

# PCB design tutorial with Eagle

## ***Introduction***

This exercise covers the use of Eagle (Easily Applicable Graphical Layout Editor) PCB design software to design an electronic schematic and lay out a printed circuit board (PCB). Eagle is a PCB design software package consisting of a schematics editor, a PCB editor and an autorouter module. The software comes with an extensive library of components, but a library editor is also available to design new parts or modify existing ones. Eagle is made by CadSoft (<http://cadsoft.de>), and is available in three versions. The light version is limited to one sheet of schematics and half eurocard format (80x100 mm), but can be used under the terms of the freeware licence for non-commercial use. This software can be downloaded from CadSoft's homepage, for Windows or Linux.

The goal of the exercise is to create a schematic and PCB layout for a system consisting of an Atmel AtMega8 microcontroller with RS-485 and RS-232 interfaces, a couple of LEDs and buttons, and with remaining I/O-pins connected to headers.

## ***The Control Panel***

Start Eagle (type "eagle" from command line).

The "Eagle Control Panel" should appear. This window is the main project manager in Eagle, and is used to open new schematics, layouts and libraries.

Create a new project named efp. Make sure the project is saved in a place where you have write access (not in the Eagle directory). You can create a description of the project by right-clicking on the project and choosing "Edit Description".

Now, right-click on the project and select "New → Schematic". This will bring up the Schematic Editor Window.

## ***Drawing the schematic***

This section describes how to draw the schematic, consisting of a power supply, an Atmel AtMega8, RS-232 and RS-485 interfaces and remaining I/O-pins connected to headers. The entire schematic can be found on page 15. This schematic should be consulted along the way to make sure you have enough room for drawing the schematic (or you will have to adjust and move the different subsystems around). The tutorial is divided into several subsections, each describing how to draw a particular subsystem of the circuit. Before we start drawing the schematic, let's look at some of the basics of editing with Eagle. The Eagle toolbar is shown on the right.

The user interface in Eagle is somewhat special when compared to other drawing utilities (and PCB layout programs). This takes a little getting time getting used to. Some of the tools will be described here, to allow the user to get to know these tools, while the tools that constitute the main part of the tutorial will be described along the way. The copy -tool can be used to easily clone a component. If you select copy and click on a component, a copy of the component will be attached to the mouse cursor, and can be placed in the schematic. If you want to copy something to a different schematic, you will need to use the cut tool. This does not delete the component from the schematic (as you might otherwise assume from the name), but merely copies it to the clipboard.



The group tool can be used to work on a group of components etc. First select the group tool and mark the components you want to modify. You can either hold the left button and drag to draw a rectangular selection, or click the left mouse button to make a polygon selection, using the right mouse button to end the polygon selection. When the selection is done, you select the tool you wish to apply, such as move, rotate, cut etc.

Then right-click the group to use the selected tool.

The change tool is used to modify the properties of various objects. Again, this is a little different in Eagle when compared to other tools (where you would normally be able to right-click on an object and change its properties from a pop-up menu). First you choose the modify -tool and select what you want to modify (style, size, layer etc.), then you click on the component you want to modify. The command line interface (CLI) can be used to make this task easier. If you want to modify the value of say 10 capacitors to 100nF, you could use the change -tool and select value. Now, each time you click a component, a dialog will pop up asking for the new value, which you will have to type in. If you instead enter the command value 100nF in the CLI (the input-box just above the main drawing canvas), you can simply click on the components whose value you wish to change.

When adding components, you will notice a small black cross on each device. This is the origin or “handle” of the device, and is used to manipulate the device with various tools. So whenever you are using a tool, Eagle will apply the tool to the entity whose origin is closest to the mouse cursor. If two or more entities are very close to each other, Eagle will highlight one and ask if this is the one you want to modify. Click left button to accept or right button to cycle to the next entity. When you use the smash -tool, the name and value-texts will be detached from the device and get their own origin, allowing them to be moved individually.

### ***Adding a frame***

The first thing that should be done is to save the empty schematic. This is done to let eagle know the name of the schematic and extra files that will be created along the way. Now add a frame to the schematics. Click the Add-button . The “add”-window pops up. Select the frame named DINA4 L from the frames-library. You can use the search-box at the bottom to search all libraries for the frame. In the search-box, the wildcard character \* can be used. Click OK, and the component (frame) will be attached to the mouse cursor.

Place the frame at the coordinates (0,0) which is marked by the black cross in the schematic editor. After adding a component, the default action is to add another component of the same type. Hitting ESC once takes you back to the Add -dialog. ESC once more takes you back to the schematics editor, leaving add mode.

### ***Zoom and keybindings***

After adding the frame, select zoom to fit. The default keybinding for this is Alt-F2. If you are using Linux, this key combination may not work as expected. In this case, you should define a different key-binding. Select Assign from the Options -menu. The assign-window shows the keybindings. Click new and select the key “F” and the modifier “CTRL”, then type the assigned command “window fit”. The commands assigned here are the same as the commands that can be typed in the Eagle command line interface.

Go back to the schematic editor and try the new keybinding. Also try executing the command by typing “window fit” on the command-line (the input-box in the top part of the schematic editor window). Note that command history is available in the command line interface, by pressing the arrow-keys. It may also be a good idea to assign keybindings for the normal cut, copy and paste commands, as well as an alternative keybinding (Ctrl-L) for redrawing the screen:

```
CTRL-C Copy
CTRL-X Cut
CTRL-V Paste
CTRL-L window;
```

The next sections will take you through creating each of the subsystems.

### ***Power supply***

Start drawing the power supply in the upper left corner of the frame, according to Figure 1.

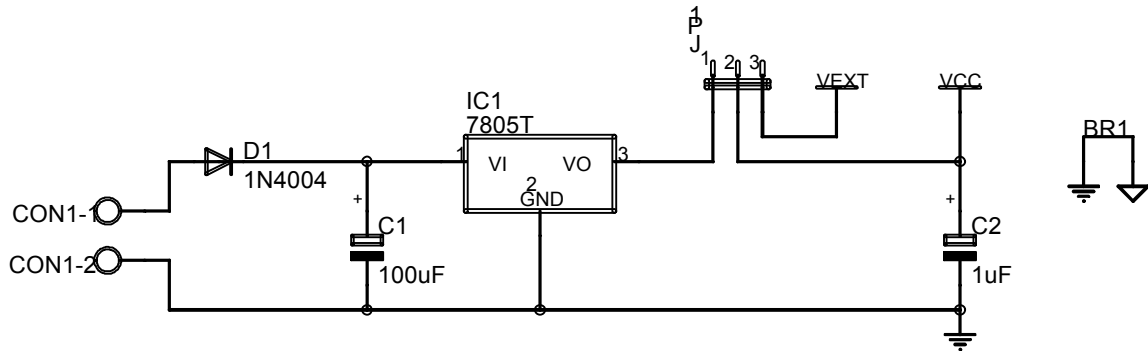


Figure 1: The power supply schematic

Use the add -command to add the following components (these might not be exactly the same in the Eagle libraries you use). Then use the net command to draw the connections.

Part	Value	Device	Package	Library
C1	100uF	CPOL-EUE 2.5-6	E2,5-6	rcl
C2	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
CON1		W237-02P	W237-132	con-wago-508
D1	1N4004	1N4004	DO41-10	diode
IC1	7805T	7805T	TO220H	linear
JP1		JP2E	JP2	jumper
BR1		BRIDGE		supply
GND1		GND1		supply
VCC		VCC		supply
VEXT		VEXT		supply
AGND		AGND		supply

The last four “components” are net identifiers. These make the schematic more readable, by eliminating a lot of wires for the supply connections. The GND1 symbol uses the net name GND, and will therefore connect to other components with hidden pins with this name. This means that some components will be connected to VCC and GND although there are no visible pins connected to these nets. VCC, VEXT and AGND have net-names equal to the symbol name.

Later in this tutorial you will need to add these net identifiers to other parts of the schematic, so you will need to remember how to find the symbols again. If the library supply.lbr is not available, the default Eagle-libraries supply1.lbr and supply2.lbr provide similar functions, but use somewhat different symbols. If you use these libraries, it is important to use the symbols named GND and not GND1, since the net name of these symbols are different. Also the BRIDGE component is not available in other libraries. A jumper can be used instead, although it will have a different footprint in the PCB.



Use the *name* command to rename components (only needed for CON1).



Use the *value* command to change a component’s value. Do this for capacitors C1 and C2.



The *smash* command can be used to detach the name and value texts from the component, and allow them to be positioned with the *move* command.

All commands in Eagle can be executed from the command line interface. For instance to add a component, you can type *add* in the command input-box. Most commands can also be called with one or more

arguments, so adding the GND1-symbol can be achieved by typing `add gnd1@supply`. The argument need not contain the name of the library to use, and can also contain the wildcard character \*. If more than one component matches the request, the add-command will bring up the add-window to allow you to select the needed one. Try typing `“add gnd*”` or `“add vcc”` in the command input-box. The commands can also be abbreviated to a shorter form (as long as it is unique), so for instance `“va 22pF”` means the same as `“value 22pF”`.

The CLI-command `help` can be used to bring up the online help window, describing the parameters of a particular command, so if you want to know what arguments can be passed to the value-command, you can type `help value` in the CLI input-box. Knowing the CLI commands used for various operations is also quite useful for creating keyboard shortcuts.

## **Microcontroller**

Let’s get on to adding the microcontroller. This exercise will be using an Atmel AtMega8. Unfortunately the library named `“atmel”` does not contain the needed part, but a library called `“avr.lbr”` has been contributed to the Eagle homepage, and can be downloaded from <http://cadsoft.de/> under Download. CadSoft’s homepage also contains many other libraries that may come in handy when doing other projects with Eagle. Remember to add the library if it is not in the default library path. You can also modify the library path from the Eagle Control Panel’s *Options* → *Directories*. . . *-menu*.

Add the microcontroller (`add MEGA8-P@avr`) below the power supply. Now add the rest of the components and nets for the analog power supply, the oscillator and the reset switch and programming connector. Remember you can use the smash -tool to move the name and value-texts to more appropriate positions. If you are having trouble placing the text precisely, you can use the alternative grid (by pressing ALT while using the mouse). In the schematics editor the normal grid should always be 100mil (0.1 inch), and should always be used for placing the components. Since all the library components are designed with this grid size, this will ensure that the pins are aligned on this grid, allowing the nets to be connected correctly.

When placing or moving a component, the right mouse button can be used to rotate the device. Use the mirror -command to change the appearance of CON2, so the pins are numbered correctly.

<b>Part</b>	<b>Value</b>	<b>Device</b>	<b>Package</b>	<b>Library</b>
IC2	MEGA8-P	MEGA8-P	DIL28-3	avr
C3	100nF	C-EUC1206	C1206	rcl
C4	100nF	C-EUC1206	C1206	rcl
C5	100nF	C-EUC1206	C1206	rcl
C6	22pF	C-EUC1206	C1206	rcl
C7	22pF	C-EUC1206	C1206	rcl
CON2		M05	05P	con-amp-quick
L1	10uH	L-US0204/5	0204/5	rcl
R1	4k7	R-EU_M0805	M0805	rcl
S1	10-XX	B3F-10XX		switch-omron
XT1	14.7456MHz	XTAL/S	QS	special

When you are done the schematics should look similar to the one shown in Figure 2.

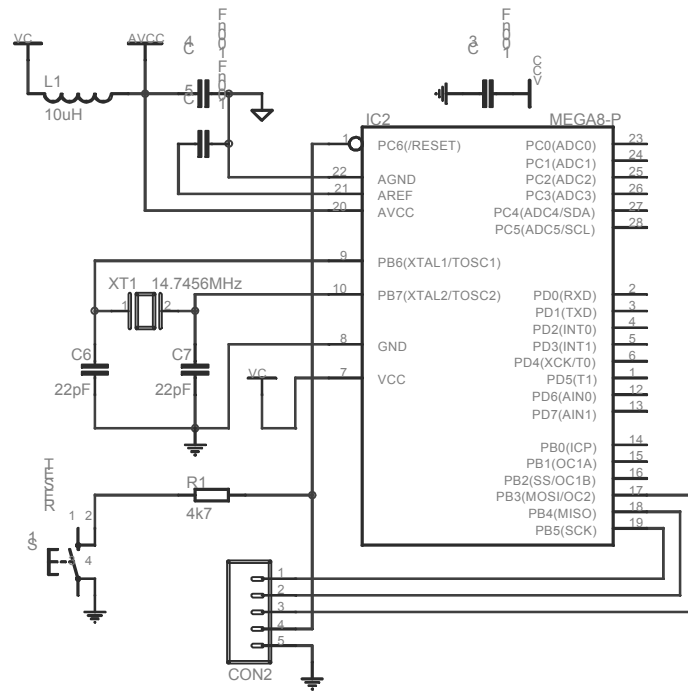


Figure 2: The microcontroller schematic

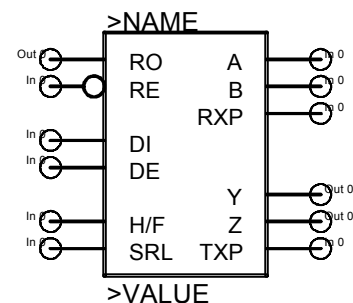
## Creating a library

The MAX3089 used for RS-485 communication cannot be found in the included libraries. You will therefore have to draw it yourself.

First we create a new library. In the Eagle Control Panel choose File → New → Library.

This opens the library editor with a new empty library. Since you probably do not have write access in the default library directory, you should immediately save the new (empty) library in a different location. This will tell Eagle to make any temporary files here as well. Save the library in your project directory as max.lbr. Go back to the control panel window and browse to Libraries → maxim → SO14. Right-click and select “copy to library”. This will import the SO14 package footprint into the open library.

Click the “symbol” button in the toolbar. In the “new”-textbox, type the name of the symbol “MAX3089”, and click OK. Confirm the creation of the new symbol. Draw a rectangle defining the component by using the “wire”-tool. Draw in layer 94 Symbols and make the rectangle 6×9 grid-points (using the default 100 mil/0.1 inch grid). Now add the pins according to the figure on the right. Select the “pin”-tool and ensure the pin-length is set to middle. Note that the circle at the end of the pin should point away from the device.



This is the end where the wire will be connected in the schematics. The pins will all be added as I/O (unless something else is selected from the drop-down box before adding the pin), and named P\$1 thru P\$10. Use the *name* tool to rename the pins. Remember to click on the circle at the end of the pin, although the name of the pin is at the other end.

Now use the change -tool to modify the pins direction according to the figure. Also add the dot to the RE-pin by using Change → Function → Dot. Finally use the *text* tool to add two strings. The string above the device should be the name of the device. The special string >NAME inserts the correct name. Similarly the text below the device should be >VALUE. You may need to use the alternate grid (ALT-key) to place the text right next to the device outline. Also change the layer of the texts to 95 (Names) and 96 (Values), by

selecting the correct layer from the drop-down box before placing the text (or by using change → layer afterwards).

Having the power-pins as a separate symbol allows these to be hidden in the schematics, which makes it more readable. Create a new symbol called VCC-GND. Add two pins, with a direction of “power” and name the pins VCC and GND. Save the library.

Now click on the Device -button in the toolbar and create a new device called MAX3089. Use the add -tool to add the two symbols you have created (MAX3089 and VCC-GND). Change the “addlevel” of the VCC-GND-symbol to request. This means the power-pins will not be added to the schematics unless you request it. The device should now look like Figure 3.

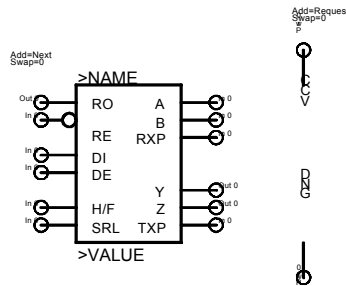


Figure 3: The MAX3089-device, including the MAX3089 and the VCC-GND symbols

Click the “New”-button in the lower right corner to add a new device variant for the created MAX3089-device. This brings up a new window, allowing you to select the footprint. Select the SO14 imported earlier and type SO as the variant name, then click OK.

Back in the library editor, the new variant has been added to the list in the righthand side of the window. Notice the yellow exclamation mark next to the variant. This indicates that the symbol and footprint is not yet connected. Click the connect-button to bring up the connect-dialog. Now select the pins and pads and hit connect, according to the following list:

H/F	1
RO	2
RE	3
DE	4
DI	5
SRL	6
GND	7
TXP	8
Y	9
Z	10
B	11
A	12
RXP	13
VCC	14

The final thing to do before saving the library is to set the prefix to IC. This is the prefix used for naming the component when it is added to the schematics. Now save the library, and go back to the schematics editor.

## RS-485 interface

To add the MAX3089, first import the library by selecting Library → Use. . . and opening the library. You should now be able to add the component (add max3089\*@max). Place the component in the upper right part of the schematic and add the rest of the components and nets for the RS-485 subsystem, as shown in Figure 4.

IC3	MAX3089SO	MAX3089SO	SO14	max
C8	100nF	C-EUC1206	C1206	rcl
CON3		ML6	ML6	con-ml
CON4		ML6	ML6	con-ml

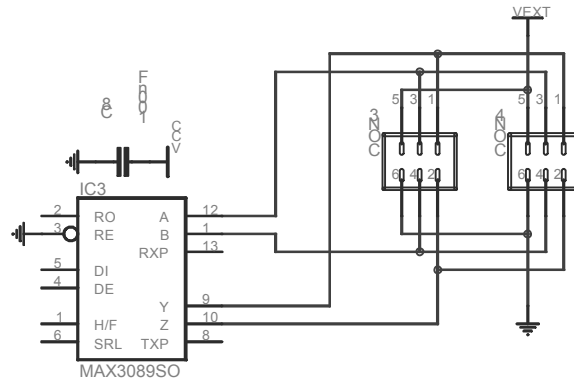


Figure 4: The RS-485 subsystem

## RS-232 interface

Add the following components for the RS-232 subsystem, as shown in Figure 5.

C9	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
C10	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
C11	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
C12	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
C13	1uF	CPOL-EUE2.5-6	E2,5-6	rcl
CON5		F09H	F09H	con-subd
IC4	MAX232	MAX232	DIL16	maxim

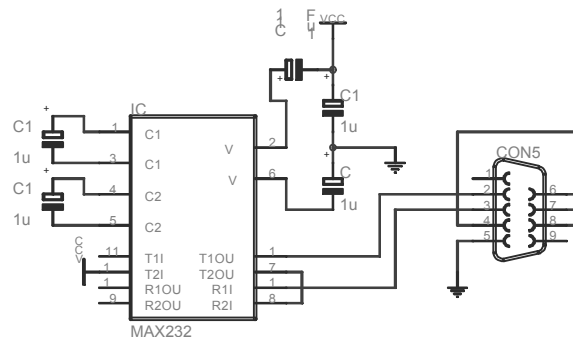


Figure 5: The RS-232 subsystem

## **Buses for the I/O-connectors**

Insert the connectors CON6, CON7 and CON8 (add ML10@con-ml). To draw the connections to the I/O-connectors, we will be using buses. A bus is simply a collection of nets drawn as a single bold line. Draw the buses according to the final diagram below. Use the label command to add labels to the buses. By default they should be named B\$1, B\$2 and B\$3.

Now use the name command to give the following names to the three buses. These names are very important, since they define which signals are available in the bus.

```
PB[ 0..2 ],VCC,GND
PD[ 2..7 ],VCC,GND
PC[ 0..5 ],AVCC,AGND
```

We can now start connecting the I/O-connectors to the bus. Use the net -command to do this. The diagram gets a little easier to read if you use an angle of 45° at the end where the net enters the bus. You will notice that connecting a net to a bus automatically prompts for the name of the net. This ensures that the nets get connected by avoiding similar but different names in each end of the bus. Now that Eagle knows which pins on the connectors map to each net, we should make this visible on the diagram. This is done simply by adding labels to each of the net connections.

Add nets connecting the pins of the microcontroller to the other ends of the buses, and give these labels as well. Notice how the buses are only a way of making the schematic more readable, and protecting against errors in the net-names. We do not need to connect the signals to the bus, as long as the nets have the proper names. This is why the nets GND, AGND, VCC and AVCC do not need to be connected at the other end of the bus.

## **LEDs and switch**

Add the following components and connect them to the bus:

```
LED1          LED3MM          LED3MM          led
LED2          LED3MM          LED3MM          led
R2   4k7      R-EU_M0805    M0805          rcl
R3   2k7      R-EU_M0805    M0805          rcl
R4   2k7      R-EU_M0805    M0805          rcl
S2                   10-XX          B3F-10XX       switch-omron
```

Then add the final jumper and resistor and hook up the RS-485 and RS-232 to the micro-controller.

```
JP2          JP2E          JP2          jumper
R5   4k7 R-EU_M0805    M0805          rcl
```

## **Electrical Rule Check**

A final thing that should be done before creating the PCB-layout is running the Electrical Rule Check (ERC). This is done by the ERC -command from the toolbar. The electrical rule check verifies various constraints regarding the connection of components. It will for instance warn about unconnected input-pins (which is why we made the tri-state inputs in the library for MAX3089 into I/O-pins. It also warns if you are connecting the supply voltage inappropriately or if you forget to connect it. You should, however, know that a DRC can not guarantee that no errors exist, but only help pinpointing some of the more obvious ones.



## ***Printing the schematic***

To print the schematic, you should use the print-command from the file-menu (or click the printer-icon on the upper toolbar). There are a couple of things that might be nice to know when printing from Eagle. They will be explained here. The printer-settings are the same for the schematic, library and board editor. While this means that you only have to set the paper-size in one place, it also has the strange side-effect of remembering if you printed the last PCB mirrored (which you will be doing for one of the layers when you do the PCB-layout), and in this case print your schematic mirrored as well, which will definitely not improve the readability.

The layers printed will be those visible when the print command is invoked. Use the Display button on the toolbar to modify the visible layer settings.

## ***PCB Layout***

To start laying out the printed circuit board, you should open the schematic in Eagle's schematic editor and click on the board button (located on the top toolbar in Eagle). You will be asked whether you want to create a new PCB design from the schematics. Confirm this. This should open Eagle's Board editor window.

Once you have created a board for a schematic, you should always have both files open when working with either the schematic or the circuit board layout. This is important, since it allows Eagle to keep the consistency between the two. This is called forward and back annotation. If you close either the schematic window or the board window and modify anything in the other window, Eagle will be unable to track the changes you have made, and help you keep the schematic and PCB consistent.

Notice how all the components from the schematic have been placed next to a white frame in the board editor. The white frame shows the maximum size of a circuit-board designed with the freeware version of Eagle. You will need to stay within these limitations. The first thing that should be added to the PCB is four 3.0 mm mounting holes. This ensures that you do not end up having troubles finding room for the mounting holes, because you have routed a lot of signals in the spot where the hole should be. It is a good idea to align the mounting holes on some nice metric positions. Switch the grid to millimeters while placing the mounting holes. You probably want to change it back afterwards, since the 0.1 inch grid is a standard distance between component pins.

## ***Placing components***

Now select the *move* tool and move each of the components and placing them within the board. Try to rotate the components while moving them (by right clicking), to untangle as many of the air-wires as possible. The air-wires are not automatically updated when moving the components. To do this you should use the Ratsnest -command. Since changing back and forth between the *move* tool and the ratsnest -tool is quite annoying, it is a good idea to define a keyboard shortcut for this action. Go to Options → assign. . . and enter the command ratsnest;move for the key-combination CTRL-E. This combination will execute the ratsnest command and change back to the move command. Notice how using a semicolon (;) allows you to have several commands carried out by a single shortcut key, which can be very useful.

When creating a new PCB Eagle adds all components to the top-side of the circuit board. Use the mirror tool to move a component to the bottom side of the PCB.

Pay special attention to L1 (the 10  $\mu$ H inductor). The physical component has a footprint similar to the electrolytic capacitors, which is somewhat bigger than the library symbol used. The pin distance, however, is correct (200 mil), but be careful not to put any other components too near L1.

The decoupling capacitors C2, C3, C8 and C10 are all connected to the VCC and GND nets. It is, however, important that they are placed as close to the IC they are decoupling as possible. So, for instance, C8 should be placed as close as possible to IC3 (the MAX3089). Consult the schematic to find out which decoupling capacitor belongs to which IC. C2 should be near the 7805 voltage regulator (IC1).

## ***Routing***

When all the components are placed appropriately, we are ready to start routing the PCB. This can either be done using the autorouter (select the auto -command from the toolbar), or using the manual routing (the route -command).

Use the autorouter with caution. In particular pay attention to the signals that should be routed on a particular side of the PCB to make room for the soldering. The autorouter can be restricted to work in only one layer by selecting the other layer as N/A. This may however generate some errors, since some components (SMD) are not routable on the allowed layer. For manual routing, select the *route* tool. Now click on an air-wire and Eagle will start routing the connection. Use right mouse button to change the bend of the routed signal. If you need to change the routing layer during routing (by inserting a via), press the middle mouse button. Holding the shift-key while starting the routing operation allows you to route a signal from anywhere, not only the end-points of the air-wires. If you keep the shift-key depressed when ending a wire, a via will be inserted. If you need to remove a routed wire, you should not use the *delete* command, since this cannot be back-annotated by Eagle. Instead, use the *ripup* -command. Note that clicking a single time on a connection rips up only this segment of the connection, while double-clicking (actually clicking an extra time on the air-wire will unroute the entire connection.

When routing manually, you switch back and forth between the route and ripup tools a lot. This can be much easier if a couple of keybindings are set up:

```
Ctrl-E  ratsnest;move
Ctrl-F  window fit;
Ctrl-L  window;
Alt-R   ripup
Ctrl-R  route
```

## ***Routing strategy***

Even with a two-layer board, it is often possible to implement reasonable ground (and power) planes using area fills on the top and bottom layers. Try to implement a ground plane on the bottom layer, and a power plane on top.

During routing, try to route as many signals as possible on the top layer, so that the ground plane on the bottom can be kept fairly solid. Also, leave the VCC and GND signals unrouted. If you are unsure about whether you routed a VCC or GND signal anyway, you can use the ripup tool from the command line (e.g. “ripup GND”).

Once the other signals are routed, make sure that each GND airwire has a connection to the bottom layer, and VCC has a connection to the top. You can place vias near component VCC and GND pads if necessary.

Finally, draw two polygons around the edges of the board, one on the top layer and the other on the bottom. Name the bottom polygon GND, and the top polygon VCC. Click ‘ratsnest’ to create the power and ground planes. Fix any unrouted signals, and you are done!

## ***Printing the PCB***

When you are done routing the PCB, you can print it. Start by experimenting with the layers that should be printed. Like the schematic editor, Eagle’s board editor prints the layers that are currently active. It is always a good idea to print a copy of all the normal layers (top, bottom, pads, vias, tOrigins, bOrigins etc.) on a single sheet of regular paper. This will allow you to see the real size of the circuit board, and can often be helpful in determining whether you have placed the components too close to each other. When the PCB is manufactured, a printout of the component placements (pads, vias, tPlace, tOrigins, tNames, tValues and tDocu) is a good way of finding out where the components should be placed. Print a similar sheet for the corresponding bottom layers.

