Starting with gEDA

at the Cambridge University Engineering Department

July 29, 2005

Contents

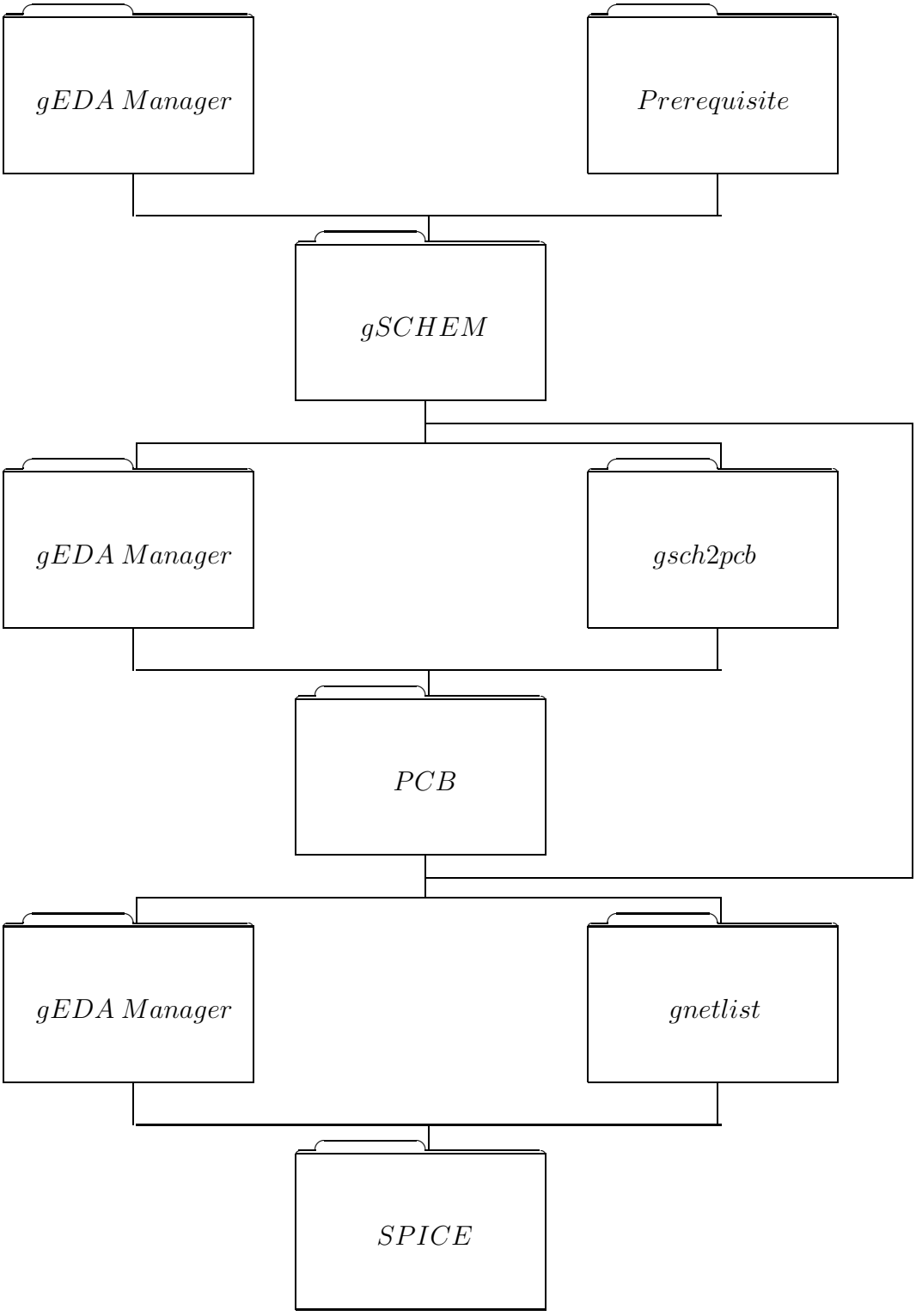

Figure 1: gEDA work flow diagram

Introduction

1 Introduction

This document is designed to guide you through the initial stages of using the gEDA tools, especially on the MDP CD.

The name 'gEDA' stands for *GPL Electronic Design Automation*: i.e. it is a open-source suite created for electronics applications design. In fact gEDA is a package containing 4 pieces of software :

- **Schematic Capture** to draw and lay out circuits.
- **PCB Layout** to create printed circuit boards.
- **Gerber Viewer** to view the files you will send to the manufacturer.
- **Spice** to simulate your circuit on your computer.

You will find all the gEDA tools on the MDP CD in:
Start Menu → Programs → Electronics

Having been, like you, a newcomer into this software, I hope that this little tutorial will help you to quickly understand how it works.

Thus, we will create the schematic and the PCB of a universal mounting board for a standard 8 pin OpAmp, for example the 741[1].

The process used is relatively linear as shown in figure 1, and Appendix A shows the basic circuit diagrams of the various layouts that can be configured on the board. The figure 2 visually presents all the software of the gEDA environment and the way they are connected. The ones highlighted in grey are the ones that we will deal with in this tutorial.

The document starts in section 4 by explaining how to generate a schematic with *Schematic Capture*, and goes on to section 4.2, which describes how to generate a bespoke component by modifying an existing one. Then you will see how to copy components and wire the schematic in sections 4.4 and 4.5, before we finally see how to print it out in the last subsection describing *Schematic Capture* (the program gschem).

Through the sections 5.1 and 5.2 you will discover how to create a PCB file with the *PCB Layout* software, and how to import a schematic designed with *Schematic Capture*. Finally, in sections 5.3 and 5.4, you will learn how to route and finish the PCB.

[1] http://mechatronics.mech.northwestern.edu/design_ref/electrical_design/opamps.html and www.uoguelph.ca/∼antoon/gadgets/741/741.html

After that, you can learn, in section7, how to simulate your circuit with *ngspice* or *gspiceui* in order to verify your circuit before any manufacturing process.

At last there is a significant library of predefined components for gschem and PCB Layout in Appendices **??** and **??** which might be useful. Some useful commands in ngspice are included in the Appendix too. If you have any questions please refer to the FAQ on page 43. Moreover, all the files used for this tutorial can be found in the Appendices.

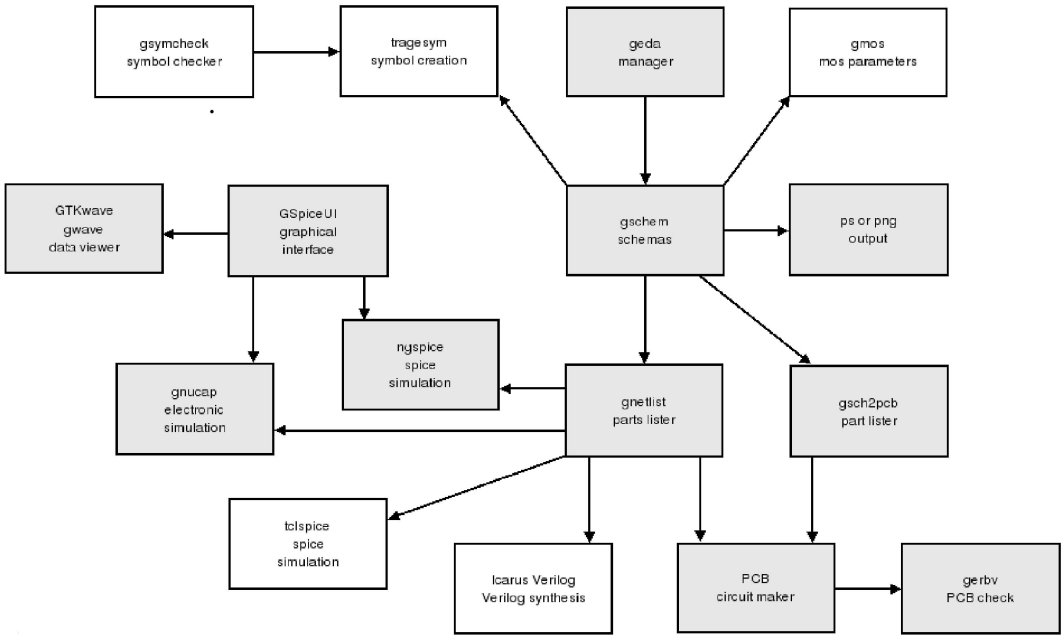

Figure 2: gEDA Overview.

gEDA Manager

2 gEDA Manager

There are currently two ways of using gEDA suite. One is to use the command line (**Xterm**) to make the link between the software packages e.g. **Schematic** → **PCB** → **Spice**. The other one is to use **gEDA Manager**, which is probably the simplest way. However, it can be useful to use a combination of both methods, as the command line interface offers more flexible options. Advanced users may soon find it necessary to use the command line for some parts of the design process.

As I want to provide you both ways of doing it, you will find below how to use the simple method, but with command line hints alongside. If you use **gEDA Manager** by itself, please ignore everything after a †.

The general process of using gEDA is as follows:

1. First of all open **gEDA Manager** from 'Start' → 'Programs' → 'Electronics'.
2. Then 'Project' → 'New'.
3. Give it a name, such as "myproject".
4. 'File' → 'New'
5. Type 'myschem' and choose the 'Schematic' type. Now you can see on the left in the 'Modules' tab that under the 'Schematic diagrams' section is a file named "myschem.sch".
6. Now double-click on "myschem.sch". **gschem** should open.
7. Draw your schematic and save it. If you need help to create it, please refer to section 4.
8. Then in **gEDA Manager** select "myschem.sch" and click on 'Action' and 'Create PCB Layout'. A file named "myschem.pcb" will be created under the section 'PCB and VLSI layouts' in the 'Modules' tab.
9. Double click on "myschem.pcb" to open the **PCB** software (if this doesn't work, use 'Action' → 'Open'). Create your PCB and print out the layouts. Please refer to section 5 for further help.
10. Select "myschem.sch" and choose 'Design rule checking' from the 'Action' menu. That will test your PCB and tell you if something is wrong with your design.
11. If you now click on 'Tool' and 'Gerber viewer' you can see the files that you had printed out in the **PCB** software. Please refer to section

12. Now select “myschem.sch” and select ‘Create SPICE netlist’ from the ‘Action’ menu.

13. Now double click on “myschem.cir”, which has just been created under the section ‘Simulations’ in the ‘Modules’ tab, and perform your SPICE analysis. Please refer to section 7 to find out how.

14. Finally select ‘Create bill of materials’ from the ‘Action’ menu to get them into a file named “myschem.bom” that you can import into any spreadsheet. Please refer to section **??** for more information.

Prerequisite

3 Prerequisite

† Before entering the gEDA suite, we must create a project file to make all the different programs in gEDA work together.

1. Open your home directory with 'Xfile Explorer' and create a new folder named 'tutorial'.
2. RMB and create a 'New File' named **project**.
3. RMB on the project file and open it with a text editor (e.g. 'emacs').
4. Type into it:
   schematics myschem.sch
   output-name mypcb
5. Save it and close it.

First steps with Schematic Capture

4 First steps with Schematic Capture

To begin with Schematic Capture click on *Start Menu* → *Programs* → *Electronics* → *Schematic Capture*

Now you should have a window named 'gschem' open.

Figure 3: gschem window.

You can see that the most useful functions have shortcuts under the menu. Thus you will find from left to right : New File, Open a file, Save a file, Undo, Redo, Add a component, Add nets, Add buses, Add Text and Select. Most functions have keyboard shortcuts as well - these are displayed in the menu.

4.1 Add a component

To add a component click on *Add* → *Component*.

In the smaller window which opens, you can choose among different libraries which may contain the component you want to add. For example, to add a resistor, click on *analog* → *resistor-1.sym* and then finally click on the place on the diagram where you want to put it.

Now it should be too small to see the resistor with precision. So you have two main options to zoom: 'zoom in' and 'zoom box'. Both of them are available in the 'View' Menu or from the keyboard shortcuts 'z' and 'w'. You can 'zoom out' or 'zoom extents' in the same menu or by using the shortcuts 'Shift+Z' and 'v+e'.

For this tutorial you will need to insert :

- 1 Resistor-1 from the Analog Library
- 1 op-amp (AOP) that we will create in the following step
- 1 Capacitor-1 from the Analog Library
- 1 Connector 4x1 from the Connector Library
- 1 Connector 3x1 from the Connector Library
- 1 pot-bourns from the Analog Library

Save this schematic onto your hard-drive as 'myschem.sch' in the folder 'tutorial'.

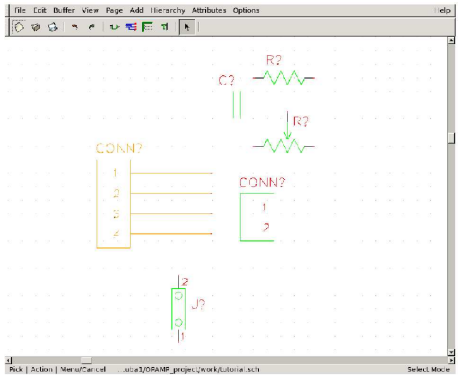

Figure 4: Components displayed in gschem.

4.2 Create a component

I said before that we will create our op-amp, but if you can see there is already one in the Analog Library! So why do we need to create a new one? Because the component given has only 5 pins to connect (pins 2,3,4,6,7) and doesn't offer the possibility of using the first and fifth pins for setting the offset[2].

So we have to add those two pins to the operational amplifier given in the library.

1. Open the symbol found in /usr/share/gEDA/sym/analog/aop-std-1.sym

   hint: on the top of the 'Open' window change the filter from "sch" to "sym".

2. Save it to your hard drive as 'myaop.sym' in the folder 'tutorial'.

3. LMB (Left Mouse Button click) on pin 4, RMB (Right mouse button) on it and select 'copy'. Now LMB again on pin 4 and LMB somewhere on a free space on the screen.

4. Repeat the above step.

5. Select the first pin you created, and select 'edit' with the RMB. Now LMB into the field 'value' of the row named 'pinnumber" and change it to '1'. Change the 'pinseq' to '1' too.

6. Do the same thing with the second pin you created, but this time change the attributes to '5'.

7. Now, to move the first pin: LMB to select it, then RMB and select 'move' (or use the 'm' key). Then LMB where you want to put it. So you can now move the pins 4 and 7 to the left and insert the pins 1 and 5, as in the figure shown below.

[2]http://mechatronics.mech.northwestern.edu/design_ref/electrical_design/opamps.html

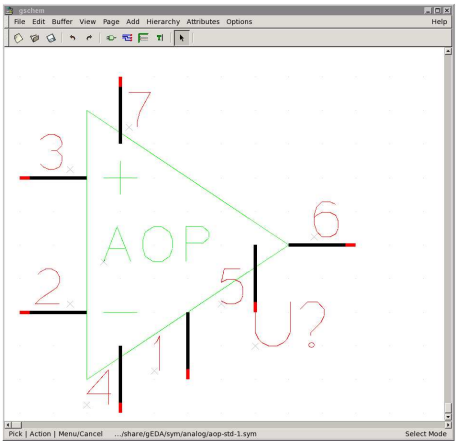

Figure 5: Pins disposal around the AOP.

8. Now, as you can see, your pins are snapped to the grid. It is important to have the red end of each pin on a point of the grid and the pin axis along a dotted line, but no need to have the white end of the pin finishing on a point. That is exactly what we want to change to have a better rendering. So, LMB on the 'Options' menu and select "Toggle Snap On/Off". Now you can LMB on pin 7, and then Long LMB on the white end of the pin to drag it up to the edge of the AOP. Do the same thing for the other pins.

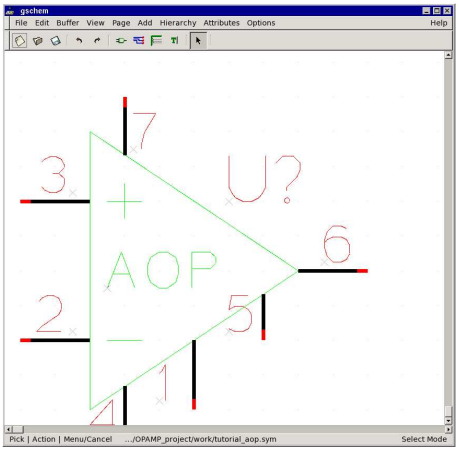

Figure 6: Final view of the operational amplifier.

9. Save the symbol and close the window (page).

10. Now, you want *gschem* to see your component into the list of components. So all you need to do is copy your file 'myaop.sym' into the directory of the library of components: /usr/share/gEDA/sym/analog **hint**:

if the computer don't want you to copy into this directory it may be because you are not allowed to read and write into this directory. So open a terminal window and type:

- su
- cd /usr/share/gEDA/sym
- chmod 664 *

Now you can copy your symbol into the directory */usr/share/gEDA/sym/analog*. An alternative to the above process is to create your own custom library. Make a new directory called 'mylib' to contain your library in your home directory, then copy 'myaop.sym' into this directory. Create another directory in your home directory called '.gEDA' (note the dot) and create a file called 'gschemrc' in this directory. Put the text `(component-library "/Home/mylib")`, including the brackets, into this file. You must shut down and restart gEDA: when it has started up again, this library should be available in gschem.

11. Re-open 'myschem.sch', and insert your AOP from the Analog library. Save your schematic.

4.3 Edit the attributes

Keeping in mind that we want to convert our schematic into a PCB layout later, you will have to tell the software that this symbol has a name (R1 for example), a value (100k) and corresponds to a certain PCB footprint (R025, a quarter-watt resistor). So in order to specify all that, you have to select the component (LMB on it) and then RMB and select 'edit' or double hit the letter 'e' once selected. Then, for each component you will have to change the 'refdes' value like we did before for the 'pinseq' and 'pinnumber' and create 2 attributes: 'value' and 'footprint', according to the following table.

NAME	REFDES	VALUE	FOOTPRINT
RESISTOR	R1	100K	R025
CAPACITOR	C1	100UF	AXIAL_LAY 300
AOP	U1	741	DIL 8 300
POT BOURNS	R10	NULL	R_AJ_V
CONNECTOR 4 1	CONN1	CONNECTOR41	CONNECTOR 4 1
CONNECTOR 3 1	CONN3	CONNECTOR31	CONNECTOR 3 1

IMPORTANT: In general, you will have to find the footprints corresponding to the components on your schematic. You should have a look at the appendix **??** to find all the footprints for all the components existing in the

"PCB Layout" software database. Moreover Appendix **??** gives you a list of all the symbols existing in the 'Schematic Capture' database. Because sometimes these two databases do not correspond, you have to use the generic components or make your own. The PCB software contains two libraries: the m4 library and the 'newlib' library. You can add your own footprints in the 'newlib' format and use them in your schematics. The m4 library has a large range of parameter-specified components, and contains nearly all 'standard' components that you might use. However, it is not easy to see how to use m4 footprints.

If you want to see the components with their layouts, look at the following web site: *http://www.inblue.com/gEDA/symlibrary*.

The 'value' attributes correspond to the value of the component if it is passive (e.g. resistor, capacitor,etc) or to the type of the component if it's active.

4.4 Copy the components

Now you've got one component completely defined of each type we need. So all you have to do is copy each component as much as necessary in order to obtain the number of components required (see table below). This is much quicker than repeatedly inserting new components.

To copy a component, select it (LMB) and then RMB and choose 'copy' (or press 'ec'). Then you have to LMB again on it and then LMB where you want to draw the copy.

COMPONENT	NUMBER
RESISTOR	7
CAPACITOR	2
AOP	1
POT BOURNS	1
CONNECTOR 4 1	2
CONNECTOR 3 1	1

Now you can re-edit the attributes of each component to change the 'refdes' attributes. Thus the resistors will have a name from R1 to R7, the capacitor from C1 to C2 and the CONNECTOR 4 1 from CONN1 to CONN2.

† If you have lots of components, manually changing the 'refdes' attributes can be tedious. In this case, you can leave the components named as 'U?', 'R?' etc. The program **`refdes_renum`** will number all of your components for you when you call it from the command line.

4.5 Wire the Schematic

Once you've got all your components fully defined you can start wiring them by adding **nets**. You can find this tool as an icon on the toolbar, or inside the 'Add' menu, or simply in pressing the hotkey 'n'. You need to obtain something like this:

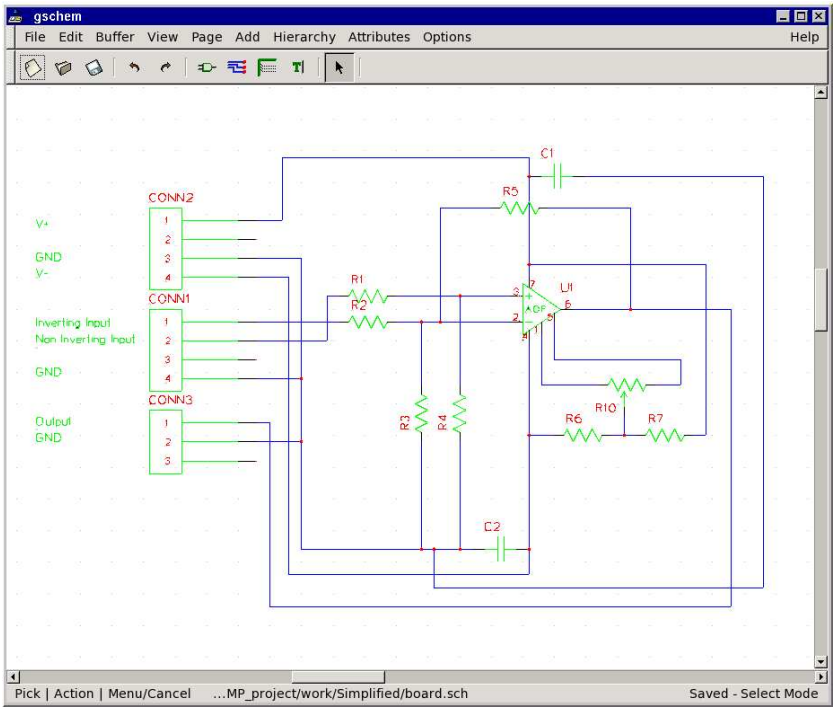

Figure 7: Final view of the schematic.

You can see that on the left of the connector there are lines of text to explain which pin of the connector is which. This is simply added with the icon 'Add Text' (that you can find in the 'Add' menu or by using the hotkey 'at'). Then all you have to do is type the text you want to add, select 'Apply' and LMB where you want to add it on the screen. That's all!

Save your schematic into "myschem.sch".

4.6 Add a title border

Now the last thing is to add the finishing touch, that is to say the title border around your schematic. In this case choose 'add a component', then 'MDP' Library and choose 'A3_title_bordered_CUED.sym'. Then double click on it or hit the keys "ee". Edit your name, the date, etc.

4.7 Print out the Schematic

In order to get your schematic into a postscript file, LMB on the 'file' menu and then 'print'. Choose your options (like A4 paper) and then press 'Ok'. A file named 'myschem.ps' has been created.

†

First steps with PCB Layout

5 First steps with PCB Layout

You've just created the schematic of your circuit. You can stop here if you don't want to make a PCB. But if you do want to, gEDA allows you to convert your schematic into a PCB fairly automatically, which can be useful.

5.1 Create a PCB file

In order to create your PCB you have to convert your schematic into the PCB format. This can be accomplished in the gEDA window by choosing 'Action' → 'Create PCB layout'. However, this is fairly limited.

† The command line approach outlined below is recommended:

1. Open a terminal window in your 'tutorial' directory.
2. Type:
   gsch2pcb myschem.sch
   which means that you want to convert a schematic to a PCB using the file named **myschem.sch** that you have created in section 4.
3. If your schematic contains elements with user-defined footprints, you need to tell gsch2pcb about these elements by specifying the `-d DIRECTORY` flag, where DIRECTORY contains the path to your user-defined footprints. Similar options are available for specifying m4 element directories.
4. If gsch2pcb can't find a user-defined symbol, insert it in gschem using the 'Embed component in schematic' option rather than the default when in the 'Add component' window. **hint:** you can use the command:
   gsch2pcb -v project
   to enter the verbose mode and see all the details. It's quite useful.
5. Now it should tell you that the file myschem.pcb had been created with all the components inside. If not, please refer to the FAQ section.
6. An alternative approach is to use the **project** file created in section 3. The syntax is then `gsch2pcb project`.

5.2 Import the schematic

1. Now open the PCB Layout Software from:
   Start Menu → Programs → Electronics → PCB Layout, or use the 'Action' menu in the gEDA window.

   You can see that in 'PCB Layout' all the icons are on the left hand side of the window. The white square in the top left corner, used for panning, is

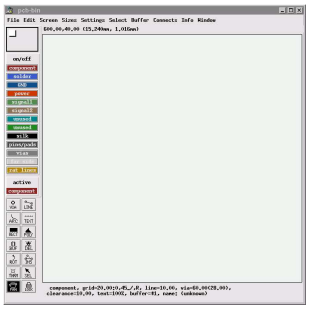

Figure 8: the PCB window.

only present in older versions of PCB. Below it is the menu to show/hide the different layers of the PCB. Radio buttons are present in newer versions of PCB to select the active layer. In the bottom left corner is the tool panel, to draw all you need on the PCB.

2. Choose 'Load layout' from the 'File' menu and select 'myschem.pcb' from your directory. You should have something like Figure 9, with all your components in the top left corner.

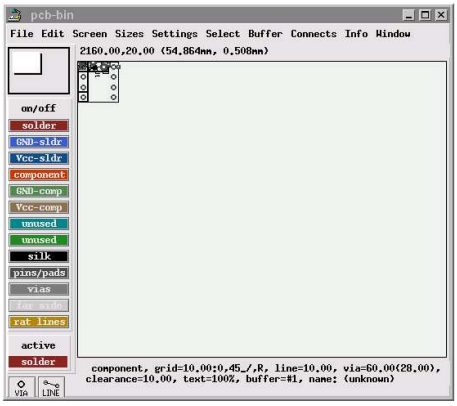

Figure 9: First opening of your PCB layout.

3. Choose the "Sel" tool from the bottom left menu.

4. Drag all the components apart with a Long LMB on it. (The component don't have to be highlighted). You should get something like Figure 10

   hint: you can rotate the components with the 'rot' tool on the bottom left menu, clicking LMB on the component you want to rotate.

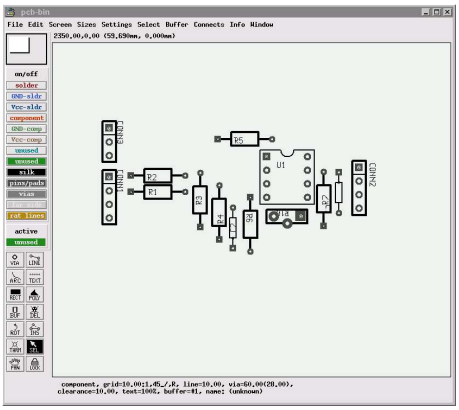

Figure 10: The disposing of your components on your PCB.

5.3 Route the PCB

1. Now that your components are placed you have to load the file that tells 'PCB Layout' the way they are linked. Go into the 'File' menu, select 'Load netlist file' and select 'myschem.net'. Choose 'Ok'.

2. Click on the 'Connects' menu and choose 'Optimize rats-nest' (or hit the hotkey 'o'). You will now see all the links between all the components. You can see too that if you move a component and reselect 'optimize rats-nest' the software re-optimizes the nets in order to get the shortest paths.

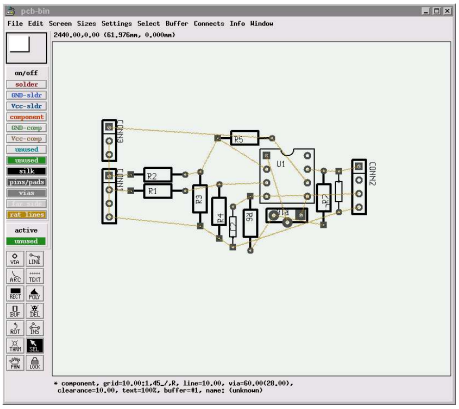

Figure 11: The optimized rats' nest.

WARNING: the computer chooses the shortest way to connect components who have to be connected together. It doesn't take care of the order they have to be connected. As a result, it might not be able to choose sensible layouts. The only way to prevent that is a good positioning of all

your components, and if it's still not suitable you will have to route some tracks by hand. Please refer to the FAQ section for that part.

3. Now select the 'Power' option under the 'Route style' button, in order to have bigger tracks for better signal integrity and ease of manufacture. The 'Route style' button can be used to change the default parameters (clearance widths etc).

4. Go into the 'Connects' menu and 'Auto-route all rats'. You can see now that the software has done your routing for you! But as you can see you've got many different colours of routing. That means that your PCB is mapped on several layers. By default, 'PCB Layout' uses up to 8 layers, until you switch off all those you don't want to use.

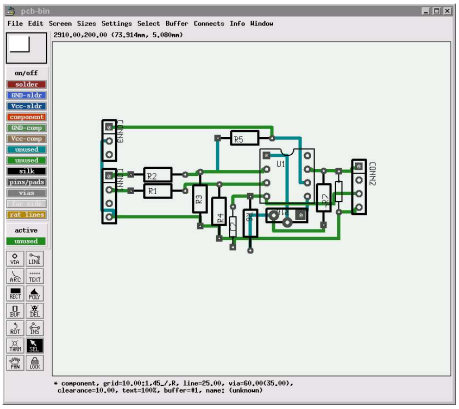

Figure 12: The PCB routed on two layers.

5. For a simple PCB, you will force the computer to route on only 1 layer. So erase the current routes with 'Connects' → 'rip up all auto-routed tracks'. And press 'o' to optimize the rats' nest.

6. Now deselect all layers except for the 'solder' layer. That will switch off those layers, and the software will tell you if it can't manage to auto-route the rats with only the selected layers.

7. Select again 'Auto-route all rats' from the 'Connects' menu and your correctly routed one layer PCB should appear. All tracks will be on the solder side of the board.

8. You can see on your PCB that some routes have strange paths. So you can select 'miter' (American for 'mitre') from the 'Connects' menu in order to have 45 degree angles and a better rendering of your PCB.

   Or you can route some (or all) rats by yourself (with the 'line' tool) before selecting the auto-routing tool. The software will know that you have

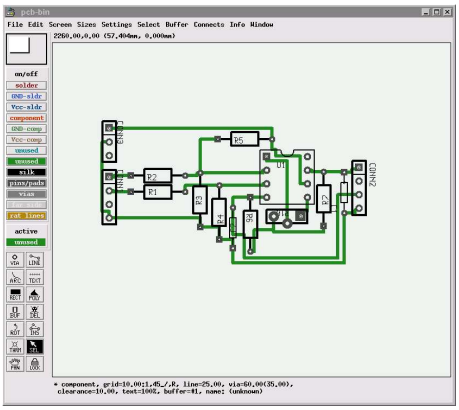

Figure 13: The PCB correctly routed.

previously routed some rats and won't route them again. It is important to press 'o' after making any changes so that PCB knows what should and shouldn't be connected (it won't let you connect tracks that it thinks should be separate).

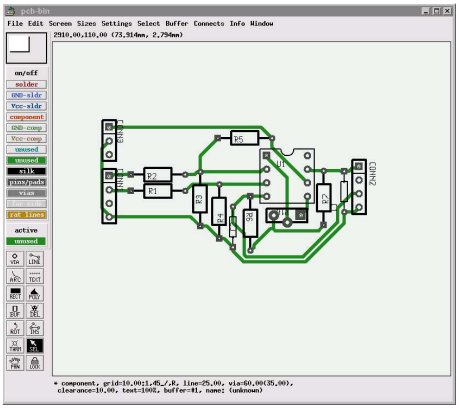

Figure 14: The PCB correctly routed and optimized.

9. Select **all** your PCB with the 'sel' tool. RMB on it and select 'cut selection to buffer', then move the PCB to the top left corner and LMB again. Finally, select the "sel" tool in order to go out of the buffer.

10. Now you can add mounting holes at each corner of your PCB by adding 4 vias with the 'via' tool. Then press 'Ctrl+H' (H for 'Hole') with the cursor over one of the vias. You see that the copper annulus has been removed. Do the same thing for the other three.

    hint: you can move around your board with the 'pan' tool.

11. Go into the 'File' → 'Preferences' menu, then 'Sizes', and change the 'pcb

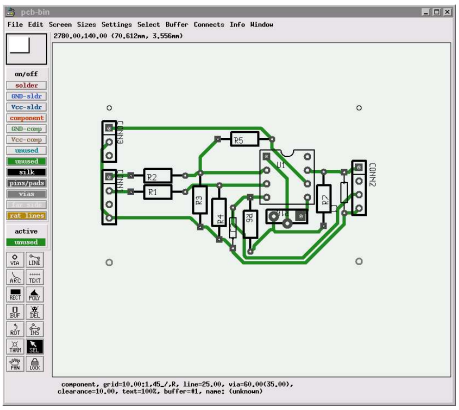

Figure 15: The PCB with mounting holes.

width' and the 'pcb height' to near the minimal value in order to have just a tiny border all around your pcb.

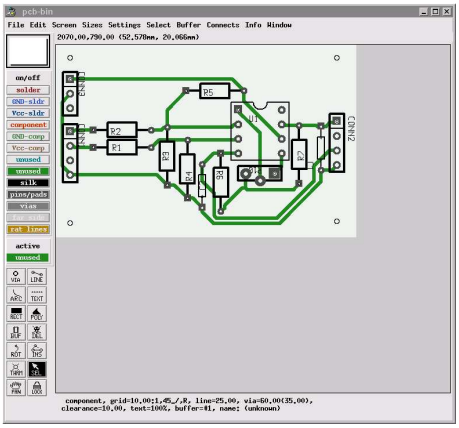

Figure 16: The PCB resized.

12. Finally we have to make the ground plane all around the wiring i.e. fill the PCB with copper everywhere except around the pins and the tracks. This will allow us (or the manufacturer) to remove as little copper as possible from the PCB (which costs a lot of money) and makes the ground reference more accurate.

    Hide all the layers except for the 'solder' layer on the screen (with LMB on the others) in order to see only the tracks i.e. disable the 'silk' and the 'pins/pads'.

13. Now, Shift+LMB on some tracks to select them (they should become blue) and press the hotkey 'Shift+J' (which means 'Toggle clearance mode on all selected') to switch on the clearance mode around the track. The

key 'j' by itself will toggle the clearance mode for the single element currently under the mouse pointer.

14. Now if you select the 'rect' tool from the bottom left Menu and make a rectangle above the track you just have selected, you will see something like Figure17.

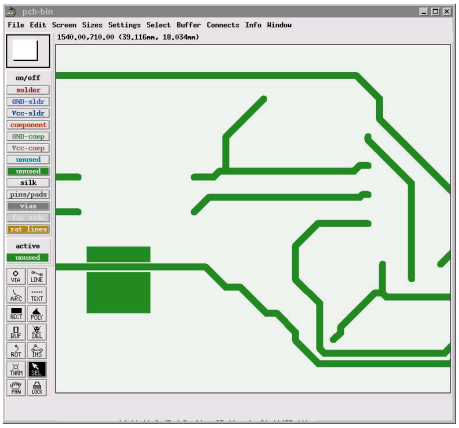

Figure 17: The clearance around a track.

15. Go into the 'Edit' menu and select 'Undo the last operation' to erase the rectangle.

16. Finally toggle on the clearance option for **all** the tracks, and draw **one** rectangle which covers all the PCB, but leave a 100mil gap all around free. If you just activate from the 'On/Off' menu the 'silk' and the 'pins/pads' you should have something like Figure **??**.

    Warning: if one track touches the ground plane where it's not a ground track, you will make a short-circuit!!

5.4 Print out the PCB

In order to get your PCB into a postscript file, LMB on the 'file' menu and then 'print layout'. Be sure that the 'device driver' is set to 'Postscript', and choose your options (like A4 paper) and then press 'Ok'. Many files are created dealing with the routing, the way the components are displayed, etc.

hint: if you want to convert one into a pdf file, go into your 'tutorial' directory and type in a terminal window:

ps2pdf myschem.ps myschem.pdf

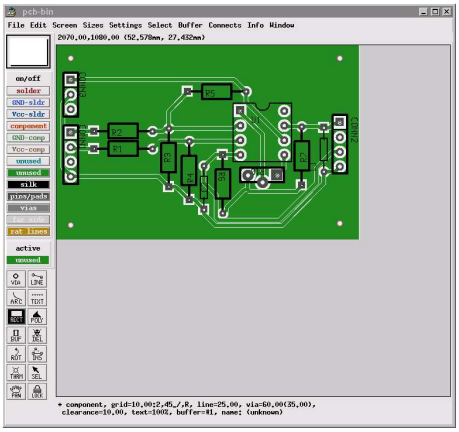

Figure 18: The PCB finished.

Your pdf file has been created under the name myschem.pdf into your directory.

In the same way eg. 'file' and 'print layout', choose this time 'Gerber' from the options menu. Choose 'A4' as above and select 'Ok'. Now many '.gbr' have been created. See section 6.

REALLY IMPORTANT NOTE: If you are printing your PostScript files on treasury paper ready for a wet PCB manufacturing process, it is very useful to know that the treasury paper will shrink by around 0.5 per cent due to the heat in the laser printer. This is only important when large or finely-spaced surface mount components are present, when some pins may become misaligned. To combat this, the whole PostScript file can be scaled with the syntax:

```
pstops '1:0@1.005' in.ps out.ps
```

which will scale by a factor of 1.005 and save the results in `out.ps`.

First steps with Gerber Viewer

6 First steps with Gerber Viewer

This section has been left in its original 'franglais', as a tribute to Alain who wrote the first version of this tutorial. This will hopefully be appreciated by the reader as a kind of reward for getting this far.

† Gerber Viewer (gerbv) is a viewer for Gerber files(".gbr"), which are generated from PCB CAD system (like "PCB Layout") in order to be sent to PCB manufacturers.

1. As "Gerber Viewer" suffer of a bug, you should open an "Xterminal" in the directory where you have generated previously the ".gbr" files in section5.4. Do do that you can use the "cd" command into the "Xterminal" or open an "Xfile explorer" into this directory and press"CTRL+T".

   Once there, type:

   *gerbv *.gbr*

   it will automatically open with "gerbv" all the ".gbr" files in this directory.

2. Click on "Files" and "Save project as". Type the name "myGBRproject" and select "Ok".

3. Now, close "Gerber Viewer" and re-open it in typing in the "Xterminal" window:

   gerbv

   or by selecting it with "Start Menu" → "Programs" → "Electronics" → "Gerber Viewer".

6.1 View the gerber files

Now that you've got your files open and saved, you can see that there are number on the right of your screen. Clicking on them will display the different layers corresponding on and off. Thus you can see each one of them, or all of them at one time(but it's not really readable)

Now you should know that each layer is in fact a ".gbr" file in your current directory, and it will be certainly those files that you will give to your PCB manufacturer. So checked them carefully to prevent you from seing errors, once the PCB in your hands!

To requestfor PCB manufacture at the CUED please refer to :

http://www-g.eng.cam.ac.uk/EDG/getting_a_pcb_produced.htm

First steps with NGSPICE

7 First steps with NGSPICE

Now that you are succeeding in creating a schematic and creating the PCB corresponding, you probably want to simulate the whole system on the computer before starting ordering or making what you need. The software "ngspice" is based on the "Spice 3" program which was created by Berkeley years ago. NGSPICE also means Next Generation Simulation Program with Integrated Circuit Emphasis. So you will see in subsection 7.4 how to simulate your circuit inside an Xterminal window with **ngspice** or you can learn how to use in subsection 7.5 the user interface provided by **gspiceui** which run ngspice in background. It depends only if you prefer command line or not.

7.1 Gathering the symbols and the models

The bad news of SPICE is you can't use your previous circuit how it is. Partly because it is not well defined for SPICE e.g. it don't tell him the eletrical behavior of each component and partly because the component itself wasn't design from scratch to be used in SPICE.

For the "most standard" components you only have to had a "model" attributes to them using **gschem**. See the figure below to see all those behavior already known by NGSPICE :

COMPONENT	MODEL
RESISTOR	RES
CAPACITOR	CAP
POLARIZED CAPACITOR	CAP
INDUCTOR	IND
DIODE	D
PMOS TRANSISTOR	PMOS
NMOS TRANSISTOR	NMOS
PNP TRANSISTOR	PNP
NPN TRANSISTOR	NPN
PFET TRANSISTOR	PJF
NFET TRANSISTOR	NJF

But if you want another component, whith a behavior more complicated, like an OpAmp 741 you will have to find the model (a ".lib" ".cir" or ".mod")on the MDP CD (**/usr/share/gEDA/models**)or on the internet[3].

[3]You can find those on the web site of the manufacturer (like texas instrument for example) but some of them are referenced on those adresses:

http://custom.lab.unb.br/pub/electronics/cookbook/spice/ti/
http://custom.lab.unb.br/pub/electronics/cookbook/spice/analog/

So, if you have a look into the directory **/usr/share/gEDA/models** you will find a file named "UA741.lib" that we will use later. In a general case, if you are looking for a component named fore example "741", perform a search into that directory to look for all the files which "contains the text" "741". Then choose the model wich seems to be the more useful for you. You should perform a search not only on the name of the files but on what the files contains, because some files contains more than one model, and so there nameis not helpful.

7.2 Creating the Schematic

Now, we want to simulate an Inverting OpAmp (see A).So, in order to do it properly, we will just create a new schematic from scratch. But you can convert the one made before if you want.

1. So open **gschem** and create a new schematic.
2. Save it as "mySPICE.sch"
3. Import components as figure shown below:

COMPONENT	LIBRARIES	COMPONENTS	NUMBER
RESISTOR	MDP	RESISTOR-1.SYM	2
OPAMP	MDP	AOP-SPICE-1.SYM	1
VIN	MDP	VSIN-1.SYM	1
VOPAMP	MDP	VDC-1.SYM	2
GROUND	MDP	GND-1.SYM	4

4. Edit the attributes of each one as below :

http://www.brorson.com/gEDA/SPICE/spice-20030321.tar.gz
http://www.elektronikschule.de/~krausg/APLAC_Schaltungssimulation/Aplac_cd_version_2.2.1/Spice-Models/
http://homepages.which.net/~paul.hills/Circuits/Spice/ModelIndex.html

COMPONENT	ATTRIBUTES	VALUE
Resistor 1	refdes	R1
	value	10K
	model	RES
Resistor 2	refdes	R2
	value	100K
	model	RES
Vin	refdes	Vin
	value	sin 0 1 1KHZ
Vcc	refdes	Vcc
	value	DC 12
Vee	refdes	Vee
	value	DC -12
OpAmp	refdes	XOA1
	value	UA741
	model-name	UA741
	file	/usr/share/gEDA/models/UA741.lib

You should know for a futur use that all the ”refdes” attributes have to begin with a specified letter regarding of the type of the component you use. All the components have already this letter predefined in the ”refdes” attribute definition (e.g. ”R?” for a Resistor). You just have to change the ”?” into a number ”1” or ”2” for example. The only components you may have to modify are the Integrated Circuits (e.g. a Chips). Their default letter is ”U”, but sometimes you will have to change it into an ”X” like for our OpAmp. Because if you open, with a text editor, the file ”ua741.mod” you will see that the first line not commented begin with a ”.subckt”. In this case, only if you see a ”.subckt” you will have to change the letter from ”U” to ”X”.

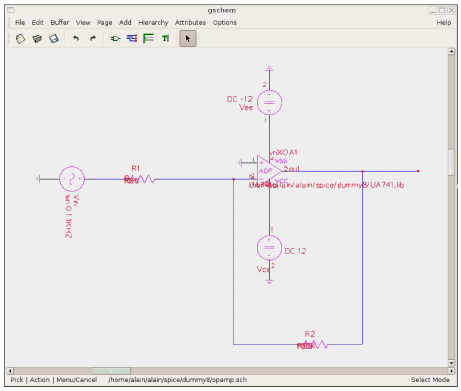

Figure 19: Schematic of the Inverting OpAmp.

5. Now, plug them all like on the figure 20.
6. In order to access easily to the value of the input and the output of the OpAmp select the line between "Vin" and "R1" and edit the attributes in clicking again on it or in typing "ee". Then create an attribute named **netname** with the value "in".
7. Select the line on the far right (which is the output of the OpAmp), and create a **netname** with the value "out".
8. Save the Schematic.
9. Close **gschem**

7.3 Generation of the netlist

†

Now in order to make one file, readable by **ngspice** which include the definition of all the nets, all the components and all the models needed, we have to use the too named **gnetlist**.

- Now open an **Xterminal** window in your "spice" directory and type :
  gnetlist -g spice-sdb mySPICE.sch
- A file named output.net should have been created in your Spice directory e.g. now you should have 3 files in your directory : ua741.mod, mySPICE.sch and ouptut.net

7.4 Simulate your circuit with NGSPICE

†

Now, all what we have done before are the prerequisite to enter now the simulation program.

In the same **Xterminal** window (or in a new one, but still pointing on your "spice" directory") type :

ngspice output.net

which will load the software **ngspice** for the file named "output.net"

tran 100us 10ms

That will run a transient analysis. Please refer to the SPICE3 online manual[4] or the Apendix B to have a list of the most useful command.

[4]http://newton.ex.ac.uk/teaching/CDHW/Electronics2/userguide/

display

Which display all the vector used. You should the see your "out" and "in" previously defined.

plot out in

Will plot the both curves in the same window. You can the see that you have well simulated an Inverting OpAmp.

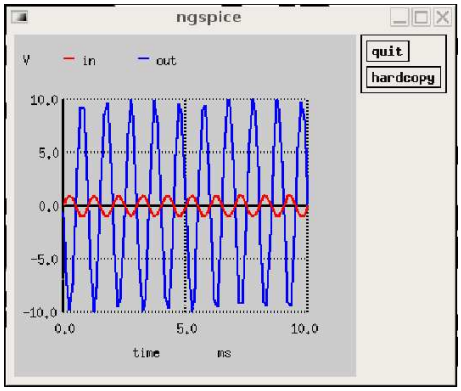

Figure 20: Plot of the Inverting OpAmp.

7.5 Simulate your circuit with GSPICEUI

So, as I have said below below you can use the more "friendly" **gspiceui** or the command line od **ngspice**. It doesn't matter whichever you use. The results will be the same.

- So open an **Xterminal** window and type :

  Gspiceui

  Now it should have open a window like in the figure21.

- Click in "File" Menu and select "Open". Then change the bottom left button from "Circuit files" to "Netlist files". Finally select your spice directory where you will find the file named "output.net". Click "Ok".

- Then on the left of the window you can see the list of the "Nodes" and "Components" of your circuit. Click on the nodes "in" and "out" that

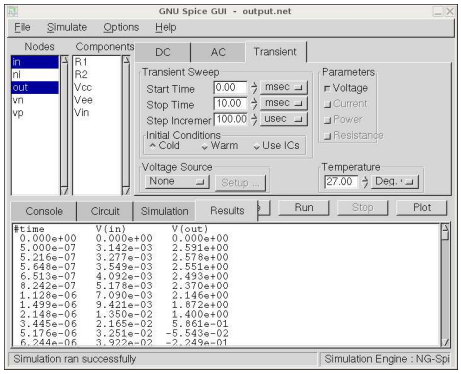

Figure 21: The GspiceUI window.

you had created previously. They should now be highlighted in blue, like in figure21.

- Then on the "Options" menu select "ngspice" instead of "GNU-Cap". Because we will use the ngspice software in the background.
- Select the tab named "Transient" in order to perform a Transient analysis. Be sure that you had parametrized all like the figure below shows :

NAME	VALUE	UNITS
START TIME	0	MSEC
STOP TIME	10	MSEC
STEP INCREMER	100	USEC
VOLTAGE SOURCE	NONE	

- Then click on "Run" which will run the analysis.
- Once finished click on "Plot" which will automatically open **gwave** to plot the curves. Two windows should have appeared.
- Drag the node "V(in)" and "V(out)" from the small **gwave** window into the big **gwave** window.
- Finally you should see something like on the figure below which is the simulation of an INverting OpAmp.

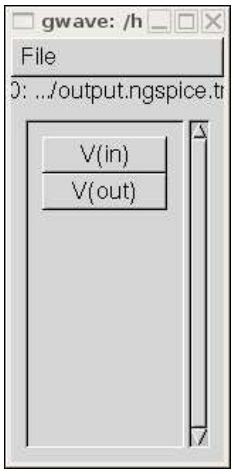

Figure 22: The small Gwave window.

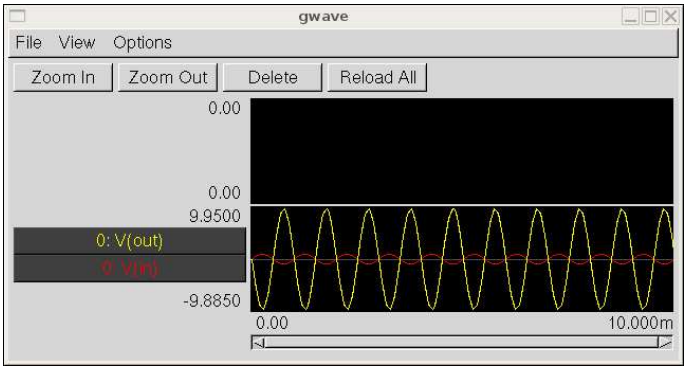

Figure 23: The simulation viewed into Gwave.

FAQ

8 Frequently Asked Questions (FAQ)

8.1 About Schematic Capture

8.1.1 I can't find the component I need!

If you can't find it, have a look on the internet if someone have already done it, else read the section4.2 or have a closer look at the appendix **??**.

8.2 About gsch2pcb

8.2.1 gsch2pcb return "Bad expression"?

If gsch2pcb print out "Bad expression" when you try to convert your schematic into a pcb, it may be because one (or several) attributes are wrongly defined e.g. wrong footprint. For exemple for a capacitor the footprint is not "DIODE_LAY" but "DIODE_LAY 300" if the distance between the 2 foot of the capacitor is 300 mils (300 of a thousandth of an inch).

8.3 About PCB Layout

8.3.1 My component look weird!

If your component is not as you expect it to be, it could be because the footprint you chose is wrong or because you had not specify the distance between the pins of the components. See section8.2.1.

8.3.2 What are the rule to make a PCB?

Prerequisite :

The majority of electronic components are still manufactured with imperial pin spacing. Thus the units in use are the "thou" and the "mil" (both, rather confusingly, 1/1000th of an inch). For example the space between the 2 rows of pin of a MAX233 is 300 mil.

Here are the basic rules, that were really helpful to me, to make a good PCB :

- Put all your components on the board
- Place your components into functional blocks on the board
- Route by hand, and first of all, the critical tracks (if needed)
- Let a clearance of 15 thou between two tracks
- Keep tracks as short as possible

- Use a single track, to go from one point to another (when routing by hand), not tracks put together end to end.
- Make your power tracks as big as possible
- Keep things symmetrical (which help to find mistakes)
- Let the power track near the ground one.
- Minimize the number of layers and the number of jumpers.

But if you want to have a closer look at how to design a PCB, there is a really good tutorial on the web at the adress:
http://alternatezone.com/electronics/pcbdesign.htm

8.3.3 What If I change my schematic after having created my PCB?

Use again the gschem2pcb tool as before and look below what you need to do regardless of what you've done:

If you had change the way your components are linked, just re- "load the netlist file" from the "file" menu into the PCB software and then "Optimize rats nest" from the "Connects" menu.

If you have added or deleted components go into the "file" menu of the PCB software and select "Load layout data to paste-buffer", select "myPCB.new.pcb" and select "Ok". LMB on the board only once! Finally drag the component(s) you had imported where you want, and suppress, if necessary, the components no more needed with the "del" tool of the bottom left corner.

8.3.4 How to route the PCB by hand?

Before all, you have to select in which layer, you want to linked your components. On the middle left side of the screen, there is a menu called "Active". LMB on the colored box below and choose the layer.

Now you have to choose the thickness of your lines. Go into the "sizes" menu and choose the routing style that suits you.

Finally click on the "lines" icon on the bottom left screen menu, and draw your lines.

hint : Before routing, "optimize rats nest" fom the "Connects" menu to help you through.

8.4 About Spice

8.4.1 gnelist print an error

If gnetlist print out an error it may simply means that one (or more) component is wrongly defined. Because you use two components may have the same ”refdes” attribute or that one attribute is missing or wrong.

8.4.2 Ngspice don’t give the result expected

If Ngspice don’t give the result expected it could be because you simply wrongly link the components between them, or that you had used a wrong components, or a wrong voltage. After having checking that, if it’s still the case, please look, if you use a model file, that the model file is setting the behavior of the component the way you want (Some model in order to be simple, don’t set the whole “real” behavior of the component). Then check that you are making the good analysis (an AC analysis is not a DC analysis...). Finally, if it’s still the case why don’t you change the scale of your plot?

Appendix

A The ways to plug an OpAmp

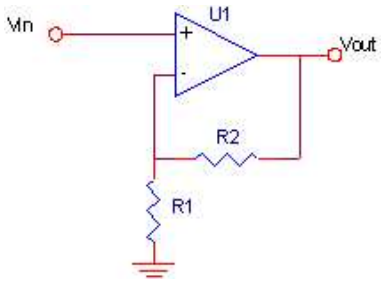

Figure 24: Non Inverting OpAmp

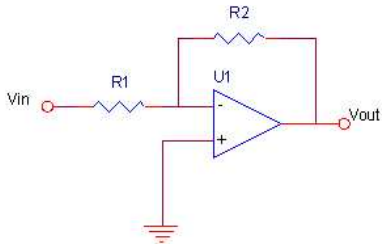

Figure 25: Inverting OpAmp

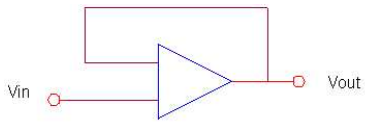

Figure 26: Buffer

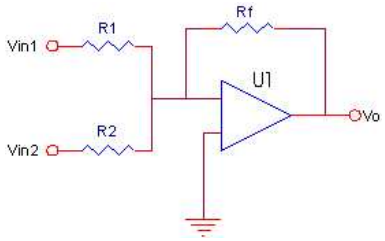

Figure 27: Summer

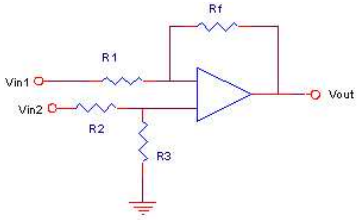

Figure 28: Difference Amplifier

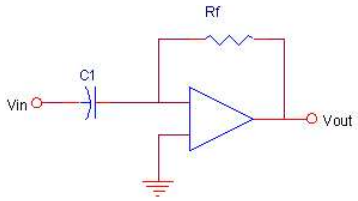

Figure 29: Differentiator

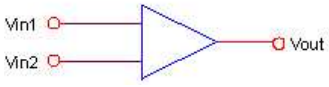

Figure 30: Comparator

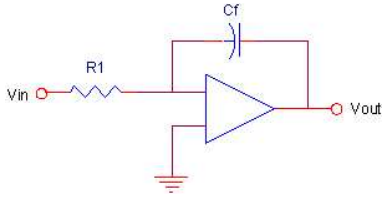

Figure 31: Integrator

B Useful commands under NGSPICE

- Perform an AC analysis :

  ac dec N FTSART FTSOP
  ac oct N FTSART FTSOP
  ac lin N FTSART FTSOP

  example :

  ac lin 100 1 100HZ

  with dec, oct and lin the type of variation (decade, octave or linear); N the number the increment and FSTART and FSTOP the startinf and final frequencies.

- Perform a DC analysis :

  dc SOURCENAME VSTART VSTOP VINCR

  example :

  dc vin 0.25 5.0 0.25

  with VSTART, VSTOP and VINCR the starting, final and incrementing values.

- Perform a Transient analysis :

  tran TSTEP TSTOP

  example :

  tran 1ns 100us

  with TSTEP and TSTOP the increment and the final time.

- Display the current vectors :

  display

- List all the nodes :

  listing

- Plot the curves :

  plot VECTOR1
  plot VECTOR1 VECTOR2 VECTOR3

For more information about all the command see:
http://newton.ex.ac.uk/teaching/CDHW/Electronics2/userguide/

C Linear Circuits and Device

Spice used for solving DC circuits problems (Examples Sheet 1)

The paragraphs below show how Spice may be used to solve awkward numerical problems in DC circuit analysis. Both examples are taken from Examples Sheet 1.

C.1 5. Determine the current in the 2W resistor in a circuit using transient analysis. [L2]

The image below shows how the labelled circuit diagram (or schematic) can be entered into the Spice package. Note that there are some differences from the symbols to which you have become accustomed. If required, ngpice offers the flexibility to let you craft symbols to suit your own preferences. Each component is uniquely labelled and a numbering system is used internally to identify connections or nodes. The designer specifies values for each component (note the resistor values - units are ohms - and current sources).

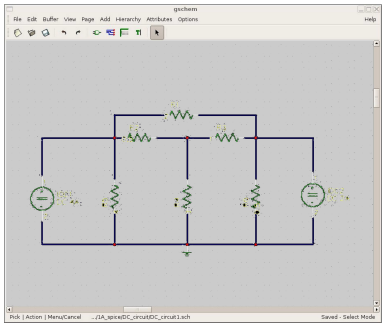

Figure 32: Schematic of the desired circuit.

After having choosing a transient analysis into ”gspiceui”, pick the R1 component, and set the length of the simulation. Pressing the ”Run” button gives the following result. The program has been instructed to plot the current as a function of time over a period of one second. As all sources are DC, there is of course no variation.

The graph shows quite clearly how the current in the 2-ohm resistor is 1 Ampere; the convention used by Spice (”current flowing into the pin”) shows that the (conventional) current flows left to right in the circuit.

C.2 9. In the circuit of Fig. 5, find the current in 11 for V1 = 7V.

Again, the image below shows how the circuit schematic can be entered into the Spice package.

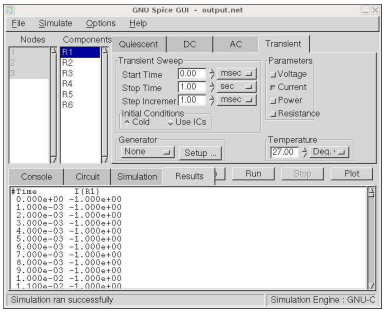

Figure 33: How the perform the analysis.

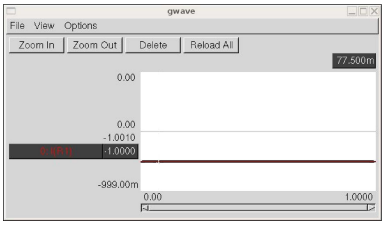

Figure 34: Plot of the result.

Pressing the ”Run” button gives the result shown below. Again, for convenience, the program has been instructed to plot the current as a function of time. All sources are DC, so no variation is seen. The graph shows that the current in R11 is 125 mA, as expected

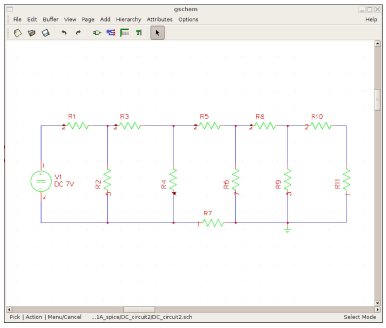

Figure 35: The new schematic.

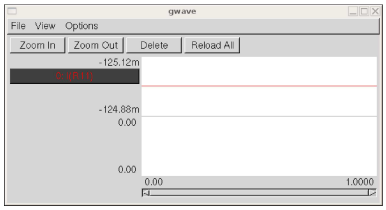

Figure 36: The new schematic.

D Simulating a FET Amplifier with Spice

D.1 Device characteristic plot

Before you begin a circuit design, it's useful to study the device characteristics for the device you're planning to use.

Every device has slightly different characteristics which must be accounted for in your circuit design.

The 2N3819 datasheet[5] (PDF = portable document format) contains characteristics and graphs for a typical device, determined by the manufacturer from measurements on a large number of samples. The ngSpice simulator encapsulates device measurements such as these in a numerical model, which may comprise many parameters.

In the results and plots shown below, we are relying on a numerical model to predict how circuits will work. This is a remarkably effective approach, widely used by professional designers, but it is vital to remember:

The results are only as good as the model used

[5] http://www2.eng.cam.ac.uk/~dmh/ptialcd/tut_spice3_jfet_2N3819.pdf

D.2 Characteristics using ngSpice

The circuit below entered into pSpice will let us plot out I-V characteristics of the device J1 corresponding to pSpice's numerical model. It's important to remember, this is a prediction of how a typical device will respond, and not a measurement as such.

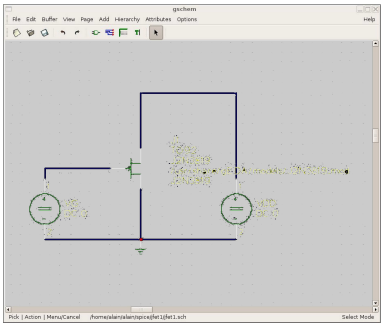

Figure 37: Plot Circuit for a 2N3819 JFET.

The data actually entered into pSpice is text-based. The schematic is just for the convenience of a human designer. Fortunately, it is fairly easy to convert a schematic diagram into data that ngSpice can accept. Here's the input that corresponds to the schematic above; it instructs ngSpice to plot the drain current in J1 as a function of VDD (the drain voltage) and VG (the gate voltage) as these are varied between prescribed limits.

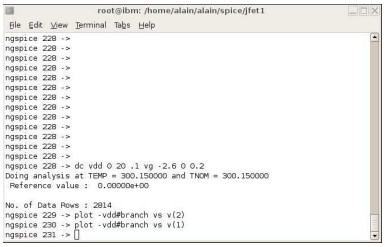

Figure 38: Command lines.

Shown below is the transfer characteristic ID vs. VGS plot for the 2N3819 n-channel JFET.

D.3 DC Operating Point (.OP)

Now that we have the characteristic I-V plot for the device, we can choose a DC operating point Q for the device. There are a number of factors to consider

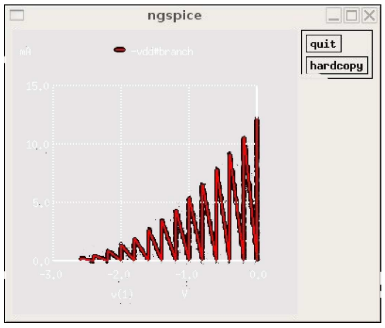

Figure 39: The spice Id vs. Vg plot for a 2N3819JFET.

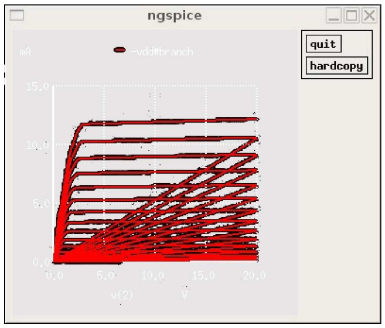

Figure 40: The spice Id vs. Vd plot for a 2N3819JFET.

when picking the operating point for our circuit. These were covered in Lectures 7-8.

First of all, we want the amplifier to work in a linear region to minimise the amount of signal distortion. The optimum region for the JFET is in the region where the drain current ID is essentially constant, the so-called constant current region.

We must avoid the region where VGS ¿ 0, and where VDS exceeds the manufacturer's limit, and we must take care that the product: ID x VDS does not exceed the maximum safe power limit, to avoid overheating. Typically, we will choose an operating point Q that lies near the centroid of the safe region.

More advanced topics

We also want to maximise the output range of our amplifier. If our operating point Q point is too close to earth potential, or to the supply voltage, the signal will clip (that is, it will be limited in amplitude by the power supply) much sooner then if it were farther away from those two voltages. For this reason, the device is typically set up to operate at midpoint bias. This means that

the JFET is biased so that the drain-source voltage is approximately halfway between the supply voltage and earth potential.

Another concern of the circuit is power consumption. Since the circuit consumes a certain amount of power even if no signal is connected to the input, typically a designer will wish to minimise the power consumption. This can be achieved by using as small a drain current as necessary to do the job that the amplifier needs to do.

In this example, we've chosen an operating point with a drain current of approximately 3 mA and a drain-source voltage of approximately 10 V.

Once the operating point has been chosen, the only other parameter of the circuit is the supply voltage, which was probably already known, and likely a factor in the selection of the operating point. We will use a supply voltage of 20 V.

The next step is to draw a line from the X intercept at the supply voltage point, through the operating point Q, all the way to the Y axis. This is the DC load line, which governs the operation of the circuit, and determines the value we choose for the drain resistor RD. It has slope -RD.

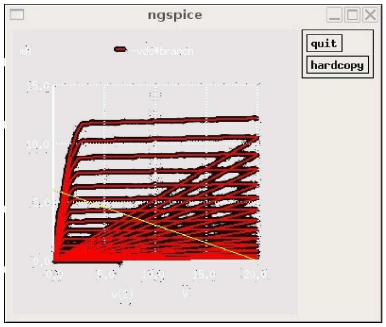

Figure 41: DC Load Line for Amplifier.

The preliminary work is now complete, and we are ready to begin selecting the components necessary to complete the circuit.

To bias the gate at the proper voltage (-1.5V according to the I-V plot), we need to provide a 1.5 V bias battery.

The operating point has been chosen as VDS = 10V and ID = 3 mA. This means that the remaining 10 volts has to be dropped across the drain resistor RD, while a drain current of 3 mA flows. This gives RD = 10/3 kohms or 3300 ohms. The design of the circuit is now complete.

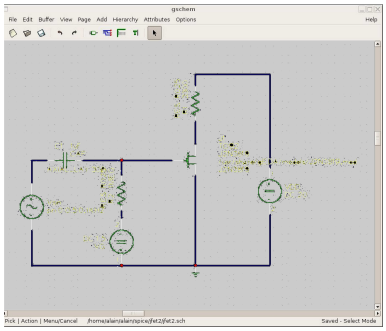

Figure 42: Circuit design details for JFET Common Source Amplifier.

D.4 Response to a sine wave of 0.25 volts peak at 100 Hz

Applying a sine wave of 0.25 V peak, or 0.5 V peak-to-peak at the input gives us the following output signal as a function of time, or transient response.

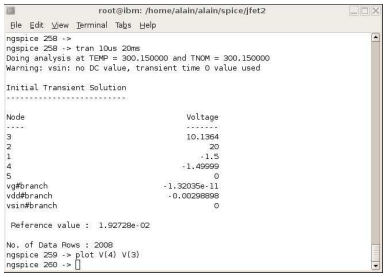

Figure 43: How to perform a transient analysis.

Figure 44: Transient response of Fixed-bias JFET Amplifier.

Hence the voltage gain is Vout/Vin = -5.8/0.5 = -11.6 approximately.

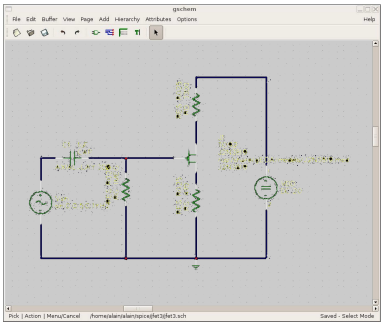

Figure 45: Schematic of the self-biased JFEt amplifier.

D.5 The self-biassed JFET amplifier

The self-biassed design allows the bias battery to be eliminated, saving an expensive and bulky battery, at the cost of only one additional resistor RS. From above, to achieve the required operating point, Q, requires the gate electrode to be biased at -1.5 volts relative to the source electrode. With self-bias, we keep the gate itself at 0 volts, and raise the potential on the source electrode to +1.5 volts - which amounts to the same thing.

RG holds the gate at earth potential, or 0 volts, since negligible DC current flows into the gate electrode.

The voltage drop across RS is given by: ID x RS. Since ID is fixed at 3 mA for the chosen operating point Q, we choose 500 ohms for RS; the closest convenient preferred value is 470 ohms.

The graph below shows the output predicted by ngSpice for the self-biased circuit.

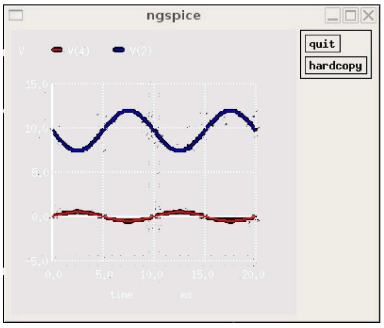

Figure 46: Vg and Vd.

The output is considerably lower than for the battery-biassed circuit with the same input applied - the gain is about -2.1/0.5 -4. Negative feedback is being applied because of the presence of the source resistor RS, which means that a fluctuating signal appears on the source electrode as the drain current varies. Consideration of Kirchhoff's voltage law shows that the gate signal

voltage vgs is actually reduced below 0.5 V p-p by presence of this signal on the source, and the output is lower as a result.

Fortunately, there is an easy solution to this - see the schematic diagram below. If we connect a bypass capacitor C3 of sufficiently large capacitance in parallel with RS, a bypass path is provided for signal currents to flow directly to earth, rather than through RS; then the full input signal appears between the gate and source, restoring the gain to the expected value. The crucial service performed by C3 is effectively to short-circuit the source resistor RS for signal frequencies only.

However, because no DC current flows through a capacitor, the steady source voltage needed to establish the gate bias (to achieve the operating point Q) is unaffected.

For the moment, we will sidestep the issue of how to select a 'sufficiently large' value for C3, and we will assume that a value of 100 mF will be satisfactory.

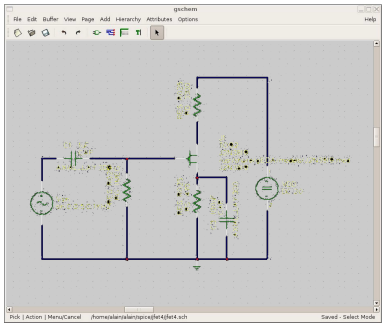

Figure 47: The self-biased schematic.

The sine wave response is as shown below, and at the frequency in use, 100 Hz, it can be seen that the gain has been restored to about -11.5. At this frequency it is apparent that the bypass capacitor C3 (100 uF) is adequate. However, if the frequency were reduced this might not be the case, since capacitive reactance is inversely proportional to frequency.

The following simulation run shows how ngSpice can be used to show the effect of varying C3, to establish how sensitive is the design's performance to the choice of value. ngSpice allows the designer to experiment with virtually any parameter or combinations of these, offering great potential for 'what-if' experiments.

The circuit schematic is almost identical to the previous one, but the annotation Cval in place of 100 uF) indicates that on this run, C3 will be varied over a range of values.

This time we plot output voltage amplitude versus frequency to give a set of frequency response graphs, one for each C3 value.

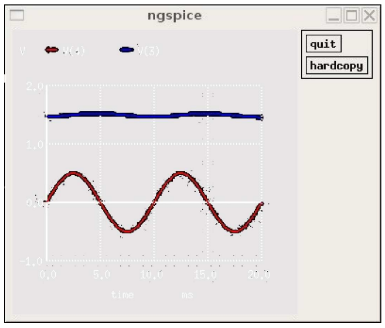

Figure 48: Vd and Vg.

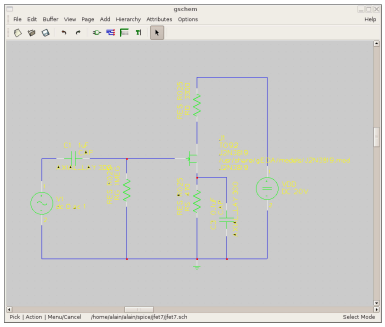

Figure 49: Circuit schematic.

The input was set to 1 volt p-p, so the vertical scale which represents the p-p output voltage at the drain is numerically equal to the gain. You can see that the transition between low gain (about -4) to high gain (about -11.8) happens at a different range of frequencies, according to the value of the bypass capacitance C3.

- With capacitance of 100 mF, the gain begins to drop off below about 10 Hz.
- With capacitance of 1 mF the gain begins to drop off below about 1000 Hz.

The plots have been labelled to show the value of bypass capacitor used. Note that the frequency scale is logarithmic.

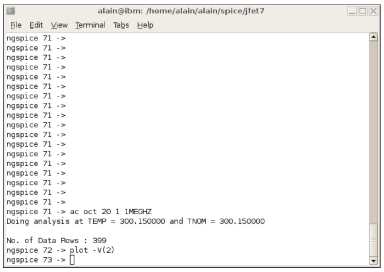

Figure 50: Perform an AC analysis.

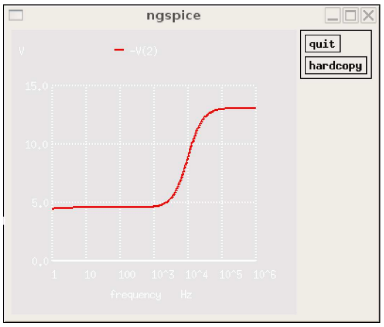

Figure 51: Voltage vs. frequency.

E Simulating a speed sensor

Here is one more example on how to use the spice simulation. The circuit below was created in order to know dynamically what is the speed (in rpm) of a steam engine (more precisely a Sterling engine).

Thus, a sensor is mounted on the fly-wheel of the engine (see on figure53), giving a signal (e.g. a square wave) at each time the wheel had performed a revolution.

So, the schematic is divided into 2 differents parts. On the left you can see the area for simulating the behavior of the sensor itself. On the right is the system which convert this signal whose voltage is proportionnal to the speed of the fly-wheel.

You are certainly wondering why there is a sinusoidal source on the far left of the schematic. It's simply because at the present time there is no way to define a customed square wave without defining each voltage of each point until the end... So we used a Schmitt trigger in order to perform the desired signal from a sinusoidal source.The simulation below are made using a transient analysis, with a stop time of 200ms and a step increment of 2ms.

Once you've got that signal, whose each peak can be larger or more spaced in the time regarding of the speed of the wheel, we want to have the same signal but with a constant width of the peaks. This way, we will be sure at

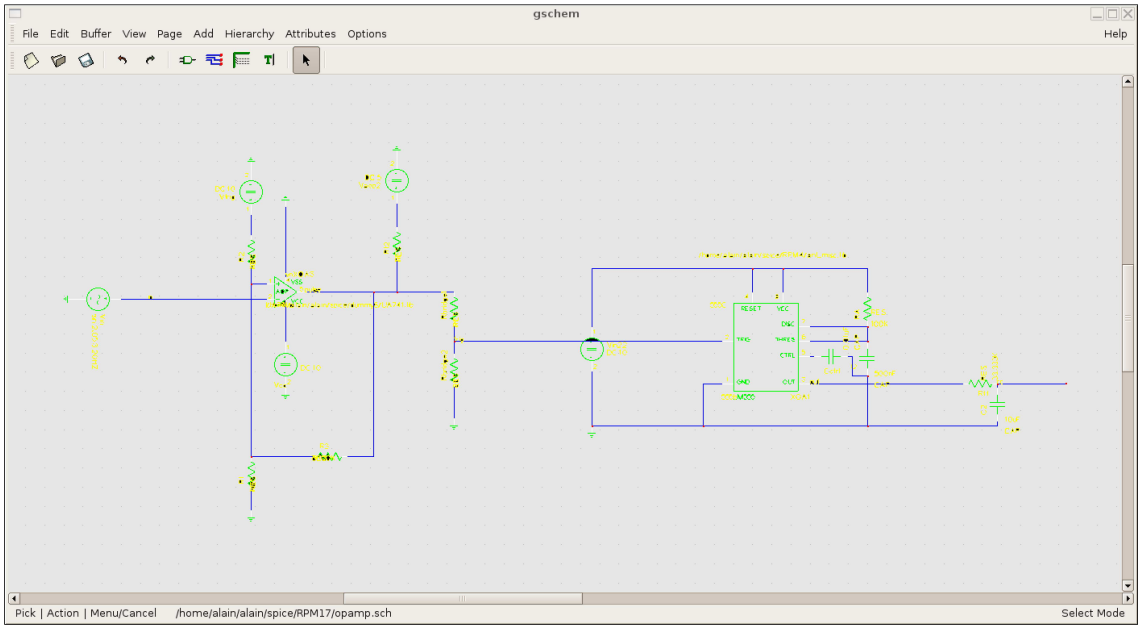

Figure 52: The schematic of the circuit.

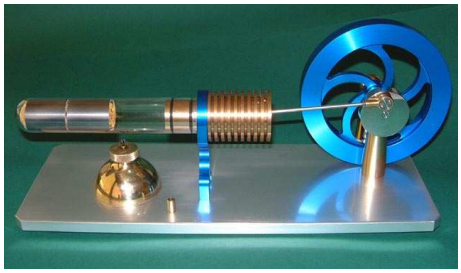

Figure 53: The Sterling engine.

each time to be able to see when the wheel at performed a revolution.

Finally, we used an RC filter to get the ”average” voltage and thus be able to see on a voltmeter (for example) the voltage corresponding to the speed of the engine.

Moreover as the frequency of the engine to be recorded is between 1HZ and 20HZ, we had set the value of the Resistor and the Capacitor of the RC filter in order to have pulsation at 3db for 1HZ. Thus, for all the frequency above 1HZ you will have a voltage greater than 0V.

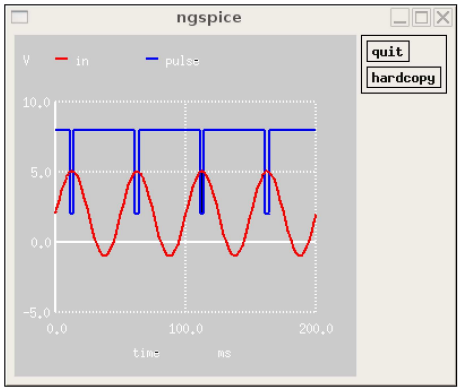

Figure 54: The Schmitt trigger role.

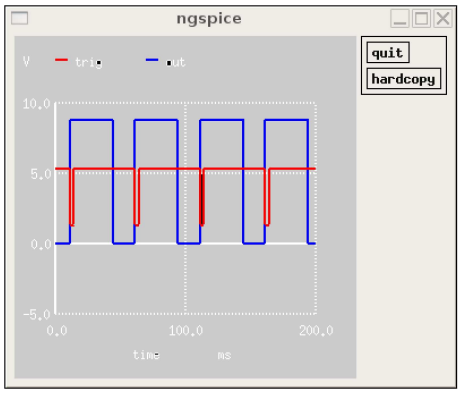

Figure 55: The 555 timer role.

Finally you will see in the figure below how each component have been defined.

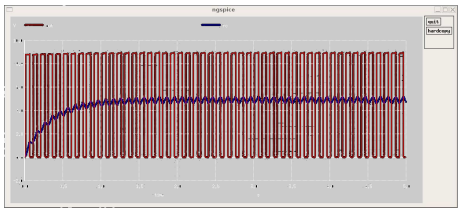

Figure 56: The RC filter role for a 20HZ frequency .

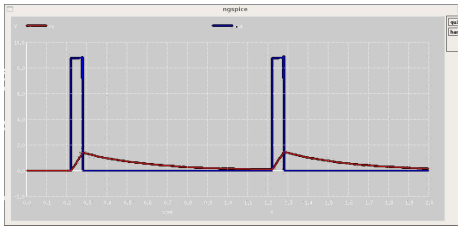

Figure 57: The RC filter role for a 1HZ frequency .

COMPONENT	ATTRIBUTES	VALUE
Rbridge2	value	2K
	model	RES
	refdes	Rbridge2
Rbridge1	value	1K
	model	RES
	refdes	Rbridge1
Rt2	value	1K
	model	RES
	refdes	Rt2
Vtrig	value	DC 10
R2	value	10K
	model	RES
	refdes	R2
R1	value	10K
	model	RES
	refdes	R1
R3	value	5000K
	model	RES
	refdes	R3
Vin	value	sin 2.05 3 20HZ
Vcc	value	DC 10
Vaop2	value	DC 5
XOA3	value	UA741
	model-name	UA741
	file	/usr/share/gEDA/models/UA741.lib
	refdes	XOA3

COMPONENT	ATTRIBUTES	VALUE
C2	value	10uF
	model	CAP
	refdes	C2
R11	value	33.333K
	model	RES
	refdes	R11
Cctrl	value	0.01uF
	model	CAP
	refdes	Cctrl
C	value	300nF
	model	CAP
	refdes	C
RA	value	100K
	model	RES
	refdes	RA
Vin22	value	DC 10
XOA1	value	555b
	model-name	555b
	file	/usr/share/gEDA/models/anl_misc.lib
	refdes	XOA1

F Content of the MDP Library

Here is the list of all the components pre-defined to be used in the CUED. But some of them don't work for **PCB Layout** or **ngSPICE** or the both. For example voltage sources aren't define for PCB, simply because you will never put them on a PCB, but linked to it via a connector. Another example is the variable resistor. You can't use it under spice. You simply have to change it into 2 resistors. In the same way if you want to simulate a diode-bridge you will have to create it with 4 diodes. For the other components I didn't simply manage to find their footprint or their model. (They may not exist yet, but what about now?)

IMPORTANT: All the components (or some of them) have a footprint and a model associated, e.g. that the real shape and the behavior is fixed. But there is plenty of models for a OpAmp741 for example, because they all simulate a peculiar behavior. In the same way, even for a resistor there is plenty of package that you can use, different sizes, different spacings,etc. So, all what is given could perhaps not correspond to what you want. It's just on way to do it. So please check and change whatever you want to get it.

COMPONENT	VALID FOR PCB	VALID FOR SPICE
74HC244	Y	Y
AOP-SPICE-1	Y	Y
AOP-SPICE-2	Y	N
CAPACITOR	Y	Y
CRYSTAL	Y	N
DIODE	Y	Y
DIODE-BRIDGE	N	N
GND	Y	Y
IDC	N	Y
INDUCTOR	Y	Y
L293	Y	Y
LED	Y	Y
LM555	Y	Y
LM7805	Y	Y
MOSFET	Y	Y
NC	Y	Y
NPN	Y	Y
PHOTODIODE	Y	Y
PHOTO-RESISTOR	N	N
PHOTO-TRANSISTOR	Y	Y
PNP	Y	Y
POT	Y	N
RELAY	Y	Y
RESISTOR	Y	Y
RESISTOR-VARIABLE	Y	N
SPST	N	N
SWITCH	N	N
VAC	N	Y
VDC	N	Y
VEXP	N	Y
VPULSE	N	Y
VPWL	N	Y
VSIN	N	Y
ZENER	Y	Y

G myschem.SCH

```
v 20050313 1
L 200 800 200 0 3 0 0 0 -1 -1
L 200 0 800 400 3 0 0 0 -1 -1
T 1050 150 11 8 0 0 0 0 1
device=AOP-Standard
L 800 400 200 800 3 0 0 0 -1 -1
P 200 600 0 600 1 0 1
{
T 150 625 5 8 1 1 0 6 1
pinnumber=3
T 150 625 5 8 0 0 0 6 1
pinseq=2
}
P 200 200 0 200 1 0 1
{
T 150 225 5 8 1 1 0 6 1
pinnumber=2
T 150 225 5 8 0 0 0 6 1
pinseq=3
}
P 800 400 1000 400 1 0 1
{
T 875 425 5 8 1 1 0 0 1
pinnumber=6
T 875 425 5 8 0 0 0 0 1
pinseq=6
}
P 300 732 300 900 1 0 1
{
T 325 750 5 8 1 1 0 0 1
pinnumber=7
T 325 750 5 8 0 0 0 0 1
pinseq=7
}
P 300 63 300 -100 1 0 1
{
T 200 -75 5 8 1 1 0 0 1
pinnumber=4
T 200 -75 5 8 0 0 0 0 1
pinseq=4
}
```

```
L 300 650 300 550 3 0 0 0 -1 -1
L 250 600 350 600 3 0 0 0 -1 -1
L 250 200 350 200 3 0 0 0 -1 -1
T 250 350 9 8 1 0 0 0 1
AOP
T 700 100 8 10 1 1 0 0 1
refdes=U?
P 500 200 500 0 1 0 1
{
T 400 25 5 8 1 1 0 0 1
pinnumber=1
T 400 25 5 8 0 0 0 0 1
pinseq=1
}
P 700 332 700 200 1 0 1
{
T 600 225 5 8 1 1 0 0 1
pinnumber=5
T 600 225 5 8 0 0 0 0 1
pinseq=5
}
```

H myOPAMP.sym

```
v 20050313 1
L 200 800 200 0 3 0 0 0 -1 -1
L 200 0 800 400 3 0 0 0 -1 -1
T 1050 150 11 8 0 0 0 0 1
device=AOP-Standard
L 800 400 200 800 3 0 0 0 -1 -1
P 200 600 0 600 1 0 1
{
T 150 625 5 8 1 1 0 6 1
pinnumber=3
T 150 625 5 8 0 0 0 6 1
pinseq=2
}
P 200 200 0 200 1 0 1
{
T 150 225 5 8 1 1 0 6 1
pinnumber=2
T 150 225 5 8 0 0 0 6 1
pinseq=3
}
P 800 400 1000 400 1 0 1
{
T 875 425 5 8 1 1 0 0 1
pinnumber=6
T 875 425 5 8 0 0 0 0 1
pinseq=6
}
P 300 732 300 900 1 0 1
{
T 325 750 5 8 1 1 0 0 1
pinnumber=7
T 325 750 5 8 0 0 0 0 1
pinseq=7
}
P 300 63 300 -100 1 0 1
{
T 200 -75 5 8 1 1 0 0 1
pinnumber=4
T 200 -75 5 8 0 0 0 0 1
pinseq=4
}
```

```
L 300 650 300 550 3 0 0 0 -1 -1
L 250 600 350 600 3 0 0 0 -1 -1
L 250 200 350 200 3 0 0 0 -1 -1
T 250 350 9 8 1 0 0 0 1
AOP
T 700 100 8 10 1 1 0 0 1
refdes=U?
P 500 200 500 0 1 0 1
{
T 400 25 5 8 1 1 0 0 1
pinnumber=1
T 400 25 5 8 0 0 0 0 1
pinseq=1
}
P 700 332 700 200 1 0 1
{
T 600 225 5 8 1 1 0 0 1
pinnumber=5
T 600 225 5 8 0 0 0 0 1
pinseq=5
}
```

I myPCB.pcb

```
# release: pcb-bin 20050127
# date:    Thu May 19 11:46:51 2005
# user:    user (MDP User)
# host:    Knoppix

PCB["" 207400 133200]

Grid[1000.00000000 0 0 0]
Cursor[134400 120000 4.000000]
Thermal[0.500000]
DRC[699 400 800 800]
Flags(0x00000000000000d0)
Groups("1,2,3,s:4,5,6,c:7:8:")
Styles["Signal,1000,4000,2000,1000:Power,2500,6000,3500,1000:Fat,4000,600

Symbol[’ ’ 1800]
(
)
Symbol[’!’ 1200]
(
SymbolLine[0 4500 0 5000 800]
SymbolLine[0 1000 0 3500 800]
)
Symbol[’"’ 1200]
(
SymbolLine[0 1000 0 2000 800]
SymbolLine[1000 1000 1000 2000 800]
)
Symbol[’#’ 1200]
(
SymbolLine[0 3500 2000 3500 800]
SymbolLine[0 2500 2000 2500 800]
SymbolLine[1500 2000 1500 4000 800]
SymbolLine[500 2000 500 4000 800]
)
Symbol[’$’ 1200]
(
SymbolLine[1500 1500 2000 2000 800]
SymbolLine[500 1500 1500 1500 800]
SymbolLine[0 2000 500 1500 800]
```

```
SymbolLine[0 2000 0 2500 800]
SymbolLine[0 2500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 4000 800]
SymbolLine[1500 4500 2000 4000 800]
SymbolLine[500 4500 1500 4500 800]
SymbolLine[0 4000 500 4500 800]
SymbolLine[1000 1000 1000 5000 800]
)
Symbol['%' 1200]
(
SymbolLine[0 1500 0 2000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1000 1000 800]
SymbolLine[1000 1000 1500 1500 800]
SymbolLine[1500 1500 1500 2000 800]
SymbolLine[1000 2500 1500 2000 800]
SymbolLine[500 2500 1000 2500 800]
SymbolLine[0 2000 500 2500 800]
SymbolLine[0 5000 4000 1000 800]
SymbolLine[3500 5000 4000 4500 800]
SymbolLine[4000 4000 4000 4500 800]
SymbolLine[3500 3500 4000 4000 800]
SymbolLine[3000 3500 3500 3500 800]
SymbolLine[2500 4000 3000 3500 800]
SymbolLine[2500 4000 2500 4500 800]
SymbolLine[2500 4500 3000 5000 800]
SymbolLine[3000 5000 3500 5000 800]
)
Symbol['&' 1200]
(
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 1500 0 2500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 3500 1500 2000 800]
SymbolLine[500 5000 1000 5000 800]
SymbolLine[1000 5000 2000 4000 800]
SymbolLine[0 2500 2500 5000 800]
SymbolLine[500 1000 1000 1000 800]
SymbolLine[1000 1000 1500 1500 800]
SymbolLine[1500 1500 1500 2000 800]
SymbolLine[0 3500 0 4500 800]
```

```
)
Symbol[’’’ 1200]
(
SymbolLine[0 2000 1000 1000 800]
)
Symbol[’(’ 1200]
(
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 1500 0 4500 800]
)
Symbol[’)’ 1200]
(
SymbolLine[0 1000 500 1500 800]
SymbolLine[500 1500 500 4500 800]
SymbolLine[0 5000 500 4500 800]
)
Symbol[’*’ 1200]
(
SymbolLine[0 2000 2000 4000 800]
SymbolLine[0 4000 2000 2000 800]
SymbolLine[0 3000 2000 3000 800]
SymbolLine[1000 2000 1000 4000 800]
)
Symbol[’+’ 1200]
(
SymbolLine[0 3000 2000 3000 800]
SymbolLine[1000 2000 1000 4000 800]
)
Symbol[’,’ 1200]
(
SymbolLine[0 6000 1000 5000 800]
)
Symbol[’-’ 1200]
(
SymbolLine[0 3000 2000 3000 800]
)
Symbol[’.’ 1200]
(
SymbolLine[0 5000 500 5000 800]
)
Symbol[’/’ 1200]
(
```

```
SymbolLine[0 4500 3000 1500 800]
)
Symbol['0' 1200]
(
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4000 2000 2000 800]
)
Symbol['1' 1200]
(
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1000 1000 1000 5000 800]
SymbolLine[0 2000 1000 1000 800]
)
Symbol['2' 1200]
(
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 2000 1000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 2500 800]
SymbolLine[0 5000 2500 2500 800]
SymbolLine[0 5000 2500 5000 800]
)
Symbol['3' 1200]
(
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 3000 2000 3000 800]
)
Symbol['4' 1200]
(
SymbolLine[0 3000 2000 1000 800]
```

```
SymbolLine[0 3000 2500 3000 800]
SymbolLine[2000 1000 2000 5000 800]
)
Symbol['5' 1200]
(
SymbolLine[0 1000 2000 1000 800]
SymbolLine[0 1000 0 3000 800]
SymbolLine[0 3000 500 2500 800]
SymbolLine[500 2500 1500 2500 800]
SymbolLine[1500 2500 2000 3000 800]
SymbolLine[2000 3000 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['6' 1200]
(
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[0 3000 1500 3000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[2000 3500 2000 4500 800]
)
Symbol['7' 1200]
(
SymbolLine[0 5000 2500 2500 800]
SymbolLine[2500 1000 2500 2500 800]
SymbolLine[0 1000 2500 1000 800]
)
Symbol['8' 1200]
(
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
```

```
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 2500 500 3000 800]
SymbolLine[0 1500 0 2500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 2500 800]
SymbolLine[1500 3000 2000 2500 800]
)
Symbol['9' 1200]
(
SymbolLine[0 5000 2000 3000 800]
SymbolLine[2000 1500 2000 3000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 1500 0 2500 800]
SymbolLine[0 2500 500 3000 800]
SymbolLine[500 3000 2000 3000 800]
)
Symbol[':' 1200]
(
SymbolLine[0 2500 500 2500 800]
SymbolLine[0 3500 500 3500 800]
)
Symbol[';' 1200]
(
SymbolLine[0 5000 1000 4000 800]
SymbolLine[1000 2500 1000 3000 800]
)
Symbol['<' 1200]
(
SymbolLine[0 3000 1000 2000 800]
SymbolLine[0 3000 1000 4000 800]
)
Symbol['=' 1200]
(
SymbolLine[0 2500 2000 2500 800]
SymbolLine[0 3500 2000 3500 800]
)
Symbol['>' 1200]
(
SymbolLine[0 2000 1000 3000 800]
```

```
SymbolLine[0 4000 1000 3000 800]
)
Symbol['?' 1200]
(
SymbolLine[1000 3000 1000 3500 800]
SymbolLine[1000 4500 1000 5000 800]
SymbolLine[0 1500 0 2000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 2000 800]
SymbolLine[1000 3000 2000 2000 800]
)
Symbol['@' 1200]
(
SymbolLine[0 1000 0 4000 800]
SymbolLine[0 4000 1000 5000 800]
SymbolLine[1000 5000 4000 5000 800]
SymbolLine[5000 3500 5000 1000 800]
SymbolLine[5000 1000 4000 0 800]
SymbolLine[4000 0 1000 0 800]
SymbolLine[1000 0 0 1000 800]
SymbolLine[1500 2000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 3000 3500 800]
SymbolLine[3000 3500 3500 3000 800]
SymbolLine[3500 3000 4000 3500 800]
SymbolLine[3500 3000 3500 1500 800]
SymbolLine[3500 2000 3000 1500 800]
SymbolLine[2000 1500 3000 1500 800]
SymbolLine[2000 1500 1500 2000 800]
SymbolLine[4000 3500 5000 3500 800]
)
Symbol['A' 1200]
(
SymbolLine[0 1500 0 5000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 2000 1000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 5000 800]
SymbolLine[0 3000 2500 3000 800]
)
Symbol['B' 1200]
```

```
(
SymbolLine[0 5000 2000 5000 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[2500 3500 2500 4500 800]
SymbolLine[2000 3000 2500 3500 800]
SymbolLine[500 3000 2000 3000 800]
SymbolLine[500 1000 500 5000 800]
SymbolLine[0 1000 2000 1000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 2500 800]
SymbolLine[2000 3000 2500 2500 800]
)
Symbol['C' 1200]
(
SymbolLine[500 5000 2000 5000 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 2000 1000 800]
)
Symbol['D' 1200]
(
SymbolLine[500 1000 500 5000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 4500 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[0 5000 2000 5000 800]
SymbolLine[0 1000 2000 1000 800]
)
Symbol['E' 1200]
(
SymbolLine[0 3000 1500 3000 800]
SymbolLine[0 5000 2000 5000 800]
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 1000 2000 1000 800]
)
Symbol['F' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 1000 2000 1000 800]
SymbolLine[0 3000 1500 3000 800]
)
Symbol['G' 1200]
```

```
(
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[500 1000 2000 1000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 2000 5000 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[2500 3500 2500 4500 800]
SymbolLine[2000 3000 2500 3500 800]
SymbolLine[1000 3000 2000 3000 800]
)
Symbol['H' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[2500 1000 2500 5000 800]
SymbolLine[0 3000 2500 3000 800]
)
Symbol['I' 1200]
(
SymbolLine[0 1000 1000 1000 800]
SymbolLine[500 1000 500 5000 800]
SymbolLine[0 5000 1000 5000 800]
)
Symbol['J' 1200]
(
SymbolLine[0 1000 1500 1000 800]
SymbolLine[1500 1000 1500 4500 800]
SymbolLine[1000 5000 1500 4500 800]
SymbolLine[500 5000 1000 5000 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['K' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 3000 2000 1000 800]
SymbolLine[0 3000 2000 5000 800]
)
Symbol['L' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 5000 2000 5000 800]
)
```

```
Symbol['M' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 1000 1500 2500 800]
SymbolLine[1500 2500 3000 1000 800]
SymbolLine[3000 1000 3000 5000 800]
)
Symbol['N' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 1000 0 1500 800]
SymbolLine[0 1500 2500 4000 800]
SymbolLine[2500 1000 2500 5000 800]
)
Symbol['O' 1200]
(
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['P' 1200]
(
SymbolLine[500 1000 500 5000 800]
SymbolLine[0 1000 2000 1000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 2500 800]
SymbolLine[2000 3000 2500 2500 800]
SymbolLine[500 3000 2000 3000 800]
)
Symbol['Q' 1200]
(
SymbolLine[0 1500 0 4500 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1500 1000 800]
SymbolLine[1500 1000 2000 1500 800]
SymbolLine[2000 1500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
```

```
SymbolLine[0 4500 500 5000 800]
SymbolLine[1000 4000 2000 5000 800]
)
Symbol['R' 1200]
(
SymbolLine[0 1000 2000 1000 800]
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[2500 1500 2500 2500 800]
SymbolLine[2000 3000 2500 2500 800]
SymbolLine[500 3000 2000 3000 800]
SymbolLine[500 1000 500 5000 800]
SymbolLine[500 3000 2500 5000 800]
)
Symbol['S' 1200]
(
SymbolLine[2000 1000 2500 1500 800]
SymbolLine[500 1000 2000 1000 800]
SymbolLine[0 1500 500 1000 800]
SymbolLine[0 1500 0 2500 800]
SymbolLine[0 2500 500 3000 800]
SymbolLine[500 3000 2000 3000 800]
SymbolLine[2000 3000 2500 3500 800]
SymbolLine[2500 3500 2500 4500 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[500 5000 2000 5000 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['T' 1200]
(
SymbolLine[0 1000 2000 1000 800]
SymbolLine[1000 1000 1000 5000 800]
)
Symbol['U' 1200]
(
SymbolLine[0 1000 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[2000 1000 2000 4500 800]
)
Symbol['V' 1200]
(
SymbolLine[0 1000 0 4000 800]
```

```
SymbolLine[0 4000 1000 5000 800]
SymbolLine[1000 5000 2000 4000 800]
SymbolLine[2000 1000 2000 4000 800]
)
Symbol['W' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 5000 1500 3500 800]
SymbolLine[1500 3500 3000 5000 800]
SymbolLine[3000 1000 3000 5000 800]
)
Symbol['X' 1200]
(
SymbolLine[0 1000 0 1500 800]
SymbolLine[0 1500 2500 4000 800]
SymbolLine[2500 4000 2500 5000 800]
SymbolLine[0 4000 0 5000 800]
SymbolLine[0 4000 2500 1500 800]
SymbolLine[2500 1000 2500 1500 800]
)
Symbol['Y' 1200]
(
SymbolLine[0 1000 0 1500 800]
SymbolLine[0 1500 1000 2500 800]
SymbolLine[1000 2500 2000 1500 800]
SymbolLine[2000 1000 2000 1500 800]
SymbolLine[1000 2500 1000 5000 800]
)
Symbol['Z' 1200]
(
SymbolLine[0 1000 2500 1000 800]
SymbolLine[2500 1000 2500 1500 800]
SymbolLine[0 4000 2500 1500 800]
SymbolLine[0 4000 0 5000 800]
SymbolLine[0 5000 2500 5000 800]
)
Symbol['[' 1200]
(
SymbolLine[0 1000 500 1000 800]
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 5000 500 5000 800]
)
Symbol['\' 1200]
```

```
(
SymbolLine[0 1500 3000 4500 800]
)
Symbol[']' 1200]
(
SymbolLine[0 1000 500 1000 800]
SymbolLine[500 1000 500 5000 800]
SymbolLine[0 5000 500 5000 800]
)
Symbol['^' 1200]
(
SymbolLine[0 1500 500 1000 800]
SymbolLine[500 1000 1000 1500 800]
)
Symbol['_' 1200]
(
SymbolLine[0 5000 2000 5000 800]
)
Symbol['a' 1200]
(
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[2000 3000 2000 4500 800]
SymbolLine[2000 4500 2500 5000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
)
Symbol['b' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[2000 3500 2000 4500 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[0 3500 500 3000 800]
)
Symbol['c' 1200]
(
```

```
SymbolLine[500 3000 2000 3000 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 2000 5000 800]
)
Symbol['d' 1200]
(
SymbolLine[2000 1000 2000 5000 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
)
Symbol['e' 1200]
(
SymbolLine[500 5000 2000 5000 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[0 4000 2000 4000 800]
SymbolLine[2000 4000 2000 3500 800]
)
Symbol['f' 1000]
(
SymbolLine[500 1500 500 5000 800]
SymbolLine[500 1500 1000 1000 800]
SymbolLine[1000 1000 1500 1000 800]
SymbolLine[0 3000 1000 3000 800]
)
Symbol['g' 1200]
(
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1500 5000 800]
```

```
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[0 6000 500 6500 800]
SymbolLine[500 6500 1500 6500 800]
SymbolLine[1500 6500 2000 6000 800]
SymbolLine[2000 3000 2000 6000 800]
)
Symbol['h' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 5000 800]
)
Symbol['i' 1000]
(
SymbolLine[0 2000 0 2500 800]
SymbolLine[0 3500 0 5000 800]
)
Symbol['j' 1000]
(
SymbolLine[500 2000 500 2500 800]
SymbolLine[500 3500 500 6000 800]
SymbolLine[0 6500 500 6000 800]
)
Symbol['k' 1200]
(
SymbolLine[0 1000 0 5000 800]
SymbolLine[0 3500 1500 5000 800]
SymbolLine[0 3500 1000 2500 800]
)
Symbol['l' 1000]
(
SymbolLine[0 1000 0 4500 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['m' 1200]
(
SymbolLine[500 3500 500 5000 800]
SymbolLine[500 3500 1000 3000 800]
SymbolLine[1000 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 5000 800]
```

```
SymbolLine[2000 3500 2500 3000 800]
SymbolLine[2500 3000 3000 3000 800]
SymbolLine[3000 3000 3500 3500 800]
SymbolLine[3500 3500 3500 5000 800]
SymbolLine[0 3000 500 3500 800]
)
Symbol['n' 1200]
(
SymbolLine[500 3500 500 5000 800]
SymbolLine[500 3500 1000 3000 800]
SymbolLine[1000 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 5000 800]
SymbolLine[0 3000 500 3500 800]
)
Symbol['o' 1200]
(
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1500 3000 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[2000 3500 2000 4500 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['p' 1200]
(
SymbolLine[500 3500 500 6500 800]
SymbolLine[0 3000 500 3500 800]
SymbolLine[500 3500 1000 3000 800]
SymbolLine[1000 3000 2000 3000 800]
SymbolLine[2000 3000 2500 3500 800]
SymbolLine[2500 3500 2500 4500 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[1000 5000 2000 5000 800]
SymbolLine[500 4500 1000 5000 800]
)
Symbol['q' 1200]
(
SymbolLine[2000 3500 2000 6500 800]
SymbolLine[1500 3000 2000 3500 800]
SymbolLine[500 3000 1500 3000 800]
```

```
SymbolLine[0 3500 500 3000 800]
SymbolLine[0 3500 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
)
Symbol['r' 1200]
(
SymbolLine[500 3500 500 5000 800]
SymbolLine[500 3500 1000 3000 800]
SymbolLine[1000 3000 2000 3000 800]
SymbolLine[0 3000 500 3500 800]
)
Symbol['s' 1200]
(
SymbolLine[500 5000 2000 5000 800]
SymbolLine[2000 5000 2500 4500 800]
SymbolLine[2000 4000 2500 4500 800]
SymbolLine[500 4000 2000 4000 800]
SymbolLine[0 3500 500 4000 800]
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 2000 3000 800]
SymbolLine[2000 3000 2500 3500 800]
SymbolLine[0 4500 500 5000 800]
)
Symbol['t' 1000]
(
SymbolLine[500 1000 500 4500 800]
SymbolLine[500 4500 1000 5000 800]
SymbolLine[0 2500 1000 2500 800]
)
Symbol['u' 1200]
(
SymbolLine[0 3000 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
SymbolLine[2000 3000 2000 4500 800]
)
Symbol['v' 1200]
(
SymbolLine[0 3000 0 4000 800]
SymbolLine[0 4000 1000 5000 800]
```

```
SymbolLine[1000 5000 2000 4000 800]
SymbolLine[2000 3000 2000 4000 800]
)
Symbol['w' 1200]
(
SymbolLine[0 3000 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[500 5000 1000 5000 800]
SymbolLine[1000 5000 1500 4500 800]
SymbolLine[1500 3000 1500 4500 800]
SymbolLine[1500 4500 2000 5000 800]
SymbolLine[2000 5000 2500 5000 800]
SymbolLine[2500 5000 3000 4500 800]
SymbolLine[3000 3000 3000 4500 800]
)
Symbol['x' 1200]
(
SymbolLine[0 3000 2000 5000 800]
SymbolLine[0 5000 2000 3000 800]
)
Symbol['y' 1200]
(
SymbolLine[0 3000 0 4500 800]
SymbolLine[0 4500 500 5000 800]
SymbolLine[2000 3000 2000 6000 800]
SymbolLine[1500 6500 2000 6000 800]
SymbolLine[500 6500 1500 6500 800]
SymbolLine[0 6000 500 6500 800]
SymbolLine[500 5000 1500 5000 800]
SymbolLine[1500 5000 2000 4500 800]
)
Symbol['z' 1200]
(
SymbolLine[0 3000 2000 3000 800]
SymbolLine[0 5000 2000 3000 800]
SymbolLine[0 5000 2000 5000 800]
)
Symbol['{' 1200]
(
SymbolLine[500 1500 1000 1000 800]
SymbolLine[500 1500 500 2500 800]
SymbolLine[0 3000 500 2500 800]
SymbolLine[0 3000 500 3500 800]
```

```
SymbolLine[500 3500 500 4500 800]
SymbolLine[500 4500 1000 5000 800]
)
Symbol['|' 1200]
(
SymbolLine[0 1000 0 5000 800]
)
Symbol['}' 1200]
(
SymbolLine[0 1000 500 1500 800]
SymbolLine[500 1500 500 2500 800]
SymbolLine[500 2500 1000 3000 800]
SymbolLine[500 3500 1000 3000 800]
SymbolLine[500 3500 500 4500 800]
SymbolLine[0 5000 500 4500 800]
)
Symbol['~' 1200]
(
SymbolLine[0 3500 500 3000 800]
SymbolLine[500 3000 1000 3000 800]
SymbolLine[1000 3000 1500 3500 800]
SymbolLine[1500 3500 2000 3500 800]
SymbolLine[2000 3500 2500 3000 800]
)
Via[10000 123000 3500 2000 3500 3500 "" 0x8000020a]
Via[10000 9000 3500 2000 3500 3500 "" 0x8000020a]
Via[194000 122000 3500 2000 3500 3500 "" 0x8000020a]
Via[194000 9000 3500 2000 3500 3500 "" 0x8000020a]

Element[0x00000000 "CONNECTOR-3-1" "CONN3" "0" 10000 24000 11000 -5000 3 1
(
Pin[0 0 6000 3000 6600 3000 "1" "1" 0x80000101]
Pin[0 10000 6000 3000 6600 3000 "2" "2" 0x80000001]
Pin[0 20000 6000 3000 6600 3000 "3" "3" 0x80000001]
ElementLine [5000 5000 5000 -5000 1000]
ElementLine [-5000 5000 5000 5000 1000]
ElementLine [5000 -5000 -5000 -5000 2000]
ElementLine [5000 25000 5000 -5000 2000]
ElementLine [-5000 25000 5000 25000 2000]
ElementLine [-5000 -5000 -5000 25000 2000]

)
```

```
Element[0x00000000 "AXIAL_LAY-300" "C1" "6.8uF" 183000 64000 -4500 17000 3
(
Pin[0 -7500 5000 3000 5600 2000 "1" "1" 0x80000101]
Pin[0 22500 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [-2500 0 2500 0 1000]
ElementLine [-2500 0 -2500 15000 1000]
ElementLine [-2500 15000 2500 15000 1000]
ElementLine [2500 0 2500 15000 1000]
ElementLine [0 15000 0 22500 1000]
ElementLine [0 -7500 0 0 1000]

)

Element[0x00000000 "CONNECTOR-4-1" "CONN1" "0" 10000 60000 11000 -5000 3 1
(
Pin[0 0 6000 3000 6600 3000 "1" "1" 0x80000101]
Pin[0 10000 6000 3000 6600 3000 "2" "2" 0x80000001]
Pin[0 20000 6000 3000 6600 3000 "3" "3" 0x80000001]
Pin[0 30000 6000 3000 6600 3000 "4" "4" 0x80000001]
ElementLine [5000 5000 5000 -5000 1000]
ElementLine [-5000 5000 5000 5000 1000]
ElementLine [5000 -5000 -5000 -5000 2000]
ElementLine [5000 35000 5000 -5000 2000]
ElementLine [-5000 35000 5000 35000 2000]
ElementLine [-5000 -5000 -5000 35000 2000]

)

Element[0x00000000 "R_AJ_V" "R10" "NULL" 151000 89000 -9000 500 2 100 0x00
(
Pin[0 0 8000 3000 8600 3000 "1" "1" 0x80004101]
Pin[-20000 0 8000 3000 8600 3000 "3" "2" 0x80004001]
Pin[-10000 5000 8000 3000 8600 3000 "2" "3" 0x80004001]
ElementLine [-25000 -5000 5000 -5000 2000]
ElementLine [-25000 -5000 -25000 5000 2000]
ElementLine [-25000 5000 5000 5000 2000]
ElementLine [5000 -5000 5000 5000 2000]

)

Element[0x00000000 "R025" "R3" "100K" 77000 97000 -4000 -17000 1 100 0x000
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80000101]
```

```
Pin[0 -40000 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [0 -40000 0 -30000 2000]
ElementLine [0 -10000 0 0 2000]
ElementLine [-5000 -10000 5000 -10000 2000]
ElementLine [5000 -30000 5000 -10000 2000]
ElementLine [-5000 -30000 5000 -30000 2000]
ElementLine [-5000 -30000 -5000 -10000 2000]

)

Element[0x00000000 "R025" "R7" "100K" 168000 96000 -2000 -16000 1 100 0x00
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80000101]
Pin[0 -40000 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [0 -40000 0 -30000 2000]
ElementLine [0 -10000 0 0 2000]
ElementLine [-5000 -10000 5000 -10000 2000]
ElementLine [5000 -30000 5000 -10000 2000]
ElementLine [-5000 -30000 5000 -30000 2000]
ElementLine [-5000 -30000 -5000 -10000 2000]

)

Element[0x00000000 "AXIAL_LAY-300" "C2" "6.8uF" 101000 105000 -3000 -5000
(
Pin[0 7500 5000 3000 5600 2000 "1" "1" 0x80000101]
Pin[0 -22500 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [-2500 0 2500 0 1000]
ElementLine [2500 -15000 2500 0 1000]
ElementLine [-2500 -15000 2500 -15000 1000]
ElementLine [-2500 -15000 -2500 0 1000]
ElementLine [0 -22500 0 -15000 1000]
ElementLine [0 0 0 7500 1000]

)

Element[0x00000000 "R025" "R5" "100K" 90000 32000 12000 -2000 0 100 0x0000
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80004101]
Pin[40000 0 5000 3000 5600 2000 "2" "2" 0x80004001]
ElementLine [30000 0 40000 0 2000]
ElementLine [0 0 10000 0 2000]
ElementLine [10000 -5000 10000 5000 2000]
```

```
ElementLine [10000 5000 30000 5000 2000]
ElementLine [30000 -5000 30000 5000 2000]
ElementLine [10000 -5000 30000 -5000 2000]

)

Element[0x00000000 "DIL-8-300" "U1" "0" 126000 45000 8000 3000 0 100 0x000
(
Pin[0 0 6000 3000 6600 2800 "1" "1" 0x80000101]
Pin[0 10000 6000 3000 6600 2800 "2" "2" 0x80000001]
Pin[0 20000 6000 3000 6600 2800 "3" "3" 0x80000001]
Pin[0 30000 6000 3000 6600 2800 "4" "4" 0x80000001]
Pin[30000 30000 6000 3000 6600 2800 "5" "5" 0x80000001]
Pin[30000 20000 6000 3000 6600 2800 "6" "6" 0x80000001]
Pin[30000 10000 6000 3000 6600 2800 "7" "7" 0x80000001]
Pin[30000 0 6000 3000 6600 2800 "8" "8" 0x80000001]
ElementLine [20000 -5000 35000 -5000 1000]
ElementLine [-5000 -5000 10000 -5000 1000]
ElementLine [35000 -5000 35000 35000 1000]
ElementLine [-5000 35000 35000 35000 1000]
ElementLine [-5000 -5000 -5000 35000 1000]
ElementArc [15000 -5000 5000 5000 0 180 1000]

)

Element[0x00000000 "R025" "R6" "100K" 114000 75000 2000 12000 3 100 0x0000
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80000101]
Pin[0 40000 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [0 30000 0 40000 2000]
ElementLine [0 0 0 10000 2000]
ElementLine [-5000 10000 5000 10000 2000]
ElementLine [-5000 10000 -5000 30000 2000]
ElementLine [-5000 30000 5000 30000 2000]
ElementLine [5000 10000 5000 30000 2000]

)

Element[0x00000000 "R025" "R4" "100K" 91000 106000 -2000 -12000 1 100 0x00
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80000101]
Pin[0 -40000 5000 3000 5600 2000 "2" "2" 0x80000001]
ElementLine [0 -40000 0 -30000 2000]
```

```
ElementLine [0 -10000 0 0 2000]
ElementLine [-5000 -10000 5000 -10000 2000]
ElementLine [5000 -30000 5000 -10000 2000]
ElementLine [-5000 -30000 5000 -30000 2000]
ElementLine [-5000 -30000 -5000 -10000 2000]

)

Element[0x00000000 "R025" "R1" "100K" 26000 71000 13000 -3000 0 100 0x0000
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80004101]
Pin[40000 0 5000 3000 5600 2000 "2" "2" 0x80004001]
ElementLine [30000 0 40000 0 2000]
ElementLine [0 0 10000 0 2000]
ElementLine [10000 5000 10000 -5000 2000]
ElementLine [30000 5000 10000 5000 2000]
ElementLine [30000 -5000 30000 5000 2000]
ElementLine [10000 -5000 30000 -5000 2000]

)

Element[0x00000000 "R025" "R2" "100K" 26000 59000 12000 -2000 0 100 0x0000
(
Pin[0 0 5000 3000 5600 2000 "1" "1" 0x80004101]
Pin[40000 0 5000 3000 5600 2000 "2" "2" 0x80004001]
ElementLine [30000 0 40000 0 2000]
ElementLine [0 0 10000 0 2000]
ElementLine [10000 5000 10000 -5000 2000]
ElementLine [30000 5000 10000 5000 2000]
ElementLine [30000 -5000 30000 5000 2000]
ElementLine [10000 -5000 30000 -5000 2000]

)

Element[0x00000000 "CONNECTOR-4-1" "CONN2" "0" 194000 53000 13000 -3000 3
(
Pin[0 0 6000 3000 6600 3000 "1" "1" 0x80000101]
Pin[0 10000 6000 3000 6600 3000 "2" "2" 0x80000001]
Pin[0 20000 6000 3000 6600 3000 "3" "3" 0x80000001]
Pin[0 30000 6000 3000 6600 3000 "4" "4" 0x80000001]
ElementLine [5000 -5000 5000 5000 1000]
ElementLine [-5000 5000 5000 5000 1000]
ElementLine [-5000 -5000 5000 -5000 2000]
```

```
ElementLine [5000 -5000 5000 35000 2000]
ElementLine [-5000 35000 5000 35000 2000]
ElementLine [-5000 -5000 -5000 35000 2000]

)
Layer(1 "solder")
(
)
Layer(2 "GND-sldr")
(
)
Layer(3 "Vcc-sldr")
(
)
Layer(4 "component")
(
)
Layer(5 "GND-comp")
(
)
Layer(6 "Vcc-comp")
(
)
Layer(7 "unused")
(
)
Layer(8 "unused")
(
Line[26000 59000 13000 59000 2500 1000 0x000000a0]
Line[13000 59000 12000 60000 2500 1000 0x00000080]
Line[10000 60000 12000 60000 2500 1000 0x00000080]
Line[179999 56000 180499 56500 2500 1000 0x00000080]
Line[189000 54000 186500 56500 2500 1000 0x000000a0]
Line[189000 54000 193000 54000 2500 1000 0x00000080]
Line[4750 83676 10000 88926 2500 1000 0x000000a0]
Line[10000 70000 12000 70000 2500 1000 0x00000080]
Line[101000 112500 101000 115000 2500 1000 0x00000080]
Line[158000 55000 159000 56000 2500 1000 0x000000a0]
Line[156000 55000 158000 55000 2500 1000 0x000000a0]
Line[10000 90000 10000 88926 2500 1000 0x00000080]
Line[4750 39250 4750 83676 2500 1000 0x000000a0]
Line[4750 39250 10000 34000 2500 1000 0x000000a0]
Line[77000 97000 70000 90000 2500 1000 0x000000a0]
```

```
Line[194000 53000 193000 54000 2500 1000 0x00000080]
Line[91000 106000 82000 97000 2500 1000 0x000000a0]
Line[82000 97000 77000 97000 2500 1000 0x000000a0]
Line[186500 56500 180499 56500 2500 1000 0x00000080]
Line[154102 123250 108898 123250 2500 1000 0x000000a0]
Line[183000 86500 183000 94352 2500 1000 0x000000a0]
Line[94500 106000 101000 112500 2500 1000 0x000000a0]
Line[183000 86500 190500 86500 2500 1000 0x000000a0]
Line[126000 75000 108500 75000 2500 1000 0x000000a0]
Line[152176 119750 178250 93676 2500 1000 0x000000a0]
Line[101000 82500 108500 75000 2500 1000 0x000000a0]
Line[101000 94250 109250 102500 2500 1000 0x000000a0]
Line[109250 102500 109250 118176 2500 1000 0x000000a0]
Line[110824 119750 152176 119750 2500 1000 0x000000a0]
Line[178250 93676 178250 83324 2500 1000 0x000000a0]
Line[188574 73000 194000 73000 2500 1000 0x000000a0]
Line[114000 75000 111499 75000 2500 1000 0x00000080]
Line[101000 115352 101000 112500 2500 1000 0x00000080]
Line[66000 71000 71000 66000 2500 1000 0x00000080]
Line[123999 65000 122999 66000 2500 1000 0x00000080]
Line[123999 65000 126000 65000 2500 1000 0x00000080]
Line[122999 66000 71000 66000 2500 1000 0x000000a0]
Line[77000 57000 77000 45000 2500 1000 0x000000a0]
Line[122000 24000 10000 24000 2500 1000 0x000000a0]
Line[74499 57000 72499 59000 2500 1000 0x000000a0]
Line[122999 55000 120999 57000 2500 1000 0x000000a0]
Line[122999 55000 126000 55000 2500 1000 0x00000080]
Line[66000 59000 72499 59000 2500 1000 0x000000a0]
Line[130000 32000 130000 39000 2500 1000 0x000000a0]
Line[122000 24000 130000 32000 2500 1000 0x000000a0]
Line[130750 39750 130750 39750 2500 1000 0x00000080]
Line[132499 41499 132499 41499 2500 1000 0x00000080]
Line[135499 44499 135499 44499 2500 1000 0x00000080]
Line[114000 115000 116500 112500 2500 1000 0x000000a0]
Line[141000 61074 126000 46074 2500 1000 0x000000a0]
Line[126000 45000 126000 46074 2500 1000 0x00000080]
Line[128076 98999 131000 98999 2500 1000 0x000000a0]
Line[141000 61074 141000 94000 2500 1000 0x000000a0]
Line[156000 75000 156000 84000 2500 1000 0x000000a0]
Line[156000 84000 151000 89000 2500 1000 0x00000080]
Line[116500 110575 128076 98999 2500 1000 0x000000a0]
Line[116500 110575 116500 112500 2500 1000 0x000000a0]
Line[132251 100250 131000 98999 2500 1000 0x00000080]
```

```
Line[168000 96000 163750 100250 2500 1000 0x000000a0]
Line[163750 100250 132251 100250 2500 1000 0x000000a0]
Line[109250 118176 110824 119750 2500 1000 0x000000a0]
Line[26000 71000 13000 71000 2500 1000 0x000000a0]
Line[12000 70000 13000 71000 2500 1000 0x00000080]
Line[131000 98999 131000 89000 2500 1000 0x000000a0]
Line[120999 57000 74499 57000 2500 1000 0x000000a0]
Line[101000 94250 101000 82500 2500 1000 0x000000a0]
Line[179999 56000 159000 56000 2500 1000 0x000000a0]
Line[183000 56500 180499 56500 2500 1000 0x000000a0]
Line[156000 65000 130000 39000 2500 1000 0x000000a0]
Line[90000 32000 77000 45000 2500 1000 0x000000a0]
Line[178250 83324 188574 73000 2500 1000 0x000000a0]
Line[194000 83000 190500 86500 2500 1000 0x000000a0]
Line[183000 94352 154102 123250 2500 1000 0x000000a0]
Line[94500 106000 91000 106000 2500 1000 0x00000080]
Line[101000 115352 108898 123250 2500 1000 0x000000a0]
Line[70000 90000 10000 90000 2500 1000 0x000000a0]
Polygon(0x00000010)
(
[0 0] [207000 0] [207000 133000] [0 133000]
)
)
Layer(9 "silk")
(
)
Layer(10 "silk")
(
)
NetList()
(
Net("unnamed_net11" "(unknown)")
(
Connect("CONN1-1")
Connect("R2-1")
)
Net("unnamed_net10" "(unknown)")
(
Connect("CONN1-2")
Connect("R1-1")
)
Net("unnamed_net9" "(unknown)")
(
```

```
Connect("R7-1")
Connect("R6-2")
Connect("R10-2")
)
Net("unnamed_net8" "(unknown)")
(
Connect("R10-1")
Connect("U1-5")
)
Net("unnamed_net7" "(unknown)")
(
Connect("R10-3")
Connect("U1-1")
)
Net("unnamed_net6" "(unknown)")
(
Connect("CONN3-1")
Connect("R5-2")
Connect("U1-6")
)
Net("unnamed_net5" "(unknown)")
(
Connect("R2-2")
Connect("R5-1")
Connect("R3-2")
Connect("U1-2")
)
Net("unnamed_net4" "(unknown)")
(
Connect("R1-2")
Connect("R4-2")
Connect("U1-3")
)
Net("unnamed_net3" "(unknown)")
(
Connect("CONN2-3")
Connect("R6-1")
Connect("U1-4")
Connect("C2-2")
)
Net("unnamed_net2" "(unknown)")
(
Connect("CONN1-4")
```

```
Connect("CONN3-2")
Connect("CONN2-4")
Connect("R3-1")
Connect("R4-1")
Connect("C2-1")
Connect("C1-2")
)
Net("unnamed_net1" "(unknown)")
(
Connect("CONN2-1")
Connect("R7-2")
Connect("U1-7")
Connect("C1-1")
)
)
```

J mySPICE.sch

```
v 20040111 1
C 27800 59200 1 0 0 aop-spice-1.sym
{
T 28500 60000 5 10 1 1 0 0 1
refdes=XOA1
T 27800 59200 5 10 1 1 0 0 1
model-name=UA741
T 27800 59200 5 10 1 1 0 0 1
file=/home/alain/alain/spice/dummy8/UA741.lib
T 27800 59200 5 10 1 1 0 0 1
value=UA741
}
C 22500 59500 1 270 0 gnd-1.sym
C 28200 56800 1 0 0 gnd-1.sym
C 28600 61900 1 180 0 vdc-1.sym
{
T 27900 61250 5 10 1 1 180 0 1
refdes=Vee
T 27900 61450 5 10 1 1 180 0 1
value=DC -12
}
C 28000 57100 1 0 0 vdc-1.sym
{
T 28700 57550 5 10 1 1 0 0 1
value=DC 12
T 28000 57100 5 10 1 1 0 0 1
refdes=Vcc
}
N 28300 59200 28300 58300 4
{
T 28300 59200 5 10 1 1 0 0 1
netname=vp
}
N 28300 60000 28300 60700 4
{
T 28300 60000 5 10 1 1 0 0 1
netname=vn
}
C 28400 62200 1 180 0 gnd-1.sym
N 28800 59600 32000 59600 4
```

```
{
T 28800 59600 5 10 1 1 0 0 1
netname=out
}
N 27400 55300 27400 59400 4
N 30600 55300 30600 59600 4
N 27800 59400 25600 59400 4
{
T 27800 59400 5 10 1 1 0 0 1
netname=ni
}
N 29900 55300 30600 55300 4
C 27500 59900 1 270 0 gnd-1.sym
C 22800 59700 1 270 0 vsin-1.sym
{
T 23450 59000 5 10 1 1 270 0 1
refdes=Vin
T 23250 59000 5 10 1 1 270 0 1
value=sin 0 1 1KHZ
}
C 24700 59300 1 0 0 resistor-1.sym
{
T 24900 59600 5 10 1 1 0 0 1
refdes=R1
T 24700 59300 5 10 1 1 0 0 1
value=10K
T 24700 59300 5 10 1 1 0 0 1
model=RES
}
N 24700 59400 24000 59400 4
{
T 24700 59400 5 10 1 1 0 0 1
netname=in
}
C 29000 55200 1 0 0 resistor-1.sym
{
T 29200 55500 5 10 1 1 0 0 1
refdes=R2
T 29000 55200 5 10 1 1 0 0 1
value=100K
T 29000 55200 5 10 1 1 0 0 1
model=RES
}
```

```
N 27400 55300 29000 55300 4
```

K output.net

```
**************************************************************
* Spice file generated by gnetlist                           *
* spice-sdb version 12.29.2003.c by SDB --                   *
* provides advanced spice netlisting capability.             *
* Documentation at http://www.brorson.com/gEDA/SPICE/        *
**************************************************************
R2 ni out 100K
R1 in ni 10K
Vin in 0 sin 0 1 1KHZ
Vcc vp 0 DC 12
Vee vn 0 DC -12
XOA1 0 ni vp vn out UA741
*vvvvvvvv  Included SPICE model from /home/alain/alain/spice/dummy8/UA741
*--------------------------------------------------------------------------
* connections:   non-inverting input
*                |  inverting input
*                |  |  positive power supply
*                |  |  |  negative power supply
*                |  |  |  |  output
*                |  |  |  |  |
*.subckt uA741    1 2 3 4 5
*
.subckt UA741    1 2 3 4 5
  c1   11 12 8.661E-12
  c2    6  7 30.00E-12
  dc    5 53 dx
  de   54  5 dx
  dlp  90 91 dx
  dln  92 90 dx
  dp    4  3 dx
  egnd 99  0 poly(2) (3,0) (4,0) 0 .5 .5
  fb    7 99 poly(5) vb vc ve vlp vln 0 10.61E6 -10E6 10E6 10E6 -10E6
  ga    6  0 11 12 188.5E-6
  gcm   0  6 10 99 5.961E-9
  iee  10  4 dc 15.16E-6
  hlim 90  0 vlim 1K
  q1   11  2 13 qx
```

```
  q2   12  1 14 qx
  r2    6  9 100.0E3
  rc1   3 11 7.957E3
  rc2   3 12 7.957E3
  re1  13 10 2.740E3
  re2  14 10 2.740E3
  ree  10 99 19.69E6
  ro1   8  5 150
  ro2   7 99 150
  rp    3  4 18.11E3
  vb    9  0 dc 0
  vc    3 53 dc 2.600
  ve   54  4 dc 2.600
  vlim  7  8 dc 0
  vlp  91  0 dc 25
  vln   0 92 dc 25
.model dx D(Is=800.0E-18)
.model qx NPN(Is=800.0E-18 Bf=62.5)
.ends


*^^^^^^^^  End of included SPICE model from /home/alain/alain/spice/dummy
*
.END
```

L Bibliography

References

[1] http://mechatronics.mech.northwestern.edu/design_ref/electrical_design/opamps.html

[2] www.uoguelph.ca/∼antoon/gadgets/741/741.html

[3] http://alternatezone.com/electronics/pcbdesign.htm

[4] htt://geda.seul.org

[5] http://www.brorson.com/gEDA/SPICE/intro.html

[6] http://www.inblue.com/gEDA/symlibrary

[7] http://newton.ex.ac.uk/teaching/CDHW/Electronics2/userguide/

[8] http://www.elektronikschule.de/∼krausg/APLAC_Schaltungssimulation/Aplac_cd_version_
Models/

[9] http://homepages.which.net/∼paul.hills/Circuits/Spice/ModelIndex.html

[10] http://www.brorson.com/gEDA/SPICE/spice-20030321.tar.gz

Index