0	0	0
☐	N01300	2	0	0	0
☐	N04589	2	0	0	0
☐	N02621	2	0	0	0
☐	N03798	4	0	0	0
☐	N06956	3	0	0	0
☐	N00797	2	0	0	0
☐	N04614	2	0	0	0
☑	GND	36	0	0	0
☐	N00534	2	0	0	0
☐	N01526	2	0	0	0
☐	N00834	2	0	0	0
☐	N01846	2	0	0	0
☐	N01538	2	0	0	0
☐	N01292	3	0	0	0

- Click on **Make all visible/invisible** one or two times to make all of the nets invisible. Then click on the GND checkbox to make only that net visible.
- Notice that the trace width for every net is "0". That means that the project default of 10 mils will be used. Let's change the trace width for GND to 15 mils.



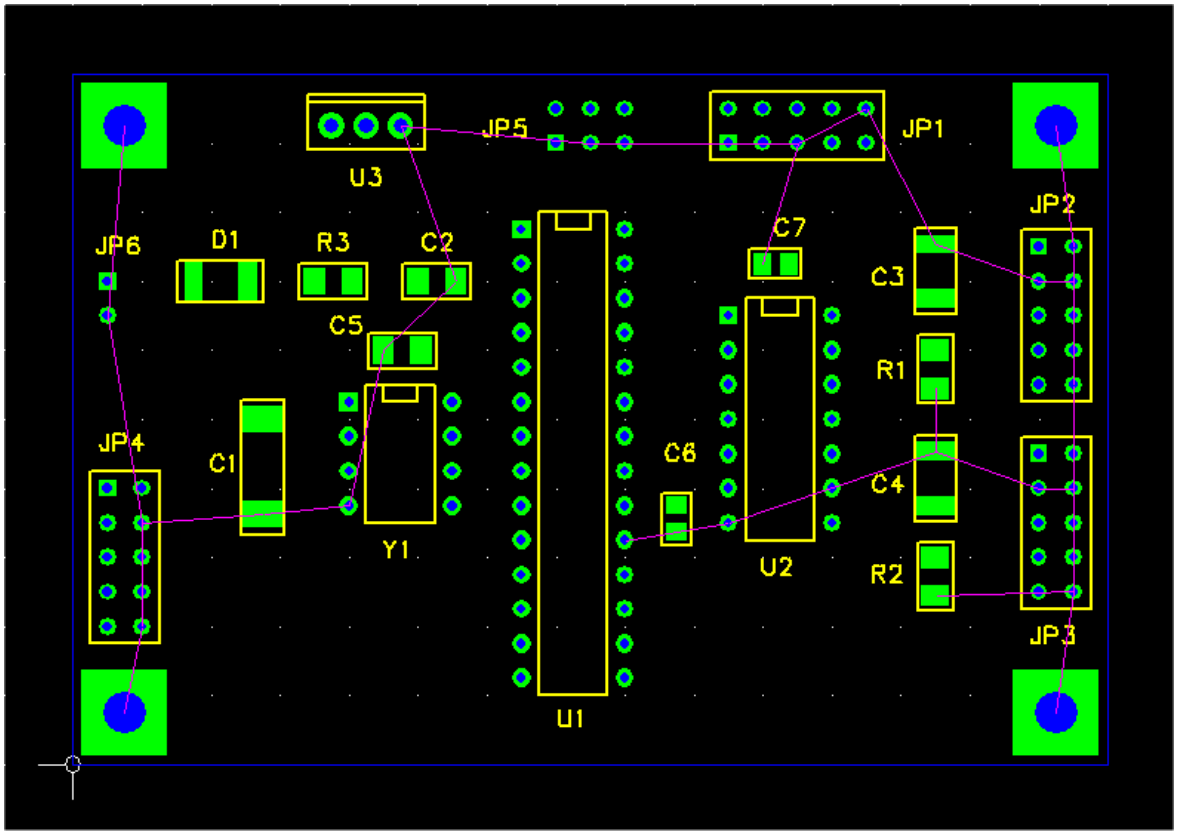
- Select the GND net by clicking on its name, which should highlight it. Then click on the **Change widths** button. The following dialog should appear.
- Select "15" for the new width for the GND net by selecting that value from the drop-down menu, or enter "15" into the text box. Leave the **Use default via for this trace width** radio button selected. If we wanted to override the default via, we could select **Set via widths** instead, and set the via width and hole size explicitly in the text boxes. Click **OK** to exit the dialog.

- The **View/Edit Netlist** dialog should now look like:

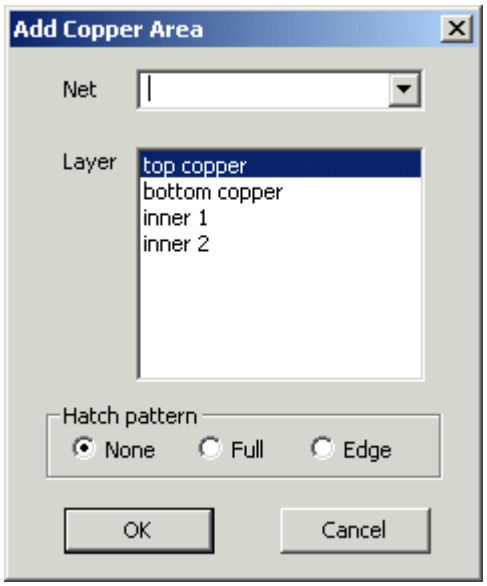


Vis	Name	Pins	Width	Via W	Hole
☐	N03791	4	0	0	0
☐	VCC	12	0	0	0
☐	N02594	2	0	0	0
☐	N01384	2	0	0	0
☐	N02122	2	0	0	0
☐	N07325	2	0	0	0
☐	N02277	2	0	0	0
☐	N01342	2	0	0	0
☐	N00809	3	0	0	0
☐	N01300	2	0	0	0
☐	N04589	2	0	0	0
☐	N02621	2	0	0	0
☐	N03798	4	0	0	0
☐	N06956	3	0	0	0
☐	N00797	2	0	0	0
☐	N04614	2	0	0	0
☑	GND	36	15	28	14
☐	N00534	2	0	0	0
☐	N01526	2	0	0	0
☐	N00834	2	0	0	0
☐	N01846	2	0	0	0
☐	N01538	2	0	0	0
☐	N01292	3	0	0	0

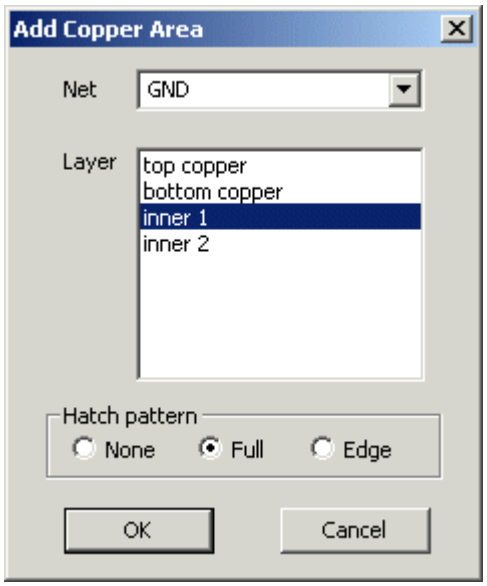
- Click **OK** to exit the dialog. The layout window should now show only the ratlines for the GND net, as shown.



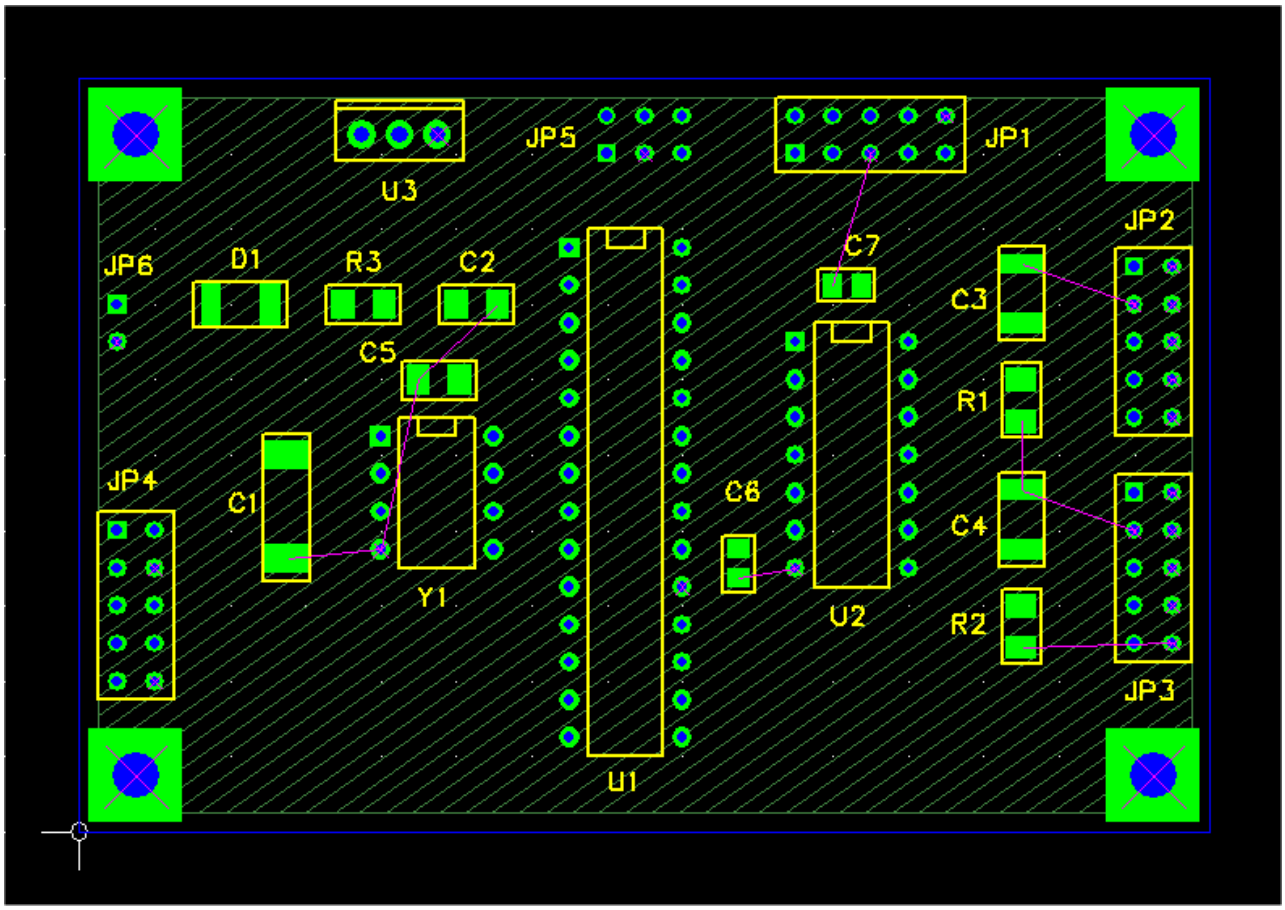
- Now we will add the copper area for the GND plane. First, set the routing grid to 50 mils. Then select **Add > Copper Area**. The following dialog should appear.



- We will use the "inner 1" copper layer for the GND plane. Select "GND" from the **Net** drop-down list, or type "GND" into the text box. Select "inner 1" from the **Layer** list. Set a hatch pattern of **Full**.

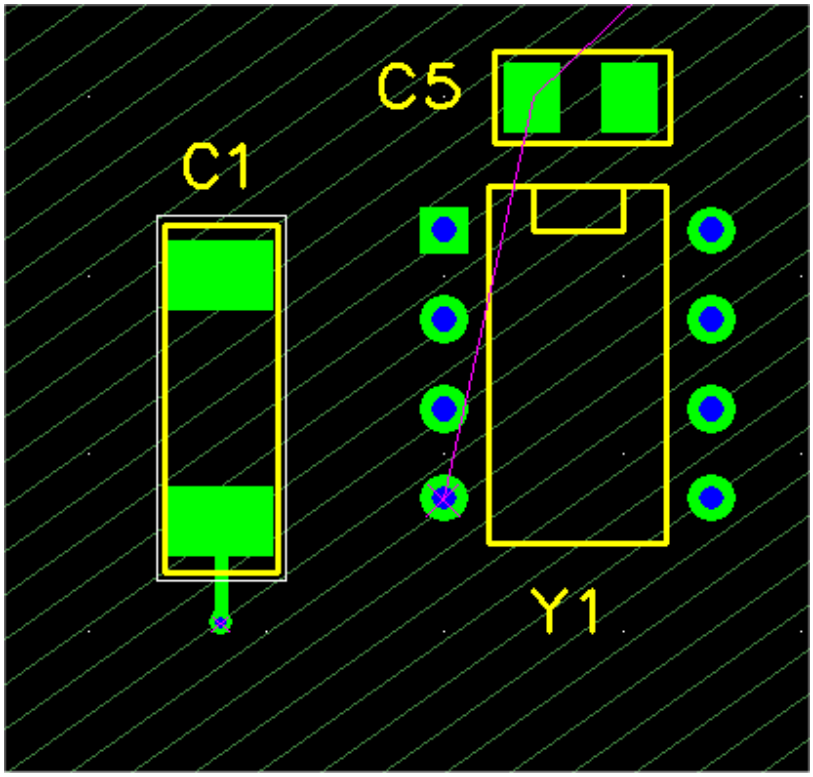


- Click **OK** to exit the dialog and start drawing the polyline for the copper area. The cursor should change to a cross-hair. We want to draw the outline of the copper area 50 mils inside the board outline. Place the cursor at the lower-left corner of the copper area, which will be at X = 50, Y = 50. Left-click to place the first corner. Then move the cursor to X = 50, Y = 1950 and left-click to place the second corner. Place the third corner at X = 2950, Y = 1950 and the fourth corner at X = 2950, Y = 50. After placing the fourth corner, right-click to close the polyline. Now your board should look like:



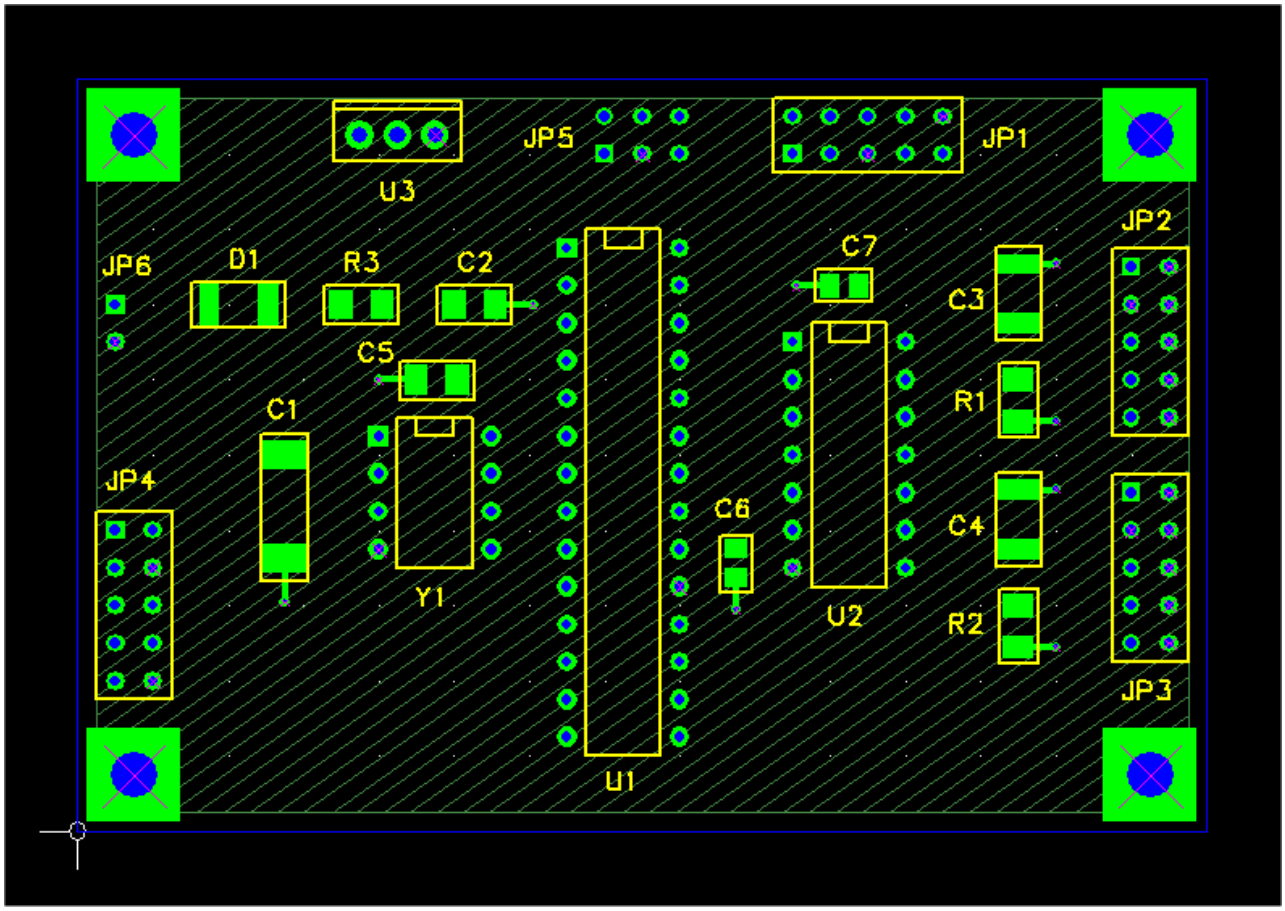
- Notice that the copper area has been drawn with a diagonal hatch pattern, in the color for layer "inner 1". Also, most of the ratlines for the GND net have disappeared, and X-shaped symbols in the ratline color have appeared on the through-hole pins in the net. These indicate internal connections of these pins to the GND copper area, using plated-through holes with thermal reliefs.
- We still have to connect the SMT pins to the GND copper area. For this, we will use **stub traces**, which are short traces that begin on a pin and end with a via to the GND plane.
- Let's start with the lower pin of C1. Select the pin by clicking on it. A white box with an "X" through it should appear around the pin, indicating that it has been selected, and the status bar should contain a description of the pin.

- Now start drawing the stub trace by pressing F3 ("Start Stub"). The cursor should change to a cross-hair, and you will be dragging the end of a trace segment from the pin, on the top copper layer. You may wish to zoom in with the scroll wheel or the Page Up key. Move the cursor a short distance below the pin, and left-click to set a vertex. Then right-click to end the trace. A via should appear at the vertex, as shown. Notice that there is a thermal-relief symbol on the via, indicating an internal connection to the GND copper area.



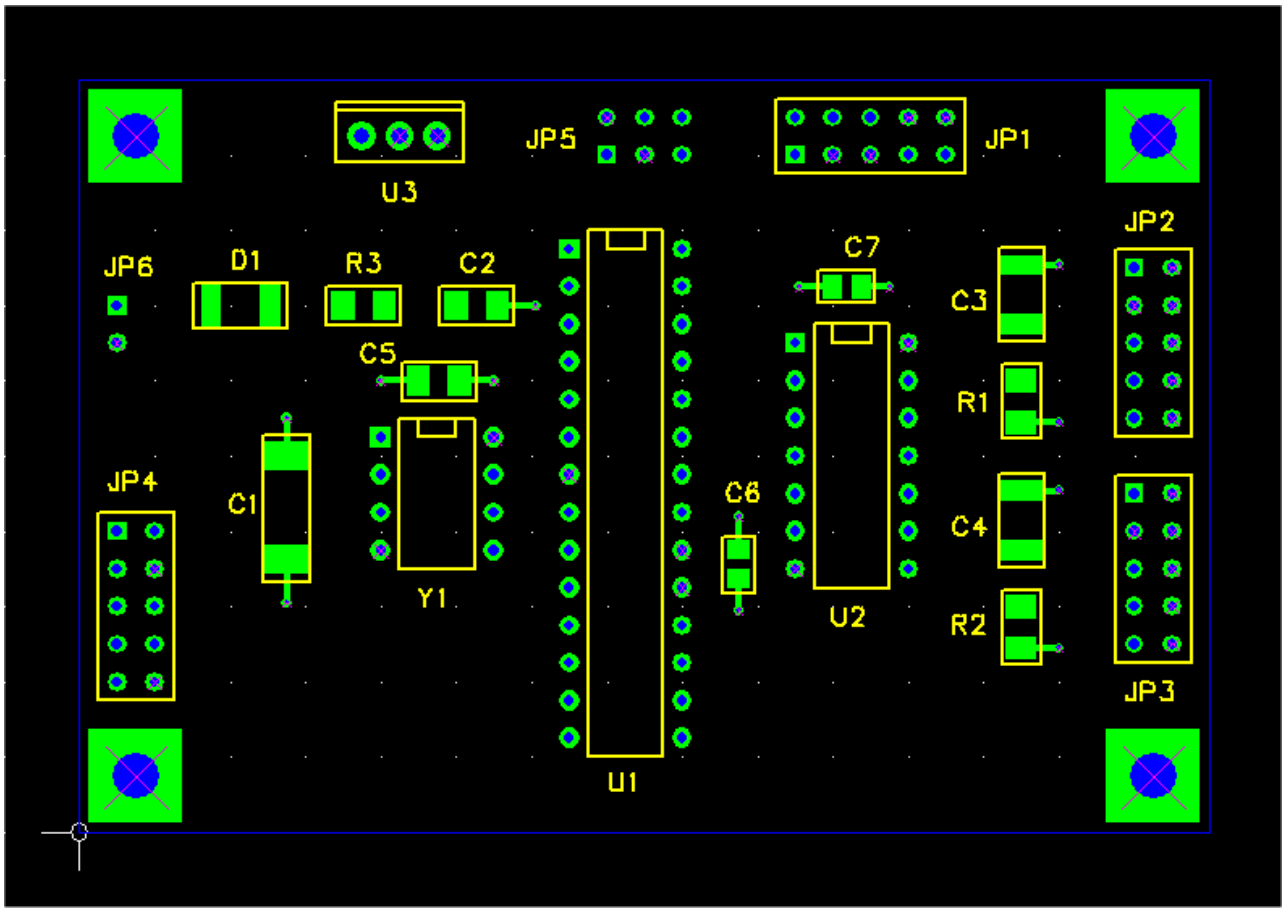
- If you are unhappy with your trace, you can delete it by selecting the trace segment or the end via and pressing F7 ("Delete Connection"). You can move the end via by selecting it and pressing F4 ("Move Vertex"). You can add segments by selecting the end via and pressing F2 ("Add Segment").
- Since stub traces are usually used to connect pins to copper areas on other layers, the end via is added automatically when you end a stub trace. If you want, you can delete the via by selecting it and pressing F3 ("Delete Via"). This may be useful if you are connecting a pin to a copper area on the same layer, or if you are using a stub trace for some other purpose such as a shield trace.

- Now that you know how to do it, add stub traces to all of the SMT pins which have ratlines. You may need to move some of the reference designators to keep them free of the end vias, since it is generally a bad practice to have a silk-screen item overlapping a pad or via. Your board should look something like:



- Congratulations, you have now routed the entire GND net, which is by far the largest net in the design. The copper area made it easy.
- Since we won't be doing any routing on the "inner 1" copper layer, and since the hatch pattern may be annoying when we are working on other layers, let's make it invisible. Select **Layers** from the **View** menu to bring up the **View/Edit Layers** dialog, and uncheck the **Visible** checkbox next to **Inner 1**. Click **OK**, and the copper area should disappear. Alternatively, we could have changed the hatch pattern by selecting a side and right-clicking, then selecting **Hatch style** from the context menu.
- Now, let's add a copper area for net VCC on the "inner 2" copper layer. You just need to repeat the same steps that you performed for the GND area. These are:
  - Make the VCC net visible.
  - Change the VCC trace width to 15 mils.
  - Draw the VCC copper area on the "inner 2" layer (you can use the same corner positions as the GND area).
  - Add stub traces for all of the SMT pads on the VCC net. There should be four, on C1, C5, C6 and C7.
  - Make the "inner 2" layer invisible.

- Select **Project > Nets...** to launch the **View/Edit Netlist** dialog. Make both VCC and GND visible. Your board should look like:

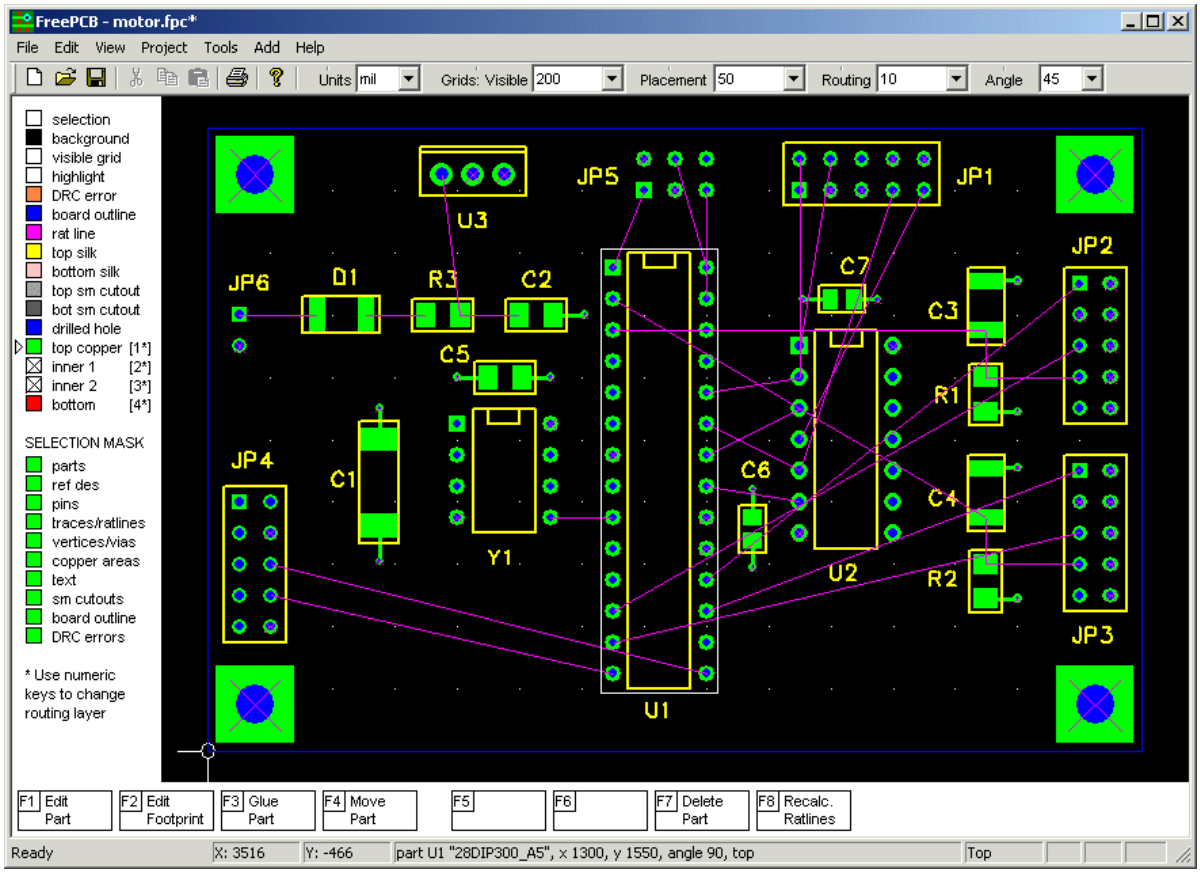


In the next section we will route the rest of the nets on the top and bottom copper layers.

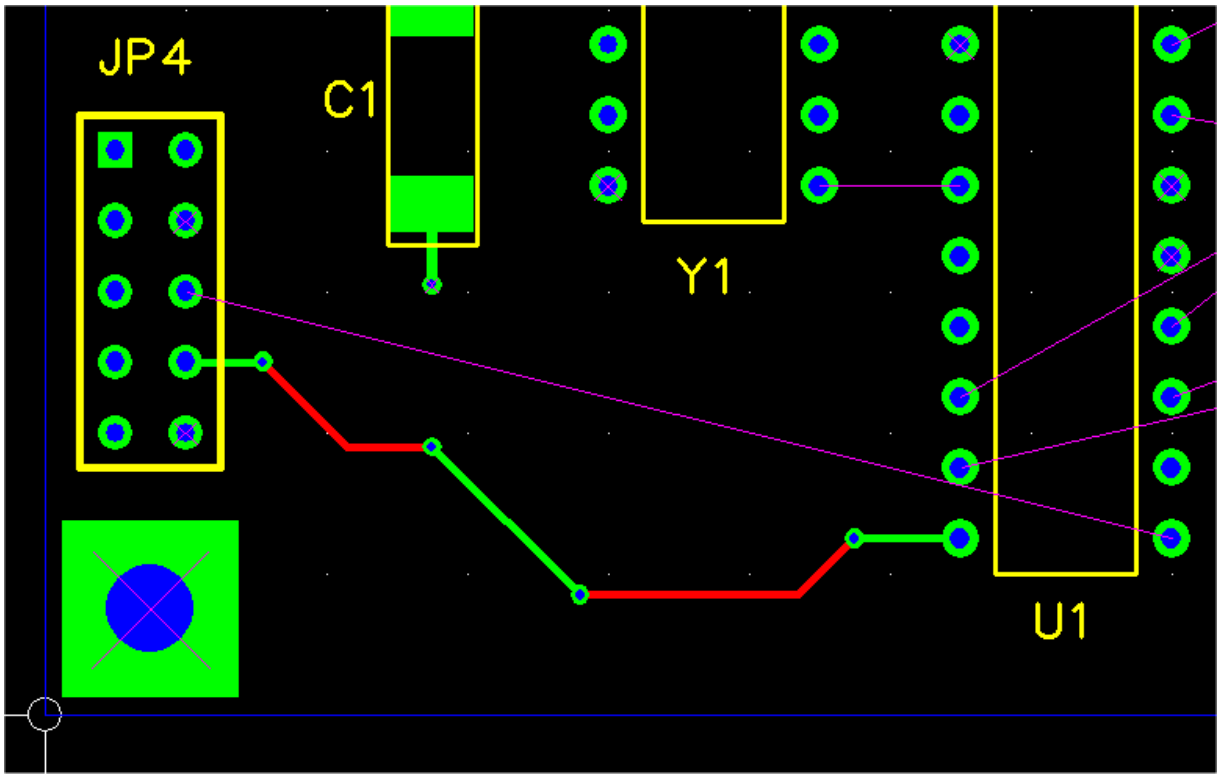
7.9 Routing

Ok, let's route our PCB. Before starting, you might want to review **Section 5.13: Nets, Ratlines and Routing**.

- **Optional:** If you want to try some of the ratline editing features that were described in **Section 5.13: Nets, Ratlines and Routing**:
  - Start by saving your project, so you can reload it if you screw up.
  - From the menus, select **Project > Nets...** to pop up the **View/Edit Netlist** dialog. Make all of the nets invisible except GND.
  - Select a ratline and delete it by pressing F7 ("Delete Connect"). Then press F8 ("Recalc. Ratlines"), and the ratline should reappear.
  - You can add a new ratline between pins on the same net by selecting one of the pins and using F4 ("Connect Pin"), which allows you to draw a ratline to the other pin. Then press F3 ("Lock Connect") to lock it. Now if you press F8 ("Recalc. Ratlines"), some other ratline on the net should disappear.
  - You can unlock the locked connection with F3. Then if you press F8 the connections should revert back to their original state.
- OK, back to business. Use the **View/Edit Netlist** dialog to make all of the nets visible.
- If necessary, use the **View/Edit Layers** dialog to make the inner 1 and inner 2 copper layers invisible
- Now your board should look something like:

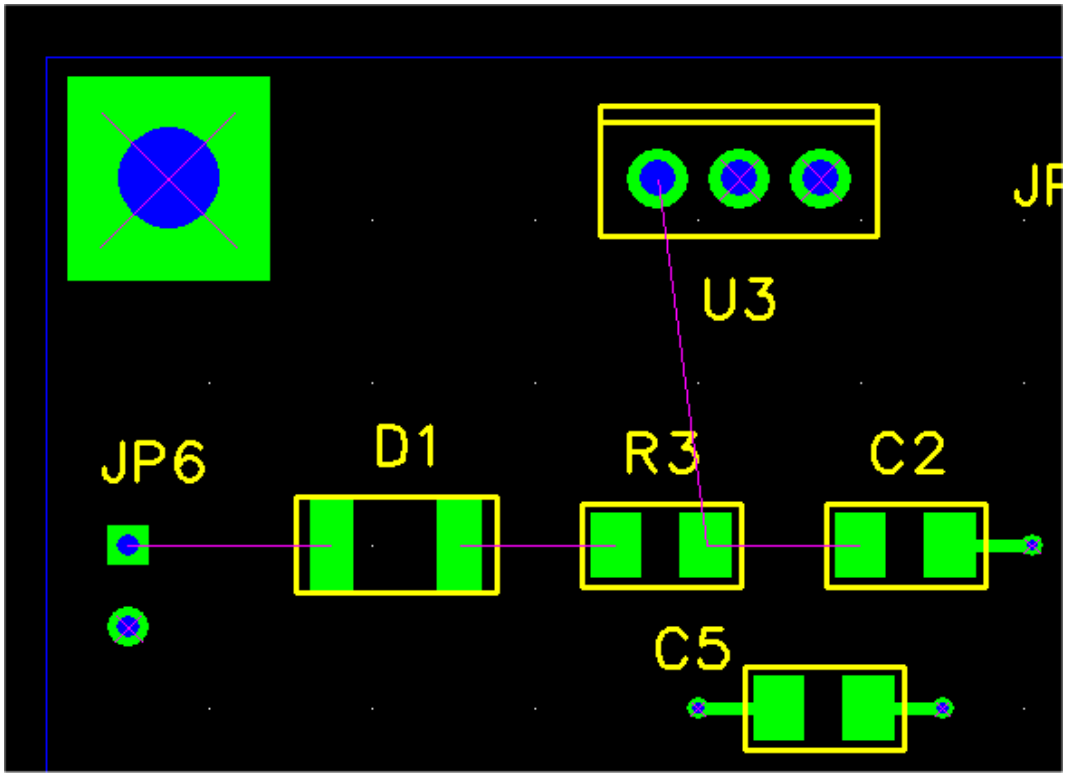


- Select a reasonably small value for the routing grid such as 10 mils.
- We will be routing traces on the top and bottom copper layers, since inner 1 and inner 2 were used for the VCC and GND planes. The currently active layer is shown on the status bar ("Top" in the screenshot above). Press the "4" key to switch to the bottom layer (since there are 4 layers). Press the "1" key to switch back to the top layer. Keys "2" and "3" would select the inner layers.
- Now select one of the longer ratlines, such as the one in the lower left from JP4.8 to U1.14. Press F4 ("Route Segment") to start routing. The cursor should change to a cross-hair, and you will be dragging a trace segment from whichever pin was closest to the cursor when you pressed F4. Since the active layer is "Top", the trace segment will be colored green. Notice that the angle of the segment will snap to multiples of 45 degrees.
- Place the first vertex in the trace by left-clicking the mouse. Now you will be dragging a new segment from the vertex. Press "4" to switch to the bottom layer. The segment will become red, and a via should appear at the vertex.
- Continue adding segments until you are ready to complete the trace by drawing a segment from the last vertex to the end pin. Press F4 ("Complete Segment") to add the last segment, or just left-click on the end pin. Your trace should look something like the screenshot below. Notice that I have used a few more segments and vias than necessary.

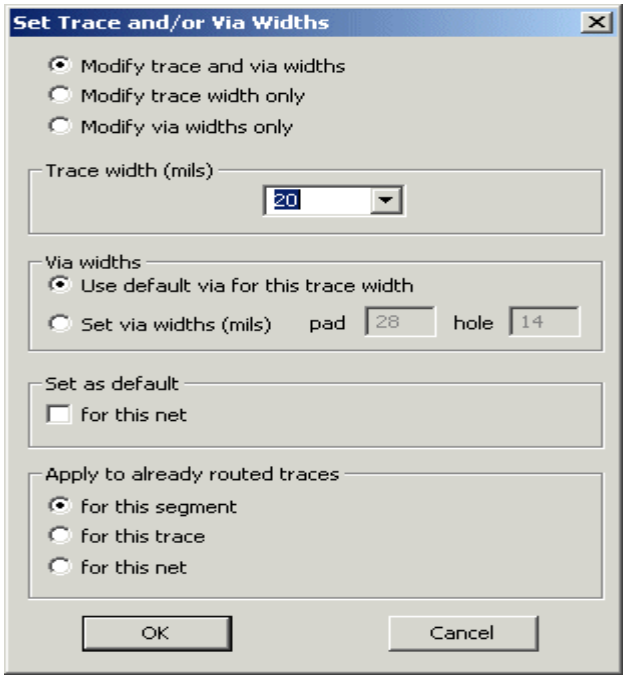


- Now let's try editing the trace which you just drew. Select one of the vertices by clicking on it. A small white box should appear around it to indicate that it has been selected, and information about the vertex should appear in the status bar. The editing options for a vertex are:
    - F1 ("Set Position") - pop up a dialog to set the X and Y coordinates of the vertex explicitly.
    - F4 ("Move Vertex") - start moving the vertex by dragging it with the mouse. The snap-angle will not be in effect, but the routing grid will.
    - F5 ("Delete Vertex") - remove the vertex, and unroute the two adjacent segments, which will be replaced by a single ratline segment.
    - F6 ("Unroute trace") - unroute the entire trace, which will revert to a ratline.
    - F7 ("Delete Connect") - delete the trace, without replacing it with a ratline.
    - F8 ("Recalc. Ratlines") - regenerate the ratlines for the net.
- Now select one of the trace segments, which should turn white indicating that it has been selected. The editing options for a trace segment are:
    - F1 ("Set Width") - pop up a dialog allowing you to set the width of the net, trace or trace segment.
    - F5 ("Unroute Segment") - replace the segment with a ratline.
    - F6 ("Unroute trace") - unroute the entire trace, which will revert to a ratline.
    - F7 ("Delete Connect") - delete the trace, without replacing it with a ratline.
    - F8 ("Recalc. Ratlines") - regenerate the ratlines for the net.
- Most of these editing options are fairly self-explanatory, so you can try them out on your own. Try deleting a vertex or trace segment, and then re-routing the resulting ratline. Ratlines between vertices are routed just like ratlines between pins.

- Now let's change the widths of some traces. Zoom in on the components in the upper left corner, like this:

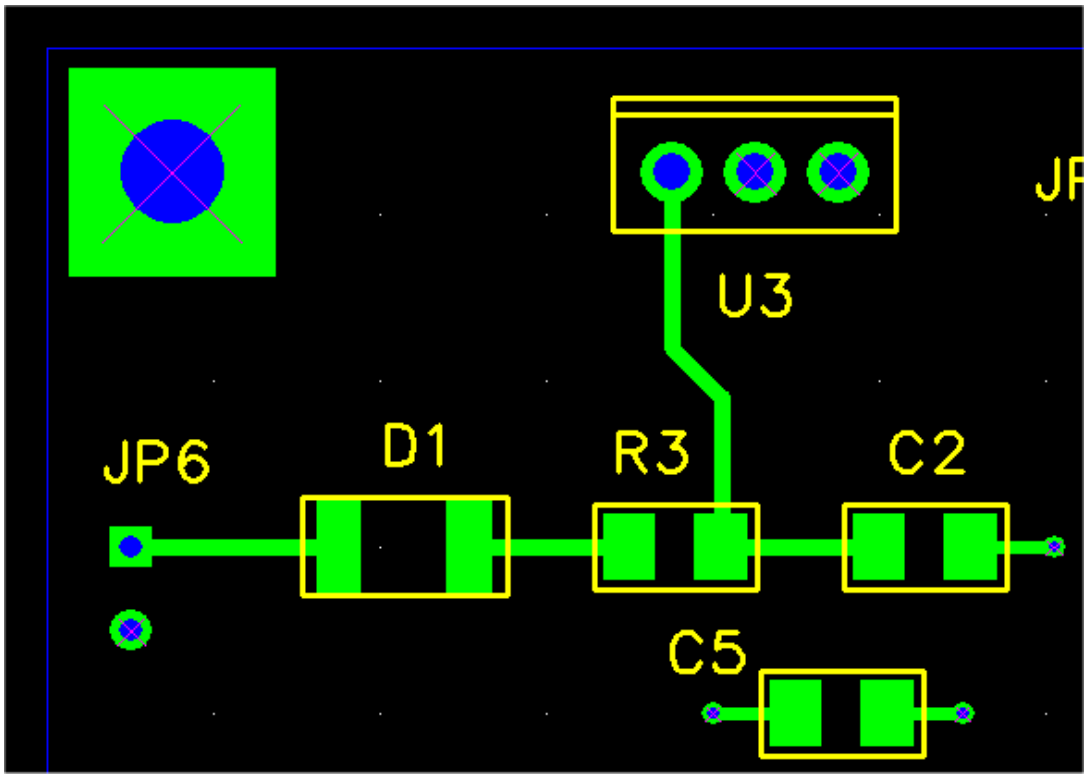


- The traces between JP6, D1, R3, C2 and U3 are power traces, so let's make them wider than the default 10 mils. Select the ratline between JP6.1 and D1.1. The status bar should indicate that this is net "N00534". Press F1 ("Set Width"). The following dialog should pop up:

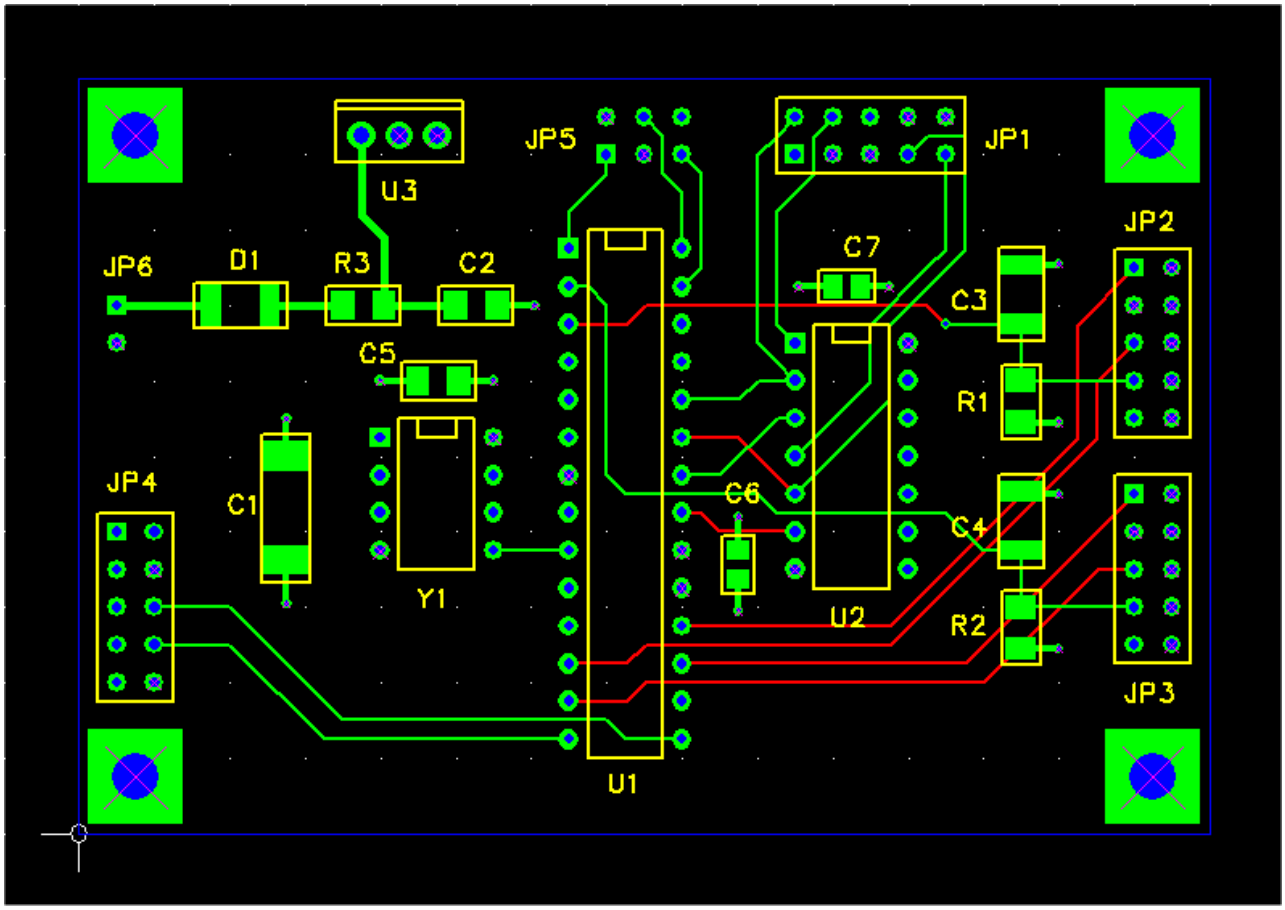


- The drop-down menu for **Trace width (mils)** has two options, 10 and 15 mils. Lets use 20 mils instead, by typing "20" directly into the text box. Click **OK** to exit the dialog.

- Now route the trace. Since the D1 is an SMT part, the trace segment which connects to D1.1 must be on the top layer. If you start routing from this pin, the active layer will be automatically set to "Top". If you start routing from JP6.1, which is a through-hole pin, you can start on either "Top" or "Bottom" but you must be on "Top" to complete the trace to D1.1.
- In similar fashion, set the width of the other power traces to 20 mils and route them. Your board should look like:



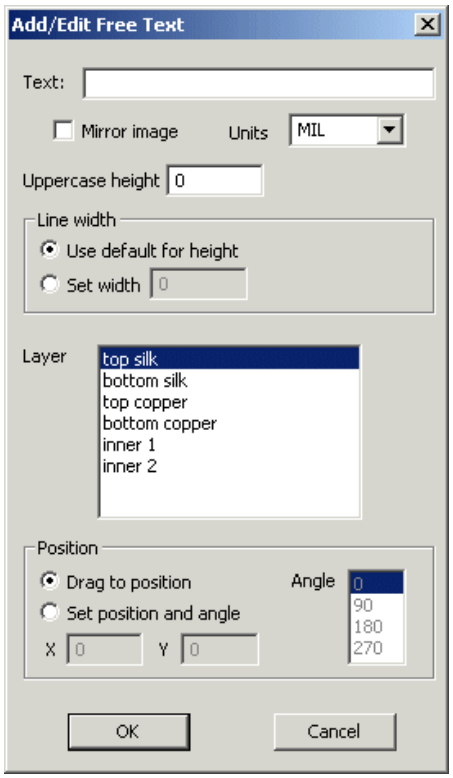
- Now route the rest of the traces on the board. Feel free to get creative with trace widths and routing if you like. If we were actually going to produce the board, it might look something like this.



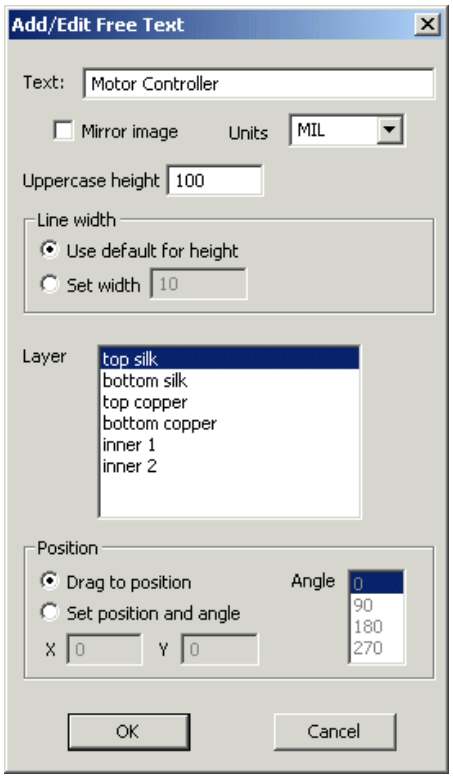
7.10 Adding Text

To complete our masterpiece, let's add some text on the top silk-screen layer to identify the board.

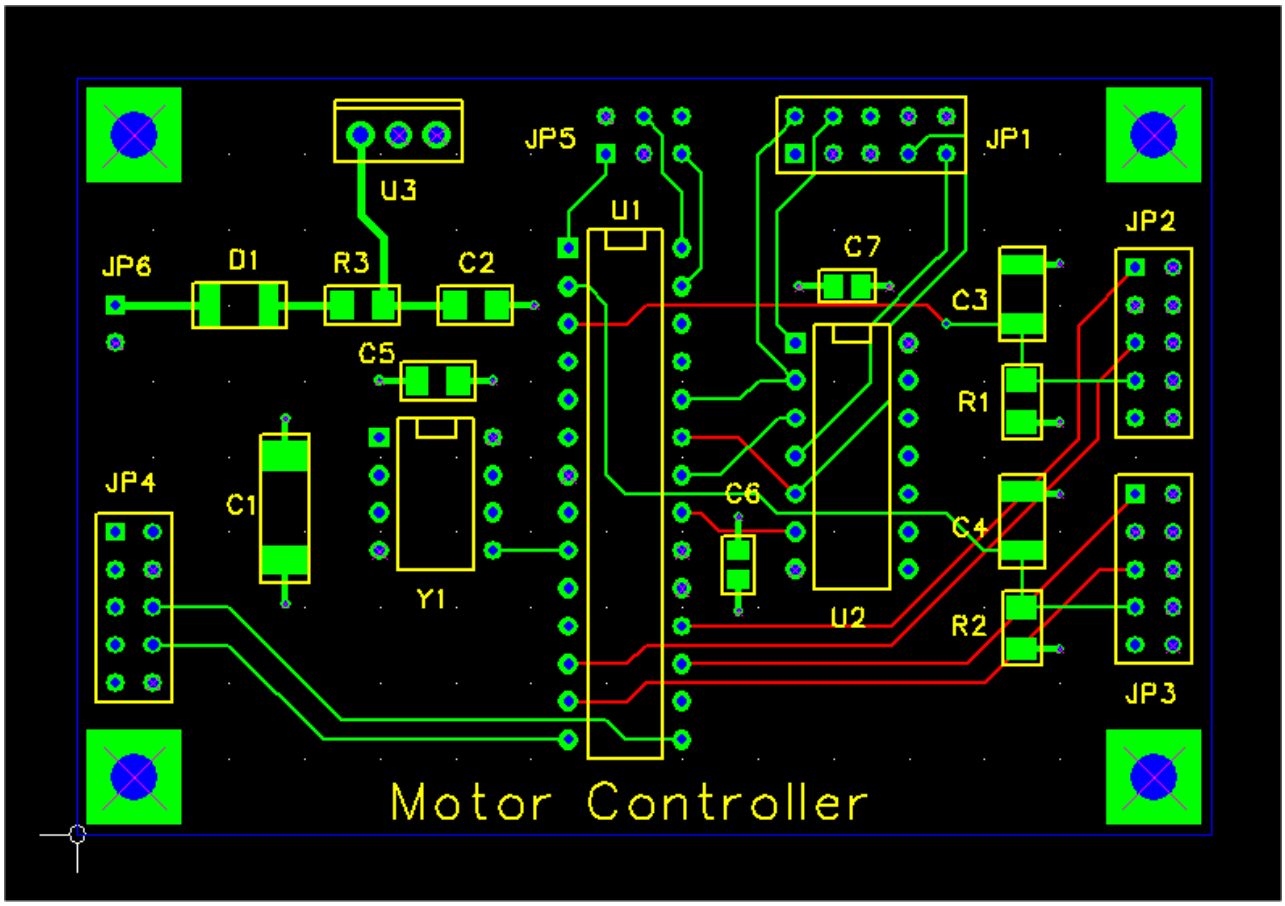
- Select **Text** from the **Add** menu to pop up the **Add/Edit Free Text** dialog.



- Enter "Motor Controller" in the **Text** field. Make sure that the **Units** are set to "MIL". Enter "100" in the **Uppercase Height** field, as shown below.



- Click **OK**. You will find yourself dragging a rectangle which represents the bounding box for the text. Place this near the bottom of the PCB, and left-click.



This completes our PCB layout. In the next section, we will create the Gerber and drill files which you would send to the manufacturer to fabricate your board.

7.11 Checking Design Rules

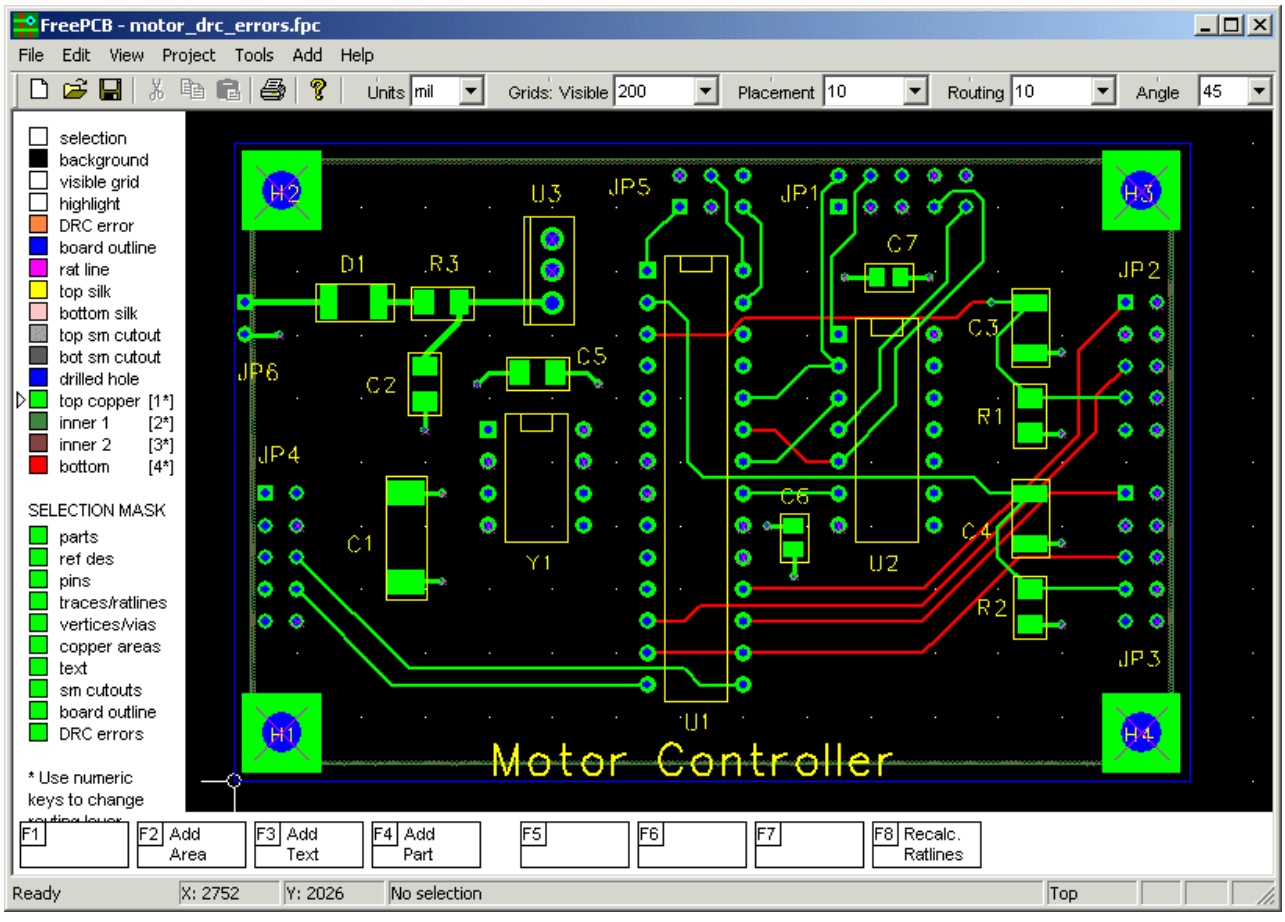
Design Rules are a set of rules that establish lower limits for dimensions such as trace widths and clearances. They are necessary because the PCB manufacturing process is subject to certain limitations and tolerances, and the design has to allow for this. For example, you can't use trace widths of 1 mil in your layout, because it is physically impossible to etch them reliably.

If you are having your PCBs made for you, the board house should provide design rules based on their process. For example, Advanced Circuits recommends the following for their low-cost process:

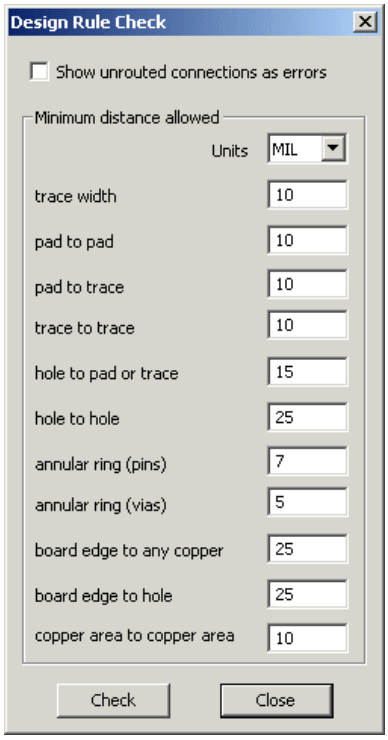
Minimum trace width	0.008 inch
Minimum clearance between copper features	0.008 inch
Minimum distance from copper to edge of PCB	0.014 inch
Minimum annular ring width (pins)	0.007 inch
Minimum annular ring width (vias)	0.005 inch
Minimum silkscreen line width	0.008 inch

The **Design Rule Checker** (or **DRC**) is a tool that you can use to make sure that your project doesn't contain any violations of these rules. These are referred to as **DRC Errors**. Since your tutorial project may not actually contain any DRC Errors, I would suggest that you close your project and instead open

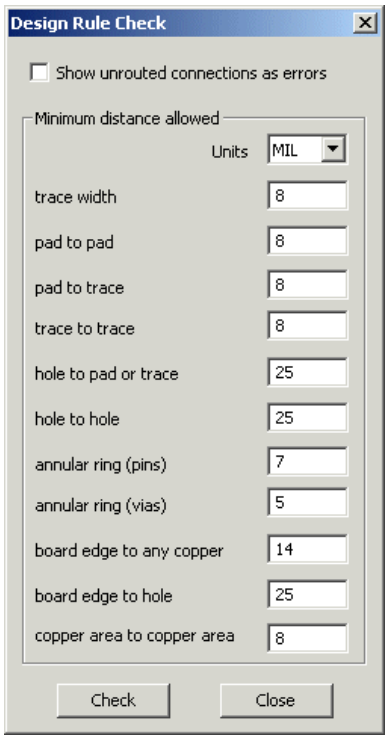
C:\freepcb\tutorial\motor_drc_errors.fpc, as shown below. This contains several typical errors. See if you can spot them.



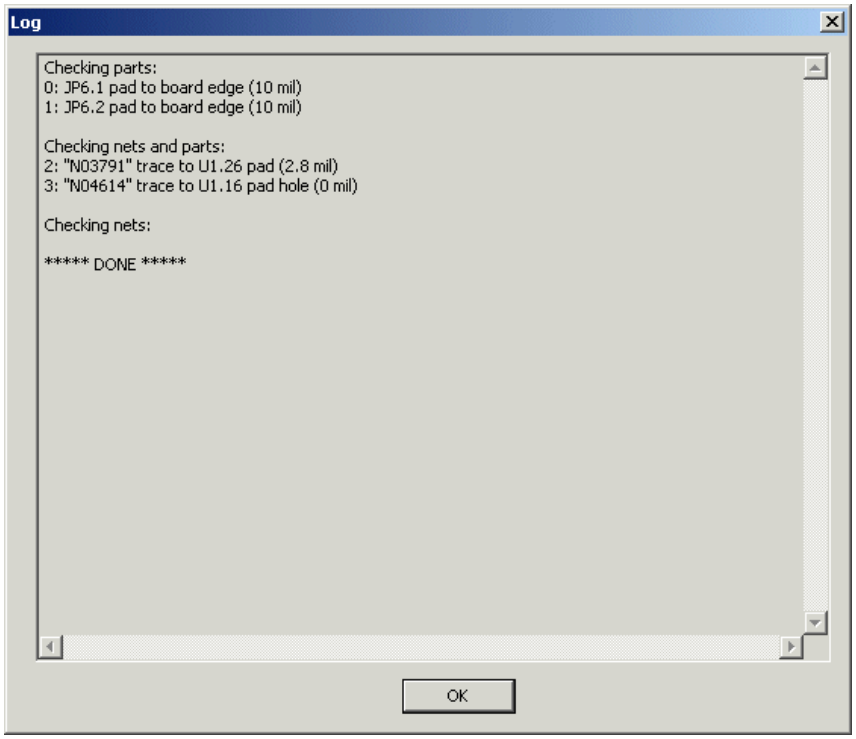
Now let's run the Design Rule Checker. You can launch it by selecting the **Tools > Design Rule Check** menu item. This will pop up the following dialog:



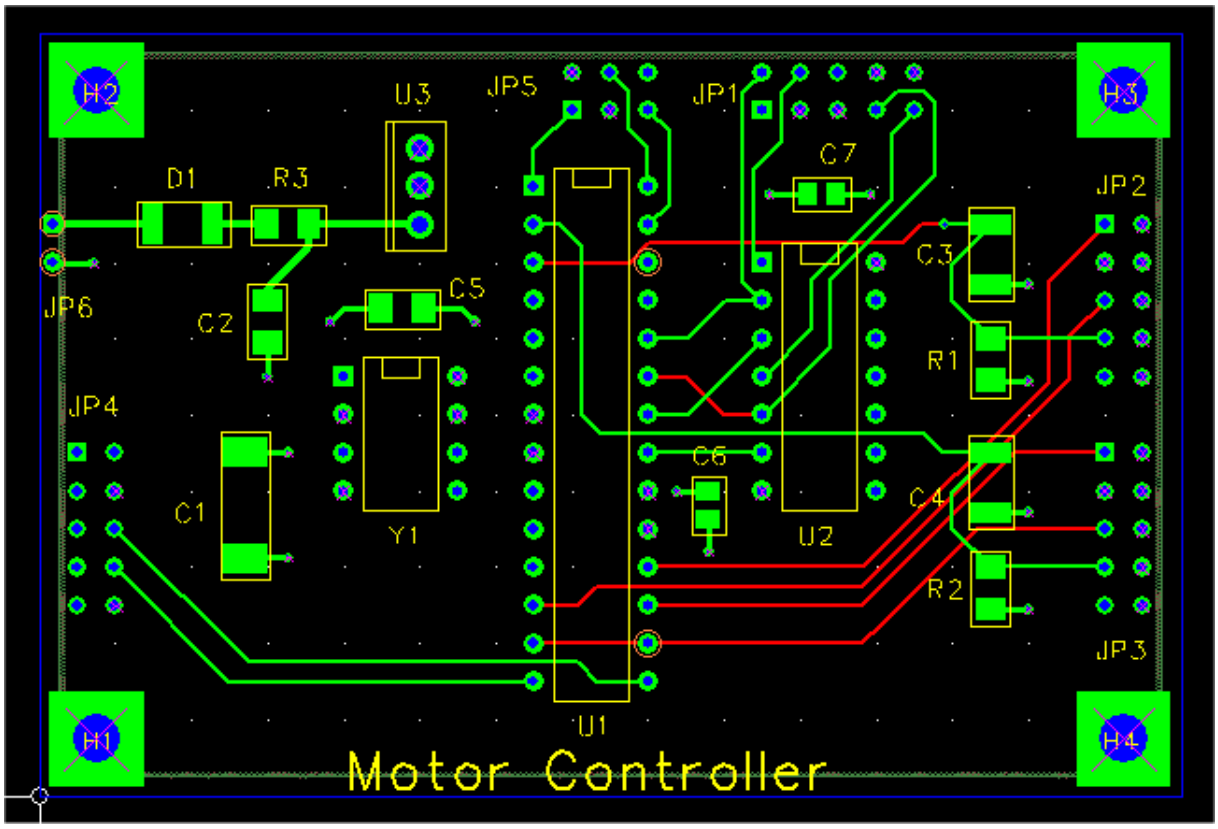
We will modify the fields in the dialog to match the Advanced Circuits values. **Trace width** should be set to 8 mils. The **pad to pad**, **pad to trace**, **trace to trace** and **copper area to copper area** values should be set to 8 mils. The **annular ring (pins)** values should be set to 7 mils, and the **annular ring (vias)** value should be set to 5 mils. The **board edge to any copper** field should be set to 14 mils. The **hole to pad or trace**, **board edge to hole** and **hole to hole** values are not given, so let's use 25 mils which seems reasonable. Your dialog should look like:



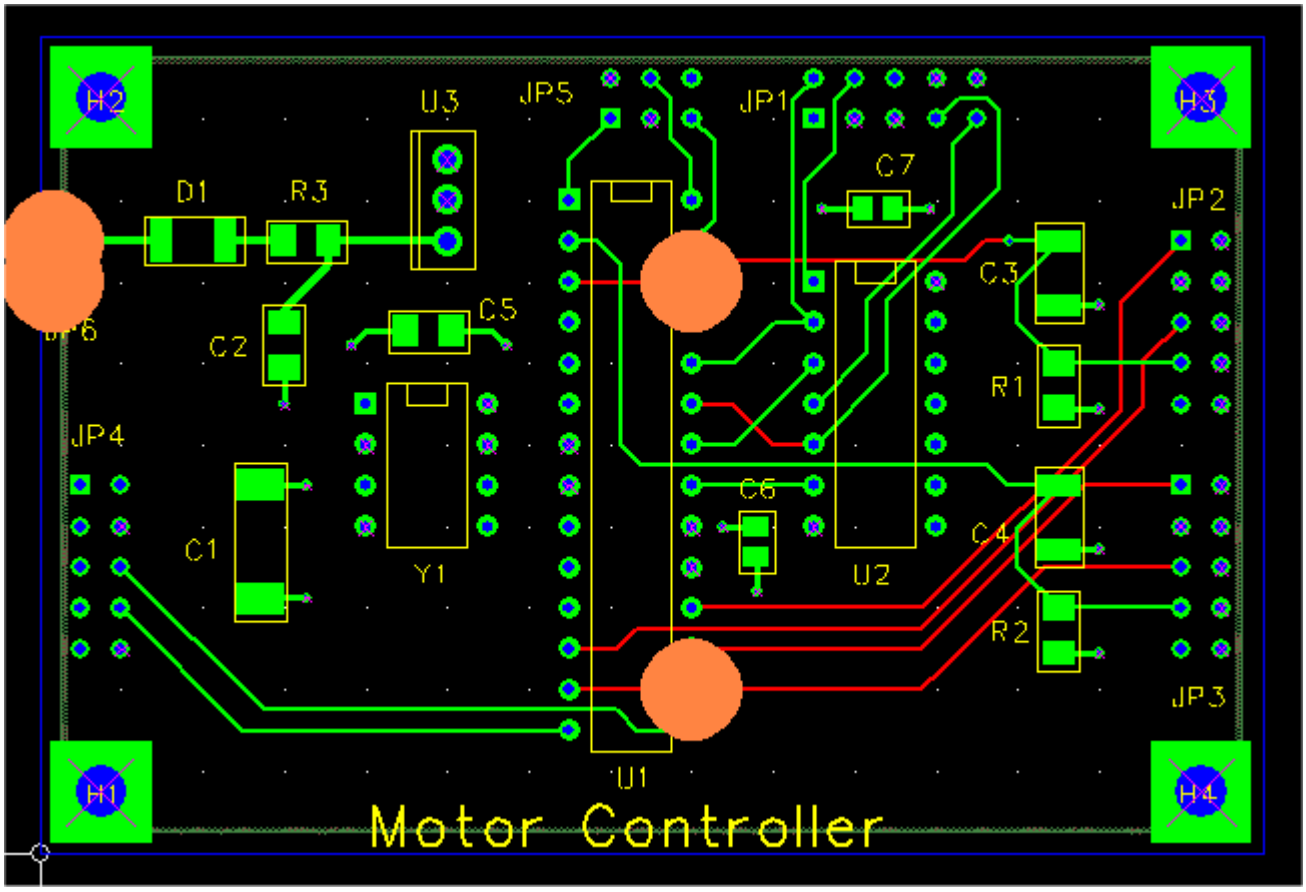
Now click OK to run the checker. A new dialog should pop up with a list of errors, as shown below.



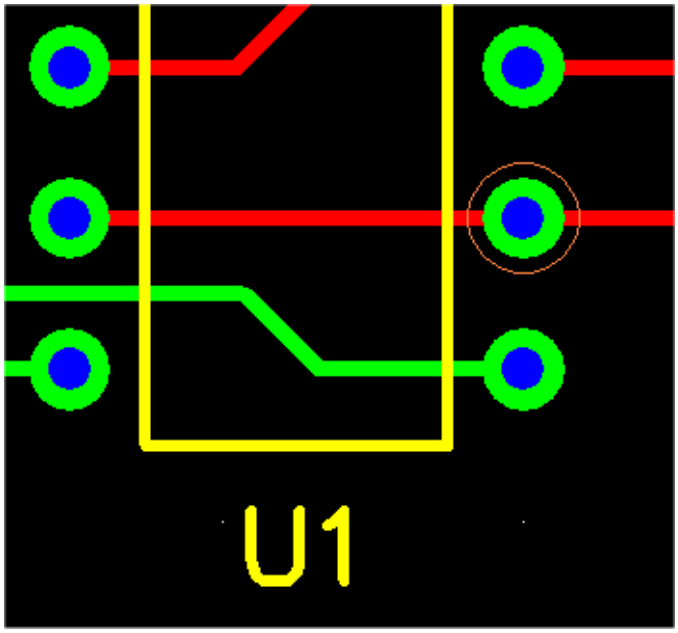
Errors 0 and 1 are violations of the minimum clearance between pads and the board edge. Errors 2 and 3 are trace-to-pad violations. If you look closely at the layout window, you will see that symbols consisting of small orange rings have been placed at the site of each error, as shown below.



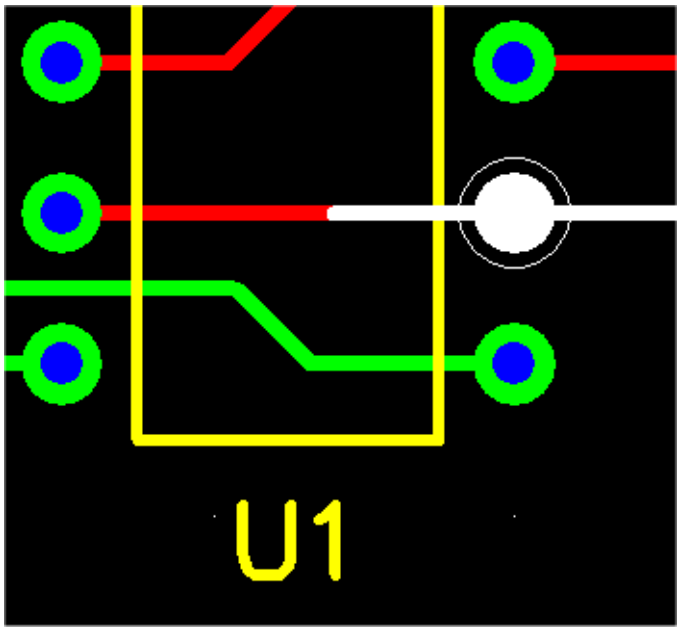
These small circles may be hard to see, but if you press the "d" key on the keyboard they become much larger solid circles, as shown.



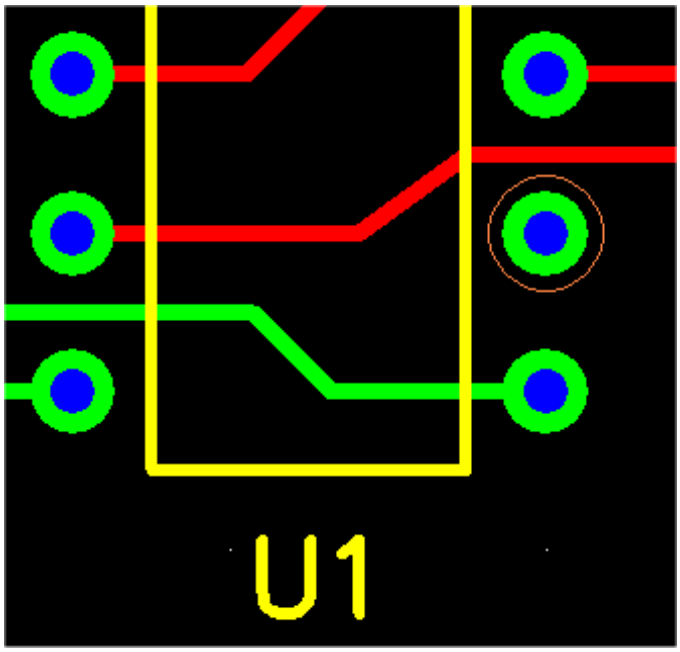
Now zoom in on one of the errors, such as the one at U1.16.



Click on the DRC Error ring to highlight it. The PCB elements that caused the violation will also be highlighted, as shown. Also, a description of the error will appear in the status bar.

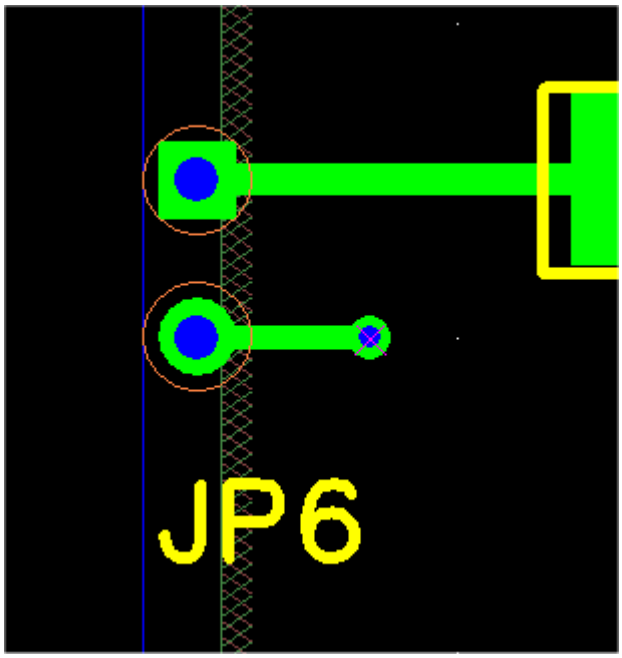


In this case, the error is a clearance violation between pad U1.16 and the trace segment passing through it, which are not on the same net. Oops!

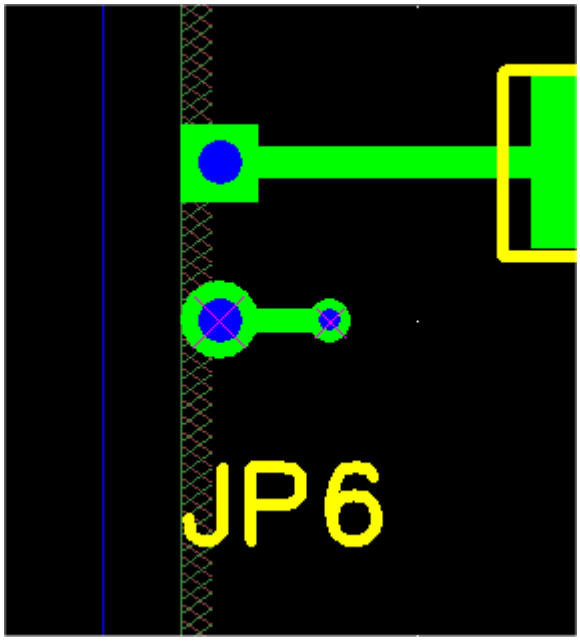


You can fix the error by rerouting the trace to avoid the pad, as shown. Note that the DRC Error symbol doesn't automatically disappear, but you can delete it by selecting it and pressing the "Delete" key, or by re-running the Design Rule Checker.

Now let's zoom in on the DRC Errors on the left side of the board, and select one of them.



In this case, the error is a pad-to-board-edge violation. Basically, JP6 is too close to the edge of the board. We can solve the problem by moving it further to the right, as shown.



I will leave it up to you to find and fix the last error. Then you can run the Design Rule Checker again, which should not find any errors.

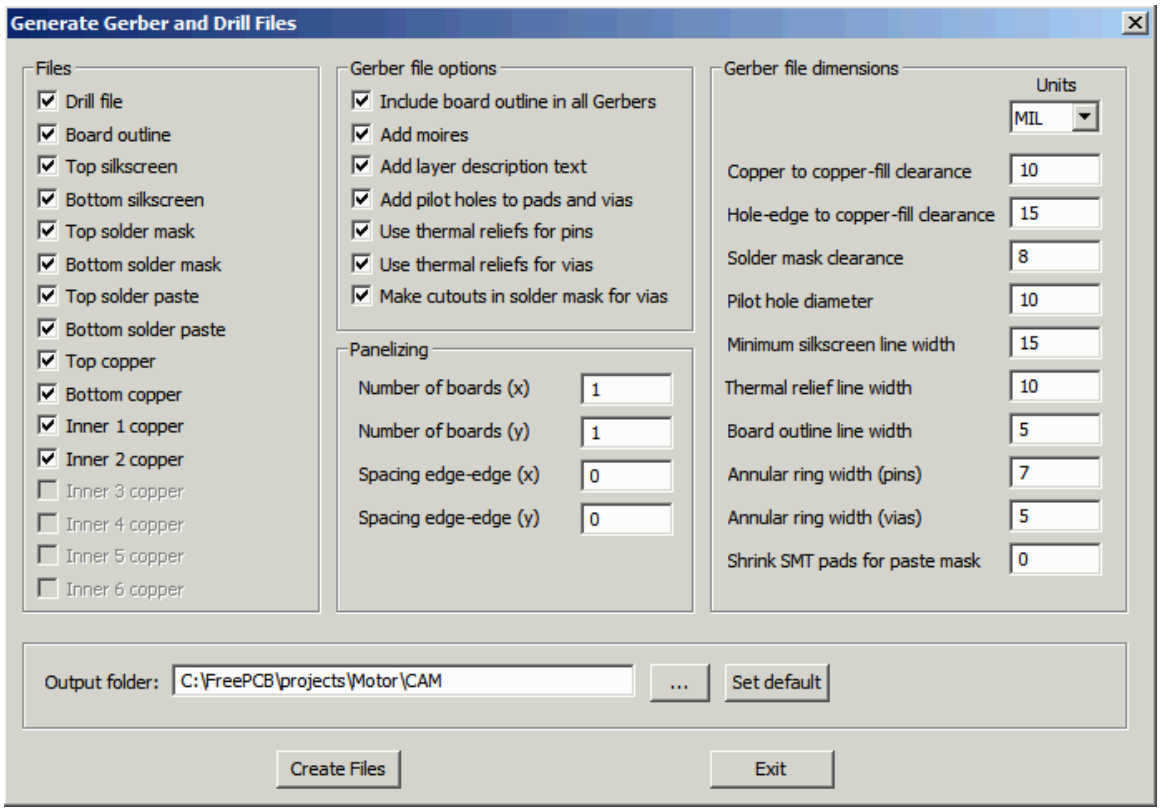
You should reload your **motor.fpc** project before beginning the next section.

When you close the **motor_drc_errors.fpc** project, I would suggest that you do NOT save your changes, so you or someone else can use it again.

7.12 Making Gerber and Drill Files

The final step in this tutorial is making the **Gerber files** and **Drill file** which you would send to the PCB manufacturer. You might want to review **Section 5.19: Exporting Drill and Gerber Files** before proceeding.

Select **Generate CAM files...** from the **File** menu. "CAM" is an acronym for "Computer-Aided Manufacturing". The following dialog should appear.



In the **Files** section, select the files that you wish to create by checking or unchecking the boxes next to them.

In the **Gerber file options** section, you can select:

- **Include board outline** - add the board outline to each Gerber file
- **Add moires** - add moire patterns (also called "targets") to each Gerber file
- **Add layer description text** - add a text string describing the layer to each Gerber file
- **Add pilot holes** - add pilot holes to pads and vias on the top and bottom layers
- **Use thermal reliefs for pins** - use thermal reliefs to connect through-hole pins to copper areas
- **Use thermal reliefs for vias** - use thermal reliefs to connect vias to copper areas
- **Make cutouts in solder mask for vias** - make openings for via pads in the solder mask layers

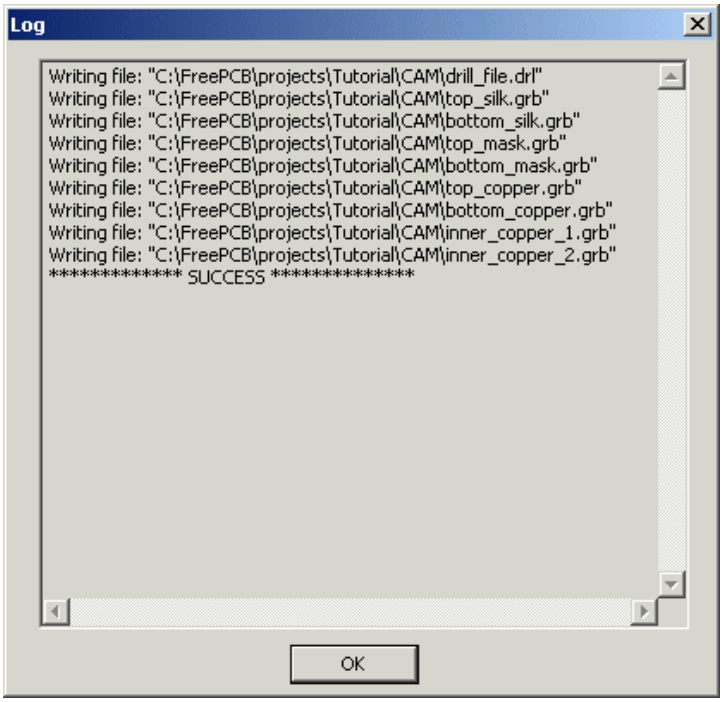
In the **Panelizing** section you can select how many copies to be produced in each direction and the spacing between copies.

The **Gerber file dimensions** section allows you to set numeric parameters for the Gerber files. When you make a real board, you would get these from the PCB manufacturer. The values shown above are fairly typical of a low-cost, multilayer process.

- **Copper to copper-fill clearance** - the clearance that will be created around traces and vias that pass through copper areas.
- **Hole-edge to copper-fill clearance** - the clearance that will be created around holes through copper areas.
- **Solder mask clearance** - the clearance that will be created around pads and vias in the solder masks.
- **Pilot hole diameter** - the diameter of pilot holes in pads, if you chose to include these.
- **Minimum silkscreen stroke width** - the minimum line width that will be used for items on the silkscreen layers.
- **Thermal relief line width** - the width of the lines in thermal reliefs, which connect pads to copper areas.
- **Board outline line width** - the width of the lines used to draw the board outline.
- **Annular ring width (pins)** - the width of annular rings placed around inner layer pin holes for thermal relief connections.
- **Annular ring width (vias)** - the width of annular rings placed around inner layer via holes for thermal relief connections.
- **Shrink SMT pads for paste mask** – paste mask pad size reduction.

The **Output folder** is the destination folder for the files which will be created. By default, this is a subfolder of the project folder named "CAM". You can change it if you like. The folder will be created if it doesn't already exist (although its parent folder must exist).

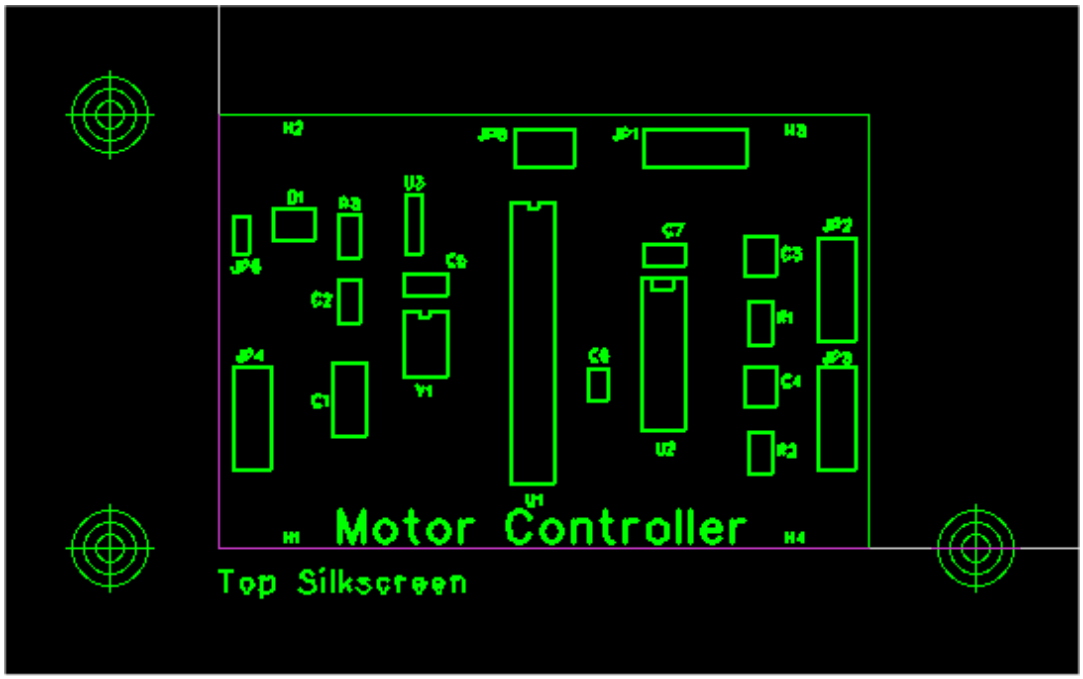
Now click **Create files** to make the files. The following dialog should appear.

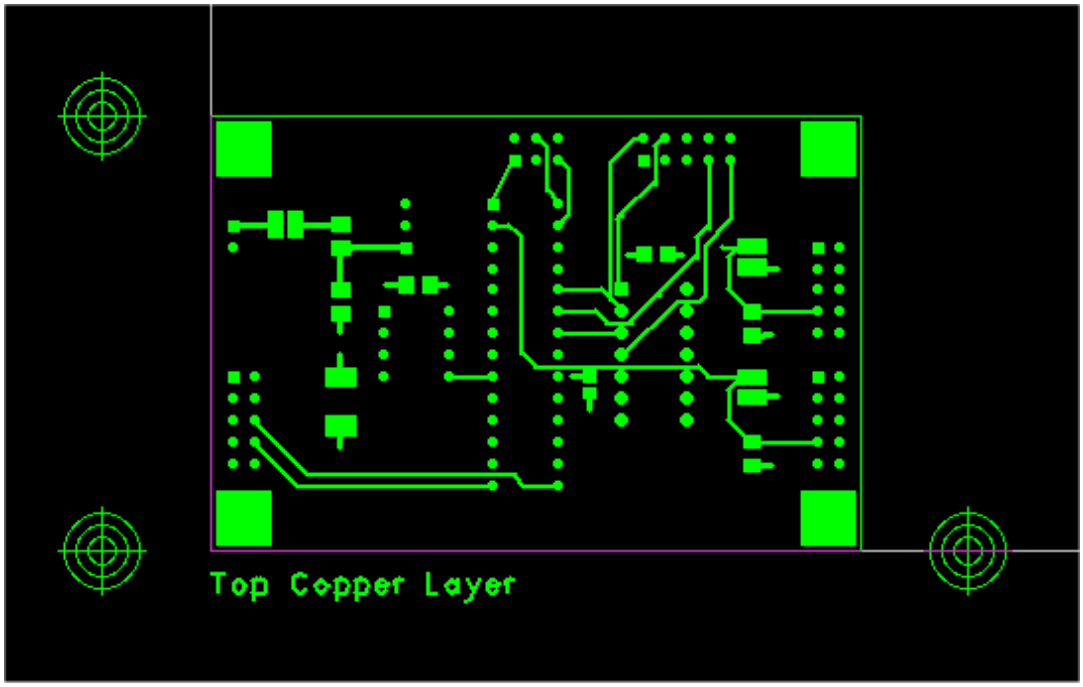


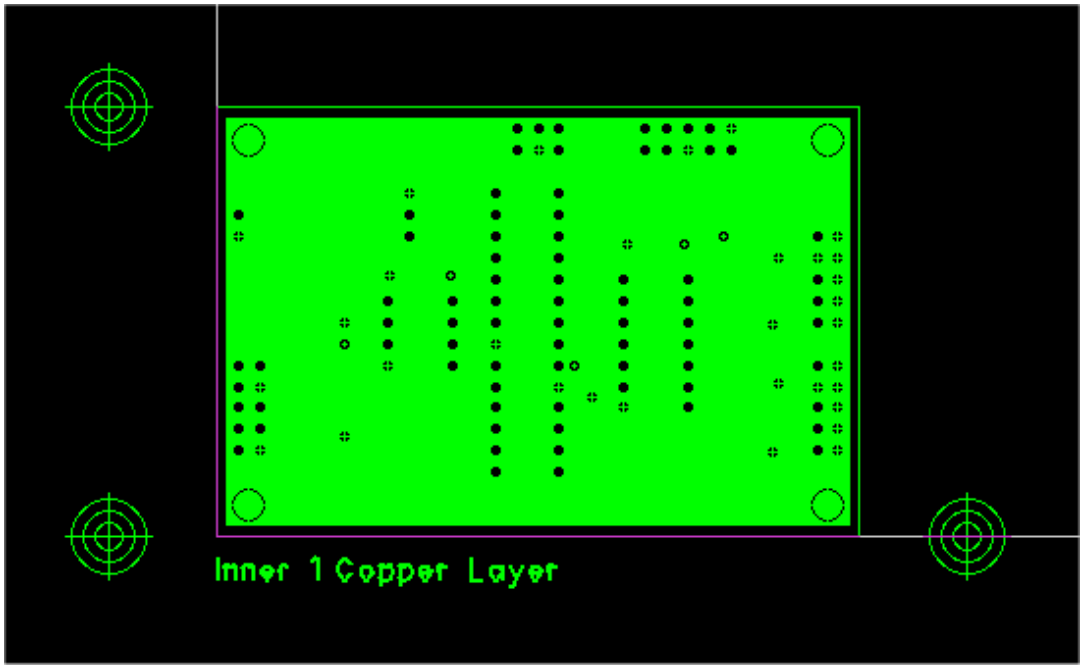
If there were any errors, messages will appear in the dialog. Usually, an error will result in FreePCB aborting that file and going on to the next one. Click **OK** to dismiss the dialog.

Now you can view your Gerber and drill files using a Gerber Viewer such as **ViewMate** by **PentaLogix** (which is free, and available here). ViewMate can also print **check plots** on your printer. You should **ALWAYS** check your files before using them to make a PCB. Also, read **Section 5.20.3: Drill Sizes** for important information about the drill file.

Screenshots of the Gerber files for the top silk-screen, top copper and inner 1 copper layers of our tutorial project are shown below. These look pretty crude in the screenshots. To appreciate just how precise they really are, you should open your own files in ViewMate and zoom in on some of the smaller features.







Well, that's the end of the tutorial. I have tried to cover most of the major features of FreePCB. Now you should be ready to create a board of your own.

8. File Formats

8.1 Project File

The **project file** (*.fpc) is an ASCII text file containing all of the information for a single project. It is composed of multiple **sections**, where each section starts with a name surrounded by square brackets, such as "[options]". Within each section, there are multiple **items** where each item starts with a colon-delimited **keyword**, such as "units:" There will usually be one or more **parameters** following each keyword. There may also be items nested below other items. This relationship is shown by indenting the lines for the nested items.

The parameters that follow a keyword are strings delimited by whitespace. In order to allow whitespace within a parameter, such as a space within a filename, the first parameter following a keyword may be surrounded by double-quotes (e.g. "test file"), and whitespace between the quotes will not be considered a delimiter. Subsequent parameters cannot contain embedded whitespace.

Many of the parameters represent dimensions, such as line lengths or widths. By default, dimensions are expressed in nanometers. A suffix may be added to a dimension to indicate other units, such as "100MM" or "23.5MIL". For footprints, there is an item called "units:" which changes the default units for all of the footprint parameters, so that suffixes are not required.

Here is a list of all of the sections and items that may appear in a project file. For each item, the keyword is shown, followed by a list of parameters where the description of each parameter is surrounded by "< >", optional items are surrounded by “{ }”. This list is current as of software version 1.327.

[options]

version: <the version number of the freepcb.exe application that created the file, such as "1.328">

file_version: <the version number of the oldest freepcb.exe application that can read the file, such as "1.312">

project_name: <the name of the project, usually the same as the project file name without the ".fpc" extension>

full_library_folder: <the absolute path to the default library folder for the project>

CAM_folder: <the absolute path to the folder for CAM files , or "" if not yet defined>

ses_file_path: <the absolute path to the last .ses file imported , or "" if not yet defined>

dsn_bounds_poly: <the board outline index used for the bounds for the .dsn file>

dsn_signals_poly: <the board outline index used for the bounds for signals the .dsn file>

autosave_interval: <the interval between autosaves in msec, or "0" if autosave disabled>

netlist_import_flags: <the flags set in the last "import netlist" dialog>

units: <the currently-selected units for the project, either "MM" or "MIL">

visible_grid_spacing: <the currently-selected visible grid spacing, in nanometers>
    visible_grid_item: <the first item in the menu of visible grids>
    visible_grid_item: <the next item in the menu of visible grids>
      ...
    visible_grid_item: <the last item in the menu of visible grids>

placement_grid_spacing: <the currently-selected placement grid spacing, in nanometers>
    placement_grid_item: <the first item in the menu of placement grids>
    placement_grid_item: <the next item in the menu of placement grids>
      ...
    placement_grid_item: <the last item in the menu of placement grids>

routing_grid_spacing: <the currently-selected routing grid spacing, in nanometers>
    routing_grid_item: <the first item in the menu of routing grids>
    routing_grid_item: <the next item in the menu of routing grids>
      ...
    routing_grid_item: <the last item in the menu of routing grids>

snap_angle: <the currently-selected snap angle in degrees>

fp_visible_grid_spacing: <the currently-selected visible grid spacing for the Footprint Editor, in nanometers>

fp_visible_grid_item: <the first item in the menu of visible grids for the Footprint Editor>
fp_visible_grid_item: <the next item in the menu of visible grids for the Footprint Editor>
...
fp_visible_grid_item: <the last item in the menu of visible grids for the Footprint Editor>

fp_placement_grid_spacing: <the currently-selected placement grid spacing for the Footprint Editor, in nanometers>
fp_placement_grid_item: <the first item in the menu of placement grids for the Footprint Editor>
fp_placement_grid_item: <the next item in the menu of placement grids for the Footprint Editor>
...
fp_placement_grid_item: <the last item in the menu of placement grids for the Footprint Editor>

fp_snap_angle: <the currently-selected snap angle for the Footprint Editor, in degrees>

fill_clearance: <for Gerber files, the copper-copper clearance for copper areas>

mask_clearance: <for Gerber files, the solder-mask clearance>

thermal_width: <for Gerber files, the width of the cross-hairs for thermal reliefs>

min_silkscreen_width: <for Gerber files, the minimum width of silk-screen lines>

board_outline_width: <for Gerber files, the width of board-outline lines>

hole_clearance: <for Gerber files, clearance from holes to copper areas>

pilot_diameter: <for Gerber files, the pilot hole diameter>

annular_ring_for_pins: <for Gerber files, the width of annular rings for through-hole pins>

annular_ring_for_vias: <for Gerber files, the width of annular rings for vias>

shrink_paste_mask: <for paste mask Gerber files, pad size reduction>

cam_flags: <bitwise ORed flags for CAM options>

cam_layers: <bitwise ORed flags for CAM files to generate>

cam_drill_file: <flag to make a drill file>

cam_units: <units to use for CAM parameters in dialog>

cam_n_x: <for panelizing, number of boards horizontally>

cam_n_y: <for panelizing, number of boards vertically>

cam_space_x: <for panelizing, horizontal spacing of boards>

cam_space_y: <for panelizing, vertical spacing of boards>

drc_check_unrouted: <flag to treat unrouted connections as DRC errors>

drc_trace_width: <minimum trace width for DRC>

drc_pad_pad: <minimum pad-pad clearance for DRC>

drc_pad_trace: <minimum pad-trace clearance for DRC>

drc_trace_trace: <minimum trace-trace clearance for DRC>

drc_hole_copper: <minimum hole-copper clearance for DRC>

drc_annular_ring_pins: <minimum annular ring width for pins for DRC>

drc_annular_ring_vias: <minimum annular ring width for vias for DRC>

drc_board_edge_copper: <minimum copper-board_edge clearance for DRC>

drc_board_edge_hole: <minimum hole-board_edge clearance for DRC>

drc_hole_hole: <minimum hole-hole clearance for DRC>

drc_copper_copper: <minimum copper_area-copper_area clearance for DRC>

default_trace_width: <default trace width for project>

default_via_pad_width: <default via pad width for project>

default_via_hole_width: <default via hole diameter for project>

n_width_menu: <number of trace width menu items>
    width_menu_item: <index of first item> <trace width> <via pad width> <via hole diameter>
    width_menu_item: <index of this item> <trace width> <via pad width> <via hole diameter>
      ...
    width_menu_item: <index of last item> <trace width> <via pad width> <via hole diameter>

n_copper_layers: <number of copper layers in project>
    layer_info: <name of first layer> <index into layer array> <color: red> <color: green> <color: blue> <flag if visible>
    layer_info: <name of next layer> <index into layer array> <color: red> <color: green> <color: blue> <flag if visible>
      ...
    layer_info: <name of last layer> <index into layer array> <color: red> <color: green> <color: blue> <flag if visible>

[footprints]

name: <name of footprint>
    {author: <name of author>}
    {source: <description of source>}
    {description: <description of footprint>}
    units: <units: "NM", "MM" or "MIL">
    sel_rect: <for selection rectangle: left> <bottom> <right> <top>
    ref_text: <for reference text: height> <left> <bottom> <angle> <line width>
    {text: <text string> <height> <x> <y> <angle> <line width> <flag if mirrored> <layer>}
      ...
    {outline_polyline: <line width> <starting x> <starting y>
        next_corner: <next x> <next y> <side style>
        next_corner: <next x> <next y> <side style>
          ...
        close_polyline: <if closed polyline, last side style>}
      ...
    n_pins: <number of pins>
        pin: <pin name> <hole_diameter> <x> <y> <angle>
            top_pad: <shape> <width> <length/2> <length/2> <corner radius>
            inner_pad: <shape> <width> <length/2> <length/2> <corner radius>
            bottom_pad: <shape> <width> <length/2> <length/2> <corner radius>
        pin: <pin name> <hole_diameter> <x> <y> <angle>
            top_pad: <shape> <width> <length/2> <length/2> <corner radius>
            inner_pad: <shape> <width> <length/2> <length/2> <corner radius>
            bottom_pad: <shape> <width> <length/2> <length/2> <corner radius>
              ....
        pin: <pin name> <hole_diameter> <x> <y> <angle>
            top_pad: <shape> <width> <length/2> <length/2> <corner radius>
            inner_pad: <shape> <width> <length/2> <length/2> <corner radius>
            bottom_pad: <shape> <width> <length/2> <length/2> <corner radius>

[board]
outline: <number of corners in board outline> <outline index>
    corner: <index> <x> <y> <style of next side>
    corner: <index> <x> <y> <style of next side>
      ...
    corner: <index> <x> <y> <style of next side>

  ...

[solder_mask_cutouts]

sm_cutout: <number of corners in solder mask cutout> <fill style> <layer>
    corner: <index> <x> <y> <style of next side>
    corner: <index> <x> <y> <style of next side>
      ...
    corner: <index> <x> <y> <style of next side>

  ...

[parts]

part: <reference designator>
    ref_text: <height> <line width> <angle> <x> <y>
    package: <package identifier from netlist file, or "">
    shape: <footprint name>
    pos: <position: x> <y> <side> <angle> <flag if glued>

[nets]

net: <name of net> <# pins> <# connections> <# areas> <default trace width> <default via pad width> <default via hole dia> <visibility>
    pin: <index of first pin in net> <name, such "U1.5">
    pin: <index of next pin in net> <name, such "U1.5">
      ...
    pin: <index of last pin in net> <name, such "U1.5">
    connect: <index of this connection> <index of start pin> <index of end pin> <# segments> <flag if locked>
        vtx: <first trace vertex: index> <x> <y> <first/last vertex, pad layer> <force via flag> <via pad width> <via hole dia> <tee id>
        seg: <first trace segment: index> <layer> <width> <via width> <via hole diameter>
        vtx: <next trace vertex: index> <x> <y> <first/last vertex, pad layer> <force via flag> <via pad width> <via hole dia> <tee id>
        seg: <next trace segment: index> <layer> <width> <via width> <via hole diameter>
          ...
        seg: <last trace segment: index> <layer> <width> <via width> <via hole diameter>
        vtx: <last trace vertex: index> <x> <y> <first/last vertex, pad layer> <force via flag> <via pad width> <via hole dia> <tee id>
    area: <index of this copper area> <# corners> <layer> <hatch style>
        corner: <index of first corner> <x> <y> <style of next side> <flag if last corner of this contour>
        corner: <index of next corner> <x> <y> <style of next side> <flag if last corner of this contour>
        ...
        corner: <index of last corner> <x> <y> <style of next side> <flag if last corner of this contour>

[texts]

text: <text string> <x> <y> <layer> <angle> <mirror flag> <height> <line width>

8.2 Library Files

A **library file** (*.fpl) is an ASCII text file containing one or more footprints. The format for this file is exactly the same as the [footprints] section of the project file, described above.

8.3 Configuration File

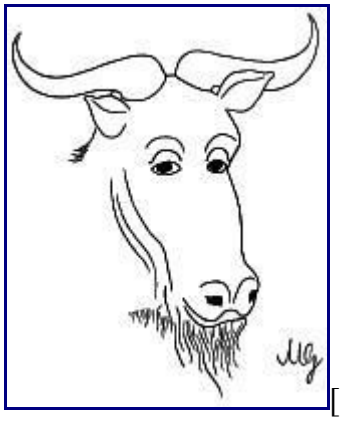
The **configuration file** (default.cfg) is an ASCII text file containing the default parameters that are used for new projects. It is located in the same folder as the executable "freepcb.exe". It uses exactly the same format as the [options] section of the project file, as described above, with 2 extra items:

parent_folder: <the absolute or relative path to the default parent folder for new projects>
library_folder: <the absolute or relative path to the default library folder for new projects>

9. License

Copyright notice:

GNU General Public License



[Czech | English | Japanese]

- *What to do if you see a possible GPL violation*
- *Translations of the GPL*
- *GPL Frequently Asked Questions*
- The GNU General Public License (GPL) in plain text format
- The GNU General Public License (GPL) in Texinfo format
- The GNU General Public License (GPL) in LaTeX format
- The GNU General Public License (GPL) as an appendix in DocBook format

---

Table of Contents

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and

the date of any change.

- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Program

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `` `show w' `` and `` `show c' `` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `` `show w' `` and `` `show c' ``; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
