Design and Simulation of DC-DC converters using open source tools

Prof. L. Umanand
Department of Electronics System Engineering
Indian Institute of Science, Bangalore

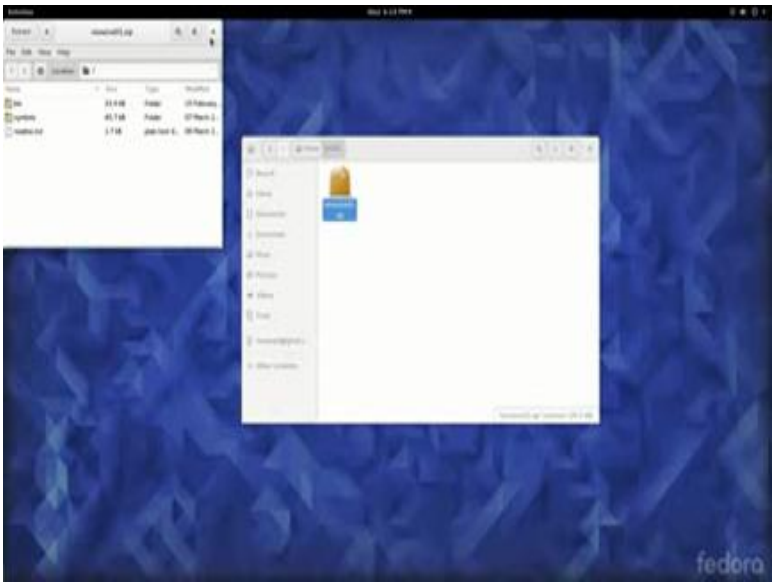
Lecture – 03
Setting up gEDA, ngSpice and Octave

In this video capsule we shall look into the installation process for installing the open source software packages under our system such that you will be able to make effective simulation for this course. The plan is to install gEDA software toolset. This is the electronic design automation toolset and then we shall install ngSpice a very popular package for circuit simulation and then after this, we shall install octave. Octave is a MATLAB like environment which is very helpful in scripting and automating many of your design tasks. So these three set of software packages we need to install I will show you by taking a walkthrough and then configuring the system to our needs.

I have a fedora 23 Linux desktop. So, I will show you by taking a walkthrough the process in this particular operating system. My students have checked it out on ubuntu also. So, I guess that it should be a fine even in other Linux distributions. On windows I have not tried nor have my students tried. So, I am not too aware on how it will perform on windows. So, I suggest that if anybody of you are trying it on windows that is EDA ngSpice and Octave. You can try and then let us know how it performs, or you can use alternative packages whatever is available on the windows platform. With this we shall start now the process of installing these software packages.

We shall begin the installation process by first downloading the resource from the Google drive this is the fedora desktop and let us open the folders and here I have created a dc-dc converter folder and within that folder I have downloaded this resource 0 1 dot zip folder file. This is located in Google drive and the link for this is given in the course website.

(Refer Slide Time: 03:19)



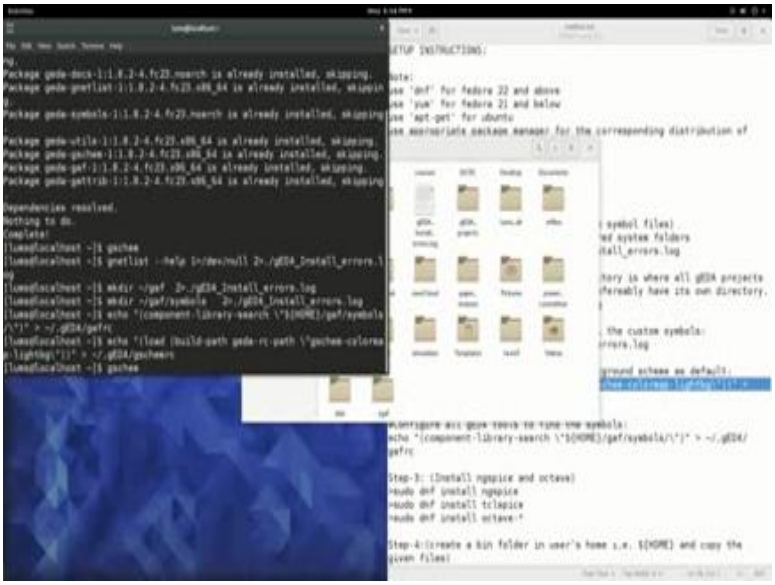
This has two folders and one file readme dot text file there is a bin folder and a symbols folder. Of course, I will explain to you later what are the functions and how we will use that one, for now extract that unzip it and within that you see this folder. Readme dot text gives you a step by step instruction on how to go above this process of installation. So, open that readme dot text file and keep it to one side and all we have to do is follow through the steps the commands are given here you just may have to copy and paste it if it is a fedora system or equivalently type in the command line commands if it is (Refer Time: 04:35) system.

(Refer Slide Time: 04:25)

Again go to the desktop and open a terminal let me adjust the size, that I will be able to see that. Now let us copy this and paste it here. It will ask for the password and then execute it. So, in my case I have already installed all these packages it will search and look for the packages in the repository and install them. So, you have a gschem check, the gschem docs, netlist symbols, utils gschem, the gEDA, g schematic. This is what we will be using at great length in this particular course. So, dependency is resolved nothing to do everything has been installed. In your case if you are not yet installed gEDA you will see that it will give a list of packages and ask you whether to install or not you say yes and it will get installed.

Next we need to configure gEDA you can. In fact, open g schematics and see what happens what (Refer Time: 06:03). So, let me type g schem enter and you will see a g y something like this opens up. There is a status dialog which opens which just gives you an indication of a the paths which from where it takes this is the g schematic r c file gaf r c file where it is located it is located in user share g e d a; however, these are in the this need root permission to modify these files; however, if you want to customize you need to have a copy of these files in your user dot gEDA folder which I will tell you about shortly. Close this; this is how the gEDA user interface will look like. This is having a dark background some people may like this background fine, but I would prefer to have a light background and I would like to have the default g schem opening with a light b g.

(Refer Slide Time: 07:37)



Now, here are the components and this is where you have libraries where a list of components coming here and then you can choose these components to place on to a sheet. Close this for now; we know that gEDA is installed working and running and it is opening up. At this point I would like to configure it I will show you how to configure it to make it into a light background one. You may want to change it or you may not want to change it you may want to retain the dark background it is up to you, but let me show you how to go about doing that.

So, you see here in the readme dot text file, g netlist just go through this command syntax copy and then paste. So, obvious this is just a gEDA command. What it would do; is that it will create a dot gEDA folder in the users (Refer Time: 08:37) directory and into that directory is where we would like to have the r c files. So, if I go into home and do control h or alternate the height it will show the hidden files with the dot prefix. So, you see that this has got created dot gEDA and in into of this you have logs and a geometry file. Here we have to put the g schem r c in the gaf r c files anyway. We will come back to that later and we will see what to do next.

Next we would like to make a gaf directory and this is where we would like to place the symbols of gEDA. The symbol the custom symbols that we make that has to be placed within this. So, we will give it a place. So, just copy this and paste it here on to the terminal and press enter. So, what it would have done now is create this gaf directory g a f it is right now empty, but we want to fill it up with symbols; custom symbols. Next we create within gaf another sub directory called symbols or that is where we shall place our symbols; copy, paste, enter. This if 2 than redirect it to errors dot log is just an indication if there are any errors to see you can go and look at this log file see what happened there are no errors nothing will be entered there.

So, now after having run this particular line within gaf you would have this and inside that let us place the symbols. So, what we shall do we go into this dc-dc converter folder and within that we have the resource folder and in that we have the symbols folder. In the symbols folder there are three sub folders; one is called a block, a bond and a comps. A block has many symbols associated with add clock gain integration, multiplication, P I D, P W M bridge, P W M single phase sample and hold s r latch summations so and so forth. So, these are custom blocks more something about logic oriented not much of power flow and interesting to have them.

I will come to this later. We may not be using this, but anyway you I would like to share that with you. The a comps are the component block the d folder the generic components capacitor, default divert, a power divert, a source, a flow source, current source, a gyrator a inductions, power switch, a resistance, s c r signal switch, a transformer, a transformer bridge transfer for bridge, another fly back transformer, forward converter transformer, a push pull transformer and a normal transformer. So, these are symbols which will be useful this is particularly symbols used for a bond graph simulation. So, people who are familiar with bond graphs and have done bond graph modeling and simulation they may use it, but right now we shall not use it, but; however, I will share this thing this from the folder also with you, you copy all these, copy and go into the gaf go into symbols and paste them and now we are in business. We have all the symbols in the proper directory now we have to assign the proper path.

So, now that we have done that. We will assign the path and create the respective files. Now there are two commands here; one is a command component library search where it will search through this path dollar within braces home is environment variable which will take the root for a root users path then into gaf and into symbols. Now this should be the path of all the component library search and they should get reflected in your g schematic when you open it and this is pushed into a file which is residing in dot gEDA and into a gaf r c file if gaf r c is not there it will get created. So, this is your own custom gaf r c file it will look search for there and then see this and then set the path environment path. So, we can have this copy and paste it here and run that one.

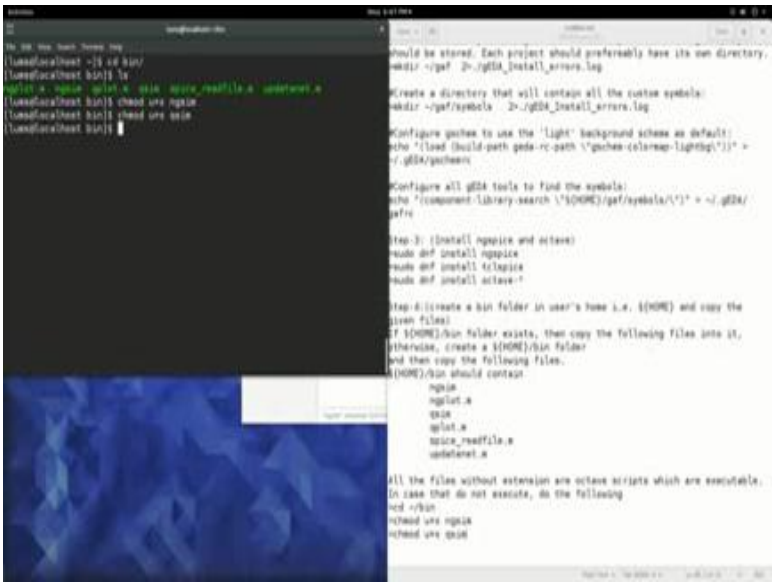
So, once you run that if you go down into the hidden dot gEDA file I have unhidden that. So, into that you will see that there is a gaf r c file that is created. So, into the gaf r c file see whatever we have typed common library search that would have coming here. So, when gEDA opens up it will look into this file see this path declaration and then appropriately remember the path.

Next we would like to make the background light. So, that is this load build par g d a gschem r c into that gschem r c we will put this; g sym color map light b g. So, instead of that b g we want a light b g and then put that into dot g d a gschem r c five. So, we will copy that and paste it here. So, that would have got entered in there. So, you see here gschem r c has been created and within that this has been copied. So, it will look into there and automatically put in the proper background color.

Now, I will do control h and hide all the doc prefix files. So, that it does not clatter up the folders. Now again now you clatter gschem. Now you see the background is light. You see that it takes the r c from the local users dot g d a directory gschem r c also from the local. So, which working properly you can open it to full screen. You can use the wheel button of the mouse to zoom in zoom out and then go into the components now you see that our custom libraries a block, a bond, components are reflecting here. So, if I open it see they are all coming in here the symbols and that is very nice and then the component. Now close this and back again to the terminal. Going back to the readme dot text file the next part of the simulation is installing ngSpice.

So, installing ngSpice is pretty straight forward. You just copy this; d n f installs ngSpice and let me clear this terminal screen control l and that is cleared paste.

(Refer Slide Time: 18:07)



So, pseudo (Refer Time: 18:12) now install ngSpice enter the password and of course, in my case here it is already installed and case you yours is not installed it will save at this particular package has to be installed whether to install or not click yes and then you will be true and then after that you need to install t c l spice also though actually you may not t c l spice there are some libraries in t c l spice that are useful for ngSpice and kindly to install t c l spice too and here again it is already installed here and then you will get such a message.

So, now ngSpice is also in place and if you want to test that you just type in ngSpice dash dash help. So, that will give you menu of options and how to use go through that and I would recommend that would down load the ngSpice manual and read through it if you are not if you do not know how to spice. It is very very similar to this spice of the windows. It is actually a world out of the (Refer Time: 19:40) spice and quite universally used by many people. Look at I just (Refer Time: 19:51) ngSpice and it will go into the ngSpice and run. It is ngSpice version 26 plus and developed by (Refer Time: 20:04) group then you have the ngSpice environment prompt.

So, I will quit from this here knowing that ngSpice is working. And now we have one more package to install and that is octave. Octave is a very very powerful tool you should have that it is a open source equivalent of MATLAB sometimes I find it much more powerful than MATLAB any way that is a good package to have on your system. Paste it here I was used d n f install I have used d n f everywhere while during this process because this fedora 23. Fedora 22 onwards we have to use d n f for the package installer or the package installer before fedora 22 it use to be m.

So, people having fedora version earlier than 22 kindly use m y u m and in the case ubuntu people can use the app get package installed. So, use this press enter and the rest the package installer will do the job and here again you see everything is already installed and dependency are resolved nothing else to do. Now, to test whether octave is running properly or not just type octave; you will now see a g u i popping up very nice g u I it looks much like the MATLAB g u I this is where the work space window is and this is where you will probably windows all the work.

So, octave is working this is (Refer Time: 21:57) version four plus. A most of the earlier versions of the octave they may not have a g u i integrated, but it will work on the terminal just like the ngSpice environment. So, if you want to work in such an environment which I also prefer you type octave dash no g u i. So, then you will get it in the terminal this is very fast and you will be able to do equivalently good programming with this terminal environment through.

So, let me quit that and with this we have installed three packages. One is the e d a electronic design automation next is the spice ngSpice and t c l spice have been installed and then octave and we have installed it on fedora system. I hope that you will not have a

problem in another distribution too kindly follow it and then look into help on the net too, if you run into problems and there is of course, a forum to discuss. There is one last job which we need to do which is the step four and that is something which you need not do, but something that I would like to share and which may be your help.

So, this is not compulsory. So, what I request you to do is if you do not have a bin folder in your user home create one; create a bin folder. So, you see that a bin folder is created then go into the d c d c there is this bin folder there are six files here n g plot, n g schem, q plot, q scheme, spice read file and this update net dot m. So, you just copy both of them and go back there into the bin and paste fine n gschem and q schem should be made executable they both are octave scripts. Octave scripts meaning that it starts with the hash exclamation user bin octave. So, it calls it runs octave directly from the terminal. So, these are useful scripts I will tell you of more about this at the time they are needed, but its better you keep it ready.

So, to make these two executable in the read me file I have put these two comments change mode u plus x q schem. So, they will be made executable. So, what you have to do go into the terminal c d bin. So, you will see that it has the six files c h mode u plus x n gschem. So, that will be made executable c h mod q plus x using that also will be made executable. So, that is it your software installation portion is over.

Next, we can start on building our simulation schematics and simulate the circuits. This portion this installation portion is one time. So, it is worth putting the effort to do this work neatly and properly now and then forget about it. Next only you can thing on circuits and simulation.